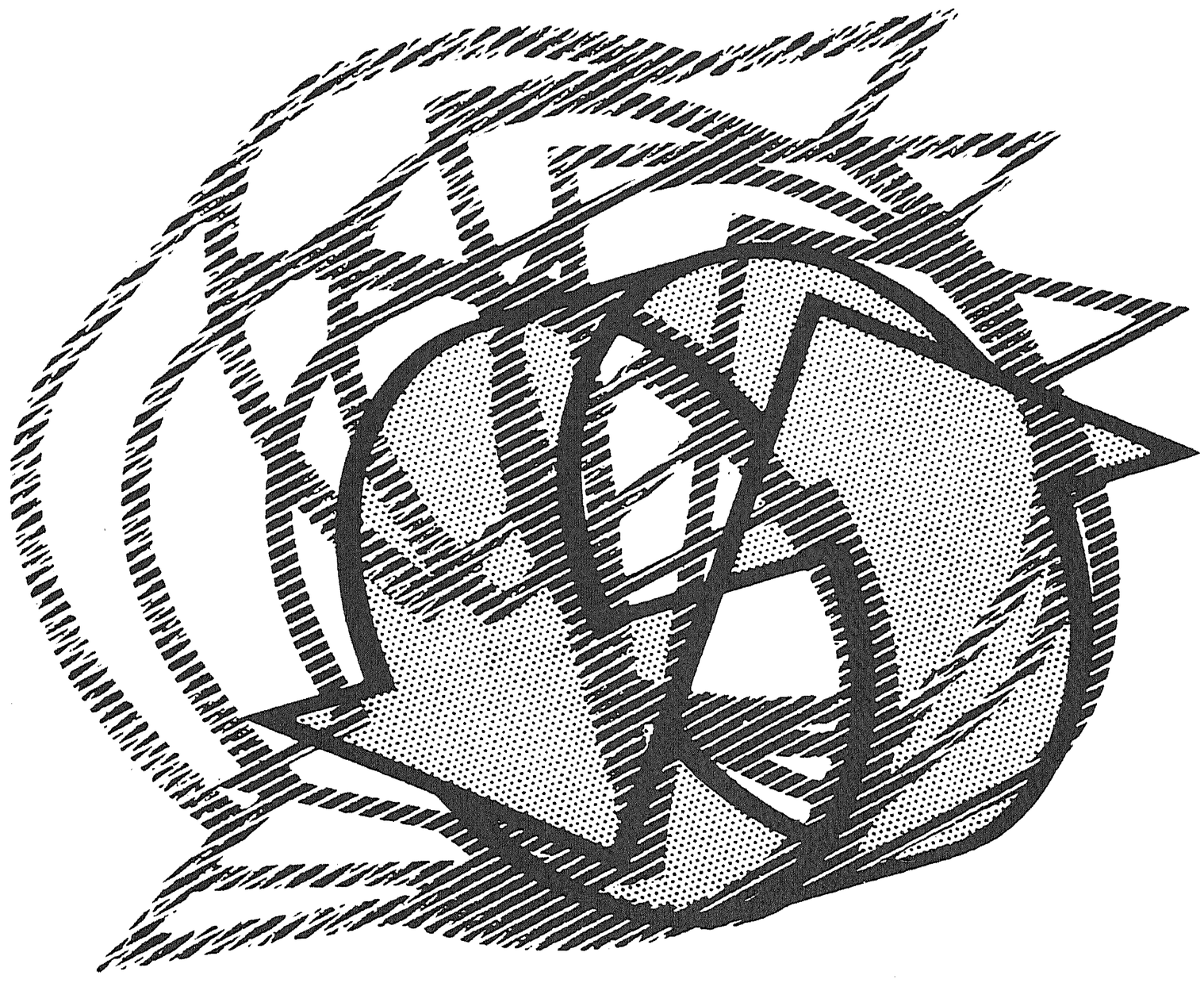
CWI Monographs 1

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

Mathematics and Computer Science

edited by
J. W. de Bakker
M. Hazewinkel
J. K. Lenstra



North-Holland

CWI Monographs

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

The CWI is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

CWI Monograph 1

Mathematics and Computer Science

Proceedings of the CWI symposium
November 1983

edited by
J. W. de Bakker
M. Hazewinkel
J. K. Lenstra

1986

North-Holland
Amsterdam · New York · Oxford · Tokyo

ISBN: 0 444 70024 2

Publishers:
Elsevier Science Publishers B.V.
P.O. Box 1991
1000 BZ Amsterdam
The Netherlands

Sole distributors for the U.S.A. and Canada:
Elsevier Science Publishing Company, Inc.
52 Vanderbilt Avenue
New York, N.Y. 10017
U.S.A.

Cover: Tobias Baanders

Printed in the Netherlands

Preface

As of September 1, 1983 the Mathematisch Centrum (Mathematical Centre) in Amsterdam changed its name to Centrum voor Wiskunde en Informatica (CWI), which translates as Centre for Mathematics and Computer Science. It seemed time to acknowledge the fact that Computer Science has always been an integral part of our Centre since its inception in 1946 and that this will continue to be the case.

To mark the occasion two events were arranged: a 'Name Change Day' on August 31, 1983 and a Symposium on Mathematics and Computer Science on November 25, 1983. On the occasion of the Symposium five leading experts surveyed various topics touching upon both mathematics and computer science. This was also the theme of the 'Name Change Day'. Four of the five surveys presented at the Symposium (by A.J. BADDELEY, C.B. JONES, L. LOVÁSZ and J.T. SCHWARTZ) and the two lectures delivered on the Name Change Day (by M. HAZEWINKEL and L.G.L.T. MEERTENS) have been collected in this volume. The fifth speaker at the Symposium, D.S. SCOTT, found himself unable to compose a text in time. The six contributions are complemented with a number of papers written by various scientists involved - in one way or another - in our Centre.

This volume may serve to underline our conviction that the two structural sciences, mathematics and computer science, should not be separated, as has happened in many schools of science. Each will lose by ignoring the other. Both are developing remarkably vigorously at the present time, and it is a often observed historicoscientific fact that a successful and fast development leaves no time and little inclination for matters of synthesis and interrelations with related activities. It would be a mistake, however, to interpret this phenomenon as a sign that things are growing apart. The history of the interrelations between physics, chemistry and mathematics, in both recent and

much older history, illustrates the point.

The name change also marked a change in the publishing activities of the CWI. Next to the 'Tracts' and 'Syllabi', there now exists the series 'CWI Monographs' of which this is, we hope, a worthy first volume. The volume was produced by the phototypesetting system of the CWI and it is a pleasure to thank here all those who put substantial efforts into it: the typing staff, especially Mrs. J. Kustina and Mrs. L. Brown, the computer typesetting group, notably H. Noot, and the desk editor, W.A.M. Aspers. In addition, we are grateful to all those who contributed in other ways to the events mentioned and the appearance of this volume.

J.W. DE BAKKER
M. HAZEWINKEL
J.K. LENSTRA

Contents

Stochastic Geometry and Image Analysis

A.J. Baddeley
Division of Mathematics and Statistics, CSIRO,
P.O. Box 218, Lindfield NSW 2070 Australia

We list recent ideas in stochastic geometry which are closely related to image analysis. These include the synthesis of stochastic models of images, techniques for evaluating models and algorithms, general concepts of 'geometrical information' and the theory of random sets, problems of image irregularity and errors in observation, techniques of geometric integration theory, and fractional dimensional irregularity.

1. Introduction

The development of computerized image processing and image analysis already seems to have prompted considerable study of the relations between geometry, probability theory and computer science. Rosenfeld [29, preface] observes that all image processing algorithms must be based explicitly or implicitly on mathematical models of the images to be processed. Some of the newer stochastic image models presented in [29] are based on Markov processes, random fields, random mosaics (tessellations) and stochastic grammars. Apart from image modeling, we imagine other mathematical contributions should include a theoretical background for the comparison of algorithms, and mathematical techniques for the treatment of image models.

Independently of such requirements, many concepts related to image analysis have evolved in other areas, notably in stochastic geometry, stereology and geometric integration theory. *Stochastic geometry* is that part of probability theory dealing with random subsets of a geometrical space, and interactions between probability and geometry. This includes all stochastic image models, at least in principle, but some frequently studied models are: elementary constructions of random lines, circles or triangles; spatial schemes such as random mosaics and random coverings of the plane; and general random processes and random sets. The main body of theory concentrates on *uniformly random* models, for which there are simple explicit solutions. However, the last decade has seen the introduction of more flexible techniques and a completely general theoretical foundation for random sets.

This paper summarizes some recent work in stochastic geometry (drawing also on stereology and geometric integration theory) which could be connected with image analysis. Section 2 introduces the range of random image models in stochastic geometry, and outlines the classical theory of uniformly random

models. The more recent combinatorial theory (Section 3) has an application to problems of image complexity. Section 4 discusses the Kendall-Matheron abstract theory of random sets, which has many similarities to tenets of image analysis. J. Serra's mathematical morphology and image analysis theory is touched upon in Section 5. Recent thoughts about image irregularity and observation errors (Section 6) are developed using geometric integration theory. Finally Section 7 speculates on the usefulness of fractal (fractional dimensional) models of image irregularity.

2. Classical stochastic geometry

Detailed surveys of stochastic geometry can be consulted in the literature [24, 3, 7, 32, 35] and we shall give here a very brief sketch. Probability models available for generating random geometrical objects (hence random image models) can be classified as:
(a) elementary constructions;
(b) stochastic processes;
(c) theory of random sets.

(a) Elementary constructions are the simple geometrical figures of Euclid with an added component of randomness, as for example the output of a computer graphics program when the input is a random number generator. Points, lines, triangles, circles and other figures are determined by $n<\infty$ real parameters so that a random figure can be defined as a probability distribution on the n-dimensional parameter space. Of course we may also construct the random line joining two random points, and so on. Using parametrisations of the rotation and translation groups we may generate random positions of an arbitrary object. Typical problems include finding the probability that two random figures (or a random figure and a fixed figure) will intersect; the mean area of length of overlap between figures; and the probability that N random figures will completely cover a specified region.

Even the simplest problems for random figures lead to difficult multiple integrals. An exception to this rule is that *uniformly distributed* random figures often lead to simple explicit solutions. For example, a random two-dimensional point $X = (x_1,x_2)$ is a *uniformly random* (UR) point in the region $A \subset \mathbb{R}^2$ if it has constant probability density $f(x_1,x_2) = K$. The constant must be $K = 1/\text{area}(A)$ since probability integrates to 1. For any measurable subset $B \subset A$ we find the probability

$$P(X \text{ falls in } B) = \frac{\text{area}(B)}{\text{area}(A)}\ , \tag{1}$$

which is what we understand by a 'simple explicit solution'. Now consider a random circle $C(X,r)$ of fixed radius r obtained by randomizing the centre point X. Let X be a uniformly random point in the disc D_{R+r} of radius $R+r$ and centre 0. Then the circle $C(X,r)$ always intersects D_R, the disc of radius R about 0. We say $C(X,r)$ is a uniformly random circle hitting D_R. For any (fixed) point $x \in D_R$,

$$P(C(X,r) \text{ contains } x) = P(X \text{ falls in } C(x,r)) = \frac{\pi r^2}{\pi(R+r)^2}$$

by (1), which does not depend on x. Furthermore, the mean or expected area of overlap between $C(X,r)$ and D_R is by Fubini's theorem

$$\mathbb{E}(\text{area } C(X,r) \cap D_R) = \int_{D_R} P(x \text{ lies in } C(X,r))dx$$

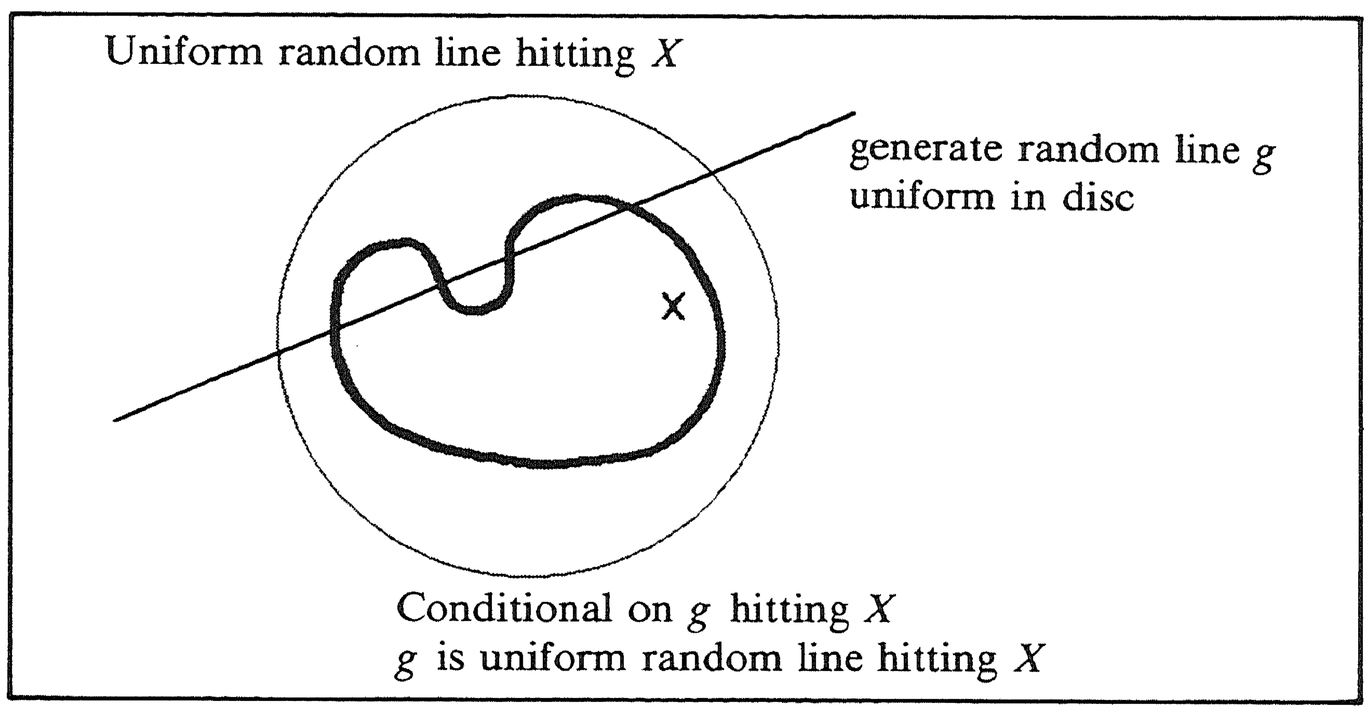
$$= \pi R^2 \frac{r^2}{(R+r)^2},$$

i.e. proportional to the product of areas of $C(X,r)$ and D_R.

Definition of a uniformly random line is less intuitive. Let parameters (p,θ) specify the line

$$\{(x,y): x\cos\theta + y\sin\theta = p\},$$

i.e. $|p|$ is the distance of the line from the origin, and θ determines its direction. A *uniformly random* (UR) line is such that (p,θ) is a uniformly distributed point in some bounded region of $\mathbb{R}\times[0,\pi)$. For example a UR line hitting the disc D_R is obtained when p and θ are independent random variables uniformly distributed over $[-r, +r]$ and $[0,\pi)$ respectively. In general for $X \subset \mathbb{R}^2$ the set of lines intersecting X is some irregular set of (p,θ) points in the allowable region. To generate a UR line hitting X, in practice, find a disc D_R circumscribing X. Generate a UR line L hitting D_R; if $L \cap X = \varnothing$, reject this attempt and generate another line L; until L hits X. Then L is UR hitting X.



Uniform random lines have the invariance property that if L is a UR line hitting X, and if $Y \subset X$, then the probability $P(L \text{ hits } Y)$ does not depend on the position or orientation of Y within X. All parts of X are equally likely to be 'sampled' by L. This fair sampling property, which characterizes the uniform distribution, can be recognised as invariance under the euclidean group of rigid motions. Another nice characterization of UR lines is based on the two-person game where A 'hides' a set Y inside X and player B draws a line L through X to find Y. Optimal strategy for B is to generate a uniform random line.

We state two fundamental results concerning UR lines. Let L be a UR line through X, a bounded measurable plane set. If $A \subset X$ is measurable then

$$\mathbb{E} \text{ length } (L \cap A) = \frac{\pi . \text{area}(A)}{K} \tag{2}$$

where $\mathbb{E}$ again denotes expected (mean) value, and K is a constant depending on X. If $C \subset X$ is a plane curve then

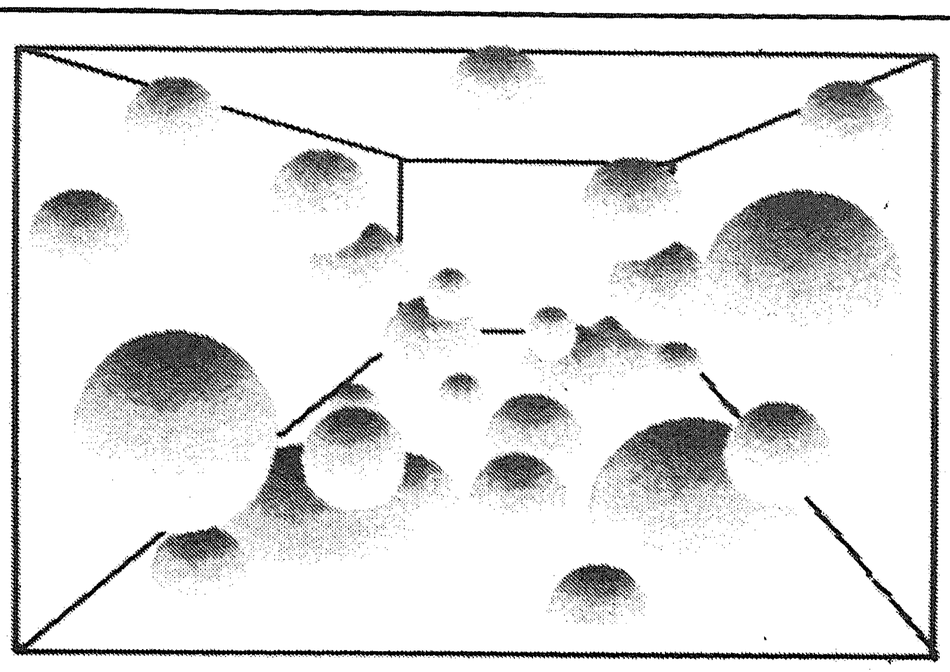
$$\mathbb{E}n(L \cap C) = \frac{2 . \text{length}(C)}{K} \tag{3}$$

where $n(L \cap C)$ is the number of intersection points between L and C. Thus, the mean amount of overlap between a UR line and a fixed figure is given by (2), (3) *regardless* of the geometrical configuration of the figure. This generality is the basis of the classical theory. Corresponding formulae hold in higher dimensions and noneuclidean spaces [30].

Apart from the obvious application of (2)-(3) to stochastic image models, we can interpret them to give methods for measurement of length and area. If an image consists of several curves, their total length can be statistically estimated by randomly rotating the image, superimposing a grid of parallel lines and counting the number of crossing points.

Statements about image complexity also follow from (2)-(3). Suppose the image consists of curves of total length l, the screen is divided into an $n \times n$ square grid, and we wish to estimate the number of grid squares which contain part of the image. Assuming the image and grid are randomly superimposed, the mean number of grid-image intersections is $\frac{4}{\pi}(n-1)l$. For large n this approximates the mean number of *squares* crossed, i.e. the mean complexity.

Stochastic image models may also be based on (b) *stochastic processes.* To generate a random pattern extending over the entire plane, divide $\mathbb{R}^2$ into squares, and place a random number of random points in each square. A random pattern of lines is a random pattern of (p, θ) points in $\mathbb{R} \times [0, \pi)$, and so on. Thus we define a *random point process* in space S as a random locally finite set of points in S, where 'locally finite' means each bounded region of S only contains a finite (random) number of (random) points. *A random line process* 'is' a random point process in $\mathbb{R} \times [0, \pi)$, or more intrinsically, is a random locally finite set of lines in $\mathbb{R}^2$. In calculations one uses the correspondence between a random point process and the system of random variables



A finite population of spheres in $\mathbb{R}^3$
or
a realisation of a random sphere process

$N(A)$ = (number of points in A), $A \subset S$, which constitute a *random measure* $N(\cdot)$ on S. A random line process is a random measure on $\mathbb{R} \times [0,\pi)$, or intrinsically, corresponds to a random capacity function $H(A)$ = (number of lines *intersecting* A), $A \subset \mathbb{R}^2$. See [18,12].

Explicit calculations are usually unsuccessful except for *uniform Poisson processes,* in which each bounded part of the process consist of independent uniformly random points/lines, and $N(A)$, $N(B)$ are independent when $A \cap B = \varnothing$. Equations (1)-(3) yield the expected values of $N(A)$, $H(A)$, the number of crossings of a fixed curve, the total length of lines overlapping A, and the number of line-line crossings inside A.

General random point processes and line processes have been studied using moments [12,19,32] and Palm probabilities [26]. For a point process the first two moment measures are the intensity measure $\mu(A) = \mathbb{E}[N(A)]$ on $\mathbb{R}^2$, and the second moment measure $\mu^{(2)}$ on $\mathbb{R}^2 \times \mathbb{R}^2$ defined by $\mu^{(2)}(A \times B) = \mathbb{E}[N(A)N(B)]$, which together contain all variance-covariance information. If the process is statistically stationary, then $\mu(A) = \lambda \text{area}(A)$ where $\lambda > 0$ is the intensity, while $\mu^{(2)}$ 'disintegrates',

$$d\mu^{(2)}(x,y) = d\gamma(y-x)d\mu(x) \quad x,y \in \mathbb{R}^2$$

and the measure γ on $\mathbb{R}^2$ describes correlations between points in the process. The correlation characteristics can be estimated from observations of the process, furnishing a general empirical approach to point- and line- processes [33]. Second-order statistics characterize many of the visible characteristics of an

image or pattern [11], but are not infallible [28,5]. A direct analysis of dependence between points or lines in a process is obtained using the Palm probabilities P^x, essentially the conditional probability distribution of the random process *given* that there is a random point at x.

A random line process or circle process subdivides the plane into a random tessellation. This is a potentially important model of random images [14, 24, 31]. Characteristics of the polygons formed by a *Poisson* line process have been determined by MILES [23], in particular the means and variances of polygon area, perimeter length and number of sides. Another important random tessellation is the Dirichlet or Voronoi tessellation: if $\{x_i, i \in Z\}$ are the points in a point process, let the tile corresponding to x_i be

$$T_i = \{y \in \mathbb{R}^2 : |y - x_i| \leqslant |y - x_j|,\ j \neq i\}.$$

The T_i are polygons tessellating $\mathbb{R}^2$. Characteristics of the Voronoi tessellation induced by a Poisson point process are given by MILES [21].

Finally, random image models can be based on (c) *the theory of random sets.* This is discussed in Section 4.

3. COMBINATORIAL THEORY

More results have recently been obtained for classical problems, by simplifying geometry and applying combinatorial probability methods [1]. We will first prove the curve length formula (3),

$$\mathbb{E}n(L \cap C) = \frac{2 \text{ length } (C)}{K}$$

where C is a plane curve, L is a UR line hitting $X \supset C$, and $n(L \cap C) =$ number of intersection points in $L \cap C$. Suppose C is a *polygonal* curve consisting of line segments $S_1, S_2, \ldots, S_n$. Let $[S_i]$ denote the event $L \cap S_i \neq \varnothing$, that is L hits S_i. Put

$$1_{[S_i]} = \begin{cases} 1 & \text{if } L \cap S_i \neq \varnothing \\ 0 & \text{if } L \cap S_i = \varnothing . \end{cases}$$

Clearly we have

$$n(L \cap C) = \sum_{i=1}^{n} 1_{[S_i]}$$

with probability 1, since $P(L \text{ contains } S_i) = 0$. But immediately

$$\mathbb{E}n(L \cap C) = \sum_{i=1}^{n} \mathbb{E}\, 1_{[S_i]} = \sum_{i=1}^{n} P([S_i]).$$

It can easily be argued that uniform random lines have $P([S_i])$ proportional to length (S_i);

$$\mathbb{E}n(L \cap C) = \alpha \sum_{i=1}^{n} \text{length } (S_i) = \alpha.\ \text{length } (C)$$

which proves (3) up to the constant factor.

The proof reveals importance of *additivity,* meaning both the linearity of the integral $\mathbb{E}$ and the additivity of the counting function $n(L \cap C)$. Together with the natural properties of uniform distributions, this property forms the basis of stochastic geometry.

Suppose now we want the *distribution* of the variable $n(L \cap C)$: computation of $P\{n(L \cap C) = k\}$ is not obvious. Consider two segments S_1, S_2 and evaluate $P([S_1] \cap [S_2])$, the probability that L intersects *both* S_1, S_2. Case 1: if S_1, S_2 have a common point, let T be the third side of the triangle. Then

$$1_{[S_1] \cap [S_2]} = \frac{1}{2}(1_{[S_1]} + 1_{[S_2]} - 1_{[T]}) \qquad \text{a.s.}$$

since if L intersects both S_1, S_2 the sum in brackets equals 2, and otherwise is zero. Case 2: if S_1, S_2 have no common point we can derive a similar expression

$$1_{[S_1] \cap [S_2]} = \frac{1}{2}(1_{[A_1]} + 1_{[A_2]} - 1_{[B_1]} - 1_{[B_2]}) \qquad \text{a.s.}$$

where A_1, A_2, B_1, B_2 are segments joining the four endpoints of S_1, S_2. But this implies that *every expression* $1_{[S_i] \cap [S_j]} = 1_{[S_i]} \cdot 1_{[S_j]}$ *can be written as a linear combination* of variables $1_{[T_k]}$, where T_k are line segments joining vertices of C.

THEOREM. *Let* $x_1, \ldots, x_n$ *be points in* $\mathbb{R}^2$, *and* s_{ij} *the line segment joining* x_i *with* x_j. *For a random line* L, *let* $[s_{ij}]$ *be the event* $L \cap s_{ij} \neq \varnothing$. *Let* $\mathfrak{A}$ *be the ring of events generated (through unions, intersections, set differences) by* $[s_{ij}]$, $1 \leqslant i < j \leqslant n$. *Then for any* $A \in \mathfrak{A}$ *there exist constants* $c_{ij}(A)$ *such that*

$$1_A = \sum_{i<j} c_{ij}(A) 1_{[s_{ij}]} \tag{4}$$

holds except when L *contains a vertex* x_i.

If L is uniformly distributed we take mean values in (4) to get

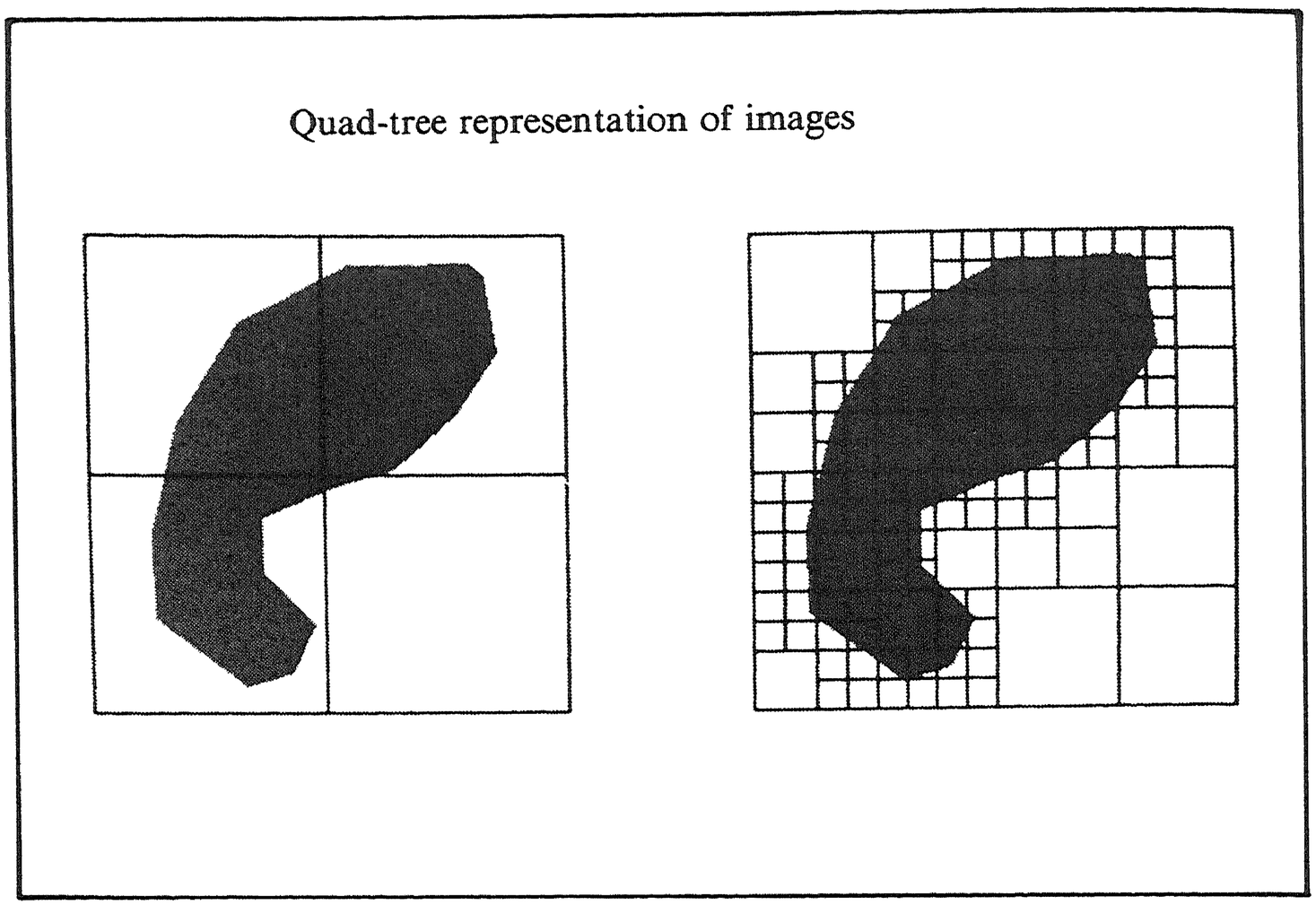
$$P(A) = 2/K \sum_{i<j} c_{ij}(A) \|x_i - x_j\| \tag{5}$$

i.e. all combinatorial probabilities for UR lines are expressible as sums of lengths of segments s_{ij}. For example, the distribution of $n(L \cap C)$ is expressible in terms of the distances between each pair of vertices of C. This is a great advance, in principle, on the classical theory which was restricted to mean values. An algorithm for the $c_{ij}(A)$ is known, and practicable for small n.

One can also take non-uniform random lines in (4), say with probability distribution Q, to obtain

$$Q(A) = \sum_{i<j} c_{ij}(A) Q[s_{ij}] \tag{6}$$

and note the coefficients $c_{ij}(A)$ are the same as above. The quantity $Q[s_{ij}]$ serves as a generalized length of s_{ij}. Thus, again in principle, nonuniform random lines are no more computationally difficult than UR lines.



Finally we present another application to image complexity, concerning the quad-tree representation of images. An image can be recorded or transmitted as tree structure, as follows. Divide the image field into four equal squares and note which squares, if any, consist of a single colour. The remaining, multicoloured squares are subdivided again into four, and the process repeats until a predetermined level of subdivision is reached. The record of subdivisions and colours forms the *quad tree.* Important questions include the average complexity (number of nodes) of the quad tree, and estimating the increase in complexity if a deeper level (finer subdivision) is added. Both problems depend on the image, but it is reasonable to suppose that in a sufficiently small square, the image boundary can be regarded as a uniformly random line. Consider a UR line hitting a square subdivided into $k \times k$ equal squares. According to (3) the mean number of subsquares crossed equals k. Furthermore using (5) we can compute the distribution of the number N of subsquares crossed. In the interesting case $k = 2$, we have $P(N = 1) = \frac{1}{2}(\sqrt{2} - 1)$, $P(N = 2) = 2 - \sqrt{2}$, $P(N = 3) = \frac{1}{2}(\sqrt{2} - 1)$. Thus the cost of adding one extra level of subdivision is to double the number of terminal nodes, on average. One fifth of the new branches will be triple.

4. Random set theory

In addition to the constructive examples of random geometry in Section 2, one can propose others such as the zero-set (or contours) of a random function. Foundations of a general theory of random sets were laid by G. Matheron [18] and D.G. Kendall [13]. Matheron's theory of random closed sets was expressly developed as a mathematical background to image analysis as well as stochastic geometry. Kendall's theory takes an abstract view of the construction of probability spaces of random sets, emphasising the variety of structures which can be chosen. The two approaches are complementary [27] and both make use of Choquet's capacity theorem.

To introduce the theory we generalize the random events $[S_i]$ which played a formative role in Section 3. For the Matheron approach, let $\mathcal{F}$ be the class of all closed sets in $\mathbb{R}^n$. If $T \subset \mathbb{R}^n$ define the *hitting set*

$$[T] = \{F \in \mathcal{F} : F \cap T \neq \varnothing\}.$$

Endow $\mathcal{F}$ with the (weakest) topology such that $[U]$ is an open subset of $\mathcal{F}$ for all open sets $U \subset \mathbb{R}^n$, and $[K]$ is closed for all compact $K \cap \mathbb{R}^n$ (see [20]). Then $\mathcal{F}$ becomes a Polish space. Define a *random closed set* as a random element of $\mathcal{F}$ with the Borel σ-algebra. Under this structure the *events* $[T]$, $T \subset \mathbb{R}^n$ are measurable when T is open, closed or indeed Borel. Intersections and unions of random closed sets are random closed sets. Area, length (where defined) and number of points (where finite) are random variables.

Kendall's approach emphasises that the definition of a random set depends on the geometrical information which is assumed to be observable. Its basic constituents are the random events $[T] = \{X \cap T \neq \varnothing\}$ where X is the random set and T is a fixed set called a 'trap'. The associated random variable

$$h(T) = \begin{cases} 1 & \text{if} \quad X \cap T \neq \varnothing \\ 0 & \text{if} \quad \text{not} \end{cases} \tag{7}$$

corresponds to a 'bit' or 'flag' indicating whether X was detected by the trap T. From the observer's point of view, the random set X is characterized by the information $\{h(T), T \in \mathcal{T}\}$ where $\mathcal{T}$ is the class of all traps available to the observer. Define a *trapping system* $\mathcal{T}$ on a space S to be a class of nonempty subsets of S, which cover S, satisfying certain properties analogous to separability and local compactness. A *random $\mathcal{T}$-set* in S is a random function

$$h : \mathcal{T} \to \{0, 1\}$$

i.e. a stochastic process of 0-1 variables $h(T)$, $T \in \mathcal{T}$, subject to a consistency condition which enables h to be interpreted in the form (7). Note the probability structure depends completely on the choice of trapping-system. If $S = \mathbb{R}^n$ and $\mathcal{T} =$ open sets, a random $\mathcal{T}$-set is a random closed set in Matheron's sense. Smaller trapping-systems may be inadequate to distinguish all closed sets. A set X is indistinguishable (to the observer) from its *$\mathcal{T}$-closure,*

$$\text{clos}\,(X, \mathcal{T}) = \left[\bigcup_{X \cap T = \varnothing} T \right]^c = \bigcap_{X \cap T = \varnothing} T^c$$

(c denotes complement) and we need only consider $\mathcal{T}$-*closed* sets $X = \text{clos}(X, \mathcal{T})$. For example if $\mathcal{T} = \{\text{open halfplanes of } \mathbb{R}^2\}$ the $\mathcal{T}$-closed sets are the convex sets of $\mathbb{R}^2$. Thus random $\mathcal{T}$-sets in this case 'are' random convex sets; and the customary representation of convex sets by support functions can be derived from $h(T)$.

Random set theory provides solid foundations for investigating both stochastic geometry and the observation and processing of images. For example, convergence of random sets is a natural concept in the general theory which has been applied to assess errors committed in digitizing an image, approximation of one image by another, and the stability of image processing transformations [31, Chapter VII] and to derive the statistically important laws of large numbers and a central limit theorem for repeated observations of images [2,36]. The general setting also permits more involved discussion of the probabilistic properties of image models, such as infinite divisibility and the semi-Markov property [18,19]. It is a basic result that the probability distribution of a random set X is determined by its *avoidance function*

$$Q(A) = \text{Prob}\,(X \cap A = \varnothing), \quad A = \bigcup_{i=1}^{n} T_i, \quad T_i \in \mathcal{T}$$

and the introduction of Q makes for a coherent approach to image models [31,18].

The strongest link between image analysis and random set theory is surely the trapping system. Any image is given to us through an array of detectors (and perhaps subjected to edge detection processing, etc.) which can be formalised as a trapping system. Further, the relationships between various forms of image information (e.g. digitized versions on different lattices; grey tones) can be studied by varying $\mathcal{T}$ in the stochastic model. The author feels that the great potential of this method is yet unexplored.

5. Mathematical morphology

The work of J. Serra [31] establishes a coherent methodology for image analysis which avoids the fragmentary character of most other approaches. Mathematical morphology developed in parallel with random set theory, beginning with Matheron's [17] geostatistical work and Serra's invention of the 'texture analyzer' image processing devices. The result is a combination of sound theoretical criteria with practical experience. We can only convey the flavour of the subject here.

Transformations of sets arise in many stochastic geometry problems. Consider the probability distribution of the random distance $d(x, A)$ from a fixed set $A \subset \mathbb{R}^2$ to a random point $x \notin A$. Clearly $P\{d(x, A) \leqslant r\}$ equals the probability that X falls in the region $A_{(r)} = \{x \in \mathbb{R}^2 : d(x, A) \leqslant r\}$ which we dub the *r-envelope* of A. Equivalently $A_{(r)}$ is the set formed by placing a disc of radius r around every point $a \in A$. The envelope transformation $A \to A_{(r)}$ is the simplest example of a set transformation. If $A = D_R$ is a disc then $A_{(r)} = D_{R+r}$, while in general the shape of $A_{(r)}$ is more rounded (with smaller holes) than that of A. It is argued that the function $f_A(r) = \text{area}\,(A_{(r)})$ reflects essential

characteristics of the geometry of A. If A is convex then $f_A(r) = \pi r^2 + r.$ length (∂A) + area (A), while if A is a finite set of points then f_A is piecewise quadratic with a behaviour reflecting the sizes of gaps between the points. A series of images $A_1, \ldots, A_n$ could be differentiated or discriminated using the derived functions $f_{A_1}(r), \ldots, f_{A_n}(r)$.

The envelope operation can be performed on a discrete grid of points. A simple algorithm is to scan the entire grid and, for each point x whose digital neighbourhood includes a point of the current image A, we mark x for inclusion in the new image $A_{(r)}$. Furthermore we can watch this process of expansion for increasing r by repeating the algorithm, since $(A_{(r)})_{(s)} = A_{(r+s)}$. This is done by texture analyzers.

The *Minkowski sum* of two sets $A, B \subset \mathbb{R}^2$ is defined as

$$A \oplus B = \{a+b : a \in A,\ b \in B\}$$

in the sense of vector addition. If B is the disc D_r then $A \oplus D_r = A_{(r)}$, the r-envelope. More generally $A \oplus B$ is the superposition of translated copies of B centred on each of the points of A, if we take the origin 0 as the 'centre' of B. Shifted copies of A are obtained when B is a single point, $A \oplus \{b\} = \{a+b : a \in A\}$. Defining $\check{B} = \{-b : b \in B\}$ one can interpret $A \oplus B = \{x \in \mathbb{R}^2 : (x \oplus \check{B}) \cap A \neq \varnothing\}$, the set of all 'centres' of shifted copies of $\check{B}$ which intersect A. Hence the transformation $A \to A \oplus B$ also has a clear interpretation in stochastic geometry, and can be claimed to reflect important characteristics of the geometry of A. This and other set transformations can be implemented on a discrete grid by including or removing points x according to the state of the entire digital neighbourhood of x.

Minkowski subtraction of $A, B \subset \mathbb{R}^2$ is defined by

$$A \ominus B = (A^c \oplus B)^c$$

i.e. the complement A^c is enlarged by B. For example, if $B = D_r$ is a disc, $A \ominus D_r = \{x \in A : d(x, A^c) \geqslant r\}$ is the *inner parallel set.* In general $A \ominus B = \{x \in A : x \oplus \check{B} \subset A\}$ is the set of all centres of copies of $\check{B}$ *contained* in A. This has a natural interpretation and the function $g(r) =$ area $(A \ominus D_r)$ is claimed to contain essential information about the geometry of A. Define two further set transformations, the closure

$$A^B = (A \oplus \check{B}) \ominus B$$

and opening

$$A_B = (A \ominus \check{B}) \oplus B.$$

Thus A_B is the union of all copies of B contained in A; and A^B is the result of a similar operation on A^c. A set is B-closed, $A^B = A$, iff it is $\mathcal{T}$-closed in the sense of Section 4 where $\mathcal{T}$ is the class of all translated copies of B. Apart from their natural interpretation in stochastic geometry, A^B and A_B can be used to develop a rigorous definition of size and size distribution for images [18,31].

The mathematical morphology approach to an image processing problem is to select an image transformation (built from $\oplus, \ominus$, A^B, A_B etc.) suitable to

the application, and make numerical analyses of the transformed images. One chooses transformations either by experience, intuition about the scientific problem, or by setting down criteria which the transformation must satisfy.

Some limitations of mathematical morphology as it currently stands call for brief comments. The texture analyser is designed on a hexagonal point lattice for the digitized image. Naturally the theory is strongly dependent on this choice of instrumentation, and probably does not answer all questions about random image models that are required in different applications. Associated with the choice of instrumentation is the adoption [31, pp 8-15] of a list of theoretical principles which notably excludes *rotational* stability. A hexagonal grid has only three basic directions and there have been difficulties with the analysis of image orientation or directionality. There may also be practical reasons for employing a rectangular grid or other system of image detection - for example, satellite data may already be in this form. Another problem with all image analysis based on stochastic geometry is that images are not sharply divided black and white sets, but grey tone functions. This is a drawback to the widespread use of texture analyzers. Mathematical morphology for grey-tone functions is under development [31, Chapter XII].

The author suspects one can be led astray by excessive analysis of a single image, when this image is to be representative of a larger population. This applies particularly in stereology, where the planar image is a random plane section $X \cap E$ of a three-dimensional body X which is the real object of interest. It is then important that the sampling procedure used to generate $X \cap E$ should be known, and appropriate. Statistical inferences depend on the sampling method used. It is not quite sufficient to base image analysis on considerations of the trapping-system and other geometrical structures, without incorporating statistical models for the origins of data.

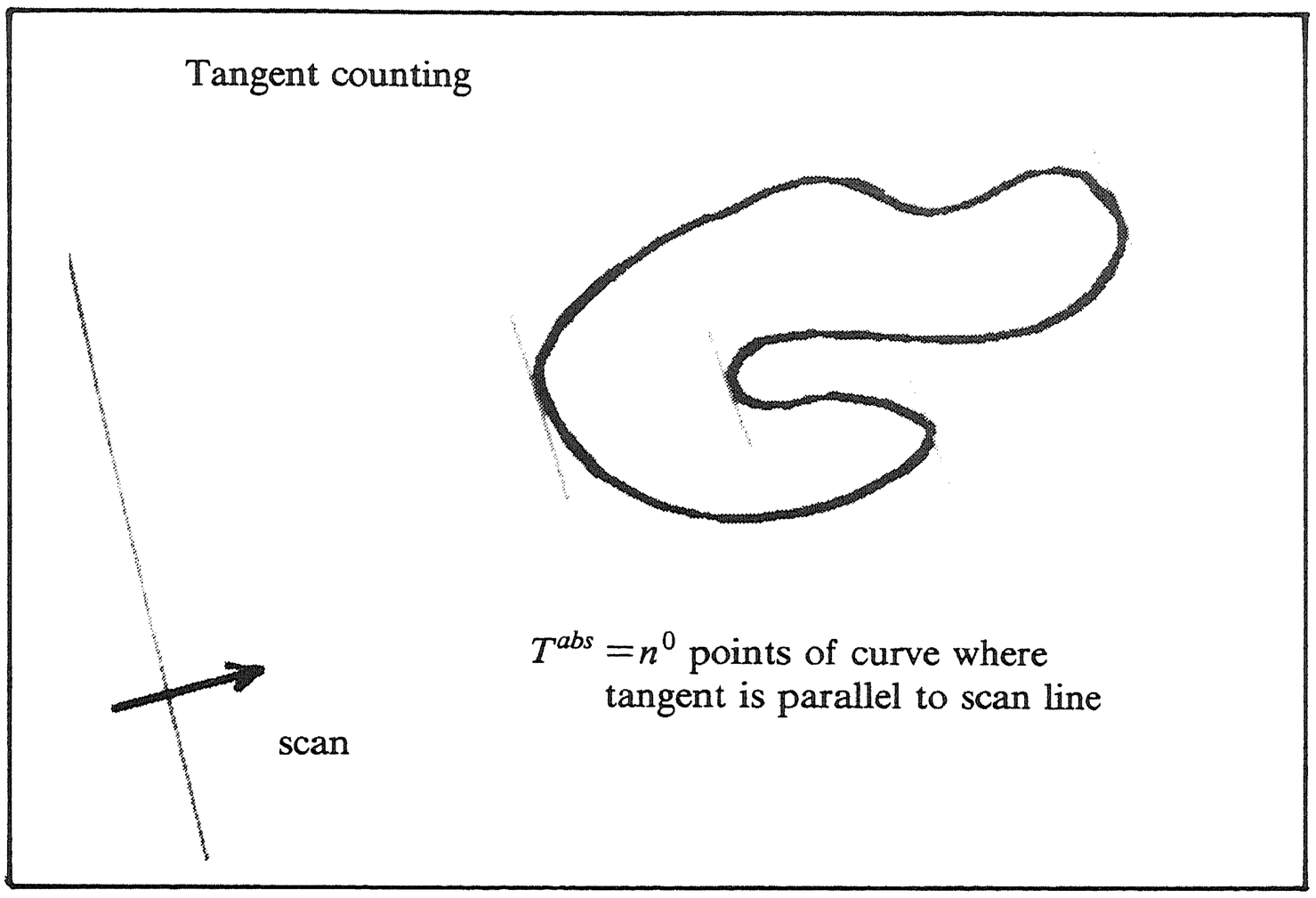
6. Image irregularity, observation errors and geometric measure theory

Elementary formulae from stochastic geometry (see (1)-(3) in Section 2) are widely used in stereology for measuring curve lengths, estimating surface areas and so on. Yet these results were derived for ideal smooth curves and it is a priori doubtful whether they apply to irregular images or images observed under error.

An extreme example is *tangent counting*. Let C be a twice differentiable plane curve, $\theta \in [0,\pi)$ and $T^{abs}(\theta)$ = number of tangents to C parallel to direction θ. This would be found by scanning a straight line across the image (parallel to θ) and counting the positions where the image is tangent to C. We have

$$\int_0^{\pi} T^{abs}(\theta)d\theta = \int_C |\kappa(s)|\,ds \tag{8}$$

where $\kappa(s)$ is the curvature of C at point s. If the scan direction θ is generated at random (uniformly), $\pi T^{abs}(\theta)$ is a statistically unbiased estimator of the total



absolute curvature of C. Additionally if C is itself a random plane section of a curved surface, then T^{abs} yields an estimate of the total 'absolute' surface curvature.

Even assuming that real images are differentiable, the tangent count is unstable in the sense that small perturbations (kinks, ripples) in C may cause large changes in T^{abs} and κ. More realistically if C is the boundary of a finite union of convex compact sets (hence, almost everywhere differentiable) T^{abs} does not share the properties usually required of a good statistic. SERRA [31, p.141 ff] nevertheless shows that a precise and useful interpretation can be given to the tangent count or 'convexity number' of such curves, and that this can be approximately determined from a digitized image.

Practical stereologists and image analysts follow procedures for counting 'tangents' to image curves, even when these are irregular, thick or fuzzy, broken or digitized. A tangent counting algorithm may be built into the image analyzing device. Mathematicians should be discussing the performance of such algorithms, their relation to real geometry, and the effects of observation errors.

Standard proofs of (1)-(3) and (8) do not accommodate a discussion of perturbations or errors, being applications of Fubini's theorem to simple geometrical models. We need the more powerful methods of geometric measure theory [6], principally the *coarea formula*. Briefly, let M, N be m- and n-dimensional domains (rectifiable surfaces), $m \geqslant n$, and let $p : M \to N$ be a Lipschitz-continuous map. For almost every $x \in N$, $p^{-1}\{x\} = \{z \in M : p(z) = x\}$ is an

$m-n$ dimensional rectifiable set. If $m=n$, then $p^{-1}\{x\}$ is a finite set. There is a function $J^n p$ defined on M called the approximate Jacobian of p, such that the *coarea formula*

$$\int_M f(z)(J^n p)(z)d\mathcal{H}^m z = \int_N \int_{p^{-1}\{x\}} f(z)d\mathcal{H}^{m-n}z \; d\mathcal{H}^n x \tag{9}$$

holds for any $\mathcal{H}^m$-integrable function $f: M \to \mathbb{R}$, where $\mathcal{H}^k$ is the k-dimensional Hausdorff measure ('k-dimensional volume integration', see Section 7).

Thus (9) is a kind of generalization of Fubini's theorem which incorporates the Jacobian for a change of variables.

To prove (8), for example, let C be a twice differentiable curve, and introduce

$$C^* = \{(s,l): s \in C,\ l \text{ is the tangent to } C \text{ at } s\}.$$

This is a one-parameter set of points in $\mathbb{R}^2 \times \mathbb{R} \times [0,\pi)$. Apply the coarea formula (9) to the map

$$p: C^* \to C, \quad p(s,l) = s.$$

This has $(J^1 p)(s,l) = (1+\kappa^2)^{-\frac{1}{2}}$ where $\kappa = \kappa(s)$ is the curvature of C, and since $p^{-1}\{s\}$ is a single point (s,l) we get

$$\int_{C^*} f(s,l)(1+\kappa^2)^{-\frac{1}{2}} d\mathcal{H}^1(s,l) = \int_C f(s,l)ds$$

for any function f. Similarly, for the map

$$q: C^* \to [0,\pi), \quad q(s,l) = \text{direction of line } l,$$

we have $(J^1 q)(s,l) = (\kappa^2/(1+\kappa^2))^{\frac{1}{2}}$. Since $q^{-1}\{\theta\}$ consists of all pairs (s,l) where l is parallel to θ, we get

$$\int_{C^*} \tilde{f}(s,l)(\kappa^2/(1+\kappa^2))^{\frac{1}{2}} d\mathcal{H}^1(s,l) = \int_0^\pi \sum_{q^{-1}\{\theta\}} \tilde{f}(s_i,l_i)d\theta.$$

If $\tilde{f} = 1$, the sum on the right hand side above is $T^{abs}(\theta)$. Choosing $f(s,l) = |\kappa|$ so that the two left hand sides agree, we get equation (8).

Now suppose that C is nondifferentiable, and that the experimenter has some algorithm for counting or detecting apparent tangents to C. Let

$$\tilde{C} = \{(s,l): s \in \mathbb{R}^2, l \text{ is a line; the algorithm counts } l \text{ as a tangent to at } s\}.$$

Then under suitable conditions we may replace C^* above by $\tilde{C}$ and perform the same calculations to get

$$\int_0^\pi \tilde{T}^{abs}(\theta)d\theta = \int_\Gamma \tilde{\kappa}(s)ds,$$

where $\tilde{T}^{abs}$ is the experimentally observed tangent count, $\Gamma = p(\tilde{C})$ is the set of points at which tangents are detected, and $\tilde{\kappa} = J^1 q / J^1 p$ is a kind of gen-

eralized curvature. For example, let C be an irregular curve $C(t) = A(t)+\epsilon(t)$, $0 \leqslant t \leqslant 1$, where curve A is smooth and $\|\epsilon(t)\| < r$. If the tangent algorithm is such that $s \in A$ and l is tangent to $A \oplus D_r$, then $\Gamma = A$, and $\tilde{\kappa}$ is a function of r and the curvature of A. Thus Γ is a rectified version of C. Secondly, if C is smooth, but a tangent where $\kappa(s)$ is small may not be observed, we get

$$\mathbb{E}(\pi \tilde{T}^{abs}) = \int_C |\kappa(s)| u(\kappa(s)) ds$$

where $u(\kappa)$ = probability of detecting a given tangent at curvature κ. Further examples are explored in [4].

Thus we still have a geometrical interpretation of the image analysis algorithm when it is applied to non-ideal images. This is achieved by concentrating on intrinsic behaviour of the algorithm or observation method, encapsulated in the projection maps p,q. More generally we can regard an image analysis algorithm as an *operator* on images in the sense of generalized functions, and the mathematical prerequisites for such an approach already exist [6].

7. Fractals

Mandelbrot [15,16] explored the concept of fractal (fractional dimensional) sets initiated by Besicovitch, which have wide mathematical associations and seem to be useful models for real images. The simplest kind of fractal set is *self-similar:* if X can be divided into k disjoint sets each of which is congruent to X after magnification by a factor α, then $\Delta = \log\alpha/\log k$ is the similarity dimension of X. For curves $\Delta = 1$; for a disc $\Delta = 2$; but for the Cantor set, $k = 2$, $\alpha = 3$, $\Delta = \log 3/\log 2$ is fractional. When X is magnified, its content increases by a fractional power of the magnification. This extreme form of fractal behaviour is not generally required (except in the limit of small scale). Define for each real $t \geqslant 0$ the t-dimensional Hausdorff measure $\mathcal{H}^t$,

$$\mathcal{H}^t(X) = \lim_{\epsilon \downarrow 0} c_t . \inf\{ \sum_{i=1}^{N} (\text{diam } S_i)^t : S_1, \ldots, S_N \text{ cover } X, \text{diam } S_i < \epsilon \}$$

where the infimum ranges over (say) all families of compact sets S_i with diameters less than ϵ. The limit may be infinite. Define the Hausdorff-Besicovitch dimension of X as

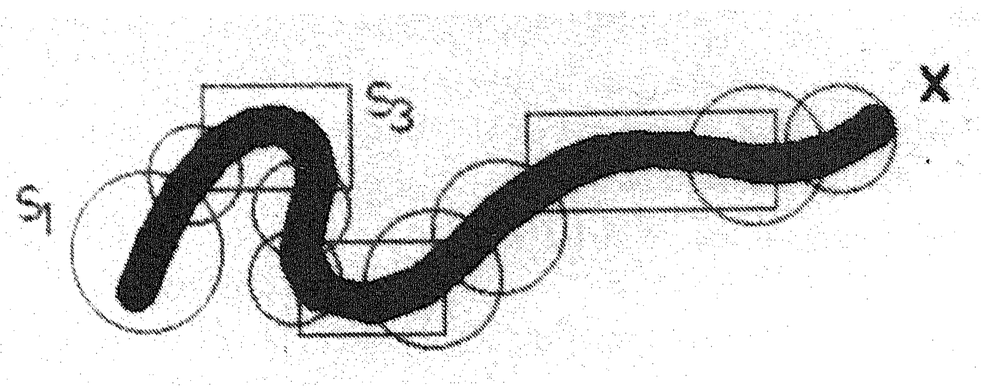
$$D(X) = \sup\{t \geqslant 0 : \mathcal{H}^t(X) < \infty\} = \inf\{t \geqslant 0 : \mathcal{H}^t(X) = \infty\}.$$

Then X is fractal if $D(X)$ is not an integer. If X is a random closed set (Section 4), the topology of $\mathcal{F}$ is such that $D(X)$ is a random variable.

Other examples of fractals include the graphs and zero sets of random continuous functions (the graph of Brownian motion is *statistically* self similar) and limit sets of iterations of quadratic maps in the complex plane. The viewer's impression of a fractal curve is one of sharp irregularity and unbounded oscillation.

Real objects and images do often behave non-linearly with magnification.

Hausdorff measure $\mathcal{H}^t$



$$\mathcal{H}^t(X) = c_t \lim_{\epsilon \downarrow 0} inf\{\sum(\text{diam} S_i)^t : S_1, \cdots, S_n \text{ cover } X \text{ diam } S_i < \epsilon\}$$

Coastlines are the best-known example. Given a picture of a fractal curve ($1<D<2$) we could estimate D as the slope of the regression line relating $\log L(\alpha)$ to $\log\alpha$, where $L(\alpha)$ is an estimate of length obtained at magnification α. Applied to coastlines this has produced a range of fractional dimensions, which seem to reflect degrees of irregularity. A more serious application concerns the measurement of lung membrane surface area [34, p. 156] from plane section curves. Conflicting estimates based on different magnifications have been reconciled and a consistent estimate of D obtained.

Many real phenomena and images have been described as 'fractal' and their empirical values of D determined. The theory of ideal fractals has not kept pace with this development of approximate fractals. Any empirical value of D is a partial description of the image, at certain scales only, and over different scales the 'dimension' may vary. This should not be an objection to the use of fractals as a geometrical model (naturally any model is confined to a chosen scale), but the meaning of a fractal approximation needs to be clarified [10]. We have already observed that Hausdorff dimension fits into the general theory of random closed sets, and indeed $D(X)$ represents an asymptotic index of the frequency of intersections between X and small *traps* $D_r, r \to 0$. It seems to the author that fractional dimensional irregularity could be better understood from the empirical and statistical viewpoint of stochastic geometry.

REFERENCES

1. R.V. AMBARTZUMIAN (1982). *Combinatorial Integral Geometry,* J. Wiley and Sons, Chichester.
2. Z. ARTSTEIN, R.A. VITALE (1975). A strong law of large numbers for random compact sets. *Ann. Probability 5,* 879-882.
3. A. BADDELEY (1982). Stochastic geometry: an introduction and reading list. *Internat. Statist. Review 50,* 179-193.
4. A. BADDELEY (1983). Applications of the coarea formula to stereology. In *[8],* 1-17.
5. A. BADDELEY, B.W. SILVERMAN. A cautionary example on the use of second order methods for analyzing point patterns. To appear in *Biometrics.*
6. H. FEDERER (1969). *Geometric Measure Theory,* Springer, Heidelberg.
7. R. FORTET, M. KAMBOUZIA (1975). Ensembles aléatoires, répartitions ponctuelles aléatoires, problèmes de recouvrement. *Ann. Inst. Henri Poincaré 11(B),* 299.
8. H-J.G. GUNDERSEN, E.B. JENSEN (eds.) (1983). *Proceedings of the Second International Workshop on Stochastic Geometry and Stereology,* Memoir No. 6, Department of Theoretical Statistics, University of Aarhus.
9. E.F. HARDING, D.G. KENDALL (eds.) (1973). *Stochastic Geometry: a Tribute to the Memory of Rollo Davidson,* J. Wiley and Sons, Chichester.
10. C.C. HEYDE. On some new probabilistic developments of significance to statistics: martingales, long range dependence, fractals and random fields. To appear.
11. B. JULESZ (1975). Experiments in the visual perception of texture. *Scientific American 232, 4,* 34-43.
12. O. KALLENBERG (1983). The invariance problem for stationary line and flat processes. In *[8],* 105-114.
13. D.G. KENDALL (1974). Foundations of a theory of random sets. In *[9],* 322-376.
14. M.G. KENDALL, P.A.P. MORAN (1963). *Geometrical Probability,* Statist. Monographs and Courses No. 10, Griffin, London.
15. B.B. MANDELBROT (1976). *Fractals: Form, Chance and Dimension,* W.H. Freeman, San Francisco.
16. B.B. MANDELBROT (1982). *The Fractal Geometry of Nature,* W.H. Freeman, San Francisco.
17. G. MATHERON (1967). *Eléments pour une Théorie des Milieux Poreux,* Masson, Paris.
18. G. MATHERON (1974). *Random Sets and Integral Geometry,* J. Wiley and Sons, New York.
19. K. MATTHES, J. KERSTAN, J. MECKE (1978). *Infinitely Divisible Point Processes,* J. Wiley and Sons, New York.
20. E. MICHAEL (1951). Topologies on spaces of subsets. *Trans. Amer. Math. Soc. 71,* 152-182.

21. R.E. MILES (1970). On the homogeneous planar Poisson process. *Math. Biosci. 6,* 85-127.
22. R.E. MILES (1972). The random division of space. *Suppl. Advan. Appl. Prob. 4,* 243-266.
23. R.E. MILES (1973). The various aggregates of random polygons determined by random lines in a plane. *Advan. Math. 10,* 256-290.
24. R.E. MILES (1981). A survey of geometric probability in the plane, with emphasis on stochastic image modeling. In *[23],* 277-300.
25. R.E. MILES, J. SERRA (eds.) (1978). *Geometrical Probability and Biological Structures: Buffon's 200th Anniversary,* Lecture Notes in Biomathematics, No. 23, Springer, Heidelberg.
26. F. PAPANGELOU (1974). The conditional intensity of general point processes and an application to line processes. *Z. Wahrscheinlichkeitstheorie 28,* 207-226.
27. B.D. RIPLEY (1976). Locally finite random sets: foundations for point process theory. *Annals of Probability 45,* 983-994.
28. B.D. RIPLEY (1981). *Spatial Statistics,* J. Wiley and Sons, New York.
29. A. ROSENFELD (Ed.) (1981). *Image Modeling,* Academic Press, New York. *Note:* largely reprinted in *Computer Graphics and Image Processing,* Vol. 12.
30. L.A. SANTALÓ (1976). *Integral Geometry and Geometric Probability,* Addison-Wesley, New York.
31. J. SERRA (1982). *Image Analysis and Mathematical Morphology,* Academic Press, New York.
32. D. STOYAN (1979). Applied stochastic geometry: a survey. *Biom. Journal 21,* 693-715.
33. D. STOYAN, J. MECKE (1983). *Stochastische Geometrie,* Wissenschaftlicher Taschenbücher Bd. 275, Akademie-Verlag, Berlin (DDR).
34. E.R. WEIBEL (1979). *Stereological Methods, Vol. 1,* Academic Press, New York.
35. W. WEIL (1983). Stereology: a survey for geometers. In: *Convexity and Its Applications,* Birkhäuser, Basel, edited by P. GRUBER and J.M. WILLS.
36. W. WEIL (1982). An application of the central limit theorem for Banach space valued random variables to the theory of random sets. *Z. Wahrscheinlichkeitstheorie 60,* 203-208.

Systematic Program Development

C.B. Jones
Department of Computer Science
University of Manchester

The so-called 'Vienna Development Method' includes techniques for specifying programs and for justifying a design with respect to a specification. The key points of this development method are described and contrasted to some other approaches. It is argued that such development methods should be guided by intuition from computer science.

1. Introduction

This paper provides an overview of one method which can be used to systematically develop computer programs. The so-called 'Vienna Development Method' (*VDM*), on which the paper is based, is described in a number of publications (e.g. [6], [7], [16]). *VDM* uses mathematical notation in order to be unambiguous but, at least in its applications, does not use particularly deep mathematical results.

Some of the techniques used in *VDM* deviate from those in wide use (the 'accepted' approach). Such heresies are, in the opinion of the originators of *VDM*, the result of relying strongly on intuition gained from the application of the method to non-trivial problems.

The overall concern is how to create programs which are (known to be) correct with respect to their specifications. The idea of proving programs correct is now widely discussed. To support a formal proof, a specification must be recorded in a formal notation. *Post-facto* proofs of the correctness of large programs are likely to be unattainable. It is now widely accepted that correctness criteria are needed for individual design steps: thus it is the program design which is justified rather than (just) the finished code. Using formal concepts in this way makes it possible to relax proofs to the level of 'rigour' used in most mathematical arguments. The knowledge of what and how to formalize ensures that sloppiness will not arise. The aim is to record program development in a formalizable notation.

One reason for concentrating on justification during design is the large impact which errors made early in the development cycle have on the productivity of programmers. Much more emphasis is put here on data structures and their refinement than on programming language control constructs. This is precisely because experience suggests that data structure decisions occur in the

early stages of design.

This paper focuses on the formal aspects of program development: there are clearly many considerations which are not going to be helped by such formal methods. In order to avoid misunderstanding, it will be best to consider the development method as providing a structure in which a design can be documented. How the original specification is agreed, how the design ideas are generated, or how the designers are managed, is not intended to be constrained.

2. Data types

There are a lot of publications concerned with the specification of data types and these often confuse, rather than answer, the question 'what is a data type'? An answer is given below in terms of behaviours but, before making this precise, it is necessary to clarify the scope of the intuitive idea as seen from *VDM* experience.

When considering data types, the natural mathematical comparison is with something like natural numbers: a set of values and a collection of total functions. With care, the functions (operators) can be defined without mentioning the values. Data types such as lists (sequences) fit this mould fairly well - although it becomes much harder to ignore the partial functions such as those yielding head and tail. Just as with integers, it is natural to build up complex list expressions which denote list values; list values can be compared; and, in a procedural language, list values can be assigned to variables. A data type like stack is different. The operators might be called 'operations' in order to recognize that they change the state of the stack by 'side effect'. This state can be an important aid in writing a specification. Since the result of some stack operations depends on the history of other operations which have been applied, the state defines equivalence classes of such histories. Furthermore, comparison and assignment of stack values might not be considered desirable. Finally, with many such data types, the problem of partial and even non-deterministic operators cannot be avoided.

It is, of course, possible to present data types, even within the many complications, in the mould of more mathematical objects. One must, however, be careful when forcing things into artificial moulds. With stacks it is natural to think of the POP operation both returning a value and having a side-effect on the state; it is doubtful whether the separation into two functions each yielding one result contributes to understanding. Where non-determinism is involved, this separation is even more questionable.

The behaviours (cf. [13], [23], [24]), with which it is intended to explain data types, must recognize the side-effects as well as partial and non-deterministic operations. The appropriate model would appear to be on the one hand a set of terms, built up from operations which do not yield error; and on the other hand, a definition of the meaning of any such term given as a relation on its inputs and outputs.

The *specification* of data types is an area where *VDM* deviates from the most widely accepted approach - that is the property oriented (algebraic, axiomatic)

style. 'Algebraic presentations' appear well-suited for basic data types like lists. Their attraction would appear to have more to do with mathematical tractability than with their ability to handle the full range of data types needed in computing science.

There is an interesting question concerning the interpretation of equations. KAMIN [20], BOTHE [8] discuss the 'final' interpretation. In some sense, the set of generating operators of a data type can be seen to create a model on which the other operators are defined. The question can then be asked whether this is the most convenient model.

VDM takes a model oriented approach to describing data types. Thus, if a relational database system is to be specified, the state might be modelled using (among other things) a mapping from relation names to sets of tuples.

The problem of representing mappings from keys to data occurs over and again in computer systems. Unlike the database example, this could easily be specified via its properties - it will however serve as an example which is small enough to be carried through to its implementation. In the model oriented specification given below, pre-conditions are used to indicate the partial operations and, in general, non-determinism is handled by (relational) post-conditions.

3. SPECIFICATION

The examples given in this paper use the *VDM* notation. In most cases, this should have obvious meaning, a full description can be found in the references.

The top-level specification of the mapping from Keys to Data is made trivial by the availability of a suitable class of objects in *VDM*. Thus:

$$\text{Mpk} = \textbf{map}\ \text{Key}\ \textbf{to}\ \text{Data}$$

The initial object in Mpk is:

$$m_0 = []$$

The operations can be defined:

INSERT(K: Key, D: Data)

ext wr M: Mpk

pre $k \notin$ **dom** m

post $m = \overleftarrow{m} \cup [k \mapsto d]$

FIND(K: Key) D: Data

ext rd M: Mpk

pre $k \in$ **dom** m

post $d = \overleftarrow{m}(k)$

DELETE(K: Key)

ext wr M: Mpk

pre $k \in$ **dom** m

post $m = \overleftarrow{m} \setminus \{k\}$

(The 'hooked' variables in post-conditions denote the value of the corresponding variable at the beginning of the operation; undecorated lower case names refer to the value of corresponding (upper case) variables at the point where the assertion is relevant.)

Each of these operations has been specified to be partial in order to illustrate the proof rules below. Although the operations are deterministic, relational post-conditions have been written. The experience with *VDM* suggests that non-determinism often arises during design: even with this small example, this point is illustrated below. Notice that selective access to the state is given by means of **rd/wr** externals.

The most serious objection raised against model oriented specifications is that they might overspecify a system. This notion of 'bias' has been made precise in earlier papers and a test has been devised to check that overspecification is avoided:

> A model specification is based on some set of objects. The model is biased (with respect to some given set of operations) if there exist different elements of the set of objects which cannot be distinguished by any of the operations (i.e. have the same behaviour).

Thus bias can be seen as preserving a part of the history (of an object) which cannot be detected by the operations: the equivalence class induced by the state is not coarse enough. (A representation which passes the bias test might contain redundant information such as duplication. The test only rules out non-unique representations.)

In practice, this test has uncovered few examples of bias in existing specifications. It is important to recall that the notion is defined with respect to a set of operations. In [16] it is shown how deleting the operations available for stacks dictates simplifications to the model based solely on this bias test. (One tantalizing example where the removal of bias might be interesting is the use of locations in the environments of denotational semantics).

In some cases, bias can be eliminated by the use of data type invariants. But invariants play a larger part in discussing representations (cf. Binnode below).

4. Representation

This section is concerned with showing that the behaviour of one data type models the behaviour of another. Normally, a representation (of an abstraction) is chosen because it is closer (than the abstraction) to the final implementation or because it can be manipulated efficiently. But the new representation is just a data type (cf. Bintree below) and is specified in the way described above. The problem of proofs about partial functions leads to a digression in this section.

The mappings (Mpk) can be represented by binary trees:

Bintree = [Binnode]

Binnode :: Bintree Key Data Bintree

where

invBinnode(mk-Binnode(lt,mk,md,rt)) $\hat{=}$

$$(\forall\ lk \in \text{collkeys}(lt).\ lk < \text{mk}) \wedge (\forall rk \in \text{collkeys}(rt).\ \text{mk} < rk)$$

Initial (Bintree) object: $t_0 = \textbf{nil}$

The set of objects satisfying the definition written with '::' is defined as:

$$\text{Binnode} = \{\text{mk}-\text{Binnode}(lt,mk,md,rt) \mid lt,rt \in \text{Bintree} \wedge mk \in \text{Key} \wedge md \in \text{Data}\}$$

The use of the constructor function (mk-Binnode) as the parameter of the data type invariant (or below in **cases**) provides a way of naming the values of the sub-fields of a constructed object.

The definition of the function which collects the keys is:

collkeys: Bintree $\rightarrow$ **set of** Key

collkeys(t)$\hat{=}$

cases t:

nil $\rightarrow$ {}

mk-Binnode(lt,mk,md,rt) $\rightarrow$ collkeys(lt) $\cup$ {mk} $\cup$ collkeys(rt)

From here on 'Binnode' (and thus 'Bintree') is taken to be the set of objects which satisfy the invariant: these are called the 'valid' objects.

Notice that Bintree is biased with respect to the operations of Mpk: there

are different ways of arranging the tree which, viewed as a mapping, are indistinguishable. Thus Bintree would not be used in the (abstract) specification; as a step of refinement, however, it is quite acceptable.

Following [15] it is now possible to build a 'theory' of the Bintree data type. Firstly, two lemmas are stated without proof:

LEMMA(collkeys1). $\forall t \in$ Bintree . collkeys(t) $\in$ **set of** Key

LEMMA(collkeys2). $\forall nd \in$ Binnode .

(**let** mk-Binnode(lt,mk,md,rt) = nd **in**

is-disj(collkeys(lt),{mk}) $\wedge$

is-disj(collkeys(lt),collkeys(rt)) $\wedge$

is-disj({mk},collkeys(rt)) $\wedge$

($\forall k \in$ collkeys(nd) .

($k < mk \Rightarrow k \in$collkeys(lt))$\wedge$($mk < k \Rightarrow k \in$collkeys(rt))))

The 'collkeys1' lemma claims that the function is total; 'collkeys2' defines some obvious results which are used in proofs below.

The next step in developing the Bintree theory is to define some functions; to locate a key in a tree:

findb(K: Key, T: Bintree) D: Data

pre $k \in$ collkeys(t)

findb(k,mk-Binnode(lt,mk,md,rt)) $\hat{=}$

if $k = mk$ **then** md

else if $k < mk$ **then** findb(lt,k)

else findb(rt,k)

Notice that the pre-condition shows that the set of keys is non-empty and therefore the (recursive) definition of findb can be written assuming that the tree is not equal to **nil**.

This function is not total, the pre-condition shows an inter-relation between the arguments which cannot be expressed by limiting either set. (It would, of course, be possible to shift the problem by defining a subset of the cross

product of the two sets via set comprehension.) Thus the lemma on the findb function is:

LEMMA(findb1). The function findb is total (w.r.t. its pre-condition) on valid Bintrees.

$$\forall k \in \text{Key}, t \in \text{Bintree} \,.\, k \in \text{collkeys}(t) \Rightarrow \text{findb}(k,t) \in \text{Data}$$

This logical expression manifests a problem which has to be faced by specification languages. Since findb is partial, the consequent of the implication can be undefined when the antecedent is false. Conventional ('two-valued') logic does not cope with this problem. The most common 'solution' is to define extra conditional logical operators. Objections to this approach and an alternative solution are presented in [3]. The proof theory for that system is given in Appendix I. The system has a number of properties worth mentioning here:

a) the and/or operators are commutative;
b) the 'law of the excluded middle' does *not* hold;
c) the deduction theorem does not hold (without additional assumptions).

Perhaps the most important property of the system is what it can *not* prove. It is not possible to show that:

$$x/0 = 1 \vee x/0 \neq 1$$

But it is possible to derive:

$$x = 0 \vee x/x = 1$$

which is equivalent to:

$$x/x = 1 \vee x = 0$$

Given that the proof system is deliberately weaker, it is not surprising that some proofs are more difficult than in normal logic. Thus in [14] a short proof is given of:

$$\frac{(E1 \vee E2) \wedge (E1 \vee E3)}{E1 \vee (E2 \wedge E3)}$$

by using the 'law of the excluded middle'. In the logic of partial functions the following natural deduction style proof is required:

	from $(E1 \vee E2) \wedge (E1 \vee E3)$	
1	$E1 \vee E2$	$\wedge$ -*E* ,pr
2	$E1 \vee E3$	$\wedge$ -*E*, pr
3	**from** $E1$	
	infer $E1 \vee E2 \wedge E3$	$\vee$ -*I*,pr3
4	**from** $E2$	
4.1	**from** $E3$	
4.1.1	$E2 \wedge E3$	$\wedge$ -*I*, pr4,pr4.1
	infer $E1 \vee E2 \wedge E3$	$\vee$ -*I*,4.1.1
	infer $E1 \vee E2 \wedge E3$	$\vee$ -*E*,2,3,4.1
	infer $E1 \vee E2 \wedge E3$	$\vee$ -*E*,1,3,4

This same proof style can be used to prove lemma 'findb1'. The proof uses structural induction. The induction rule for Bintree can be stated:

p (**nil**),

$t = \text{mk-Binnode}(lt, mk, md, rt),\ \text{invBinnode}(t), p(lt), p(rt) \vdash p(t)$

$t \in \text{Bintree} \vdash p(t)$

The proof uses the abbreviation:

$p(k,t)$:

$k \notin \text{collkeys}(t) \vee \text{findb}(k,t) \in \text{Data}$

A formal proof is:

from $t \in$ Bintree, $k \in$ Key

1	$k \notin$ collkeys (**nil**)	collkeys, **set**
2	$p\,(k,$**nil**$)$	$\vee$ -*I*,1,p
3	**from** $t =$ mk-Binnode$(lt,mk,md,rt) \wedge$ invBinnode$(t) \wedge p\,(k,lt) \wedge p\,(k,rt)$	
3.1	$k \in$ collkeys$(t) \vee k \notin$ collkeys(t)	collekeysl,**set**
3.2	**from** $k \notin$ collkeys(t)	
	infer $p\,(k,t)$	$\vee$ -*I*,pr3.2,p
3.3	**from** $k \in$ collkeys(t)	
3.3.1	$k < mk \vee k = mk \vee mk < k$	collkeys1,pr3.3,Key
3.3.2	**from** $k = mk$	
3.3.2.1	findb$(k,t) = md$	findb,pr3.3.2
	infer findb$(k,t) \in$ Data	
3.3.3	**from** $k < mk$	
3.3.3.1	$k \in$ collkeys(lt)	collkeys2, pr3.3,pr3.3.3
3.3.3.2	findb$(k,lt) \in$ Data	pr3,p,-1
3.3.3.3	findb(k,t) = findb(k,lt)	findb,pr3.3.3
	infer findb$(k,t) \in$ Data	-2,-1
3.3.4	**from** $mk < k$ — similar —	
	infer findb$(k,t) \in$ Data	
3.3.5	findb$(k,t) \in$ Data	$\vee$ -*E*,3.3.1, 3.3.2,3.3.3,3.3.4
	infer $p\,(k,t)$	$\vee$ -*I*,-1,p
	infer $p\,(k,t)$	$\vee$ -*E*,3.1,3.2,3.3
	infer $k \notin$ collkeys$(t) \vee$ findb$(k,t) \in$ Data	indn,2,3,p

One of the nice features of such natural deduction proofs is that they can be used in a less-than-formal way by considering the outer levels of the proof.

Similar considerations arise with the function for inserting values into trees — the proof is more interesting because of the need to strengthen the statement of the lemma in order to obtain an induction hypothesis which carries through.

insb(K: Key, D: Data, T: Bintree) R: Bintree

pre $k \notin$ collkeys(t)

insb(k,d,t)$\hat{=}$

cases t:

nil $\rightarrow$ mk-Binnode(**nil**,k,d,**nil**)

mk-Binnode(lt,mk,md,rt) $\rightarrow$

if $k < mk$ **then** mk-Binnode(insb(k,d,lt),mk,md,rt)

else mk-Binnode(lt,mk,md,insb(k,d,rt))

LEMMA(insb1). The function insb is total (w.r.t. its pre-condition) on (valid) Bintrees; it preserves the invariant:

$\forall t \in$ Bintree, $k \in$ Key, $d \in$ Data .

$k \in$ collkeys (t) $\vee$ insb(k,d,t)$\in$ Bintree

Proof by structural induction

abbreviation $p(k,t)$:

$k \in$ collkeys(t) $\vee$

insb(k,d,t) $\in$ Bintree $\wedge$

collkeys(insb(k,d,t)) = collkeys(t) $\cup$ $\{k\}$

	from $t \in$ Bintree, $k \in$ Key, $d \in$ Data	
1	insb(k,d,**nil**) = mk-Binnode(**nil**,k,d,**nil**)	insb
2	invBinnode(insb(k,d,**nil**))	1,inv
3	collkeys(insb(k,d,**nil**)) = $\{k\}$	1,collkeys
4	= collkeys(**nil**) $\cup$ $\{k\}$	3,collkeys
5	p(k,**nil**)	$\wedge$ -I,2,4, $\vee$ -I,p
6	**from** t = mk-Binnode(lt,mk,md,rt) $\vee$ invBinnode(t)$\wedge p(k,lt)\wedge p(k,rt)$	
6.1	$k \in$ collkeys(t) $\vee$ $k \notin$ collkeys(t)	collkeys1,**set**
6.2	**from** $k \in$ collkeys(t)	
	infer $p(k,t)$	$\vee$ - I,pr6.2,p
6.3	**from** $k \notin$ collkeys(t)	
6.3.1	collkeys(t) = collkeys(lt)$\cup\{mk\}\cup$ collkeys(rt)	pr6,collkeys
6.3.2	$k<mk \vee mk<k$	-1,pr6,3,Key
6.3.3	**from** $k<mk$	
6.3.3.1	$k \notin$ collkeys(lt)	pr6.3,6.3.1
6.3.3.2	invBinnode(insb(k,d,lt))	-1,p,pr6
6.3.3.3	collkeys(insb(k,d,lt)) = collkeys(lt)$\cup\{k\}$	-2,p,pr6
6.3.3.4	insb(k,d,t) = mk-Binnode(insb(k,d,lt),mk,md,rt)	insb,pr6.3.3
	infer $p(k,t)$	$\wedge$ - I,-1,-2, $\vee$ -I,p
6.3.4	**from** $mk<k$ — similar — **infer** $p(k,t)$	
	infer $p(k,t)$	$\vee$ - I,6.3.2,6.3.3,6.3.4
	infer $p(k,t)$	$\vee$ - E,6.1,6.2,6.3
	infer $p(k,t)$	indn,5,6

The function delb is harder to write but it and its proof could be tackled by enthusiastic readers.

With this theory of Bintrees, the actual proofs relating to refinement are all trivial. The task is to show that a series of operations on Bintree 'model' those on the abstract mapping (Mpk) data type. The key to these proofs is to relate the underlying objects by a function which retrieves the abstraction from the representation. In this case:

retrm: Bintree $\rightarrow$ Mpk

retrm(t) $\hat{=}$ **cases** t:

nil $\rightarrow$ []

mk-Binnode(lt, mk, md, rt) $\rightarrow$

merge([$k \mapsto d$], retrm(lt), retrm(rt))

The reason for the choice of direction of this function is precisely because the increase in bias results in many representatives corresponding to one abstraction.

One requirement on retrieve functions is that they be total (on valid representations). This follows here since the invariant guarantees that the domains of the mappings to be merged are disjoint. Another property required of the representation and its retrieve function is that the representation be 'adequate':

$\forall m \in$ Mpk . $\exists t \in$ Bintree . retrm(t) $= m$

Intuitively, this requires that there must be at least one representation for each abstract state. A proof can be performed by induction on the domain of the mapping — since only existence is required, a completely imbalanced tree can be generated.

Also note:

$\forall t \in$ Bintree . **dom**(retrm(t)) = collkeys(t)

This ensures that the domains of the following functions are 'large enough'.

The relation can now be seen to preserve:

retrm(t_0) $= m_0$

$\forall t \in$ Bintree, $k \in$ Key, $d \in$ Data .

$$k \notin \text{collkeys}(t) \Rightarrow \text{retrm}(\text{insb}(k,d,t)) = \text{retrm}(t) \cup [k \mapsto d]$$

$\forall t \in$ Bintree, $k \in$ Key .

$$k \in \text{collkeys}(t) \Rightarrow \text{findb}(k,t) = (\text{retrm}(t))(k)$$

The INSERT operation on Bintrees can be defined:

INSERTB(K: Key, D: Data)

ext wr T: Bintree

pre $k \notin$ collkeys(t)

post t = insb(k,d,$\overleftarrow{t}$)

The general form of the rule to show that operations 'preserve invariants' (valid results exist) is:

$$\forall \overleftarrow{\sigma} \in \Sigma \text{ . } \text{preOP}(\overleftarrow{\sigma}) \Rightarrow \exists \sigma \in \Sigma \text{ . } \text{postOP}(\overleftarrow{\sigma}, \sigma)$$

In this case, the result follows from lemma 'insb1'.

There are two rules which show that an operation on a representation models one on an abstraction. The domain rule ensures that the pre-condition is not too restrictive:

$$\forall t \in \text{Bintree} \text{ . } \text{preOP}(\text{retrm}(t)) \Rightarrow \text{preOPB}(t)$$

The range rule ensures that no contradictory results can arise:

$$\forall \overleftarrow{t},\ t \in \text{Bintree} \text{ . } \text{preOP}(\text{retrm}(t)) \wedge \text{postOPB}(\overleftarrow{t}, t) \Rightarrow \text{postOP}(\text{retrm}(\overleftarrow{t}),\ \text{retrm}(t))$$

Here, both results are immediate consequences of the theory above. Indeed, the form of the rules might appear unnecessarily general. It is worth remembering that both operations can be non-deterministic and that the operation on the representation should be allowed to have a larger-than-required pre-condition. To illustrate the former point, notice that the post-condition of INSERTB could be:

post $\text{retrm}(t) = \text{retrm}(\overleftarrow{t}) \cup [k \mapsto d]$

This is highly non-deterministic and could, for example, cover tree balancing.

For the model of the FIND operation, one might use:

FINDB(K: Key) D: Data

ext rd T: Bintree

pre $k \in$ collkeys(t)

post d = findb(k,$\overleftarrow{t}$)

Here the invariant is preserved since there is no state change and, for the same reason, the modelling proofs are simpler.

In order to illustrate the way in which design steps can be isolated by the proofs, a sketch is given of a second step of data refinement. The trees used above cannot be constructed directly in a language like Pascal. Instead each node must be created as a record on the heap and nested trees must be represented by pointers. Thus:

Heap = **map** Ptr **to** Binnoderep

Root = [Ptr]

Binnoderep:: *LP*: [Ptr]

MK: Key

MD: Data

RP: [Ptr]

Without formalizing the statement, it is clear that the mapping should be well-founded and that all used keys should be in the domain of the map.

The remainder of the refinement step can be seen from:

collkeysh: Root x Heap $\rightarrow$ **set of** Key

collkeysh(p,m)$\hat{=}$

if p = **nil then** {}

else(let mk-Binnoderep(lp,mk,md,rp)=$m(p)$ **in**

collkeysh(lp,m)$\cup\{mk\}\cup$ collkeysh(rp,m))

findbhn: Key x Ptr x Heap $\rightarrow$ Binnoderep

pre $k \in$ collkeysh(p,m)

findbhn(k,p,m)$\hat{=}$

let mk-Binnoderep(lp, mk, md, rp) = $m(p)$ **in**

if k = mk **then** $m(p)$

else if k < mk **then** finbhn(k, lp, m)

else findbhn(k, rp, m)

FINDBH(K: Key) D: Data

ext rd RT: Root

rd HP: Heap

pre $k \in$ collkeysh(rt, hp)

post $d = MD(\text{findbhn}(k, \overleftarrow{rt}, \overleftarrow{hp}))$

The point about isolation of development steps can now be made in two directions. The argument that the Heap representation is valid with respect to the Bintree does not rely on the earlier argument that Bintree is a valid representation for the map Mpk. Furthermore, the development, in the next section, of code to match the specifications on Heap is insulated from understanding Bintree.

5. Decomposition

A specification of a system should be abstract both in the data types it uses and in the fact that post-conditions make it possible to define (non-constructively) what result is required. The preceding section shows how the data can be refined to the point where it can be represented in the programming language. However, the post-conditions must eventually be realized by (i.e. decomposed into) sequences of statements. This section shows how such decomposition can be supported by correctness arguments.

This area is, of course, widely discussed in the literature and textbooks like [14] tend to focus on this topic. (Even [16] makes the mistake of covering it before data refinement.) But, here again, *VDM* falls into heresy: the well-known Hoare-logic (cf. [2]) cannot be used because of *VDM*'s reliance on relational post-conditions. Peter Aczel (in [1]) writes:

'For example, the program

$$\textbf{while}(y+1)^2 \leqslant x \textbf{ do } y := y+1$$

meets the specification having precondition $y=0$ & $x \geqslant 0$ and postcondition $y^2 \leqslant x < (y+1)^2$.
It is a familiar fact that this specification does not explicitly express all that we have in mind. For example if the above program is prefixed by $x:=0$; the resulting program will still formally meet the specification, although the implicit understanding that x is supposed to remain fixed has been violated. One natural convention is to make this understanding explicit by using special symbols for variables that are to remain fixed throughout a computation. But a more flexible and powerful approach ... is to allow the postcondition of a specification to depend on the starting state of a computation. So, in our example the postcondition should be

$$(y^2 \leqslant x < (y+1)^2) \quad (x = \overleftarrow{x})$$

where we use $\overleftarrow{x}$ to denote the value of x at the start of the computation. In his book, Cliff Jones presents some rules for proving the total correctness of programs for his notion of specification. His rules appear elaborate and immemorable compared with the original rules for partial correctness of Hoare. Moreover, they are not complete.'

In short, [16] had the right idea but used awful notation! Fortunately PETER ACZEL suggested a far better way to present the proof rules (cf. Appendix III).

The most important of the proof rules is that for the repetitive construct. PETER ACZEL's presentation of this rule is:

$$\frac{\{P \wedge B\} S \{P \wedge R\}}{\{P\} \textbf{while } B \textbf{ do } S \{P \wedge R^r \wedge \sim B\}}$$

If all mention of R is suppressed, this is identical to the Hoare rule. However, R is very important - it is a relational (i.e. two state) predicate which captures input/output behaviour. If R is transitive and well-founded, and R^r is its reflexive closure, then this rule captures total correctness for relational post-conditions. It is interesting to contrast the use of R with the 'variant' in [12]: it can be seen that here, as well as providing a termination proof, R is used in the correctness argument.

These proof rules are used on a number of examples in [18]; again, an important property is that a proof at one stage provides specifications which completely isolate the justification of the next stage. (It is, in fact, the inbedding of specified operations within a program construct which requires that the rules are capable of handling non-determinism. In [17] proof rules are justified with respect to a relational denotational semantics; two semantic functions are given — one for the relational meaning and one for the termination set; a

satisfaction (*sat*) ordering is defined between such semantic objects; the principal programming language constructs are shown to be monotone in the *sat* ordering.)

Here, the rules are used to justify annotated programs. This goes back to the style in [21] but shows how to incorporate relational post-conditions. Furthermore, a style has been adopted which emphasizes the link to the natural deduction proofs shown above.

For the task of multiplying two numbers (using successive addition), the top-level development might be:

ext $I,J,M,:Z$
|
pre **true**
|
| **pre** **true**
| |
| | **if** $I<0$ **then**
| | |
| | **pre** **true**
| | |
| | $I,J:=-I,-J$
| | |
| | **post** $i\geqslant 0\wedge i\star j=\overleftarrow{i}\star\overleftarrow{j}$
| |
| **post** $i\geqslant 0\wedge i\star j=\overleftarrow{i}\star\overleftarrow{j}$
| ;
| **pre** $i\geqslant 0$
| — see below —
| **post** $m=\overleftarrow{i}\star\overleftarrow{j}$
|
post $m=\overleftarrow{i}\star\overleftarrow{j}$

This could be formally justified using the conditional and composition rules.

The actual multiplication step can be achieved by:

```
ext wr I,J,M : Z
|
pre i ≥ 0
M := 0
;
pre i ≥ 0
|
|   while I ≠ 0 do
|   |
|   inv i ≥ 0
|   |
|   |
|   rel m + i*j = m~ + i~*j~ ∧ i ≤ i~
|   |
|   |   extr wr I,J : Z
|   |   |
|   |   pre i ≠ 0
|   |   |
|   |   |   while is-even(I) do
|   |   |   |
|   |   |   inv i ≥ 1
|   |   |   |
|   |   |   |
|   |   |   rel i*j = i~*j~ ∧ i ≤ i~
|   |   |   |
|   |   |   |       I,J := I/2, J*2
|   |   |   |
|   |   post i*j = i~*j~ ∧ i ≤ i~
|   |   |
|   |   ;
|   |   |
|   |   M,I := M + J, I - 1
|   |
post m = m~ + i~*j~
|
post m = i~*j~
```

This could be formally justified by (two applications) of the while rule. Notice how the inner specification (preserving $m + i \star j$) naturally introduces non-determinism in a deterministic program: this inner specification can be realized

by the final multiple assignment yielding a linear algorithm; the same specification is satisfied by the $\log_2 i$ algorithm.

Clearly this presentation says little about discovering invariants (cf. [14]) — indeed, it would appear that there is more work to be done since both **inv** and **rel** must be found. Experience so far suggests that there are natural ways of seeking the required predicates but this is not the place to pursue this topic.

As an illustration of a more interesting problem, consider the task (suggested by TONY HOARE) of describing how a hand calculator performs division of I by J — result in Q leaving remainder I. In a first stage (SL) J is shifted left until it is larger than I — the number of shifts is counted in N. The second stage (SR) shifts J back and at each step keeps the expression $J \star Q + I$ constant. There are two places this must be done: shifting at SRS and re-establishing $i<j$ by stepping down I at SRC. The following presentation is made simpler by assuming that all variables are natural numbers.

pre $j \neq 0$

ext I: **rd** J, Q, N: **wr**

$Q := 0;\ N := 0;$

while $J \leqslant I$ **do**

inv

rel $j \star 10^{\overleftarrow{n}} = \overleftarrow{j} \star 10^{n}$

$J, N := J \star 10, N + 1$

post $j = \overleftarrow{j} \star 10^{n} \wedge i < j \wedge q = 0$ *SL*

;

pre 10^{n} **div** $j \wedge i < j$ *(SR)*

while $N \neq 0$ **do**

inv 10^{n} **div** $j \wedge i < j$

rel $j / 10^{n} = \overleftarrow{j} / 10^{\overleftarrow{n}} \wedge j \star q + i = \overleftarrow{j} \star \overleftarrow{q} + \overleftarrow{i} \wedge n < \overleftarrow{n}$

$N, J, Q := N - 1, J / 10, Q \star 10;$ *SRS*

while $J \leqslant I$ **do** *SRS*

ext J: **rd** I, Q: **wr**

inv $(0 \leqslant i)$

rel $j \star q + i = \overleftarrow{j} \star \overleftarrow{q} + \overleftarrow{i} \wedge i < \overleftarrow{i}$

$I, Q := I - J, Q + 1$

post $j = \overleftarrow{j} / 10^{n} \wedge j \star q + i = \overleftarrow{j} \star \overleftarrow{q} + \overleftarrow{i} \wedge i < j$

post $\overleftarrow{j} \star q + i = \overleftarrow{i} \wedge i < \overleftarrow{j}$

As a final example of a decomposition proof, the binary tree problem is picked up from the preceding section. The Pascal equivalents of the data objects are:

```
type Ptr = ↑ Binoderep

Binnoderep = record

             LP: Ptr

             MK: Key

             MD: Data

             RP: Ptr

             end

ROOT: Ptr
```

The FINDB function can now be coded as follows:

function FINDBH(*K*: Key) Data

{**ext rd** *RT*: Ptr

rd *HP*: Heap}

{**pre** $k \in$ collkeysh(rt)}

var *P*: Ptr;

begin

$P := RT$

;

{**pre** $k \in$ collkeysh(p)}

while $K \neq P\uparrow MK$ **do**

{**inv** $k \in$ collkeysh(p)}

{**rel** findhn(k,p) = findbhn(k,$\overleftarrow{p}$) $\wedge$ depth(p) $<$ depth($\overleftarrow{p}$)}

with $P \uparrow$ **do**

if $MK < K$ **then** $P := LP$

else $P := RP$;

{**post** p = findbhn(k,$\overleftarrow{p}$)}

;

FINDBH := $P\uparrow MD$

{**pos** d = MD(findbhn(k,$\overleftarrow{rt}$))}

end

6. Conclusions

The material in [16] and [7] focusses on different parts of *VDM*; this paper brings in ideas like the logic of partial functions which are not mentioned in either book; perhaps the time has come to answer the question 'What is *VDM*?' The only reasonable answer would appear to be that it is a specification technique which is formal enough to support implementation justification. Specifications are written at different levels of design and rules indicate the proof obligation required to relate any level to the preceding level. One important characteristic of *VDM* is that the choice of techniques is motivated by computing science and not (just) mathematical elegance. In some cases (the proof rules in the appendix) the story has a happy ending in that elegance is married to utility.

There are a number of topics which need further research. New proof rules (e.g. for recursive procedures) must be developed in the relational post-condition style. The proofs here have all assumed that parameters are passed by value; the obvious (recursive) procedure for INSERTBH needs to share a variable parameter with part of the tree. Proof rules for more powerful parameter passing will require care (cf. [22]). The specification of exception conditions is a topic which has received some attention (e.g. [9], [11], [10]) — one of the difficulties is to remain relatively language independent. The exit mechanism of the *VDM* metalanguage might be a suitable procedural construct.

The need, in a development method, for the isolation property in a development method is referred to above. To achieve this for parallel, interfering, processes is difficult — some first steps are described in [18], [19]; similar ideas are being pursued in a temporal logic setting by Howard Barringer, Amir Pnueli and Ruurd Kuiper ([4], [5]).

In the area of data types, model oriented specifications need a proper basis for parameterized data types and their refinement.

Acknowledgements

My debt to Peter Aczel should be clear from the preceding discussion. The binary tree example was first developed by Elizabeth Fielding. The work here is supported by SERC research grants and has benefitted from discussions at meetings of IFIP WG 2.3. My thanks are due to Julie Hibbs for coping with another of my bizarre manuscripts.

REFERENCES

1. P. ACZEL (1982). *A Note on Program Verification* (manuscript).
2. K.R. APT (1981). Ten years of Hoare's logic: a survey - part 1. *ACM Trans. Program. Lang. Syst. 3,* no 4.
3. H. BARRINGER, J.H. CHENG, C.B. JONES (1983). A logic covering undefinedness in program proofs. Accepted for publication in *Acta Informatica.*
4. H. BARRINGER, R. KUIPER (1983). Towards the hierarchical, temporal logic, specification of concurrent systems. *Proceedings of the STL/SERC 'Workshop on The Analysis of Concurrent Systems',* Cambridge.
5. H. BARRINGER, R. KUIPER, A. PNUELI (1984). Now you may compose temporal logic specifications. *Proceedings of the 16th ACM Symposium on the Theory of Computing,* Washington.
6. D. BJORNER (1981). The *VDM* principles of software specification and program design. *Formalization of Programming Concepts,* Springer-Verlag LNCS 107, 44-74.
7. D. BJORNER, C.B. JONES (1982). *Formal Specification and Software Development,* Prentice-Hall International.
8. K. BOTHE (1981). A comparative study of abstract data type concepts. *Elektronische Informationsverarbeitung und Kybernetik, EIK 17,* 4/6, 237-257.
9. C. BRON, M.M. FOKKINGA, A.C.M. DE HAAS (1976). *A Proposal for Dealing with Abnormal Termination of Programs,* Memo no. 150, TH Twente.
10. I.D. COTTAM (1984). The rigorous development of a system version control program. *IEEE Trans. on Software Engineering SE-10,* no 2, 143-154.
11. F. CRISTIAN (1984). Correct and robust programs. *IEEE Trans. on Software Engineering SE-10,* no 2, 163-174.
12. E.W. DIJKSTRA (1976). *A Discipline of Programming,* Prentice-Hall series in automatic computation.
13. H. GANZINGER (1983). Parameterized specifications: Parameter passing and implementation with respect to observability. *ACM Trans. Program. Lang. Syst. 5,* no 3.
14. D. GRIES (1981). *The Science of Programming,* Springer-Verlag.
15. C.B. JONES (1979). Constructing a theory of a data structure as an aid to program development. *Acta Informatica 11,* 119-137.
16. C.B. JONES (1980). *Software Development: A Rigorous Approach,* Prentice-Hall International.
17. C.B. JONES (1981). *Development Methods for Computer Program Including a Notion of Interference,* Oxford University, PRG 25.
18. C.B. JONES (1983). *Specification and Design of (Parallel) Programs* (invited paper), IFIP 1983, Paris.

19. C.B. JONES (1983). Tentative steps toward a development method for interfering programs. *ACM Trans. Program. Lang. Syst. 5,* no 4, 596-619.
20. S. KAMIN (1983). Final data types and their specification. *ACM Trans. Program Lang. Syst. 5,* no 1, 97-123.
21. J.C. KING (1971). Proving programs to be correct. *IEEE Trans. on Computers, c-20,* no 11.
22. J.C. REYNOLDS (1983). *The Craft of Programming,* Prentice-Hall International.
23. D. SANNELLA, M. WIRSING (1983). A kernel language for algebraic specification and implementation. To appear in *Proc. Intl. Conference on Foundations of Computing Theory,* Bergholm, Sweden.
24. O. SCHOETT (1982). *A Theory of Program Modules, their Specification and Implementation.* Edinburgh private communication.

APPENDIX I: LOGIC PROOF RULES

See [3] for discussion.

Conventions

(1) $E, E1, \ldots$ denote logical expressions (predicates);
(2) $x, y, \ldots$ denote variables of proper elements in a universe;
(3) $c, c1, \ldots$ denote constants of proper elements in a universe;
(4) $s, s1, \ldots$ denote terms which may contain partial functions.
(5) $p(x)$ denotes a formula in which x occurs free;
(6) $p(s/x)$ denotes a formula obtained by substituting all occurrences of x by s in p. If a clash between free and bound variables would occur, then suitable renaming is performed before the substitution.
(7) $p[s2/s1]$ denotes a formula obtained by substituting some occurrence of $s1$ by $s2$. If a clash between free and bound variables would occur, then suitable renaming is performed before the substitution.

Basic operators

Name	*Rule*	
$\vee\text{-}I$	$\dfrac{Ei}{E1 \vee E2}$	$(1 \leqslant i \leqslant 2)$
$\vee\text{-}E$	$\dfrac{E1 \vee E2,\ E1 \vdash E,\ E2 \vdash E}{E}$	
$\sim\vee\text{-}I$	$\dfrac{\sim E1,\ \sim E2}{\sim(E1 \vee E2)}$	
$\sim\vee\text{-}E$	$\dfrac{\sim(E1 \vee E2)}{\sim Ei}$	$(1 \leqslant i \leqslant 2)$
$\sim\sim\text{-}I$	$\dfrac{E}{\sim\sim E}$	
$\sim\sim\text{-}E$	$\dfrac{\sim\sim E}{E}$	
contr	$\dfrac{E1, \sim E1}{E2}$	
$\exists\text{-}I$	$\dfrac{p(s/x), s = s}{\exists x \,.\, p(x)}$	
$\exists\text{-}E$	$\dfrac{\exists x \,.\, p(x), p(y/x) \vdash E}{E}$	(y is arbitrary and not free in E)

$\sim\exists - I$ $\quad \dfrac{\sim p(x)}{\sim \exists x \,.\, p(x)}$ (x is arbitrary)

$\sim\exists - E$ $\quad \dfrac{\sim \exists x \,.\, p(x), s = s}{\sim p(s/x)}$

$=$-subs $\quad \dfrac{s1 = s2, p}{p[s2/s1]}$

$=$-contr $\quad \dfrac{\sim(s = s)}{E}$

$=$-cons $\quad \dfrac{}{c = c}$

$=$-var $\quad \dfrac{}{x = x}$

consts $\quad \dfrac{\sim t}{E} \quad \dfrac{}{t} \quad \dfrac{u}{E} \quad \dfrac{\sim u}{E}$

Definitions of other connectives

$E1 \wedge E2$ for $\sim(\sim E1 \vee \sim E2)$

$\forall x \,.\, p(x)$ for $\sim(\exists x \,.\, \sim p(x))$

$E1 \Rightarrow E2$ for $E1 \vee E2$

δE for $E \vee \sim E$

f for $\sim t$

Derived rules

Name	*Rule*	
$\wedge - I$	$\dfrac{E1, E2}{E1 \wedge E2}$	
$\wedge - E$	$\dfrac{E1 \wedge E2}{Ei}$	$(1 \leqslant i \leqslant 2)$
$\sim\wedge - I$	$\dfrac{\sim Ei}{\sim(E1 \wedge E2)}$	$(1 \leqslant i \leqslant 2)$
$\sim\wedge - E$	$\dfrac{\sim(E1 \wedge E2), \sim E1 \vdash E, \sim E2 \vdash E}{E}$	

comm $\dfrac{E1 \vee E2}{E2 \vee E1}$ $\dfrac{E1 \wedge E2}{E2 \wedge E1}$

$\vee$-ass $\dfrac{(E1 \vee E2) \vee E3}{E1 \vee (E2 \vee E3)}$

$\wedge$-ass $\dfrac{(E1 \wedge E2) \wedge E3}{E1 \wedge (E2 \wedge E3)}$

It is now legitimate to use n-fold versions of $\vee$-I/E, etc. For example:

$\wedge$-I $\dfrac{E1, E2, \ldots, En}{E1 \wedge E2 \wedge \ldots \wedge En}$

$\vee\wedge$-dist $\dfrac{E1 \vee (E2 \wedge E3)}{(E1 \vee E2) \wedge (E1 \vee E3)}$

$\wedge\vee$-dist $\dfrac{E1 \wedge (E2 \vee E3)}{(E1 \wedge E2) \vee (E1 \wedge E3)}$

$\Rightarrow$-I $\dfrac{E1 \vdash E2, \delta E1}{\sim E1 \vee E2}$ or $\dfrac{E1 \vdash E2, \delta E1}{E1 \Rightarrow E2}$

$\Rightarrow$-E $\dfrac{\sim E1 \vee E2, E1}{E2}$ or $\dfrac{E1 \Rightarrow E2, E1}{E2}$

$\Rightarrow$ $\dfrac{E}{E1 \Rightarrow E}$ $\dfrac{\sim E}{E \Rightarrow E1}$

$\forall$-I $\dfrac{p(x)}{\forall x \,.\, p(x)}$ (x is arbitrary)

$\forall$-E $\dfrac{\forall x \,.\, p(x), s = s}{p(s/x)}$

$\sim\forall$-I $\dfrac{\sim p(s/x), s = s}{\sim \forall x \,.\, p(x)}$

$\sim\forall$-I $\dfrac{\sim \forall x \,.\, p(x), \sim p(y/x) \vdash E}{E}$ (y is arbitrary and bound in E)

$\dfrac{\forall x \,.\, p(x)}{p(y/x)}$ $\dfrac{\sim \exists x \,.\, p(x)}{\sim p(y/x)}$

$=$ comm $\dfrac{s1 = s2}{s2 = s1}$

$=$ trans $\dfrac{s1 = s2, s2 = s3}{s1 = s3}$

Name	*Rule*
$\Delta - I$	$\dfrac{E \quad \sim E}{\Delta E}$
$\Delta - E$	$\dfrac{\Delta E, E \vdash E\,1, \sim E \vdash E\,1}{E\,1}$
$\sim\Delta - I$	$\dfrac{\Delta E \vdash E\,1, \Delta E \vdash \sim E\,1}{\sim \Delta E}$
$\sim\Delta - E$	$\dfrac{\sim \Delta E \vdash E\,1, \sim \Delta E \vdash \sim E\,1}{\Delta E}$
==-reflx	$\dfrac{}{s == s}$
==-subs	$\dfrac{s\,1 == s\,2, p}{p\,[s\,2/s\,1]}$
$\sim$==-*I*	$\dfrac{s\,1 == s\,2 \vdash E,\ s\,1 == s\,2 \vdash \sim E}{\sim (s\,1 == s\,2)}$
$\sim$==-*E*	$\dfrac{\sim (s\,1 == s\,2) \vdash E,\ \sim (s\,1 == s\,2) \vdash \sim E}{s\,1 == s\,2}$
==comm	$\dfrac{s\,1 == s\,2}{s\,2 == s\,1}$
== trans	$\dfrac{s\,1 == s\,2,\ s\,2 == s\,3}{s\,1 == s\,2}$
== $\rightarrow$ =	$\dfrac{s\,1 == s\,2,\ si = si}{s\,1 = s\,2}$ $\quad (1 \leqslant i \leqslant 2)$

APPENDIX II: RULES FOR DATA TYPES

INVARIANT PRESERVATION

$$\forall \overleftarrow{\sigma} \in \Sigma \;.\; \text{preOP}(\overleftarrow{\sigma}) \Rightarrow \exists \sigma \in \Sigma \;\text{postOP}(\overleftarrow{\sigma}, \sigma)$$

REPRESENTATION

retr: Rep $\rightarrow$ Abs

total

adequacy (onto)

$$\forall a \in \text{Abs} \;.\; \exists r \in \text{Rep} \;.\; \text{retr}(r) = a$$

domain

$$\forall r \in \text{Rep} \;.\; \text{preOPA}(\text{retr}(r)) \Rightarrow \text{preOPR}(r)$$

result

$$\forall \overleftarrow{r}, r \in \text{Rep} \;.\; \text{preOPA}(\text{retr}(\overleftarrow{r})) \wedge \text{postOPR}(\overleftarrow{r}, r) \Rightarrow \text{postOPA}(\text{retr}(\overleftarrow{r}), \text{retr}(r))$$

APPENDIX III: RULES FOR SEQUENTIAL PROGRAMS

{Presentation due to PETER ACZEL}

P, B etc	predicates of single states
$\overleftarrow{P}$	assertion formed from P by decorating all free variables with hooks
R	relational predicate
R^r	reflexive closure of R
I_x	identity on all but x
$\mid$	operator between relational predicates corresponding to relational composition

GENERAL

$$\frac{P \Rightarrow P', \{P'\}S\{R'\}, R' \Rightarrow R}{\{P\}S\{R\}}$$

$$\frac{\{P\}S\{R\}}{\{P\}S\{\overleftarrow{P} \wedge R\}}$$

ASSIGNMENT

$$\{\Delta E\}x := E\{x = \overleftarrow{E} \wedge I_x\}$$

CONDITIONAL

$$\frac{\{P \wedge B\}S1\{R\}, \{P \wedge \sim B\}S2\{R\}}{\{P\}\ \textbf{if}\ B\ \textbf{then}\ S1\ \textbf{else}\ S2\{R\}}$$

COMPOSITION

$$\frac{\{P\}S1\{R1 \wedge P1\}, \{P1\}S2\{R2\}}{\{P\}S1;S2\{R1 \mid R2\}}$$

ITERATION

R is transitive and well-founded,

$$\frac{\{P \wedge B\}S\{P \wedge R\}}{\{P\}\ \textbf{while}\ B\ \textbf{do}\ S\{P \wedge R^r \wedge \sim B\}}$$

Algorithmic Aspects of Some Notions in Classical Mathematics

László Lovász
Institute of Mathematics
Eötvös Loránd University
Budapest, Hungary H-1088

1. Introduction

Throughout the history of mathematics, two main lines of results can be identified: a *descriptive* line and an *algorithmic* line. However, the relative significance of these directions has varied considerably. Mathematics started out as a clearly algorithmic discipline in the ancient Egypt and Babylon, dealing mainly with recipes for inventory, land measurements etc. The descriptive (Theorem-Proof) side of mathematics was the great invention of the ancient Greeks. It would be difficult to follow up the shifts of the centre of gravity of mathematical science between its algorithmic and descriptive sides, and it would take certainly someone who is more learned in the history of mathematics than I am. But it is clear that the applications of mathematics have put emphasis on its algorithmic aspects, while the study of its foundations has gone far in the non-algorithmic direction (think of axiomatic set theory). Recently, the development of computers gave new dimensions to the algorithmic line: the possibility of carrying out billions of operations, rather than maybe thousands, enormously increases the possibilities of algorithmic approaches. The fact that these billions of operations have to be carried out without the direct supervision of humans made it imperative to carefully think about questions like how the input and output of an algorithm are given, what has to be stored and in what form, what are the time and space requirements etc. Most notions of classical mathematics were not made with these kinds of questions in mind and this sets us the task of re-thinking our mathematics from an algorithmic point of view. This is of course an enormous work; it is perhaps not yet begun and it is questionable whether it will ever be finished. In this paper I take just one question: 'What is a real number from an algorithmic point of view?' I will try to convince the reader that by thinking about this question one is lead to algorithmic problems which are non-trivial; and the

solution of these problems yields algorithms for very concrete problems in number theory, numerical analysis and combinatorics.

2. What is a real number?

Let us start with an easier question: what is an integer? More precisely: if an algorithm has integers as its input, how do we measure the running time and input length of this algorithm?

There are at least three non-equivalent answers which can be given here:

(1) We may say that an integer t contributes $|t|$ to the input size. In this case the parameter t is in *unary encoding*.

(2) We may say that an integer t contributes $1+\lceil\log_2(1+|t|)\rceil$ to the input size (this is the number of digits in the binary expansion of t, plus 1 for the sign of non-zero integers). In this case parameter t is in *binary encoding*.

(3) We may say that an integer t contributes 1 to the input size. In this case the variable t is in *arithmetic encoding*.

To each of these notions there corresponds a natural way to define the *number of steps* in an algorithm. In cases (1) and (2), every arithmetic operation must be carried out bitwise. So for example, the multiplication of two k-digit integers by the usual procedure takes about $2k^2$ steps. If the input size is measured by (3), then however it is better to count one arithmetic operation (addition, subtraction, multiplication, comparison) as one step. Note that these arithmetic operations are polynomial-time in all three encodings (this is not the case with exponentiation, since for this the size of the output is not polynomially bounded).

These conventions to measure input size and running time lead to different notions of complexity of certain algorithms.

Every algorithm which is polynomial in the binary encoding is also polynomial in the unary, since for the unary encoding, the input size is larger. This is, of course, not true the other way around: for example, the trivial algorithm for primality testing (scanning all smaller integers to find a divisor) is polynomial in the unary encoding but exponential in the binary.

There are many algorithms which are polynomial in the binary sense but not in the arithmetic sense: for example, the euclidean algorithm to find the g.c.d. of two integers. A less trivial example is Khachiyan's algorithm for linear programming: here the existence of a linear programming algorithm which would be polynomial in both the binary and arithmetic sense is an open question.

An example of an algorithm which is polynomial in the arithmetic sense, but not in the binary sense, is the computation of $2^{a_1 \cdot a_2 \cdots a_n}$, where the input $a_1,\ldots,a_n$ is a string of 2's and 3's. The Gaussian elimination algorithm to compute the determinant of a matrix is trivially polynomial in the arithmetic sense, but it is not easy to see that it is also polynomial in the binary sense: one has to show that the integers involved do not grow too big (Edmonds [3]). If an algorithm is polynomial in the arithmetic sense and one can show that the number of digits in any integer occuring during the run of the algorithm can

be bounded by a polynomial in the input size, then it is also polynomial in the binary sense.

In what follows, we shall consider the binary encoding of numbers and measure the running time of an algorithm accordingly.

Let us go on and discuss more general numbers. Rational numbers do not cause any problem: they can be represented as the ratio of two integers. Accordingly, we shall define the *input size* of a rational number $r = a/b$ (where $b>0$ and $(a,b) = 1$) by

$$<r> = 1+\lceil\log_2(1+|a|)\rceil+1+\lceil\log_2(1+|b|)\rceil.$$

We can extend this notion to *complex rational numbers,* i.e. complex numbers r with $\mathrm{Re}\ r, \mathrm{Im}\ r \in \mathbb{Q}$, by letting $<r> = <Re\ r>+<Im\ r>$.

We now come to the more subtle question of irrationals. One may of course take the point of view that they 'do not exist from an algorithmic point of view', but this is sometimes too restrictive. So let us explore how a real number can be defined by approximating it by rationals. This approach is similar to that of BISHOP [2], but our motivation is not a constructive philosophy of mathematics but rather a tool to formalize and analyse algorithms involving real computations.

So we define a real number as a black box α; if we plug in a rational number $\epsilon>0$, then it gives us back a rational number r (an approximation of α with error ϵ). The box has the following tag:

(1)

Real Number α
MANUFACTURER'S GUARANTEE: For any two inputs $\epsilon_1, \epsilon_2>0$, the outputs r_1 and r_2 satisfy $\lvert r_1-r_2\rvert<\epsilon_1+\epsilon_2$.

It is obvious that if we have such a box (and it works as its manufacturer guaranteed it), then this does indeed determine a unique real number. Such a black box ('oracle') can now be included in any algorithm as subroutine. Since we do not know how the box works, we shall count one call on this oracle as one step. If we are interested in polynomial-time computations then, however, an additional difficulty arises: the output of the oracle may be too long, and it might take too much time just to read it. So we shall assume that the black boxes we allow also have the following additional tag:

(2)

ADDITIONAL GUARANTEE:
For any input $\epsilon>0$, the output r satisfies $<r> \leqslant <\epsilon>^k$.

(Here k is a constant explicitely given in unary encoding.) An oracle with guarantees (1) and (2) will be called a *real number box.* The number k is the *input size* of the box, and it has to be added to the input size of any problem in which the box is used as a subroutine.

It is not difficult to assemble a box named $\sqrt{2}$, or a box named π, etc. In fact, one can realize these boxes so that they work in polynomial time. Furthermore, if we have two boxes named 'α' and 'β' then it is easy to design a box for '$\alpha+\beta$' which satisfies the right kinds of guarantees and which works in polynomial time; similarly for '$\alpha-\beta$'. The situation is more complicated with division. There is no difficulty with designing a box for 'α/β', provided we can compute a positive lower bound on $|\beta|$ in polynomial time. But what happens if we cannot compute such a bound? This then means that β cannot be distinguished from 0 in polynomial time. This leads us to the discovery that the equality of two real numbers cannot be determined from the black box description above. At the first sight this seems to be a serious handicap of this model, but it in fact reflects a real requirement in numerical analysis: stability. Namely, if an algorithm contains, any a branching depending on the condition $\alpha = \beta$, then it should also be correct if the condition is replaced by $|\alpha-\beta|<\epsilon$ for any sufficiently small ϵ; since if α and β are only approximately known, their exact equality cannot be decided.

It is now obvious to extend this model to complex numbers: there the oracle puts out, for any rational $\epsilon>0$, a complex rational number r such that (1) and (2) are fulfilled. Such an oracle will be called a *complex number box.*

3. WHAT IS AN ALGEBRAIC NUMBER?

Let us come back to the number $\sqrt{2}$. This notation expresses the following: 'the (unique) root of the equation $x^2-2 = 0$ on the semiline $(0,\infty)$'. More generally, we can encode any real algebraic number by a triple $(f;a,b)$ where f is a polynomial with rational coefficients, and a and b are rational numbers such that f has a unique root in the interval (a,b). Such a triple will be called a *real algebraic number triple.* We define the input size of a polynomial $f(x) = a_0+a_1x+...+a_nx^n$ by $<f> = <a_0>+<a_1>+...+<a_n>$, and the input size of an algebraic number triple $(f;a,b)$ by $<f>+<a>+<b>$. (Note that $<f>\geqslant \deg(f)$.)

This encoding of algebraic numbers is not unique. We can make it more unique by requiring, say, that f is an irreducible polynomial with relatively prime integral coefficients, and also normalizing a and b somehow, but this would make many things awkward. The crucial thing is that *one can decide in*

polynomial time whether or not two real algebraic number triples define the same number.

To see this, let $(f;a,b)$ and $(g;c,d)$ be two real algebraic number triples. Let (u,v) denote the intersection of the intervals (a,b) and (c,d). (If this intersection is empty then the two triples define different numbers.) Compute $h = \text{g.c.d.}(f,g)$. This can be done by the euclidean algorithm; the euclidean algorithm for two polynomials is trivially polynomial in the arithmetic sense, and with some effort one can show that it is also polynomial in the binary sense, i.e. the coefficients of the polynomials which occur never have more than a polynomial number of digits. Having computed the polynomial h, all we have to do is to decide whether it has a root in the interval (u,v): this is easy by computing the values $h(u)$ and $h(v)$, and checking whether or not they have the same sign (to be precise, we have to deal with the case when one of $h(u)$ and $h(v)$ is 0 seperately; but this is easily settled and the details are left to the reader).

It is somewhat more difficult to compute an algebraic number triple describing the sum (difference, product, ratio) of two algebraic numbers described by algebraic number triples, but it can be done in polynomial time. Note however that if $\alpha_1,\ldots,\alpha_n$ are algebraic numbers described by appropriate algebraic number triples then the sum $\alpha_1+\ldots+\alpha_n$ may be of degree exponential in the input size of $\alpha_1+\ldots+\alpha_n$. So the sum of a variable number of algebraic numbers cannot be computed in polynomial time.

One can also try to describe algebraic real numbers as special real numbers. Consider a real number box, i.e. an oracle satisfying (1) and (2), and suppose it has one more tag:

(3)

ADDITIONAL GUARANTEE:
The number defined by this box is algebraic.

It is easy to realize that this guarantee is meaningless: you can never catch the manufacturer cheating. In fact, after a finite number of calls on a real number oracle, there will always be even a rational number satisfying all the previous answers! So one has to require a stronger guarantee from the manufacturer, say the following:

(4) ADDITIONAL GUARANTEE:
The number defined by this box is the root of a rational polynomial with input size at most m.

We call a real number box with property (4) an *algebraic number box.* The *input size* for this algebraic number box is $k+m$ (k from tag (2) and m from tag (4)).

So now we have two different ways to describe algebraic numbers. Are these two equivalent? In a sense the answer to this question is in the affirmative, as shown by the following theorem (A. SCHÖNHAGE [14], R. KANNAN, A.K. LENSTRA, L. LOVÁSZ [8]).

THEOREM 3.1. *(a) Given an algebraic number triple, an algebraic number box describing the same number and working in polynomial time can be designed.*
(b) Given an algebraic number box, one can compute an algebraic number triple describing the same number in polynomial time.

PROOF (sketch). (a) is easy: given an algebraic number triple $(f;a,b)$, we can find by binary search a rational number r, $a<r<b$ such that $|r-\alpha|<\epsilon$, where α is the unique root of f in the interval (a,b), and $\epsilon>0$ is any given rational number. It also follows by an easy computation that $<r> < 5<a>+5<b>+<\epsilon>$. This shows that a box for α can be realized by a polynomial-time algorithm.

The converse is much more involved. Suppose that we have a box with guarantees (1), (2) and (4). Let α be the real algebraic number described by this box. Consider the following matrix:

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^m \\ & \delta & & & \\ & & \delta & 0 & \\ & & & & \\ & 0 & & & \delta \end{pmatrix}$$

(where δ is a sufficiently small positive number). Let L denote the lattice generated by the colums of this matrix.

Every vector in the lattice L can be written in the form

$$v_f = \begin{pmatrix} f(\alpha) \\ \delta a_0 \\ \delta a_1 \\ \vdots \\ \delta a_m \end{pmatrix}$$

where $f(x) = a_0 + a_1 x + ... + a_m x^m$ is a polynomial with integral coefficients. If in particular f is the minimal polynomial of α (which is in L by guarantee (4)), then

$$\|v_f\| = \sqrt{f(\alpha)^2 + \delta^2 a_0^2 + \delta^2 a_1^2 + ... + \delta^2 a_m^2} < \delta 2^m$$

by a routine computation. Now if a lattice contains a vector of length l then by the algorithm described in [11], we can find a vector in this lattice of length at most $2^m.l$ in polynomial time. Let v_g be the vector found this way. Then in particular

$$|g(\alpha)| < \|v_g\| < \delta 2^{2m}$$

and

$$\|g\| \leqslant \frac{1}{\delta}\|v_g\| < 2^{2m}.$$

We claim that from this it follows that $g(\alpha) = 0$, i.e. g is a polynomial with root α. For, let $\alpha = \alpha_1, \alpha_2, ..., \alpha_k$ be the conjugates of α (i.e. the roots of f), then

$$A = g(\alpha_1)g(\alpha_2)...g(\alpha_k)$$

is an integer. On the other hand, it is a routine computation to show that $|\alpha_i| < \|f\| < 2^m$, and hence $|g(\alpha_i)| < m \cdot \|g\| . \|f\|^m < 2^{2m^2}$. Hence

$$|A| = |g(\alpha_1)||g(\alpha_2)|...|g(\alpha_k)| < \delta 2^{2m} 2^{2m^2} < 1.$$

So $A = 0$, i.e. $g(\alpha_i) = 0$ for least one i. Since f is irreducible, this implies that $f|g$, and so $g(\alpha) = 0$ as claimed.

Of course, we have cheated since the matrix we started with contains irrational entries. But a more careful computation would show that replacing α by any rational number sufficiently close to α, the same argument would work; and sufficiently good approximations of α can be obtained from the algebraic number box describing α.

We still have to find an interval in which α is the only root of g. This is easy again using the box describing the number.

A little more careful analysis of the algorithm would also show that we in fact obtain the minimal polynomial of α, i.e. we have $g = f$.

The analog of Theorem 3.1 can be proved for complex numbers as well. Here the computation of the roots of a polynomial is a little more involved but still fairly standard.

We mention two consequences of the above algorithm. First, it shows that *taking the digits of an algebraic number is a very poor random number generator.* Not only is the rest of the sequence determined by a polynomial number of elements, but it can be computed in polynomial time from this beginning segment of polynomial length. Another consequence, more on the positive side, is the following result, which was proved in a different way by A.K. LENSTRA, H.W. LENSTRA, JR. and L. LOVÁSZ [11], although the algorithm sketched here was also mentioned there:

THEOREM 3.2. *A polynomial with rational coefficients can be factorized into polynomials irreducible over the rational field in polynomial time.*

PROOF (sketch). Let f be a polynomial with rational coefficients, and let α be a root of f. Then we can design an algebraic number box for α in polynomial time, by part (a) of Theorem 3.1. By the algorithm of part (b), we can find the minimal polynomial of α in polynomial time. This is then an irreducible polynomial dividing f. Repeating this procedure, the theorem follows.

We conclude this section by some further remarks on algorithmic questions concerning algebraic numbers. It was proved by GALOIS that not every algebraic number can be expressed by radicals, and he also gave an algorithm to decide whether the roots of a given polynomial are expressable this way (equivalently, whether the Galois group of the polynomial is solvable). His algorithm was doubly exponential. Recently, LANDAU and MILLER [10] gave a polynomial time algorithm to decide this. Their method combines Theorem 3.2 with the powerful algorithms in group theory developed recently; see e.g. BABAI, KANTOR and LUKS [1].

We have remarked before that the sum of n algebraic numbers cannot be computed in polynomial time. Another way to look at this problem is that algebraic numbers often have a more compact way of encoding then their minimal polynomials: for example, to write down $\sqrt{2}+\sqrt{3}+\ldots+\sqrt{n}$ takes only about $n \log n$ space (even not using the ' ... '), while the minimal polynomial for this number has exponential degree and hence exponentially many variables. With this more compact encoding, however, it seems more difficult to handle the numbers. I do not know for example, how to decide the equality of two algebraic numbers given as polynomials of other algebraic numbers, which in turn are given by algebraic number triples.

4. MINIMIZING A CONVEX FUNCTION

Mathematical programming and numerical analysis provide a wide range of problems and methods which have not been analyzed from the point of view of computational complexity theory. Often the known methods are non-polynomial or their results, (say) a real number, is not computable in the sense discussed in section 2. We illustrate this situation on the example of the problem of minimizing a convex function.

So let f be a convex function, defined (say) on $\mathbb{R}^n$, and suppose that we want to find the minimum of f. Many known methods to solve this problem (conjugate gradient, steepest descent etc) do not provide a 'real number box' for the minimum value: to compute the minimum up to k digits, one need time exponential in k, at least if the function is not smooth enough. A further handicap of these methods is that they require the knowledge of grad f and quite often also some information about the second derivative of f.

But how is a function 'given'? In the spirit of section 2, it is natural to say that a function is a black box which, when we plug in a rational vector $x \in \mathbb{Q}^n$, puts out the value of $f(x)$. However, this value may be irrational and so more

precisely we need a black box whose input is a pair (x, ϵ) where $x \in \mathbb{Q}^n$ and $\epsilon > 0$ is a rational number. The output of the box is then a rational number r such that $|f(x) - r| \leqslant \epsilon$. As before, we need some guarantees about the way the box works:

(5)

GUARANTEE:

For any two inputs (x, ϵ_1) and (x, ϵ_2), the outputs r_1 and r_2 satisfy $|r_1 - r_2| \leqslant \epsilon_1 + \epsilon_2$. For any three inputs (x_1, ϵ), (x_2, ϵ) and (x_3, ϵ), such that $x_3 = \lambda x_1 + (1-\lambda)x_2$, the outputs r_1, r_2 and r_3 satisfy $r_3 \leqslant \lambda r_1 + (1-\lambda)r_2 + 2\epsilon$. For any input (x, ϵ), the output r satisfies $< r > < (< x > + < \epsilon >)^k$.

Even these guarantees are not enough to determine the minimum of f. For example, if we try to determine the minimum of the function 'x' (defined on $\mathbb{R}$), then after any finite number of function evaluations it is still possible that the function has a minimum, but this is smaller than any value tested so far. To overcome this difficulty, we only consider *constrained minimization*, and for the sake of simplicity we assume that we want to minimize our function f over the ball $\|x\| \leqslant R$. YUDIN and NEMIROVSKII [15] obtained the following result.

THEOREM 4.1. *Given a convex function $f : \mathbb{R}^n \to \mathbb{R}$ by an oracle with property (5), and two rational numbers $R > 0$ and $\epsilon > 0$, we can find in polynomial time a vector $y \in \mathbb{R}^n$ such that $\|y\| \leqslant R$ and for each $x \in \mathbb{R}^n$ with $\|x\| \leqslant R$ we have $f(x) \geqslant f(y) - \epsilon$.*

The little known version of the Ellipsoid Method used to prove this result is based on *shallow cuts*. The idea is that while in the basic Ellipsoid Method we cut the current ellipsoid through its center point, in the Shallow Cut Ellipsoid Method we cut the ellipsoid into two non-congruent parts, so that the larger of the two parts contains the set of optimum points. If the two parts are not 'too different' then even the larger one can be included in an ellipsoid with smaller volume, and then the sequence of the ellipsoids shrinking to the set of optimum points can be constructed. This idea seems to be opposite to the more natural version of *deep cuts*; while the deep cut version of the Ellipsoid Method provides slightly faster convergence, the shallow cut version allows the cutting hyperplane (the subgradient of f) to have an error (at the expense of slower, but still polynomial, running time). The second important ingredient of the Yudin-Nemirovskii Method is an algorithm to compute an *approximate* subgradient of a function given by a black box with property (5). While this approximation is rather poor (the time is polynomial in the error and not in the input size of the error), it is just enough for the shallow cut version of the Ellipsoid Method. For details, see the paper by YUDIN-NEMIROVSKII or the forthcoming book by GRÖTSCHEL, LOVÁSZ and SCHRIJVER.

It is a very interesting question whether one can find the minimum of a convex function given by a black box with property (5) in polynomial time by a more direct algorithm. In the 1-dimensional case, it is easy to do so by, say, the so-called Fibonacci search. It seems to be a long detour to compute approximate subgradients first, and a more geometric argument is very desirable, not to mention its possible practical superiority. However, we shall see in the next section that even in quite special cases nobody could replace the Ellipsoid Method by more direct algorithms.

We conclude this section with the remark that the algorithm sketched in this section provides a powerful tool to attack problems concerning *convex bodies.* Given a convex body, we may be interested in finding its volume, diameter, width, supporting hyperplane in a given direction, hyperplane separating it from a given point etc. The obvious assumption about the body is that we have an oracle (a *convex body box)* which tells us whether or not a given point is in the body. This box has to be supplied with the usual guarantees: it has to define a convex body, and we need two numbers r and R and a point a_0 such that the body contains the ball with radius r about a_0 and is contained in the ball with radius R about a_0. Some of the above-mentioned tasks cannot be solved in polynomial time for such a convex body. For example, we can find the volume of the body within a factor of n^{2n} in polynomial time but one can show that it cannot be computed within a factor of 1.99^n in polynomial time (G. ELEKES, unpublished). Other problems, like supporting hyperplane in a given direction, can be solved in polynomial time in the sense that an approximation of the hyperplane can be computed with error less than ϵ in time polynomial in $<\epsilon>$, $<r>$ and $<R>$. Many such questions of algorithmic geometry will be discussed in the forthcoming book on the Ellipsoid Method by M. GRÖTSCHEL, L. LOVÁSZ and A. SCHRIJVER.

5. MINIMIZING A SUBMODULAR SETFUNCTION

Let S be a finite set and let $f:2^S \to \mathbb{R}$ be any function defined on the subsets of S. We say that f is *submodular,* if the following inequality holds for any two subsets X and Y of S:

$$f(X \cap Y)+f(X \cup Y) \leqslant f(X)+f(Y).$$

Submodular functions play a very important role in combinatorial optimization; for a survey, see [13]. We also have to rely on this paper for convincing the reader that to find the minimum of a submodular setfunction is an important problem which contains, as special cases, a large variety of combinatorial optimization problems, such as minimum cut, matroid intersection, testing membership in polymatroids, and more. In this paper we shall only sketch how this problem can be reduced to minimizing a convex function over the unit cube. It is still an open problem to find a more combinatorial way to minimize such setfunctions.

So let f be a submodular setfunction, defined on the subsets of S. Of course, if f is given as the table of the values it assumes at different subsets, then its minimum is found trivially by scanning this table and the problem is obvious.

A non-trivial problem arises when f is given by an oracle, a *submodular box*, which tells us, for every subset $X \subseteq S$, the value $f(X)$; and has a guarantee:

(6)

GUARANTEE:
For any two inputs $X, Y \subseteq S$, the outputs $f(X)$ and $f(Y)$ satisfy $f(X \cap Y)+f(X \cup Y) \leqslant f(X)+f(Y)$. For any input $X \subseteq S$, the output $f(X)$ satisfies $<f(X)> \leqslant k$.

Then we have the following theorem (GRÖTSCHEL, LOVÁSZ and SCHRIJVER [6]):

THEOREM 5.1. *Given a submodular set-function by an oracle with property (6), the subset X of S minimizing this setfunction can be found in polynomial time.*

Let us sketch the algorithm proving this theorem.

We want to find the set $X \subseteq S$ for which $f(X)$ is minimal. We may assume without loss of generality that $f(\phi) = 0$, since otherwise we can subtract $f(\phi)$ from all values of f. Now we define a function $\tilde{f}: \mathbb{R}_+^S \to \mathbb{R}$ as follows. Let $c \in \mathbb{R}_+^S$. Then we can write c in a unique way in the form

$$c = \lambda_1 a_1 + \lambda_2 a_2 + \cdots + \lambda_n a_n,$$

where $\lambda_1, \ldots, \lambda_n > 0$, $a_1, \ldots, a_n$ are 0-1 vectors and $a_1 \geqslant a_2 \geqslant \cdots \geqslant a_n$. (Here e.g. a_1 is the incidence vector of the support of c.) The set

$$\tilde{f} = \lambda_1 f(a_1) + \cdots + \lambda_n f(a_n).$$

Now it is not too difficult to verify that if f is submodular then $\tilde{f}$ is convex. Furthermore, the minimum of $\tilde{f}$ over the unit cube is attained at a vertex of the cube (this is a very special property of this convex function; usually concave functions have such property!). Hence the minimum of $\tilde{f}$ over the unit cube is equal to the minimum of f over the subsets of S.

Thus it suffices to find the minimum of $\tilde{f}$ over the unit cube. But this can be accomplished in polynomial time by Theorem 4.1.

6. ALGORITHMS FOR PERFECT GRAPHS

Our second example from combinatorial optimization is the problem of finding the independence number of a perfect graph. A graph G is *perfect,* if for every induced subgraph H of G, the chromatic number of H is equal to the size of the maximum clique in H. It is known that the complement of a perfect graph is perfect, so this is equivalent to saying that for each induced subgraph H, the minimum number of cliques in H covering every point is equal to the maximum number of independent points in H. For more about perfect graphs, see

the monograph of GOLUMBIC [5].

Considering the definition of perfectness, it is an immediate question to ask whether the maximum size of an independent set of points in a perfect graph can be found in polynomial time. Note that this problem is trivially equivalent to finding the minimum number of cliques covering the points, or the maximum clique, or the chromatic number of a perfect graph. For various special classes of perfect graphs, these problems correspond to important combinatorial optimization problems and can be solved by special algorithms. So this problem includes the matching problem for bipartite graphs, the Dilworth problem for partially ordered sets, the problem of finding a maximum number of disjoint intervals in a family of intervals etc. However, no combinatorial algorithm is known to find the maximum number of independent points in a (general) perfect graph.

Using the Ellipsoid Method, GRÖTSCHEL, LOVÁSZ and SCHRIJVER [6] proved the following.

THEOREM 6.1. *The maximum number of independent points in a perfect graph can be found in polynomial time.*

The key to the proof is the following quantity, defined by LOVÁSZ [12]. Let G be any graph on $V(G) = \{1,...,n\}$, and define the family $\mathcal{A}$ of $n \times n$ matrices as the set of the symmetric matrices for which $(A)_{ij} = 1$ if $i = j$ or if i and j correspond to non-adjacent positions in the graph. Let $\Lambda(A)$ denote the largest eigenvalue of the matrix A and let

$$\theta(G) = \min\{\Lambda(A): A \in \mathcal{A}\}.$$

It is not difficult to see that this number always lies between the maximum number of independent points and the minimum number of cliques covering all points. In particular, for perfect graphs $\theta(G)$ is equal to the common value of these two numbers. Thus Theorem 6.1 follows if we prove

THEOREM 6.2. *Given a graph G and a rational number $\epsilon > 0$, we can compute in polynomial time a number r such that $|r - \theta(G)| < \epsilon$.*

It is not difficult to see that $\Lambda(A)$ is a convex function of A while A ranges over all symmetric $n \times n$ matrices. It remains a convex function if some of the variables are fixed. So to compute $\theta(G)$ it suffices to find the minimum of this convex function. Using standard methods from numerical analysis we can design a polynomial-time algorithm to evaluate the function $\Lambda(A)$ and we can also derive an upper bound on the entries of the optimizing matrix. So Theorem 6.2 follows from Theorem 4.1.

Recently M. GRÖTSCHEL, A. SCHRIJVER and the author have extended this method to obtain an algorithm for maximum independent set for certain non-perfect graphs, in particular for the so-called h-perfect graphs.

References

1. L. Babai, W.M. Kantor, E.M. Luks (1983). Computational complexity and the classification of finite simple groups. *24th Annual Symp. on Found. of Comp. Sci.*, IEEE Computer Society Press, 162-171.
2. E. Bishop (1967). *Foundations of Constructive Mathematics,* McGraw-Hill, New York - San Francisco - St. Louis - Toronto - London - Sidney.
3. J. Edmonds (1967). Systems of distinct representatives and linear algebra. *J. Res. Natl. Bureau of Standards 71B,* 241-247.
4. G. Elekes (unpublished).
5. M. Golumbic (1980). *Algorithmic Graph Theory and Perfect Graphs,* Academic Press, New York.
6. M. Grötschel, L. Lovász, A. Schrijver (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica 1,* 169-197.
7. M. Grötschel, L. Lovász, A. Schrijver. *The Ellipsoid Method and Combinatorial Optimization,* forthcoming book.
8. R. Kannan, A.K. Lenstra, L. Lovász (1984). Polynomial factorization and the nonrandomness of bits of algebraic and some transcendental numbers. *Proc. 16th ACM SIGACT Symp. on Theory of Computing,* ACM 1984.
9. L.G. Khachiyan (1979). A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR 244,* 1093-1096 (Russian); English translation: *Sovjet Math. Dokl.* 20, 191-194.
10. S. Landau, G.L. Miller (1983). Solvability by radicals is in polynomial time. *Proc. 15th Annual ACM Symp. on Theory of Computing,* ACM, 140-151.
11. A.K. Lenstra, H.W. Lenstra, Jr., L. Lovász (1982). Factoring polynomials with rational coefficients. *Mat. Ann. 261,* 515-534.
12. L. Lovász (1979). On the Shannon capacity of a graph. *IEEE Trans. Information Theory 25,* 1-7.
13. L. Lovász (1983). Submodular functions and convexity, in: *Mathematical Programming: The State of the Art,* Springer-Verlag, Berlin - Heidelberg - New York - Tokyo, 235-257.
14. A. Schönhage (1983). *Factorization of Univariate Integer Polynomials by Diophantine Approximation and by Improved Basis Reduction Algorithm,* preprint, Univ. of Tübingen.
15. D.B. Yudin, A.S. Nemirovskii (1976). Informational complexity and effective methods of solution for convex extremal problems. *Ekon. i Mat. Metodi 12,* 357-369 (Russian); English translation: *Matekon 13 (3),* 24-45.

Problems and Perspectives in Robotics[1]

Robotics Activity

J.T. Schwartz
Courant Institute of Mathematical Sciences
New York University

1. Introduction

The aim of robotics is the mechanization of that elementary 'operative' intelligence which people use unthinkingly in locating and handling ordinary objects. Research in this field has two principal aspects: sensory and manipulative. Sensory studies aim to develop techniques which make it possible to organize the raw data gathered by sensors such as video cameras and ultrasonic range finders into perceptually meaningful gestalts. Studies of manipulation deal with the tactics and strategy needed to control bodies moving slowly or rapidly through three dimensional space, both when the controlled (robot) bodies must move avoiding contact with other bodies or obstacles in their environment, and also when the controlled robot bodies need to make contact with portions of their environment or with other robots, e.g. to grip an object which is to be moved, to insert a peg into a hole, etc.

Although robotics has by now been a recognized area of computer science for several decades, it is only during the last few years that the level of activity in it has begun to expand rapidly. Nevertheless, we can confidently predict that research in this area is destined to affect computer science profoundly. Till now, computer science has been largely combinatorial and symbolic, having the manipulation of patterns and tables of data as its principal content. In robotics, however, computer science makes contact with real-world geometric and physical phenomena such as the compliance of elastic bodies, the frictional phenomena which occur when bodies come in contact, errors in modeling

1. Work on this paper has been supported in part by Office of Naval Research Grant N00014-82-K-0381, and by grants from the Digital Equipment Corporation, the Sloan Foundation, the System Development Foundation, and the IBM Corporation.

which are inevitable in the real world, the sudden changes of state which occur when bodies collide unexpectedly, and so forth. Much interesting new computer science will emerge from contact with these rich conceptual domains and, in particular, computer science will become more traditionally mathematical and 'continuous'.

Fields of science have their own internal rhythms, in which periods of slow progress conditioned by a lack of ideas or by the exhaustion of old ideas alternate with the excitement of rapid advance triggered by conceptual breakthroughs or by maturation of supporting technology. Sustaining technological development and systematic conceptual advance often go together and reinforce one another. After its slow start during the past several decades, robotics stands at the start of a period of rapid advance, current theoretical and pragmatic developments foreshadowing major progress in many of its subfields. The massive computational power created by VLSI technology is a major driving force: by making computing cycles available in whatever quantity required, the work of VLSI designers is rapidly creating most of the purely 'electronic' side of the technological base which robotics will require. Armed with this technology, robotics researchers have begun to perceive ways for the radically strengthening of the basic capabilities of robots, e.g. their ability to see, to manipulate, and to plan. As these capabilities are improved, additional work integrating them into composite software environments facilitating robot use must also follow.

This talk will try to convey something of the flavor of recent robotics research by giving a few simple examples. Before doing this, however, it is appropriate to review some of the main areas in which rapid progress seems likely. These include:

1 control of highly dynamic motions;
2 control of delicate and dextrous motions;
3 automatic planning of robot motions;
4 robot vision;
5 unsolved difficulties.

1.1. Control of highly dynamic motions

Work in this area aims at robots that can run, jump, tumble, climb, etc. Here the work of M. RAIBERT at Carnegie-Mellon University seems particularly promising, see [34]. In order to concentrate on the specifically dynamic problems of robot locomotion, this work has cleverly chosen to focus on an extreme case; dynamic control of a 1-legged robot that traverses its environment by hopping about. Since a one-legged hopper totally lacks static stability and will simply fall over if it ever stops hopping, such a mechanism poses the problem of dynamic stability in pure form. The CMU work shows that stable control along reasonably straightforward lines is nevertheless possible. This work is now being extended to multilegged robots, where equal success, leading e.g., and, to 2-legged robots which can run gracefully and rapidly, is to be expected.

1.2. Control of delicate and dextrous motions

This work aims at robots that can adjust smoothly and simply to the shapes and physical behavior of delicate 3D-bodies. For example, one wishes to be able to grasp an egg, either to draw some figure on it with a stiff pen, or to carry it to the edge of a cup, crack it, and then (more dynamically) pour its contents into the cup. Here a variety of problems arise. Sophisticated multidimensional feedback control methods are needed and are rapidly being defined. Work in computational geometry is elucidating the interesting geometric issues involved in management of mechanisms with many degrees of freedom. Theoretical attention is being directed to one of the most neglected areas of classical physics, the analysis of the frictional motions of rigid and flexible bodies. Interesting robot hands, which will provide appropriate levels of experimental challenge to control theorists, computational geometers, and robotic software designers, are being developed by several engineering groups, notably those led by S. JACOBSEN at the University of Utah, see [21] and by K. SALISBURY at MIT, see [36]. Many laboratories are attempting to develop improved touch-sensing arrays. These include Bell Laboratories, M. Raibert's laboratory at Carnegie-Mellon University, and many others: see [16], [34], [13], and also [14] for a general review of these developments.

1.3. Automatic planning of robot motions

The problem here is to develop algorithms which will allow a robot which knows the geometry of the environment in which it must move to plan the details of its motion automatically. Moreover, if the robot is grasping a body (of known geometry) it must allow for this in the motion it plans. The availability of such algorithms would simplify robot programming considerably by allowing commands having the form

MOVE FROM POSITION *a* TO POSITION *b*

to be issued without it being necessary to specify the details of an obstacle-avoiding path.

This motion planning problem has begun to yield to the efforts of theorists and algorithm designers who have found it possible to apply methods developed by topologists and algebraic geometers to this practical area. See [39], [40], [41], [29], for an account of some of this work. The algorithms given in these papers, though polynomial, have impractically high exponents; however, more recent work suggests that practical motion planning algorithms, linear in the number of obstacles present in a scene as long as the scene is of bounded local complexity (i.e., only a fixed finite number of obstacles lie in any finite sphere) are possible. These algorithms should also facilitate plan updating as individual objects are moved about in its environment by a robot.

1.4. Robot vision

As already said, the problem here is to find techniques which make it possible to organize the raw data gathered by a video camera into perceptually meaningful gestalts. In considering this deep problem, it is well to distinguish two basic ways in which it can be cast, which are different enough to lead work in quite different directions. These can be called the *model based* approach to vision, as contrasted to the *general* approach. In 'model based' vision studies, attention is confined to scenes containing only known objects or objects belonging to known parametric classes (e.g. cylinders with spherical caps and cylindrical holes bored in them, but of heights and radii not known a priori). 'General' vision studies aim to impose helpful perceptual groupings on entirely general scenes, e.g. landscapes containing shrubbery. The great advantage of the first problem is that it is entirely objective: its aim is simply to reduce a scene known to contain objects drawn from a fixed finite set $\{O_1, \ldots, O_n\}$ to a table giving the identities and orientations of all objects actually present. In contrast, the deeper 'general' vision problem has inherently psychological aspects: here one aims to devise (the image-analysis portions of) a robot 'eye' whose perceptual groupings are close enough to those formed by the human eye for easy communication and mutual understanding to be possible. In working on this problem, our aim must be to construct a mechanical eye that regards scenes as similar, and portions of scenes as coherent, just when the human eye reacts in the same way. This rules out the use of all kinds of geometric tricks and 'artificial' approaches which can be used very effectively to attack the unpsychological problem of model-based vision.

Precisely because it is narrower and more objective, the problem of model-based vision seems the right point at which to try to penetrate the overall robot vision problem. Fortunately, this problem can be put at various levels of difficulty, to which theoretical and experimental work can advance progressively. More specifically, one can consider:

- images of either 2D or 3D-bodies, which can either be seen in isolation or as parts of compound scenes;
- the bodies with which one deals can either be wholly visible or partially obscured, and can be present either in constrained or in perfectly general orientations;
- the bodies seen can either be stationary or can be allowed to move;
- the bodies seen can either conform exactly to their models, or can be affected by extra error features such as 'burrs', 'dents', 'flash', etc.

Recent work makes successful treatment of all these problems appear feasible. In an interesting series of papers ([10], [11], [12]), T. LOZANO-PEREZ of MIT and his collaborators have shown how effectively knowledge of the geometry of a body or list of candidate bodies can be used to build up an effective discrimination-tree approach to the object identification/orientation problem. These results are corroborated by work at New York University, the Stanford Research Institute (see [5]), and elsewhere. They suggest that it need not be long before a robust packaged solution of the model-based robot vision

problem appears, in the form of a camera/computer system which, when pointed at a suitably illuminated scene containing k objects drawn from a known list of candidates, will produce a continuously updated table of length $7k$, each 7-word record of which defines the identity, translational coordinates, and Euler angles of a body present in the scene.

Vision studies are also conditioned by the form of input they assume, and a variety of schemes have been developed for acquiring information-rich images when use of simpler images complicates the object identification problem. The images from which a model-based approach works can either be:

- ordinary intensity images;
- high quality silhouettes, obtained, e.g., by 'backlighting' a scene;
- 'depth images', in which each pixel records the true geometric distance of an observed point P on a body surface from the camera, or, equivalently, P's true geometric position in 3-dimensional space.

Images of this last type can be obtained by a variety of schemes, including photometric stereo [18] and use of structured light, as described below. It is worth noting that such 3D or depth images are a particularly favorable form of visual input. Their crucial advantage is that they allow images acquired by arbitrarily many eyes to be combined easily, since all depth images give the locations of surfaces in a common 3-dimensional space.

As stated, it appears likely that effective analysis of all of these kinds of images will be possible provided that only finitely many objects of shapes known a priori can be present in the scene being viewed. Beyond this, successful attack on the 'object acquisition' problem for constrained classes of depth appears feasible. This is the problem of using visual information to build a geometric model of an unknown object. This should be practicable if the object is known to have surfaces all belonging to some limited geometric class (e.g., all plane, cylindrical, spherical, or conical) and to have only a limited number of such surfaces.

Some of the techniques which can be used to solve the object identification and acquisition problems, at least for the particularly simple case of convex polyhedral objects, will be described later in this talk.

1.5. Unsolved difficulties

Though an optimistic view of near-term perspectives seems amply justified in many subareas of robotics, it is also true that significant problems abound for which no easy solution is yet obvious. Some of these unsolved problems are conceptual, e.g., how to accomplish 'general robot planning'; others narrowly technological. For example, no entirely adequate technique is yet available for powering robot systems having many independent degrees of freedom. (Even though mechanical designers are beginning to consider systems with many independently controllable subparts; an example would be a multifingered hand, each of whose fingers carries a number of still smaller fingers, intended for extremely flexible and delicate gripping.) Animal muscle tissue solves this problem easily; muscle provides indefinitely many independent controllable

activators down to the level of microscopic individual fibers. No suitable robotic analog of muscle is yet known. Electrical powering of robots creates weight problems, since the weight of numerous motors attached to moving members of a robot tends to mount rapidly. Pneumatic systems, though light, are difficult to control accurately. Hydraulic powering of very many degrees of freedom has not yet been demonstrated, though here new technologies based upon the properties of electrically or magnetically active fluids may be possible.

2. Illustrative examples of technological and mathematical problems in robotics

In the remainder of this talk, we will discuss two illustrative problems, both relatively elementary, drawn from the two major subareas of robotics, namely vision and manipulation.

We begin with a few remarks on robot vision, which try to make two complimentary points: on the one hand, specialized visual sensors can be designed to give full 3-dimensional information concerning the visible parts of the surface of every body present in an observed scene; on the other hand, provided that the bodies in a scene are known to belong to some finite sets of possible bodies for which detailed geometric models are available, very fragmentary sensory information often suffices to identify all these bodies, and determine their orientations. Together these two remarks indicate that the 'model-based' vision problem should be tractable.

2.1. 'Structured light' or 'active depth' sensors

A *structured light* or *active illumination* visual sensor is one which illuminates objects using non-uniform beams of light upon which one has impressed internal properties varying from point to point through the geometric extent of the beam. Any observable property can be used to form such a beam; among properties which one might think of exploiting are intensity, polarization, spectral distribution of optical energy, coherence, and (temporal) modulation. Structured light sensors can be used to achieve 3D-vision, that is, to build visual sensors that directly give the true position in 3-dimensional space of points on the surface of a body or bodies being observed.

The simplest form of structured light approach is the *striped light* scheme originally introduced by P. Will and K. Pennington of IBM Research. In this sensor (see figure 1) a plane P of light formed by passing collimated light from a source S through a slit is used to illuminate a body B. The resulting illuminated stripe on the body B is observed through a camera C offset from the source S. Then the 3D-position of each illuminated point X can be calculated as the unique point of intersection of the plane P with the known line L from the camera through X. Note that L is known since it is determined geometrically by the point at which the image of X falls on the retina of C. (See figure 1.)

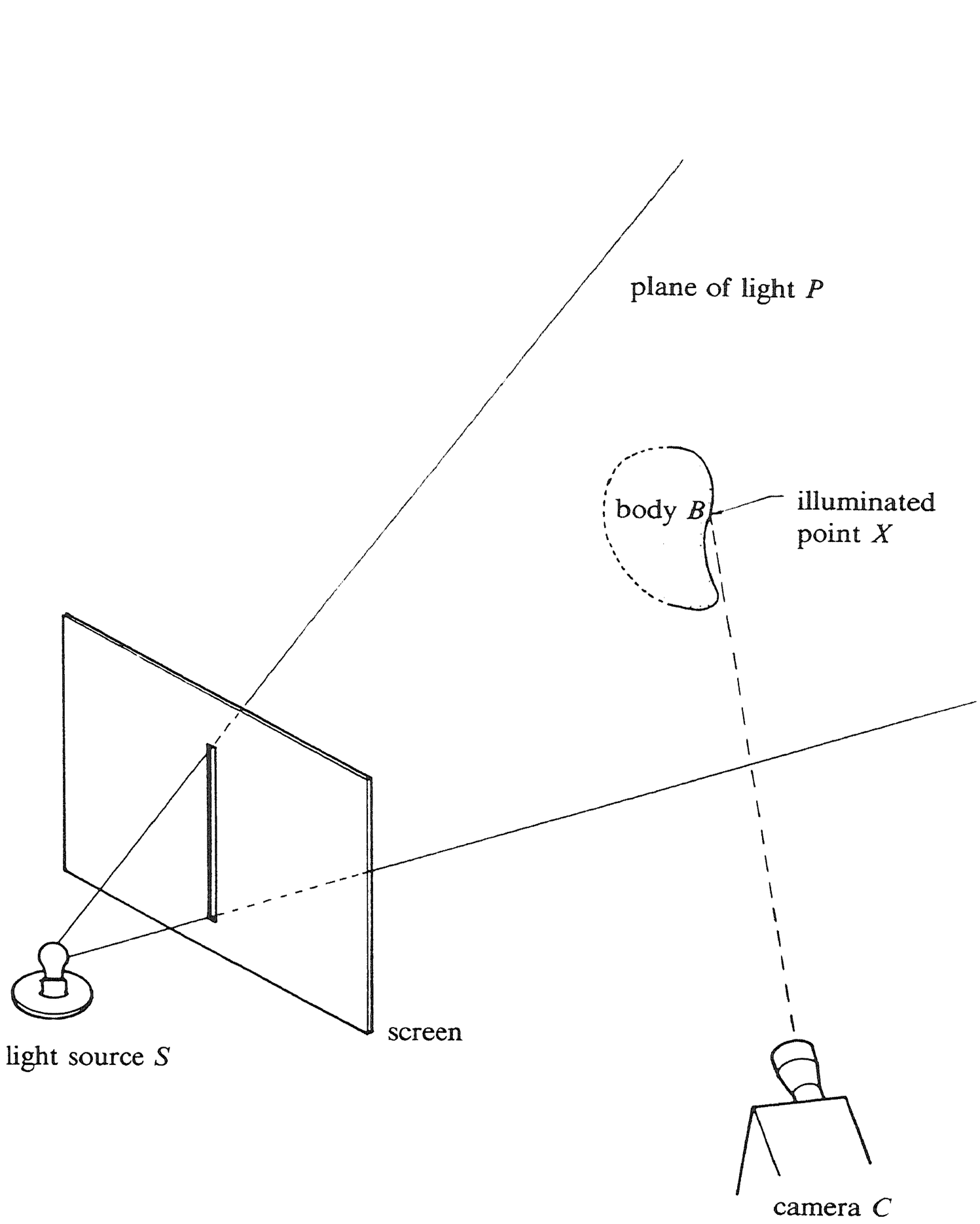


FIGURE 1. The basic Will-Pennington striped light scheme

In the Will-Pennington scheme, the whole surface of the body B can be determined by panning the plane P over the whole visual field. An improved

scheme, which allows this whole surface to be determined more rapidly, is as follows. Suppose that the body B is illuminated by a edge W of light as in figure 2. Suppose that this wedge of light is structured, in the sense that some measurable property Q is imposed upon its separate rays; this property is assumed to be constant in each vertical plane of the wedge W, but to vary monotonically from left to right across the wedge, which is to say monotonically with the geometric parameter h shown in figure 2, which defines the vertical plane in which a given point of W lies.

The internal property Q imposed upon the wedge W of light must have the following properties:

1 there must exist some convenient way of imposing the property Q on the wedge W of light, and of causing Q to vary in the desired manner;
2 Q must be detectable by the camera being used;
3 Q must be invariant under reflection.

Property 3 is required since light projected on a point P on the body B will only be observed after reflection from B, so that a property Q that was modified by reflection might be distorted in some unpredictable manner. Thus Q cannot simply be light intensity, average color, or polarization, since reflection from a body of varying or unknown color, albedo, or polarizing properties would modify these properties.

The simplest property that can be used for Q is the ratio $I2/I1$ of two intensities. More specifically, we can form two successive images of the body B, the first under uniform illumination $I1(h) = \text{const.}$, the second under an illumination $I2(h)$ that varies linearly as the geometric parameter h varies from left to right over the wedge W of light. Then we can form the ratio $Q(h)$ of the two reflected intensities. Assuming that the body B is in perfect focus, the ratio of reflected intensities must be equal to the ratio $I2(h)/I1(h)$ of incident intensities. To see this, note that reflected intensity can be regarded as the number of photons per second reflected from a given point on the surface of $B1$; since photons are reflected individually, the number of photons reflected from a point is proportional to the number projected to the point P by an incident beam.

Let the symbol $R(h)$ denote the ratio $I2(h)/I1(h)$. Then $R(h)$ is invariant in the sense explained above, and the value of $R(h)$ determines h uniquely. Hence by measuring $R(h)$ and triangulating in the manner already explained we can determine the true 3-dimensional locations of all points visible on the surface of the body.

Of course, empirical difficulties ignored in our oversimplified account complicate the situation. Among other problems, a successful version of the sensor described will have to compensate for:

- nonlinearities in the reaction to incident light of the electronic camera used;
- smearing effects caused by lack of focus;
- ambient light effects.

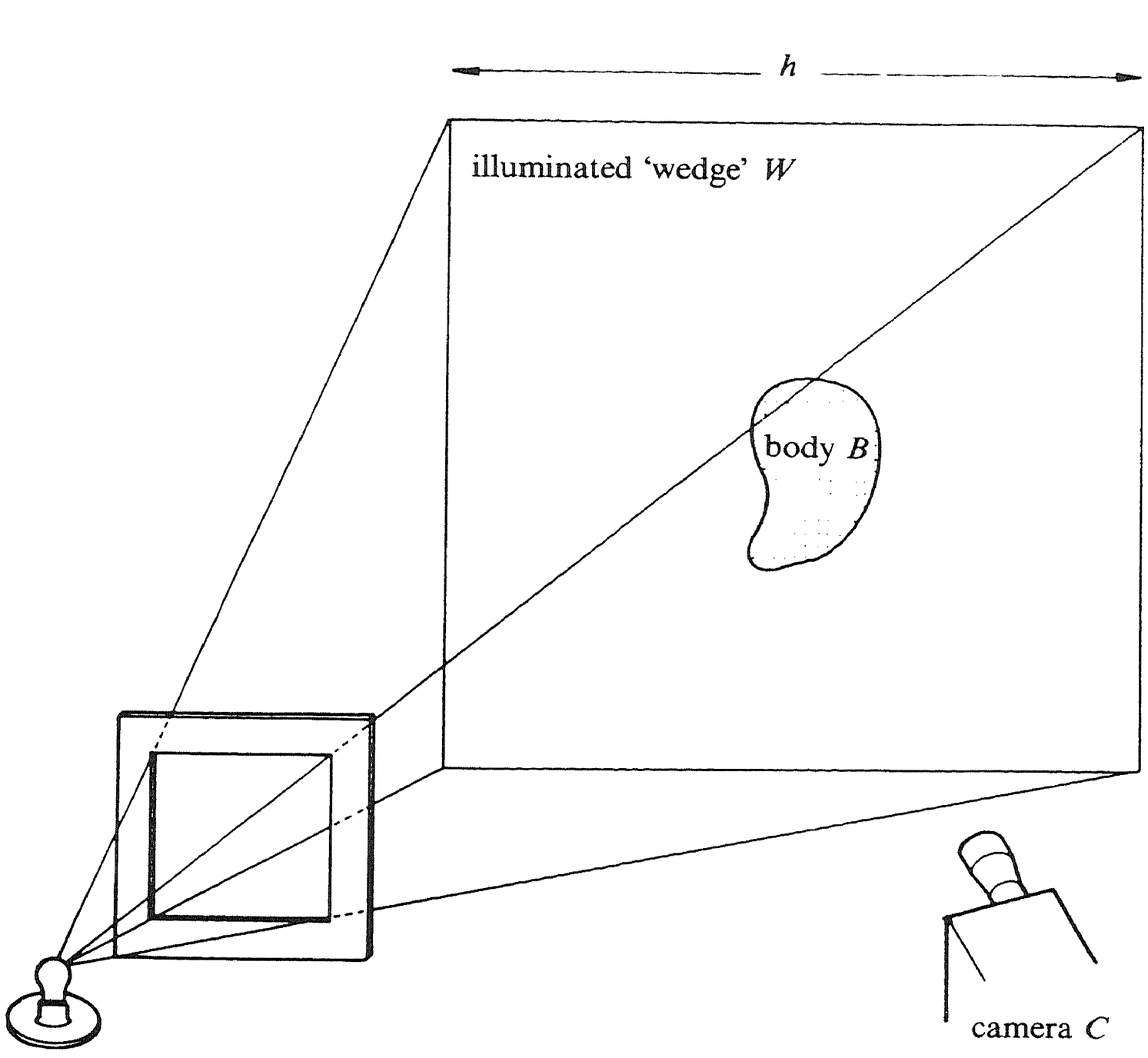


FIGURE 2. Illumination of a body by a structured wedge of light

Nevertheless all these difficulties can be overcome; accurate, high speed depth sensors should therefore become common robot attachments over the next few years.

2.2. Polyhedron recognition using silhouettes

Given presently available sensors, object silhouettes can be formed more accurately and rapidly than depth images. For this reason, it is worth considering the extent in which the silhouettes of polyhedra can be used to identify them. To this end, the following remarks on silhouettes will be helpful. Suppose that a convex polyhedron P is given a certain orientation in 3-space, and projected upon a plane Q parallel to the xz-plane, which lies entirely on one side of P. Given any such orientation, one group of P's faces will be visible from Q, while its other faces will be obscured by the body of P. The boundary between the visible and the invisible portions is a sequence of edges of P, which we will call the *3-silhouette* of P; the projection onto the xz-plane of the 3-silhouette bounds the ordinary *2D-silhouette* of P, which is always a convex polygon. If we assume that no face of P is orthogonal to the xz-plane, then just one point p of the 3-silhouette projects onto each point q of its 2D-silhouette, and p varies continuously with q. Hence the 3-silhouette is topologically a circle, and therefore divides the surface of P into exactly 2 groups of faces, each of which must be connected. The silhouette of a convex polyhedron P is therefore the projection, on the camera's image plane, of a closed sequence of edges on P.

The sequence of edges constituting the 3-silhouette of a polyhedron P as seen from a viewpoint Z can change only when Z crosses one of the face planes of P. If P has n faces, its n face planes will decompose the 3D-space exterior to P into at most $O(n^3)$ regions; hence P can have at most $O(n^3)$ distinct 3-silhouettes. Much the same argument shows that if we confine ourselves to isometric views of P, i.e., views from a point Z very distant from P, then at most $O(n^2)$ 3-silhouettes are possible.

The set of points in 3-space projecting into a given line on the retina of an observing camera is a plane in 3-space whose equation is determined by the known camera geometry. Thus, if an observed edge of the silhouette is taken to be the image of a body edge E we know a plane on which the two extremities x, y of E must lie. This gives us a pair of equations

$$L\cdot(Rx + A) = L\cdot(Ry + A) = b,$$

where R is a 3 by 3 rotation matrix and A is a 3-dimensional translation vector. From this we have $L\cdot R(x - y) = 0$, so that by using three such conditions, i.e., by observing 3 silhouette edges, which need not even be adjacent, we can determine the three independent parameters of R completely, and hence determine A also. Moreover, as soon as a provisional value is calculated for R and A, we know the silhouette that should be seen, and then any significant deviation of the observed silhouette from this calculated silhouette allows us to reject the provisional object and silhouette identification on which this calculation was based and hence, after a serial search through finitely many possible

identifications, to determine what object we are looking at, and how it is oriented.

For a closer view of the calculation required to determine R, write $x-y=z$, and express $R(x-y)=Rz$ in terms of a unit quaternion q as $qz\bar{q}$. Then $L\cdot R(x-y)=0$ takes on the form $L\cdot qz\bar{q}=0$, i.e., is a quadratic equation for a unit quaternion. We must deal with three such equations, and can treat them most easily as homogeneous quadratic equations in 4 variables, or equivalently, as a set of three inhomogeneous quadratic equations in projective 3-space, easily solved numerically by simultaneous diagonalization of two of the three symmetric coefficient matrices appearing in these equations. Similar even simpler considerations apply whenever we can observe any sufficiently large neighborhood of a silhouette corner.

The calculations just described are simple enough for continuous tracking of moving polyhedra to be feasible, simply by building up a bit of special data-reduction hardware to process silhouettes rapidly. Except in positions at which one or more edges are about to leave an object's 3-silhouette and be replaced by others, a small change in the position of a polyhedron will simply be represented by a small change in the coefficient vectors L of the various quadratic equations $L\cdot qz\bar{q}=0$ described in the preceding paragraph. Adjusted values of the quaternions q can then be calculated simply by solving a linear perturbation equation. Note also that if the position of a body translating/rotating in 3-space is tracked continuously, it will always be possible to predict when the objects 3-silhouette is about to change, and to adjust the tracking computations to allow for this change. Generally, this will simply involve identification of any three edges which are not about to leave the silhouette.

Next consider the more challenging problem of how to *acquire* knowledge of the geometry of previously unknown convex polyhedron from a sequence of silhouettes of the polygon. This can be done by turning the body about a vertical axis while forming a series of silhouette images. To see how this is possible, it is convenient to suppose that all the silhouettes used are isometric, i.e., are orthogonal projections of 3-silhouettes of the body on a fixed distant view plane, and that no two vertices of the body lie in the same horizontal plane. The polyhedron corners common to both silhouettes are simply the corners which appear at the same vertical level in both silhouettes. To locate an observed corner C in 3-dimensions, we simply require two silhouettes in which C is visible, taken at rotated position not differing by 180°. (To see this, let C's distance from the vertical axis about which the body is rotating be r and let the horizontal line through C and the axis of rotation make an angle θ with the plane of projection. Then the distance from the corresponding silhouette corner to the projected axis of rotation is $r\cos\theta$, and the equivalent distance in a silhouette formed after rotation of the body through an angle θ is $r\cos(\theta+\Psi)$. The ratio of these two distances is therefore $\cos\Psi-\sin\theta\sin\Psi$, and since Ψ is known this determines θ, and hence r, provided only that $\sin\Psi\neq 0$.)

A few remarks concerning the manner in which a polyhedron's silhouette,

assumed isometric, changes as a polyhedron P is rotated about a vertical axis will be helpful. Suppose we draw the outward-directed normal n to a given face F of P. Then F is visible from a plane of view and viewpoint Q if n points toward Q, but obscured by the body of P if n points away from Q. We want to understand how the 3-silhouette of P varies as we rotate P about a vertical axis. For this, it is convenient to assume that the plane of view is the xy-plane and to project all the normals n to P's faces F onto the xy-plane. This forms a 'direction diagram' consisting of unit vectors in the xy-plane: a face is visible from Q if the corresponding projected normal points toward the positive side of the y-axis, but invisible otherwise. Therefore the edge separating two adjacent faces $F1$ and $F2$ belongs to P's 3-silhouette if and only if the corresponding projected normal vectors point into opposite sides of the y-axis. It follows that nearly every edge e of P will appear in a 3-silhouette of P as we rotate P about the 2-axis. To find a silhouette in which e will appear, one merely has to find a rotated position in which the y-axis separates the projected normals of the two faces which meet at e. The only exceptions would be edges which separate pairs of faces having identical projected normals; these are exactly those edges parallel to the xy-plane, which are ruled out by our assumption that no two corners of P have the same vertical height.

As P rotates about the z-axis, each edge e enters the 3-silhouette and leaves the 3-silhouette exactly twice. Specifically, e enters when the first of the two faces $F1$, $F2$ which meet at e becomes visible, but then leaves when $F2$ becomes visible also; enters the 3-silhouette again when $F1$ becomes obscured, and leaves again where $F2$ becomes obscured also. Consideration of P's direction diagram makes it plain that the set of angular orientations (about the z-axis) in which a given edge belongs to P's 3-silhouette constitutes a pair of angular sectors separated by exactly 180°; in one of these two sectors exactly one of the two faces meeting at e is visible, and in the other sector exactly the other face is visible.

Similarly, the angular range in which the 3-silhouette of P is a given set of edges is a pair of angular sectors separated by 180°, each such sector being delimited by two consecutive lines in P's angle diagram. To see this, note once more that the 3-silhouette S is a topological circle separating two connected sets of faces; wherever S is the silhouette, one of these sets of faces must be visible from Q, the other obscured. Since the range in which a given edge belongs to the 3-silhouette is a pair of opposed angles, the range in which any specified set of edges belongs to this silhouette must be the intersection of such pairs, and hence must comprise itself exactly one pair of opposed angles. Obviously, neither of these two angles can contain a critical direction in which just one face goes from visible to invisible.

At each critical orientation of P, i.e. each orientation at which one of the lines in the direction diagram is perpendicular to the y-axis, one of the faces F of P will be seen end-on. The topmost and bottommost points of P then divide its edges into two groups; those on the side of P facing the xz-plane, and those on the opposite side. As P turns through the critical orientation and becomes obscured, the group of edges facing away from the xz-plane drops out

of P's 3-silhouette and is replaced by the other set of edges of P; thus the 3-silhouette changes in a very simple way at every critical orientation.

This last remark allows us to determine whether the noncritical orientations θ_1, θ_2 are separated by just one, or by more than one, critical orientations. To make this determination, find the two 3-silhouettes S_1, S_2 which correspond to these orientations, and then find all the edges which belong to just one of S_1 and S_2. Then θ_1 and θ_2 are separated by just one critical orientation if and only if these edges are all coplanar, bound a convex polygonal face F of P, and moreover if both S_1 and S_2 consist of an unbroken sequence of edges of F. (To test whether F is a face of P, one determines whether P lies entirely on one side of the plane in which F lies.) To verify this last assertion, note that the remarks made in the preceding paragraph imply that if θ_1 and θ_2 are separated by just one critical direction, then S_1 and S_2 must differ in the manner stated. Conversely, suppose that S_1 and S_2 differ in just this way. Then the regions of the surface of P bounded by S_1 and S_2 differ by just the face of F; hence, this is the only face that drops out of or enters into visibility (from the xz-direction) as P turns from orientation θ_1 to orientation θ_2, and therefore the only critical direction between θ_1 and θ_2 is the projected normal to the face F.

All this shows that we can proceed in the following systematic way to find all the edges and corners of P. Start at an arbitrary pair of mutually perpendicular orientations and find the 3-silhouette S of P. Given a sequence of orientations, test successive pairs $\theta_1 + \theta_2$ of orientations to see if they are separated by just one critical orientation. If not, take the average $(\theta_1 + \theta_2)/2$ of the two orientations, and repeat the test, replacing one of θ_1, θ_2 by $(\theta_1 + \theta_2)/2$. This bisection procedure will converge to a sequence of orientations each of whose pairs of successive members are separated by just one critical orientation each, and then every edge and corner of P will be visible in at least one of the silhouettes gathered. Plainly, the geometric information available from a slightly larger collection of silhouettes suffices to build a complete geometric model of the convex polyhedron P.

2.3. Theory of manipulation

The problem of manipulation is to use the fingers of one or more dextrous hands to touch, grasp, and manipulate objects. For example, we may wish to rotate a tool handle, turn a crank, grasp a door handle and move it to open a door, push a bead along a wire of unknown shape, turn a long thin section of rigid pipe using two hands, hand objects between grippers, etc. The overall aim of work in this area is to create commands, suitable for the programming of such operations, which are as general as possible. For example, a person can easily be instructed to pass his finger over a surface of known shape until he makes contact with the edge of a roughly circular hole, and then to push into the hole until a cylindrical side-passageway is felt. Moreover, a person can without difficulty grasp a handle geared in such a way as to constrain it to move along some curious eccentric path not known to him, and push the handle in a specified direction, along its natural path. At present, robots lack

capabilities of anything like this sophistication. Creating such abilities appears eminently feasible, but to do so a wide variety of issues must be faced. These include:

- Managing a mixture of geometric and force feedbacks;
- Adapting to bodies of imprecisely known shape;
- Managing geometric information, including both information known a priori, and information gathered progressively as tactile-sensing fingers move over the surface of an object being manipulated;
- Devising appropriate 'gaits' for continuing motion of a body gripped by fingers, e.g., for continuously turning a screwdriver handle. For example, during continuous rotation of a handle each of the gripping fingers in turn will approach the extreme limit of its geometric range, and must then be lifted out of contact with the body being manipulated and moved to a new point of contact with the body, along a path free of inadvertent collisions either with the body of the other fingers. For this, we must specify some appropriate 'gait' in which the fingers can walk over the surface of the turning body.
- Appropriate dictions must be found for conveniently describing coordinated motions of mechanical systems possessing many degrees of freedom. For example, the 4-finger hand being developed at the University of Utah has 20 separate degrees of freedom, and can assume most of the poses characterizing the human hand; curled around a suitcase handle, grasping a pen, exerting 2-finger grasp of a delicate slender object, etc., etc. How are the paths of fingers between these major categories of poses to be specified, especially in an environment which may contain constraining obstacles?

2.4. Control-theoretic issues

Control theory questions stand at the heart of any attempt to manipulate objects of imprecisely known shape using a robot mechanism whose internal dimensions, stiffnesses, and frictional resistances will not all be known precisely. For precise grips to be possible in the presence of these geometric and physical uncertainties, the gripping fingers (or arms) must be partly complaint, and feedback schemes taking both force and geometry into account are required. Moreover, since any control scheme will fail when the physical situation with which it is attempting to deal deviates too radically from the physical model which the feedback control assumes, the software system within which the control code operates must provide for the possibility of such exceptions, which the control code must anticipate and be prepared to classify in helpful ways. In the following pages, we will attempt to sketch at least a few of the mathematical ideas that enter into the practical treatment of these issues.

The basic question that feedback control theory deals with is this: We are given a physical system S, assumed to satisfy equations, e.g. ordinary differential equations, having a known or assumed form. We are also given a model M of the system; this model has approximately the same form as the true physical equations of the real physical system S, but does not agree with

it in every detail. However, suppose that M embodies our best available theoretical understanding of S, i.e. that we have no more precise model of S, and in particular do not know the precise details or coefficients of the equations of S. For example, if S is a robot manipulator, some of its geometric dimensions, masses, inertial moments, frictional resistances, etc., may not be known to us precisely. This discrepancy between model and reality creates the problem with which *feedback* control theory must cope, and it is essential that we not assume it away. However, we can assume that the actual behavior of S, for example the position and velocity of its parts, the force which it exerts upon an external body, etc. can be measured with high precision, or at least with precision substantially greater than the precision with which the model M approximates S. Using these measurements, and combining them with information taken from M, we can aim to control S with a precision approaching that of the available measurements, and hence substantially exceeding that of the best available model M.

It is revealing to note that this problem can be related to the most classical of all numerical techniques, namely Newton's method for solving nonlinear multiparameter equations. Given that the delicate motions we need to control can be regarded as infinitely slow ('quasistatic') the simplest way of modelling the situation we wish to address is to treat the control problem as that of solving a multidimensional nonlinear system of equations $y=F(u)$, known to us approximately but not precisely. The control system used can make use of repeated but not overfrequent evaluations of F (which correspond to measurements of the response of the system S being controlled), but must attain precision greater than that with which F is known. We can summarize our approximate knowledge of F by a function $f(u)$, which approximates $F(u)$ reasonably well, but not precisely. In a typical control-theoretic setting, we will be given a $y=Y(t)$ specified function of a time parameter t, and also an initial control value u_0 satisfying $Y(t_0)=F(u_0)$ to an adequate degree of precision. We are then required to find a function $u(t)$ of time satisfying $Y(t)=F(u(t))$ as closely as possible. Note again that we know the mathematical form only of $f(u)$, not $F(u)$; however individual values of $F(u)$ can be measured, i.e. by actually supplying a control parameter in which we can measure the resulting value of $F(u)$ no more than Δt seconds later.

If, in this situation, F (rather than merely f) were known to us precisely, the procedure of choice for calculating $u(t)$ would be Newton's method. Namely, letting f' denote the Jacobian matrix of f (where momentarily we assume $f=F$), we would put

$$u(t+\Delta t) = u(t)+(f'(u(t)))^{-1}(Y(t+\Delta t)-F(u(t))). \qquad (*)$$

(Thus using what the control literature sometimes calls 'feedforward'.)

If the 'model' represented by f agreed perfectly with 'reality' as represented by F, we would have $I-F'(u)f'(u)^{-1}=0$, so that $F(u(t+\Delta t))-Y(t+\Delta t)$ would stay quite close to zero if the initial condition $|Y(t_0)-F(u(t_0))| \ll 1$ and Y varies slowly enough for $|Y(t+\Delta t)-Y(t)| \ll 1$ to be satisfied also. Even when $f(u)$ and $F(u)$ differ, an easy argument shows that

$|F(u(t))-Y(t)|$ should remain quite small even if the 'error norm' $\epsilon = \epsilon(u) = |I - F'(u)f'(u)^{-1}| < 1$ on all (or at least most) of the path u traversed by the solution of $Y(t) = F(u(t))$. That is, even if the mathematical form of the function F is not known to us, the 'feedback' equation (*) can still be used to calculate the control $u(t+\Delta t)$ to be applied at time $t+\Delta t$, provided only that the value $F(u(t))$, i.e. the actual system state at time t, can be measured. These considerations simply identify the 'linear feedback matrix' customary in control theory with the best available estimate of $f'(a(t))^{-1}$ of the inverse Jacobian matrix of the nonlinear system we are attempting to control.

The known limitations of Newton's method necessarily appear as limitations of the control scheme (*), and warn us of the circumstances in which we must be prepared to deal with control failures. First of all, if the error norm $|I-(f')^{-1}F'|$ exceeds or approaches unity, errors will be reduced slowly or even magnified, and the controlled system will tend to oscillate; this is the well-known phenomenon of control instability. Secondly, Newton's method does not converge globally, but only locally in a sufficiently small region around the true solution of the equation being sought. Hence if the adjustment $Y(t+\Delta t)-y$ which we attempt to apply to y is too large, the Newton correction added to u will produce a new function value $F(u(t+\Delta t))$ which bears no useful relationship to the target $Y(t+\Delta t)$ which we aim to track. To cope with this difficulty we can limit the size of the adjustment $Y(t+\Delta t)-y$ attempted during any one control cycle. That is, we can choose a limit L_0 and replace the target $Y(t+\Delta t)$ which we try to reach at time $t+\Delta t$ by the modified target $L(Y(t+\Delta t)-y)+y$, where

$$L = \min(L_0, |Y(t+\Delta t)-y|).$$

If L is chosen to be some modest fraction of the norm of the tensor $f''(u(t))f'(u(t))^{-1}$, then an easy analysis of the effects of nonlinearity on the domain of convergence in Newton's method shows that as long as the error norm ϵ is considerably less than 1 the Newton correction scheme should reach a point quite close to the modified target. Of course, by temporarily modifying the target we abandon all hope of staying close to the prescribed path $Y(t)$ at all times. On the other hand, if Y seldom moves too rapidly to be followed, this gives us a reliable way of falling behind the original target Y during periods of rapid motion, but then catching up again when Y slows down. We never become hopelessly lost, as might happen as soon as the target Y moved too far from its current position y of the controlled system if we applied the Newton-method control heuristic in its simplest form.

Another important method of improving the performance of the control scheme outlined is to use the idea known to control theorists as 'adaptive control' and to numerical analysts as the 'quasi-Newton' method. Specifically, each control step applied gives us a direct measurement of the difference $F(u(t))-F(u(t+\Delta t))$. Even though this observation does not give us information concerning all the components of the unknown Jacobian matrix $F'(u(t))$, it does at least give us a way of estimating the directional derivative of F in the direction of the vector $u(t+\Delta t)-u(t)$, namely this equals

$F(u(t+\Delta t))-F(u(t))/\Delta t$. This information can be used during the next control cycle, e.g., by replacing the Jacobian matrix $J=f'(u(t+\Delta t))$ calculated using the available model, and substituting the matrix J^* which maps the vector $u(t+\Delta t)-u(t)$ into the measured change $F(u(t+\Delta t))-F(u(t))$, but which maps all vectors v orthogonal to $u(t+\Delta t)-u(t)$ into the same value Jv as J does.

Analysis (for which see [24]) shows that this idea leads to an accurate and efficient numerical scheme, a fact that can easily be confirmed in the control-theoretic context by simulation.

2.5. Two- and three-fingered manipulation of planar objects

For a typical application of the preceding ideas in a robotic context, we consider the problem of simultaneously controlling two, three, or more 'fingers' which move in the plane to grip and manipulate planar objects in user-specified ways. (E.g. the robot programmer may specify that the center of gravity G of an object O is to move along a curve $G(t)$, while the object's angular orientation varies as $\theta(t)$.) Since the two (or three) gripping fingers, which we shall model by their 'fingertips' (i.e. by points able to move anywhere in the plane) possess a total of four (or six) degrees of freedom, while the rigid body being manipulated has only three, it is clear that a mixture of positional and force control must be involved. In the case of a two-fingered grip, the law of static equilibrium (which remains applicable to the quasistatic motions we consider) demands that the forces exerted by the two fingers must be equal in magnitude and act in opposite directions along the line between the two points of finger-body contact. Similarly, if three fingers are involved, the lines of action of the three forces exerted by the fingers must be concurrent at a single point, since otherwise each of the forces would exert an unbalanced torque around the point of intersection of the lines of action of the other two. This point of concurrency must also lie inside the triangle spanned by the three points p_1, p_2, p_3 of finger/body contact, since otherwise there would clearly exist an unbalanced force component orthogonal to one of the sides of this triangle (see figure 3).

Finally, once the point p of concurrency of the three lines of force is specified, the three forces must have the form $\lambda_1(p-p_1)$, $\lambda_2(p-p_2)$, $\lambda_3(p-p_3)$, for three constants λ_1, λ_2, λ_3, and then force-balance equation $\lambda_1(p-p_1)+\lambda_2(p-p_2)+\lambda_3(p-p_3)=0$ shows that λ_1, λ_2, λ_3 stand in a fixed proportionality determined by p, more specifically that the λ_j are proportional to the barycentric coordinates of p relative to the three points p_1, p_2, p_3. The positive constant of proportionality appearing here plainly measures the total gripping force exerted.

Suppose that the compliance of the .nr 99 10i-th of n fingers is described by a positive-definite matrix K_i (which will be nonscalar if the fingertip compliance is anisotropic), and that, in the presence of frictional forces constraining the i-th finger to remain at a point z_i fixed on a rigid body that is free to rotate and translate, the point to which the i-th finger would move in the absence of any external constraint is set equal to y_i by manipulating the

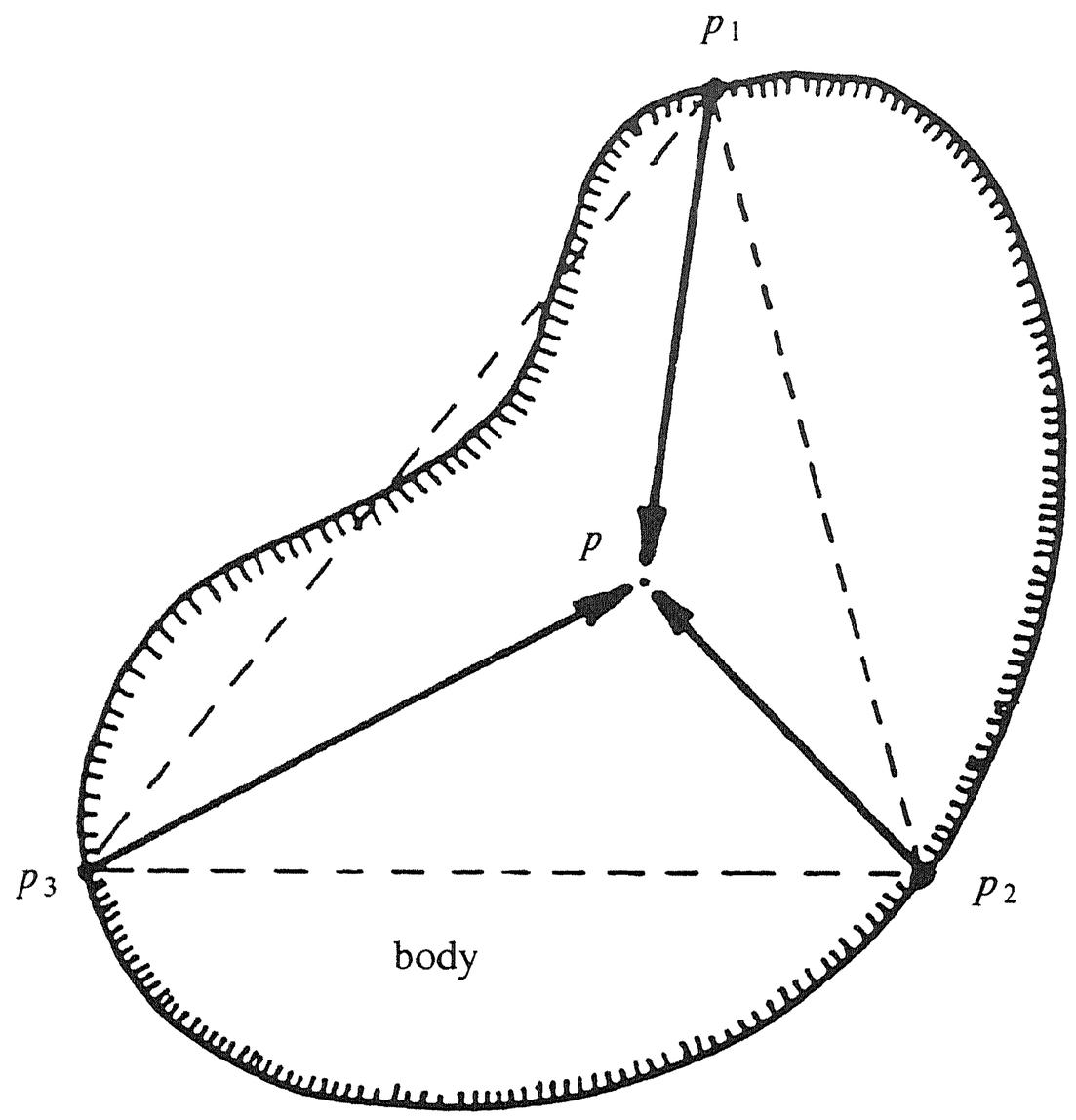


FIGURE 3. Static equilibrium of a planar body gripped frictionally at three points p_1, p_2, p_3

internal controls of the finger. The body will then rotate/translate to a new position defined by the Euclidean transformation $Ez = R(\theta)z + y$ which minimizes the energy

$$V(y_1, \ldots, y_n) = \sum_{i=1}^{n} (Ez_i - y_i) \cdot K_i (Ez_i - y_i) \tag{1}$$

Call the minimizing Euclidean transformation $E(y_1, \ldots, y_n)$, and let the force that the i-th finger exerts on the body when equilibrium is reached be $F_i(y_1, \ldots, y_n)$, so that

$$\begin{aligned} F_i(y_1, \ldots, y_n) &= \alpha_{y_i} V(y_1, \ldots, y_n) \\ &= -2 \sum_{i=1}^{n} K_i (Ez_i - y_i) + 2 \sum_{i=1}^{n} K_i \alpha y_i E(y_1, \ldots, y_n) z_i \end{aligned} \tag{2}$$

Note that in virtue of the inherent rotational invariance of the situation being considered we have

$$E(E_0y_1, \ldots, E_0y_n) = E_0E(y_1, \ldots, y_n)$$
$$V(E_0y_1, \ldots, E_0y_n) = V(y_1, \ldots, y_n)$$

and

$$F_i(E_0y_1, \ldots, E_0y_n) = (R^{-1}(\theta_0)F_i)(E_0y_1, \ldots, E_0y_n)$$

for any Euclidean transformation $E_0x = R(\theta_0)x + y_0$. The translational invariance of the energy function gives the equilibrium condition $\sum F_i = 0$, and similarly the condition that torques must balance at equilibrium follows from the rotation invariance of $V(y_1, \ldots, y_n)$.

To control the system we need to drive the quantities E and $F_1, \ldots, F_n$ along specified paths. To express the control law needed for this, we will find it useful to write a rotation through θ degrees as $\epsilon\theta J$, where J is the skew-symmetric matrix

$$J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{3}$$

Assume that the system of finger tips with which we deal is 'stiff', i.e. that the matrices K_i are all relatively large, and that the points $y_1, \ldots, y_n$ differ little from the specified fingertip positions $z_1, \ldots, z_n$. Then the amounts y, θ that the points $z_1, \ldots, z_n$ will shift and rotate are small, and so to first order in the small angle θ we can write the elastic energy $V^*(y_1, \ldots, y_n)$ stored in the fingers as

$$V^*(y_1, \ldots, y_n) = \underset{y,\theta}{\text{MIN}} \sum_{i=1}^{n} (\theta Jz_i - \eta_i - y)K_i(\theta Jz_i - \eta_i - y), \tag{4}$$

where we have written $\eta_i = y_i - z_i$. Introduce the $2n$-dimensional space E^{2n} of all vectors $y^* = (y_1, \ldots, y_n)$, and in it the symmetric form K^* and the positive definite inner product $[y^*, w^*]$ defined by

$$y^*K^*w^* = [y^*, w^*] = y_1K_1w_1 + \ldots + y_nK_nw_n, \tag{5}$$

and also define the 3-dimensional subspace S spanned by the vectors $Jz_1, \ldots, Jz_n$ and by all vectors $(y, y, \ldots, y)$, where $y \in E^2$. Then $V^*(y^*)$ is simply the squared distance of y^* from S (relative to the norm $[y^*, y^*]$). Suppose therefore that we let P_1 denote the orthogonal projection (relative to this same norm) of E^{2n} onto S, and let $P_2 = I - P_1$ be the complementary projection. Then $V^*(y^*) = [P_2y^*, P_2y^*] = y^*P_2K^*P_2y^*$ and hence the force $F_i(y^*)$ exerted on the i-th finger is $F_i(y^*) = -2E_iP_2K^*P_2y^*$, where E_i is the projection of E^{2n} onto E^2 defined by $E_i(y_1, \ldots, y_i, \ldots, y_n) = y_i$. The map $F^*(y^*) \to (F_1(y^*), \ldots, F_n(y^*))$ therefore has the representation $-2P_2K^*P_2y^*$, from which it is plain that F^* has a 3-dimensional nullspace and a range which has codimension 3 in the space of all possible n-tuples $f_1, \ldots, f_n$ of force vectors. As already noted, the translational and rotational invariance of the situation being analyzed imply that every set of force vectors in the range of F^* must satisfy three linearly independent equilibrium conditions; hence

conversely every $(f_1, \ldots, f_n)$ satisfying these equilibrium conditions belongs to the range of F^*. Moreover, the map P_1 projects the space E^{2n} in 1-1 fashion onto the subspace S of vectors of the form $((E-I)z_1, \ldots, (E-I)z_n)$, where E is any infinitesimal Euclidean transformation. Thus, if f^* denotes any set $(f_1, \ldots, f_n)$ of force vectors satisfying the conditions of static equilibrium and E any infinitesimal Euclidean transformation, while $z^* = (z_1, \ldots, z_n)$, then the equations

$$-2P_2K^*P_2\eta^* = f^*P_1\eta^* = (E-I)z^* \tag{6}$$

have a unique solution $\eta^* = (\eta_1, \ldots, \eta_n)$, and moreover if η^* is small the forces f_i and Euclidean motion E of the grasped body will be realized (to infinitesimals of second order) by putting $y_i = z_i + \eta_i$.

The manner in which a rigid body gripped by a finger can be controlled should now be clear. If the current Euclidean position and force exerted on the body are E_0, F^* then to change these to $E_1, F^* + \Delta F^*$ the necessary incremental change η_0^* in y is determined by the equations

$$-2P_2K^*P_2\eta_0^* = \Delta F^*P_1\eta_0^* = (E_1z^* - E_0z^*) \tag{7}$$

where z^* denotes the vector of positions of the gripping fingers at some initial time. If we assume that the force F^* is to follow a specified path $F^*(t)$ and the body a path $E(t)$ in position space, then on each cycle of control we measure the current vector of forces F_0^* and vector z_0^* of finger positions, and give the vector $y^* = (y_1, \ldots, y_n)$ of controls being applied the new value $(y_1 + \eta_1, \ldots, y_n + \eta_n)$, where η^* is calculated from the equations

$$-2P_2K^*P_2\eta^* = F^*(t+\Delta t) - F_0^*, \quad P_1\eta^* = E(t+\Delta t)z^* - z_0^*. \tag{8}$$

A few additional remarks explaining how to calculate P_1 and P_2 will be useful. Given a space V of vectors v with inner product $[u,v]$ and subspace V_0 spanned by vectors $v_1, \ldots, v_k$, we can write the orthogonal projection P of V onto V_0 as

$$Pv = \sum_{i,j} a_{ij}v_i[v_j, v], \tag{9}$$

where a_{ij} is simply the inverse of the matrix $[v_j, v_k]$, i.e., $\sum_j a_{ij}[v_j, v_k] = \delta_{ik}$. Hence if the inner product has the form $[u,v] = \sum K_{ij}^* u_i u_j$, then P is just the matrix $P_{mn} = \sum_{i,j,k} a_{ij}v_i^{(m)}v_j^{(k)}K_{kn}$, where $v_i^{(m)}$ is the m-th component of the i-th basis vector of V_0, and a_{ij} is the inverse of $\sum_{m,n} K_{mn}v_i^{(m)}v_j^{(n)}$. Calculation of the projections P_1, P_2 appearing in the preceding discussion is therefore straightforward.

The preceding paragraphs are phased to apply to the control of fingers exerting a slip-free grip on a two-dimensional object moving in the plane. However, only a few details need to be modified to adapt this analysis to frictionally gripped solid bodies moving in three dimensions. In the 3-dimensional case equation (4) becomes

$$V^*(y_1, \ldots, y_n) = \underset{y,L}{\mathrm{MIN}} \sum_{i=1}^{n} (Lz_i - \eta_i - y)K_i(Lz_i - \eta_i - y), \tag{10}$$

where L denotes an arbitrary real skew-symmetric matrix, corresponding to an infinitesimal rotation in E^3. The space of vectors $y^* = (y_1, \ldots, y_n)$ is now $3n$-dimensional rather than $2n$-dimensional, i.e. E^{3n} rather than E^{2n}, and the subspace S spanned by $(Lz_1, \ldots, Lz_n)$ and $(y, \ldots, y)$ is six- rather than three-dimensional, but otherwise the definition of the projections P_1 and P_2 changes hardly at all. The map $y^* \to F^*(y^*)$ has the same representation as previously, but now has a 6-codimensional rather than a 3-codimensional range. Equations (6) and (7) remain valid. Hence again we step from one control cycle to the next by measuring the vector F_0^* of forces exerted by the fingers and the vector of finger positions, and change the vector $y^* = (y_1, \ldots, y_n)$ of controls being applied to $(y_1 + \eta_1, \ldots, y_n + \eta_n)$, where η^* is calculated from the equation $P_2 K^* P_2 \eta^* = F^*(t + \Delta t) - F_0^*$, $P_1 \eta^* = E(t + \Delta t) z^* - z_0$.

References

1. G.J. Agin, T.O. Binford (1973). Computer description of curved objects. *Third International Joint Conference on Artificial Intelligence,* 629-640.
2. G.J. Agin, R.O. Duda. SRI vision research for advanced industrial automation. *2nd USA-Japan Computer Conference,* Tokyo.
3. R.C. Bolles, R.A. Cain (1982). Recognizing and locating partially visible objects: The local-feature-focus method. *Robotics Research 1, 3,* 57-82.
4. R. Bolles, J. Kremers, R. Cain. *A Simple Sensor to Gather 3-dimensional Data,* Technical Note 249, Industrial Automation Department, Stanford Research Institute, Menlo Park, Calif.
5. R.C. Bolles, P. Horaud, M.J. Hannah (1983). 3DPO: A three-dimensional part orientation system. *First International Symposium of Robotics Research,* Bretton Woods, N.H.
6. M. Briot (1979). The utilization of an 'artificial skin' sensor for the identification of solid objects. *Ninth ISIR,* Washington, D.C., 529-548.
7. R. Burridge, V.T. Rajan, J.T. Schwartz (1984). *The Peg-in-hole problem: The Phenomena of Sticking and Jamming for Nearly Rigid Bodies in Frictional Contact, and the Transition from Static to Dynamic Behavior. I. Motions in Two Dimensions,* NYU Technical Report.
8. H.G. Barrow, J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. A.R. Hanson, E.M. Riseman (eds.). *Computer Vision Systems,* 3-26, Academic Press, New York, N.Y.
9. O.D. Faugeras, M. Hebert (1983). A 3D recognition and positioning algorithm using geometrical matching between primitive surfaces. *Eighth International Joint Conference on Artificial Intelligence,* 996-1002.
10. P. Gaston, T. Lozano-Perez (1983). *Tactile Recognition of Localization Using Object Models,* MIT Artificial Intelligence Laboratory Report AIM-705.

11. W.E.L. GRIMSON, T. LOZANO-PEREZ (1983). *A combinatorial analysis of recognition and localization using object models.* MIT Artificial Intelligence Laboratory Technical Report.
12. W.E.L. GRIMSON, T. LOZANO-PEREZ (1983). *Model-based Recognition and Localization from Sparse Range or Tactile Data,* A.I. Memo 738, MIT Artificial Intelligence Laboratory.
13. L.D. HARMON (1982). Automated tactile sensing. *International Journal of Robotics Res. 1,* 3-32.
14. W.D. HILLIS (1981). A high-resolution image touch sensor. *International Journal of Robotics Res. 1,* 33-44. See also: *Active Touch Sensing,* S.M. Thesis, Dept. of Electrical Engineering and Computer Science, Mass. Inst. of Technology.
15. M.J. HANNAH. *Computer Matching of Areas in Stereo Images,* Memo AIM-239, Stanford Artificial Intelligence Laboratory, Stanford University, Stanford, Calif.
16. S. HACKWOOD, G. BENI, L. HORNAK, R. WOLFE, T. NELSON (1983). A torque-sensitive tactile array for robotics. *International Journal of Robotics Research 2,* 46-50.
17. B.K.P. HORN. Obtaining shape from shading information. P.H. WINSTON (ed.). *The Psychology of Computer Vision,* McGraw-Hill Book Company, New York, N.Y.
18. B.K.P. HORN. Understanding image intensities. *Artificial Intelligence, 8,* 201-231.
19. R.A. JARVIS (1983). A perspective on range finding techniques for computer vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence* PAMI-5, 2, 122-139.
20. A.R. JOHNSTON. *Infrared Laser Rangefinder,* NASA New Technology Report No. NPO-13460, Jet Propulsion Laboratory, Pasadena, Calif.
21. S. JACOBSEN, J. WOOD, D. KNUTTI, K. BIGGERS (1983). The Utah/MIT dextrous hand, *MIT/SDF International Robotics Research Symposium,* Bretton Woods, N.H.
22. G. KINOSHITA, S. AIDA, M. MORI (1975). Pattern classification by dynamic tactile sense information processing. *Pattern Recognition 7,* 243.
23. R.A. LEWIS, A.R. JOHNSTON (1977). A scanning laser range finder for a robotic vehicle. *Fifth Intl. Joint Conf. on Artificial Intelligence,* 762-768.
24. W. MURRAY (1972). *Numerical Methods for Unconstrained Optimization,* Academic Press.
25. T. OKADA, S. TSUCHIYA (1977). Object recognition by grasping. *Pattern Recognition 9, 3,* 111-119.
26. M. OSHIMA, Y. SHIRAI (1978). A scene description method using three-dimensional information. *Pattern Recognition 11,* 9-17.
27. M. OSHIMA, Y. SHIRAI (1983). Object recognition using three-dimensional information. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI-5, 4, 353-361.

28. H. OZAKI, S. WAKU, A. MOHRI, M. TAKATA (1982). Pattern recognition of a grasped object by unit-vector distribution. *IEEE Transactions on Systems, Man, and Cybernetics,* SMC-12, 3, 315-324.
29. C. O'DUNLAING, C. YAP (March 1983). *A Voronoi Diagram Method of Motion-planning,* NYU Technical Report.
30. C.J. PAGE, A. PUGH, W.B. HEGINBOTHAM (1976). Novel techniques for tactile sensing in a three-dimensional environment. *Sixth ISIR,* University of Nottingham.
31. R.J. POPPLESTONE ET AL. Forming models of plane- and cylinder-faceted bodies from light stripes. *Proceedings of the 4th International Joint Conf. on Artificial Intelligence,* Tsiblisi, Georgia, USSR, 664-668.
32. C.A. ROSEN, D. NITZAN. Use of sensors in programmable automation. *IEEE Computer, 10.*
33. C.A. ROSEN. Combined ranging and color sensor. *U.S. Patent No.3,* 945, 729.
34. M.H. RAIBERT, J.E. TANNER (1982). Design and implementation of a VLSI tactile sensing computer. *International Journal of Robotics Res. 1,* 3-18.
35. M. SHARIR, E. ARIEL-SHEFFI (February 1983). *On the piano movers' problem IV.* Various decomposable two-dimensional motion planning problems, NYU Technical Report.
36. J. SALISBURY, J. CRAIG (1982). Articulated hands: force control and kinematic issues. *International Journal of Robotics Res. 1,* 4-17.
37. J.T. SCHWARTZ, D. GROSSMAN (1983). The Next Generation of Robots, in *Frontiers in Science and Technology: a Report by the Committee on Science, Engineering, and Public Policy of the National Academy of Sciences,* W.H. Freeman and Co.
38. Y. SHIRAI, M. SUWA. Recognition of polyhedra with a range finder. *Proc. 2nd Int. Joint Conf. on Artificial Intelligence,* 80-87.
39. J.T. SCHWARTZ, M. SHARIR (1983). On the piano movers' problem I. The case of a rigid polygonal body moving amidst polygonal obstacles. *Comm. Pure and Appl. Math. 36,* 345-398.
40. J.T. SCHWARTZ, M. SHARIR (1983). On the piano movers' problem II. General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math. 4,* 298-351.
41. J.T. SCHWARTZ, M. SHARIR (1982). On the piano movers' problem III. Coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal obstacles. *International Journal of Robotics 2,* 46-75, Report.
42. J.T. SCHWARTZ, M. SHARIR (August 1983). *On the piano movers' problem V.* The case of a rod moving in three-dimensional space amidst polyhedral obstacles, NYU Technical Report.

Algebra of Communicating Processes

J.A. Bergstra
J.W. Klop
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

A survey of process algebra is presented including the following features: merging processes without communication, merging processes with communication, data flow networks, regular processes, recursively defined processes, abstraction mechanisms both in absence and presence of communication. Throughout the paper emphasis is on equational specifications and graph theoretic models.

INTRODUCTION

It is widely recognized that Milner's CCS constitutes a fundamental contribution to the theory of concurrency. Milner's aim is to establish laws for concurrent processes in the form of algebraic identities. We view process algebra, as developed in [3],...,[10], as a rephrasing of the basic issues of CCS. For a motivation of CCS as a theory of concurrency we refer to MILNER [14], [15]. We will not assume that the reader knows CCS. The differences with CCS in aims and techniques can be summarized as follows:

(1) We use these operators and constants:

$+$	alternative composition (sum)
$\cdot$	sequential composition (product)
$\|\|$	parallel composition (merge)
$\lfloor\!\lfloor$	left merge
$\mid$	communication merge
∂_H	encapsulation
τ_I	abstraction
δ	deadlock (failure)
τ	silent (internal) action

TABLE 1

We will briefly discuss how these operators relate to CCS. The operators $+$, $\|$, and τ have exactly the same meaning; multiplication $\cdot$ is more

general than the prefix multiplication of CCS; $\lfloor\!\lfloor$ and $\mid$ are new; δ is similar to NIL in sums (but not in products). ∂_H and τ_I are new operators. (However these are formally renaming operators in the sense of CCS.)

(2) This choice of operators allows a finite initial algebra specification of the behaviour of finite processes. Seen from CCS, $\lfloor\!\lfloor$ and $\mid$ are hidden operators involved in this specification. We feel however that $\lfloor\!\lfloor$ and $\mid$ are perfectly meaningful from an intuitive point of view.
Our presentation culminates in a system of equations ACP_τ, and passes through several smaller specifications (PA, PA_τ, ACP) involving only a subset of the operators.

(3) ACP chooses from the onset the axiomatic approach. Thus, where CCS starts with a model of processes and derives identities in that model as theorems, ACP reverses this procedure: a set of axioms is given first and its models are investigated next. In the course of our investigations we have met some twenty interesting process algebras (interesting as opposed to pathological; the axiomatic approach allows also some less useful models) and since there are so many it seems sensible to organize them as models of some axiomatic theory.

(4) We claim that ACP is more amenable to a mathematical analysis than CCS (in its original form). As an example we would like to point out the simple formulation of the Expansion Theorem (2.2), and the specification of a Stack in subsection 3.5.

The core of this presentation is the system ACP. Infinite models for ACP are constructed as projective limits of finite models, and as graph models modulo bisimulation. The projective limit models have been derived from the topological construction in DE BAKKER and ZUCKER [1], [2]. The work on process algebra originated from a problem in [2] (page 87) which was solved in [3] thereby essentially using the algebraic properties of $\lfloor\!\lfloor$. (See 1.9 below.)

ACKNOWLEDGEMENTS

The material in the first three of the four sections of this paper was presented in the *Workshop on Concurrency and Abstract Data Types* (Mook, October 1983) organized by W.P. DE ROEVER. We thank him for giving us the opportunity to organize the present material in a set of lecture notes which was the basis for this paper.

Furthermore, we thank J.W. DE BAKKER for his continuous support and J.-J.CH. MEYER, J.V. TUCKER, J. TIURYN, E. BRINKSMA, C.J. KOOMEN, H. JONKERS and H. OBBINK for many discussions on the subject of this paper.

Most of the following material has been covered in more detail in our reports [3],...,[10]. Section 4 contains new results, centering around ACP_τ, an axiom system for communicating processes with internal steps. Almost all proofs are omitted - these can be found in the above mentioned reports (except for most of Section 4).

The structure of this paper is as follows:

1. Process algebra: PA

In this section we will introduce the axiom systems PA for process algebra without communication (treated in Section 2) and without internal steps (treated in Section 4). The co-operation between processes described by PA is that of *interleaving*. As semantics for PA several 'process algebras' will be introduced of which the simplest one is the initial algebra of PA.

1.1. The axiom system PA

The axiom system PA consists of the following list of axioms:

$x+y = y+x$	A1
$x+(y+z) = (x+y)+z$	A2
$x+x = x$	A3
$(x+y)\cdot z = x\cdot z+y\cdot z$	A4
$(x\cdot y)\cdot z = x\cdot(y\cdot z)$	A5
$x\|y = x\lfloor y+y\lfloor x$	M1
$a\lfloor x = a\cdot x$	M2
$ax\lfloor y = a(x\|y)$	M3
$(x+y)\lfloor z = x\lfloor z+y\lfloor z$	M4

Table 2

1.1.1. The signature of PA. The signature of PA consists of the following ingredients:

(i) $a,b,c, \ldots \in A$, the set of *axiomatic actions* (also called 'steps' or 'events'). A is also referred to as the *alphabet*. Throughout this paper, we will assume that A is *finite*. (This is done to safeguard the algebraic nature of our considerations — e.g. infinite sums of processes are not considered here.) In the axioms of PA, 'a' varies over A.

(ii) $x,y,z, \ldots$ are *variables*, ranging over the domains of processes (process algebras) which will be constructed below.

(iii) binary *operators*. These are:

$+$ alternative composition, or sum
$\cdot$ sequential composition, or product
$\|$ parallel composition, or merge
$\lfloor$ left-merge.

The 'main' operators are $+,\cdot,\|$. Left-merge $\lfloor\!\lfloor$ is an auxiliary operator.

1.1.2. Process expressions. Process expressions or process terms are built from the $a \in A$ by means of $+,\cdot, |, \lfloor\!\lfloor$. Examples of process expressions are:

$$(a+b), \quad ((((a\cdot a)\lfloor\!\lfloor b) + (c\cdot d))\cdot e).$$

The following notational conventions will be employed: xy stands for $x\cdot y$; outermost brackets are omitted; the operator $\cdot$ has the greatest binding power; x^n stands for $xx \ldots x$ (n times); $\|$ and $\lfloor\!\lfloor$ bind stronger than $+$. So the two process expressions above may be written as

$$a + b, \quad (a^2\lfloor\!\lfloor b + cd)e.$$

1.2. Semantics of PA

A *process algebra* is a domain of processes satisfying the axioms of *PA*. The three most important process algebras for *PA* are:

(1) A_ω, *the initial algebra of PA,*
(2) $\mathbf{A}^\infty$, *the graph model of PA,*
(3) A^∞, *the standard model of PA*

It will turn out that these algebras properly extend each other (modulo isomorphism): $A_\omega \subsetneq \mathbf{A}^\infty \subsetneq A^\infty$.

1.2.1. The initial algebra A_ω. The elements of A_ω are the *process expressions modulo the equivalence given by PA.* So, in A_ω, '$a+b$' and '$b+a$' and '$a+b+a$' are the same. Likewise, the process expressions $((aa\lfloor\!\lfloor b+cd)e$ and $a(abe+bae)+cde$ denote the same element in A_ω, since using *PA* one computes

$$(aa\lfloor\!\lfloor b+cd)e = (aa\lfloor\!\lfloor b)e+cde = a(a\|b)e+cde =$$
$$a(a\lfloor\!\lfloor b+b\lfloor\!\lfloor a)e+cde = a(ab+ba)e+cde =$$
$$a(abe+bae)+cde.$$

Note that this derivation has eliminated the $\|$, $\lfloor\!\lfloor$ operators in the original process term. We have the following general fact:

THEOREM 1.1.
(i) *Using the axioms of PA as rewrite rules from left to right, every process expression can be rewritten to an expression without* $\|$ *or* $\lfloor\!\lfloor$.
(ii) *If* $PA \vdash t_1 = t_2$ *and* t_1, t_2 *do not contain* $\|$, $\lfloor\!\lfloor$, *then* $A1\text{-}5 \vdash t_1 = t_2$.

This entails that elements of the initial algebra A_ω can be thought of as process expressions built from atoms via $+$ and $\cdot$ only, modulo A1-5. Using this fact we arrive at a convenient representation of elements of A_ω:

PROPOSITION 1.1. *Modulo PA-equivalence, A_ω is inductively generated as follows:*

$$x_i \in A_\omega, \quad a_i \in A \ (i = 1, \ldots, n), \quad b_j \in A (j = 1, \ldots, m) \Rightarrow \sum_{j=1}^{m} b_j + \sum_{i=1}^{n} a_i x_i \in A_\omega.$$

EXAMPLE 1.1.

$$bab \| ab = bab \lfloor\!\lfloor ab + ab \lfloor\!\lfloor bab = b(ab \| ab) + a(b \| bab) =$$
$$b(ab \lfloor\!\lfloor ab + ab \lfloor\!\lfloor ab) + a(b \lfloor\!\lfloor bab + bab \lfloor\!\lfloor b) =$$
$$b(ab \lfloor\!\lfloor ab) + a(bbab + b(ab \| b)) =$$
$$b(a(b \| ab)) + a(bbab + (b(ab \lfloor\!\lfloor b + b \lfloor\!\lfloor ab)) =$$
$$b(a(bab + abb)) + a(bbab + b(abb + bab)).$$

Expressions like the last one, without $\|$ and $\lfloor\!\lfloor$, can conveniently be 'pictured' as finite trees:

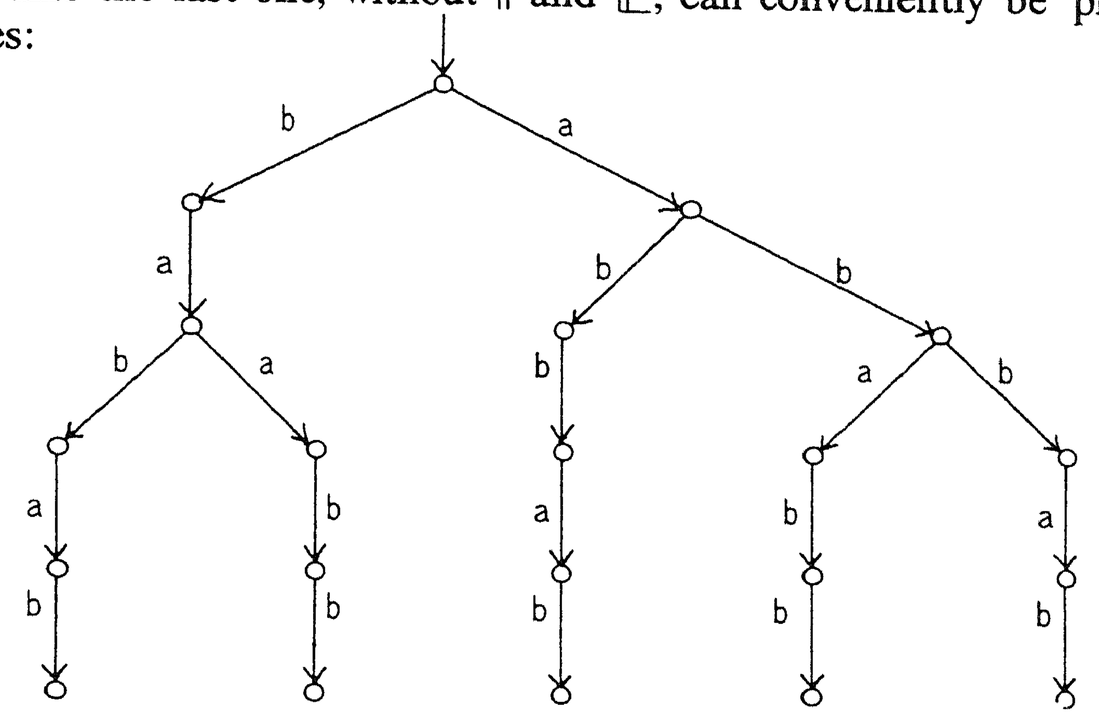


FIGURE 1

Let us note here (in advance to the definition of $\|$ for process graphs later on in this section) that the tree above, resulting from the interleaving of *bab* and *ab*, can be obtained quickly by 'unraveling' the cartesian product graph

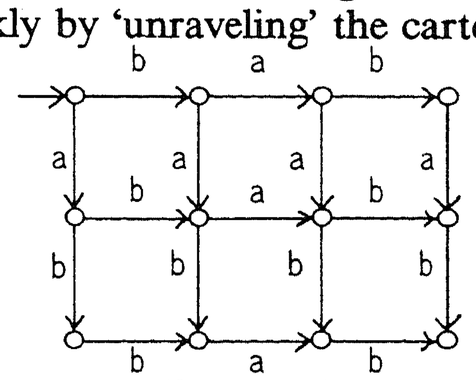


FIGURE 2

Vice versa, the above tree yields this product graph by identifying some nodes with identical subtrees. We will return to such process trees and graphs later.

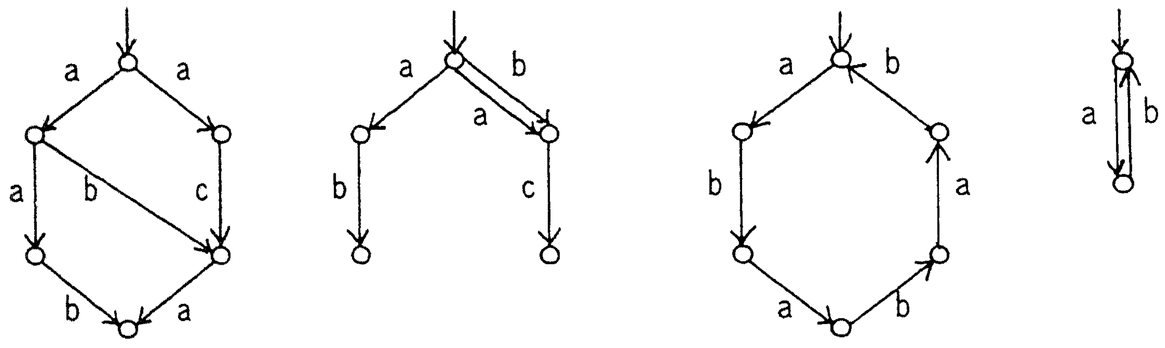
Note that *PA* does not contain the distributive law $x(y+z) = xy+xz$.

Indeed, for pairwise different atoms a,b,c the processes $a(b+c)$ and $ab+ac$ are different in A_ω. (Cf. also Example 2.2.)

We have now constructed our first process algebra as semantics of PA, the initial algebra A_ω, whose elements can also be thought of as *finitely branching and finitely deep process trees.* The fact that the processes in A_ω are only finitely deep, means that we cannot find solutions p in A_ω for recursive definitions like $p = ap$; for, p would be $aaaa$ ... or a^ω.

Therefore we will now construct process algebras which do have infinite elements, and in which solutions of recursion equations can be found.

1.2.2. The process graph model $\mathbf{A}^\infty$. A *process graph* (also called: *transition diagram*) over a set of atoms A is a *rooted, directed multigraph* whose edges are labeled by elements of A. Process graphs may be infinite and may contain cycles. *Process trees* are special cases: they are acyclic process graphs where no subgraph is shared (and containing no multiple edges); in other words, where no two edges have the same end-point. Some examples will clarify these concepts.



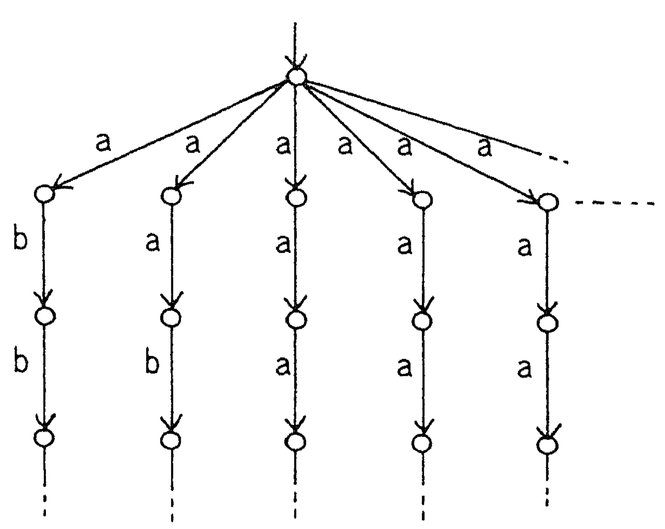


FIGURE 3

Here (i), (ii) are finite acyclic process graphs, but not trees; (iii), (iv) are finite process graphs containing cycles; (v) is an infinite process graph containing cycles and (vi) is an infinite process tree.

To construct our second process algebra $\mathbf{A}^{\infty}$, we will restrict ourselves to *finitely branching* process graphs. (This also puts a bound on the cardinality of the edges and nodes of such graphs.)

Having this large collection of finitely branching process graphs available, we note that there are 'too many' of them — some process graphs should be identified. E.g. the five graphs in figure 4 all seem to denote the same process: in each node ('state of the process') there are in all five cases infinitely many a-steps possible.

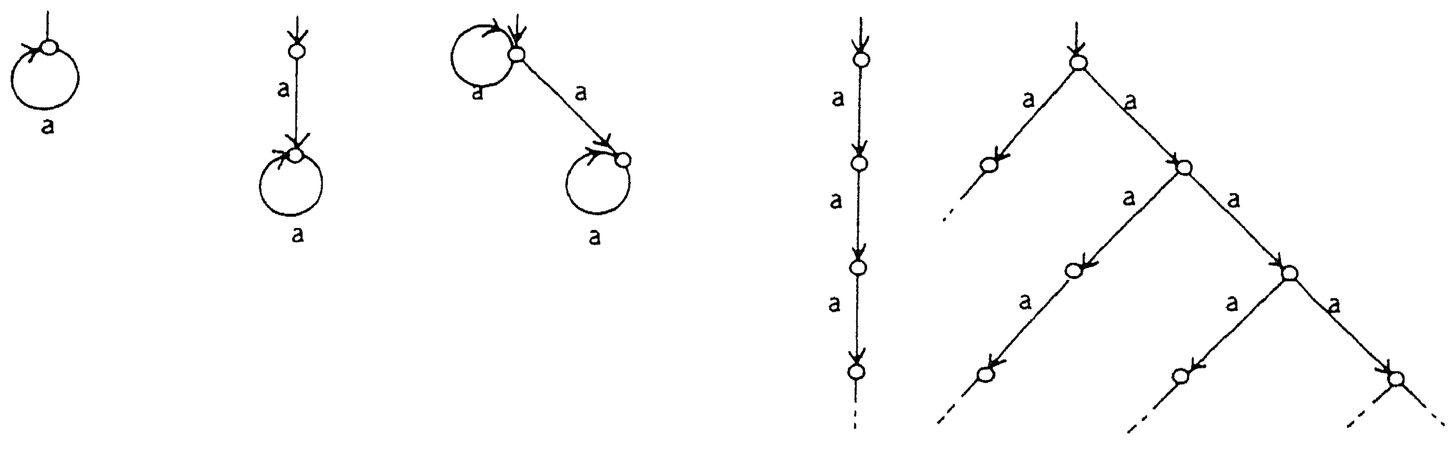


FIGURE 4

An elegant notion, introduced in PARK [16], called *bisimulation*, does indeed identify these graphs.

1.2.2.1. Bisimulation of process graphs. Bisimulation of process graphs is defined as follows.

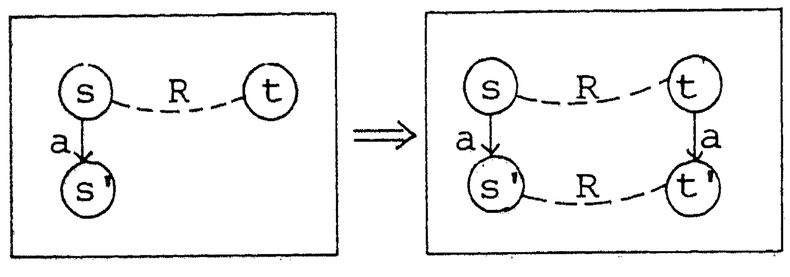
Let g_1,g_2 be process graphs with node sets Nodes (g_1), Nodes (g_2). Let s_0,t_0 be the roots of g_1,g_2 respectively. Then g_1,g_2 *are bisimilar,* in symbols:

$$g_1 \leftrightarrows g_2$$

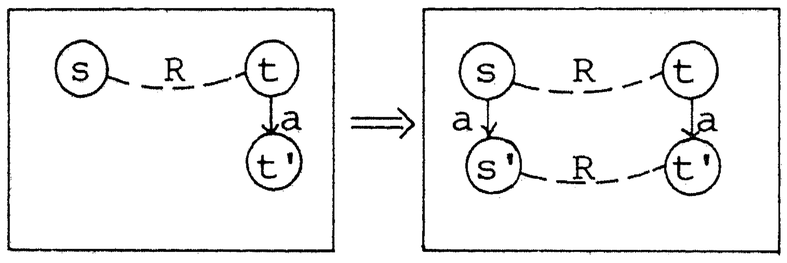
if there is a relation $R \subseteq$ Nodes $(g_1) \times$ Nodes (g_2) such that

(i) $s_0 R t_0$ (the roots are related)

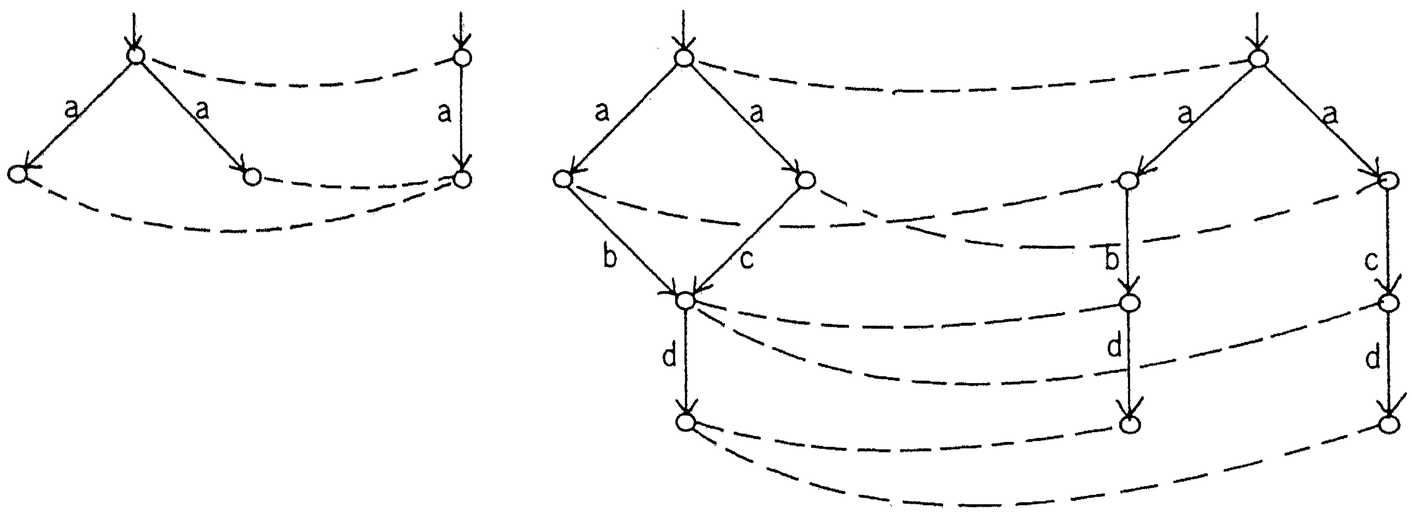
(ii) if $s \xrightarrow{a} s'$ is an edge of g_1 and sRt, there must be an edge $t \xrightarrow{a} t'$ of g_2 such that $s'Rt'$. In a diagram:



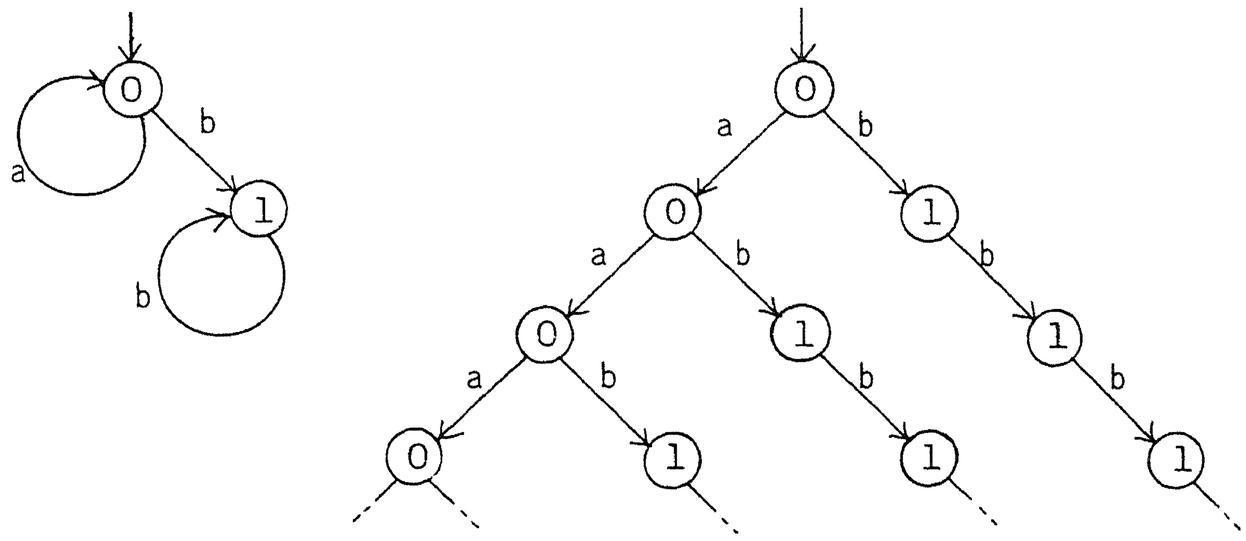
(iii) vice versa (with the role of g_1,g_2 interchanged):



EXAMPLES.



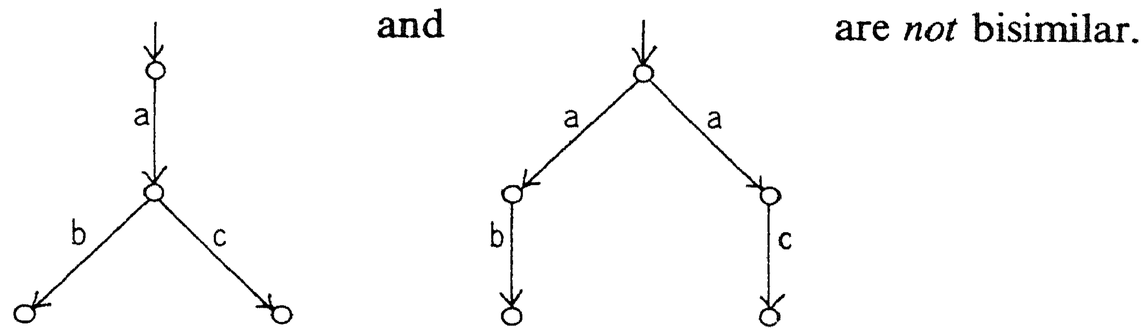
(i) (ii)



(iii)

(In figure 5 (iii) the bisimulation is given by the numbering of the nodes.)

(iv) A non-example:



are *not* bisimilar.

FIGURE 5

(Cf. our earlier remark that $A_\omega \not\vdash a(b+c) = ab+ac$.) Note that *unfolding* (or unwinding) a process graph respects bisimilarity. The same holds for *sharing* (identifying nodes with identical subgraphs).

We call the process graph with one node and no edges, the *trivial* graph. A node lying on a cycle is a *cyclic* node.

Now the second process algebra for *PA*, called the process graph algebra $\mathbf{A}^\infty$, is defined as follows.

The elements of $\mathbf{A}^\infty$ are the *finitely branching, nontrivial process graphs with acyclic roots modulo bisimulation.*

The operations $+,\cdot,\|,\lfloor\!\lfloor$ *on* $\mathbf{A}^\infty$ *are defined thus:*
(i) The *sum* g_1+g_2 is obtained by identifying the roots of g_1,g_2. E.g.:

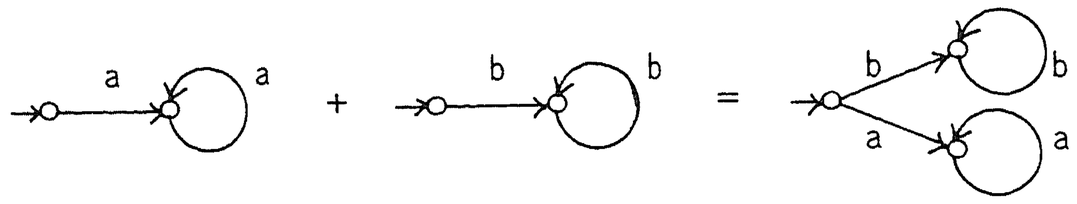


FIGURE 6

This example indicates why the roots have to be acyclic: otherwise

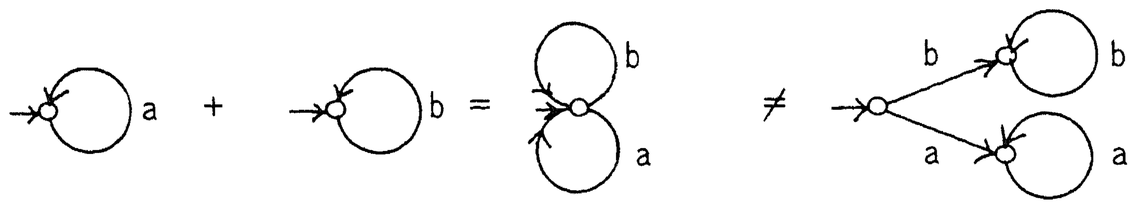


FIGURE 7

(ii) The *product* $g_1 \cdot g_2$ is obtained by appending g_2 to all end nodes of g_1.
(iii) The *merge* $g_1 \| g_2$ is the cartesian product graph as in the example:

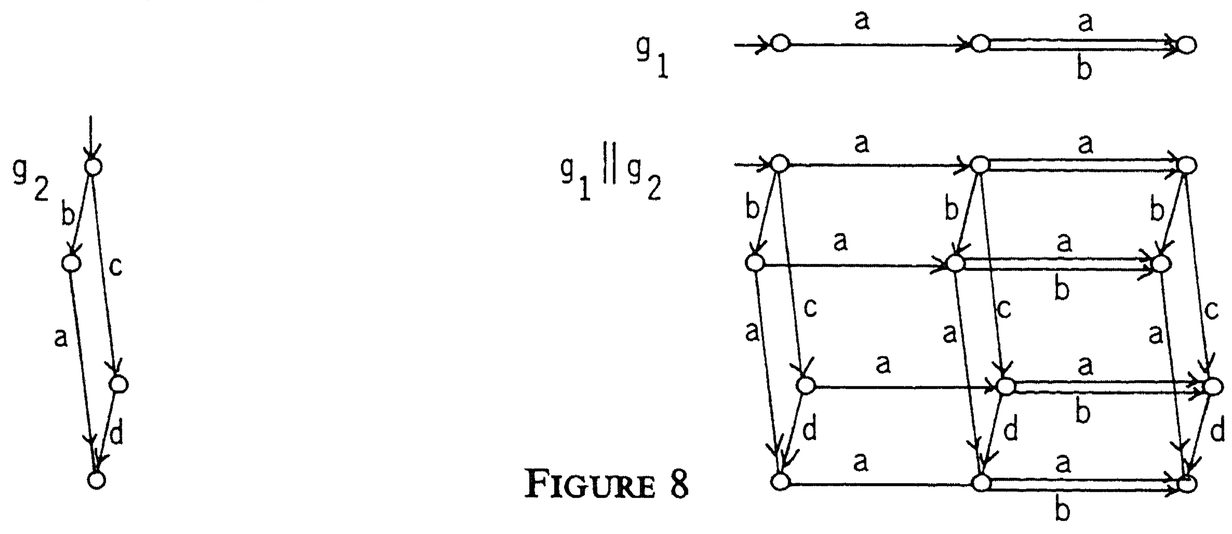


FIGURE 8

(iv) The *left-merge* $g_1 \lfloor\!\lfloor g_2$ is obtained as a subgraph or $g_1 \| g_2$ as in the example:

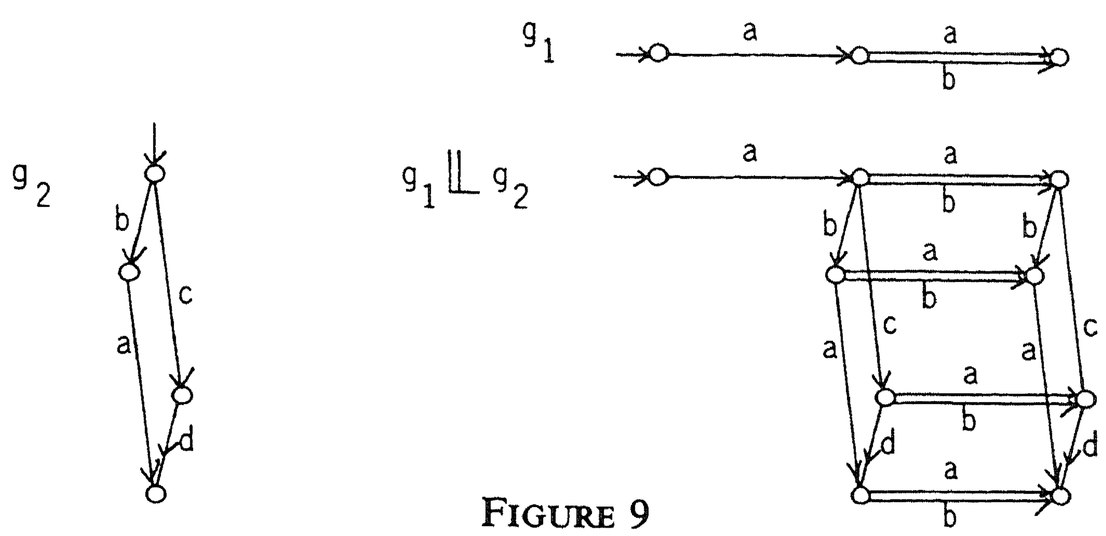


FIGURE 9

It is now easy to prove the following theorem.

THEOREM 1.2.
(i) $\mathbf{A}^\infty$ *is a process algebra (a model of PA).*
(ii) *The finite acyclic elements in* $\mathbf{A}^\infty$ *constitute a subalgebra which is (isomorphic to)* A_ω.

1.2.2.2. Approximations of processes in $\mathbf{A}^\infty$. There is a clear sense in which a (possibly infinite) process tree t can be approximated by finite process trees $(t)_n$ $(n \geqslant 1)$: $(t)_n$ is t where everything below level n is cut-off. (I.e. the branches of $(t)_n$ have at most n steps.) This notion of projection induces one in $\mathbf{A}^\infty$ in a straightforward manner: writing $[\![g]\!]$ for the bisimulation equivalence class of the process graph g (so $[\![g]\!] \in \mathbf{A}^\infty$, if g is nontrivial etc.), we define

$$([\![g]\!])_n = [\![(tree(g))_n]\!]$$

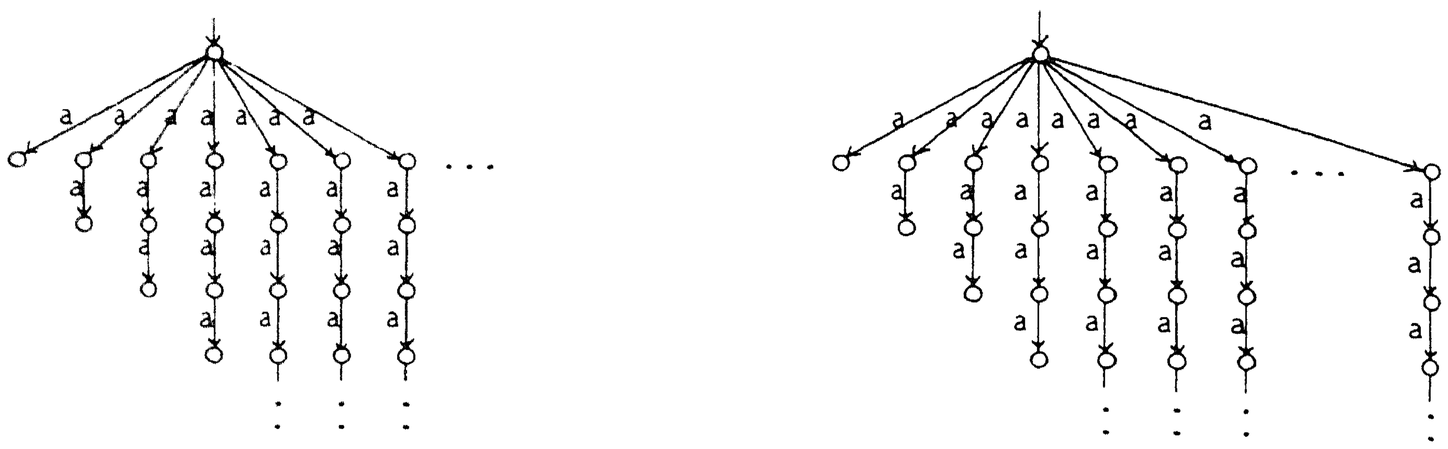
where $tree(g)$ is a tree obtained by unwinding g. To establish the precise definition of $tree(g)$ and the well-definedness of the projection operation $(\ )_n$: $\mathbf{A}^\infty \to \mathbf{A}^\infty$ is a matter of routine. From now on, we write simply g instead of $[\![g]\!]$ when dealing with elements of $\mathbf{A}^\infty$. There is the following interesting fact:

THEOREM 1.3. *Let* $g,h \in \mathbf{A}^\infty$. *Then* $g = h \Leftrightarrow \forall n\ (g)_n = (h)_n$.

So equality between finitely branching graphs (modulo bisimulation) is entirely determined by their finite approximations — i.o.w. a finitely branching graph modulo bisimulation is determined by its finite approximations.

The implication $\Rightarrow$ in this theorem is trivial; the proof of the reverse implication consists of an application of König's Lemma made possible by the condition that elements in $\mathbf{A}^\infty$ are (equivalence classes of) *finitely branching* graphs. (This is used as follows: construct the tree of all bisimulations between $(g)_n$ and $(h)_n$, for all $n \geqslant 1$. That is, on the n-th level are the bisimulations between $(g)_n$ and $(h)_n$. Ordering in the tree is: extension of bisimulations in the set-theoretic sense. Since this tree is finitely branching and infinite, it has an infinite branch which yields a bisimulation for the pair g,h.) That this condition is essential for the proposition in the theorem, follows from a

consideration of these two process graphs which have the same finite approximations:



(i) (ii)

FIGURE 10

Although the elements of $\mathbf{A}^\infty$ are attractive objects, they are notoriously lacking in algebraic nature. On the basis of our intuitive understanding of the graph model $\mathbf{A}^\infty$, we will now construct a process algebra for PA which is algebraic in nature and which will be called the standard model $\mathbf{A}^\infty$ for PA.

1.2.3. The standard model A^∞ for PA. Bearing in mind that an element $g \in \mathbf{A}^\infty$ gives rise to a sequence $((g)_1,(g)_2, \ldots)$ of approximations which by the previous theorem (1.3) determines g and for which we obviously have: $(g)_n = ((g)_{n+1})_n$, we now define without any reference to graphs: A *projective sequence* is a sequence $(p_1,p_2,p_3, \ldots, p_n, \ldots)$ of elements of A_ω such that $p_n = (p_{n+1})_n$. Here the projections $(\ )_n : A_\omega \to A_\omega$ $(n \geqslant 1)$ are defined by

$$(a)_n = a$$

$$(ax)_1 = a, \quad (ax)_{n+1} = a(x)_n$$

$$(x+y)_n = (x)_n + (y)_n.$$

Furthermore we define: *the elements of A^∞ are the projective sequences.* The operations $+, \cdot, \|, \lfloor\!\lfloor$ on A^∞ are defined coordinate-wise, thus:

$$(p_1,p_2, \ldots, p_n, \ldots) \,\square\, (q_1,q_2, \ldots, q_n, \ldots,) =$$

$$((p_1 \,\square\, q_1)_1, (p_2 \,\square\, q_2)_2, \ldots, (p_n \,\square\, q_n)_n, \ldots)$$

where $\square \in \{+, \cdot, \|, \lfloor\!\lfloor\}$. Note the outermost subscripts in the RHS, necessary to ensure that the result from applying the operation $\square$ is again a projective sequence. (The simple proof employs the fact that $(p \,\square\, q)_n = ((p)_n \,\square\, (q)_n)_n$.)

EXAMPLE 1.2.

(i) The atomic action 'a' is represented by $(a,a,a, \ldots)$.

(ii) $(a,\ a+a^2,\ a+a^2+a^3,\ \ldots,\ \Sigma_{i=1}^{n}a^i,\ \ldots)\in A^\infty$. We will refer to this element as $\Sigma_{i=1}^{\infty}a^i$. (Note however that except for this ad hoc notation we will not use infinite sums.)

(iii) Call $a^\omega=(a,a^2,a^3,\ \ldots)$. Then $a^\omega\cdot b^\omega=((a\cdot b)_1,$ $(a^2\cdot b^2)_2,\ \ldots)=a^\omega$.

(iv) $a^\omega\|b^\omega=((a\|b)_1,\ (a^2\|b^2)_2,\ \ldots)=(a+b,\ a(a+b)+b(a+b),\ \ldots)$ $=(a+b,\ (a+b)^2,\ \ldots)=(a+b)^\omega$.

(v) $[(a^\omega\|b^\omega)+a^\omega]\|b^\omega=a^\omega\|b^\omega$.

Again it is straightforward to verify that A^∞ is a model of PA.

A natural question is how A^∞ and $\mathbf{A}^\infty$ compare. The answer is that A^∞ is an extension of $\mathbf{A}^\infty$: it contains all the processes in $\mathbf{A}^\infty$ (modulo an isomorphism) but also some processes which are not finitely branching, like $\Sigma_{i=1}^{\infty}a^i$ above. Strictly speaking, we have not yet defined when an element of A^∞, a projective sequence, is finitely branching.

This can be done by assigning to a $p\in A^\infty$ a process graph $G(p)$, as follows. First we define what a 'subprocess' of $p\in A^\infty$ is.

The collection of subprocesses of p is given by

(i) $p\in Sub(p)$,

(ii) $ax\in Sub(p)\Rightarrow x\in Sub(p)$,

(iii) $ax+y\in Sub(p)\Rightarrow x\in Sub(p)$.

From the subprocesses of p (which may be thought of as the nonterminal states of the process) we can assemble a process graph $G(p)$. This process graph will be called *the canonical process graph* $G(p)$ *for* p. It is defined as follows:

(i) the set of nodes of $G(p)$ is $Sub(p)\cup\{\bigcirc\}$,

(ii) the root of $G(p)$ is p,

(iii) the edges of $G(p)$ are given by:

(1) $a\in Sub(p)\Rightarrow a\xrightarrow{a}\bigcirc$ is an edge,

(2) $ax\in Sub(p)\Rightarrow ax\xrightarrow{a}x$ is an edge,

(3) $a+y\in Sub(p)\Rightarrow a+y\xrightarrow{a}\bigcirc$ is an edge,

(4) $ax+y\in Sub(p)\Rightarrow ax+y\xrightarrow{a}x$ is an edge.

(If p has only infinite branches, the termination node $\bigcirc$ can be discarded.) So now the statement that $\Sigma_{i=1}^{\infty}a^i$ $(=(a,a+a^2,\ \ldots))$ is infinitely branching makes sense: it is meant that its canonical process graph is so. In fact, the canonical process graph of $p=\Sigma_{i=1}^{\infty}a^i$ is

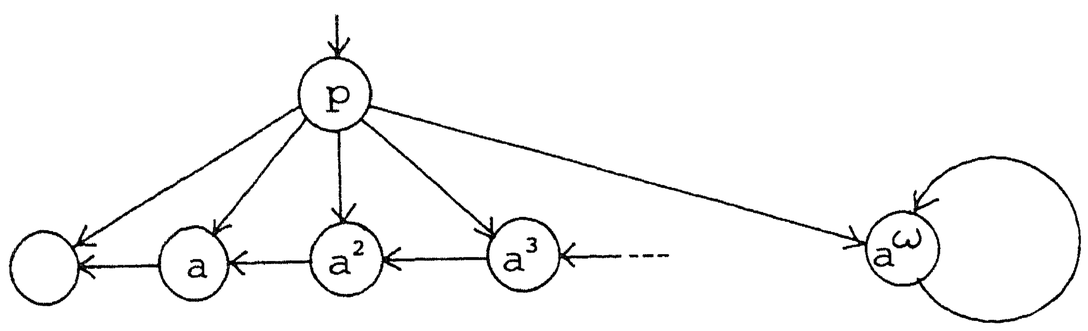


FIGURE 11

which is bisimilar to the process graph in figure 10 (ii).
(Note that $G(p)$ contains the infinite branch a^ω; for: $p=p+a^\omega=p+a\cdot a^\omega$, hence $a^\omega \in Sub(p)$.)

We conclude this section about PA with a number of remarks which give some additional information about PA and its models (but which are not strictly necessary for an understanding of the following sections).

1.3. The cardinality of $\mathbf{A}^\infty$ *and* A^∞

The cardinality of $\mathbf{A}^\infty$ and A^∞ is $2^{\aleph_0}$ for all finite A (as supposed throughout the paper). In contrast, one may consider the following. If there is no condition imposed on the branching degree of process graphs, and $\mathbf{A}^\infty$ is constructed as before, then even for a singleton alphabet A the process domain $\mathbf{A}^\infty$ would be a proper class in the sense of axiomatic set theory. This shows that in order to obtain a set-sized domain of process graphs modulo bisimulation one has to specify some cardinal as upper bound on the branching degree in advance.

1.4. The finite process algebras A_n

Some interesting *finite* process algebras (models of PA) which were not introduced above, can be obtained as follows. Define $A_n=\{(p)_n | p\in A_\omega\}$ and define as operations $\Box_n$ on the finite set A_n:

$$x\,\Box_n y = (x\Box y)_n$$

where the $\Box\in\{+,\cdot,\|,\lfloor\}$ are the operations from A_ω. Then:
$A_n(+_n,\cdot_n,\|_n,\lfloor_n)\models PA$. Now A^∞ can be defined simply as the *projective limit of the algebras* A_n $(n\geqslant 1)$.

1.5. Commutativity and associativity of merge

From the axioms of PA, the commutativity of merge follows immediately:

$$x\|y = x\lfloor y + y\lfloor x = y\lfloor x + x\lfloor y = y\|x.$$

The associativity

$$x\|(y\|z) = (x\|y)\|z$$

does *not* follow from PA. (Indeed one can construct a process algebra with nonassociative merge operator.) However, in the process algebras introduced above $(A_\omega, A_n, \mathbf{A}^\infty, A^\infty)$ the associativity does hold. A proof, by induction on the structure of the elements, can be given simultaneously with a proof of the

useful identity

$$(x \between y) \between z = x \between (y \| z).$$

1.6. Adding a zero process to PA

One can argue about the desirability of an element 0 in process algebras, with the properties

$$x + 0 = x$$

$$0x = x0 = x.$$

Naive addition of such axioms to *PA* yields an 'inconsistency', though. For consider:

$$ab = (a + 0)b = ab + 0b = ab + b$$

contrary to our intention to distinguish ab from $ab + b$.

However, with the added proviso in axiom A4:

$$(x + y)z = xz + yz \quad \text{if} \quad x, y \neq 0$$

(and adding $0 \between x = 0$, $x \between 0 = x$) this inconsistency is removed and we have a conservative extension of PA.

Yet we will not pursue this option, since we have no need for 0. One reason is found in the next remark, another reason is the wish to adhere to an equational format for process algebra as long as possible.

1.7. The (non)existence of a suitable partial order on process algebras

It would be most convenient to have a cpo structure for process algebras such as A_ω, A^∞. One could think of adding an element 0 as in the previous remark, to function as the least element in a supposed partial order $\leqslant$ on A_ω, A^∞. Moreover, such a p.o. should be 'suitable' in the sense of respecting substitution (i.o.w. being monotone in the operations).

However, a partial order on A_ω or A^∞ (extended with 0) with these properties:

$$\begin{cases} 0 \leqslant p \\ p \leqslant q \Rightarrow s(p) \leqslant s(q) \end{cases}$$

(where $s(\ )$ is some 'context'), does not exist, since it would yield the contradictory equation $aa = aa + a$:

$$aa = aa + 0 \leqslant aa + a = aa + a0 \leqslant aa + aa = aa.$$

Also there does not exist a p.o. on A_ω, A^∞ satisfying the properties

$$\begin{cases} x \leqslant x + y \\ x \leqslant y \Rightarrow s(x) \leqslant s(y). \end{cases}$$

For, this would result in the contradictory equation $a(b + c) = a(b + c) + ab$:

$$a(b+c) \leqslant a(b+c)+ab \leqslant a(b+c)+a(b+c) = a(b+c).$$

1.8. The auxiliary operator left-merge
The theory of the initial algebra $A_\omega(+,\cdot,\|,\lfloor)$, that is the set of true equations between closed terms, is finitely axiomatized by PA. Without $\lfloor$ however such a finite axiomatization of the theory of the reduct $A_\omega(+,\cdot,\|)$ does not seem possible. Of course the main advantage of $\lfloor$ is the ease in algebraical computation.

Another advantage of $\lfloor$ is the greater defining power it gives. E.g. the unique solution of the recursion equation

$$X = p\lfloor X$$

(a topic considered in detail in Section 3) can be seen as the 'ω-merge' of p, notation: $p^{\overset{\omega}{=}}$, which is intuitively

$$p\|p\|p\|....$$

i.e. the limit of the sequence p, $p\|p$, $p\|p\|p$, ... (see also the next remark). Without $\lfloor$, such a uniform definition of $p^{\overset{\omega}{=}}$ does not seem possible.

1.9. Solving equations in A^∞
In Section 3 recursion equations and systems of recursion equations will be considered under the condition that the equations are *guarded.* Here, we want to mention a theorem for the unguarded case:

THEOREM 1.4. *Let $E_X=\{X_i=T_i(X)|i=1, ...,n\}$ be a system of equations for $X=X_1, ..., X_n$. Then E_X has a solution $(p_1, ..., p_n)$ in each of the above introduced process algebras.*

In general this solution will not be unique. In the case that $n=1$ solutions can be obtained as follows:

THEOREM 1.5. *Let $X=T(X)$ be a recursion equation for X. Then a solution for X can be obtained as the limit of the iteration sequence*

$$q,\ T(q),\ T(T(q)),\ ...,\ T^n(q),\ ...$$

for arbitrary q.

(Here $\lim_{k\to\infty} T^k(q)=p$ means: $\forall n\ \exists m\ (T^m(q))_n=(p)_n$.) At present however we do not see applications for the possibility of solving unguarded fixed point equations.

2. Process algebra with communication: ACP

We will now extend the axiom system PA of Section 1 with the facility of *communication* between processes. The communication will be modeled by *actions sharing*. In PA all atomic actions were on equal footing, and capable of being performed independently. In *ACP, Algebra of Communicating Processes,* we will introduce next to this kind of independent or autonomous actions, so-called *subatomic* actions which need one or more other subatomic actions as partners in order to be executed. (Cf. the subatomic actions $C!t$ and $C?x$ in Hoare's CSP (see [12]), whose simultaneous execution amounts to the assignment $x:=t$.) The execution is then an 'ordinary' atomic action.

Using this model of shared actions, of which a particular case is *'handshaking'*, we will as an application model the process given by a *dataflow network.*

As a first illustration, consider the following processes $p=(abc)^\omega$ and $q=(efg)^\omega$.

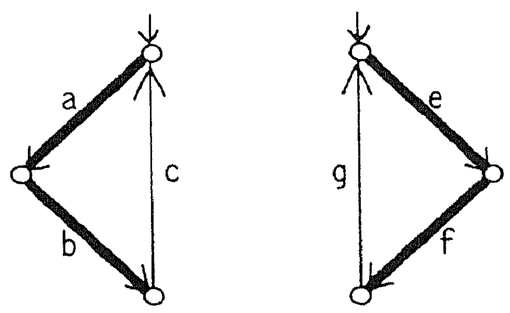


FIGURE 12

The heavy lines denote atomic actions, the steps c and g are subatomic actions and need each other to perform the action h, notation: $c|g=h$. (In Petri net notation, the process resulting from the co-operation of p,q would be given by

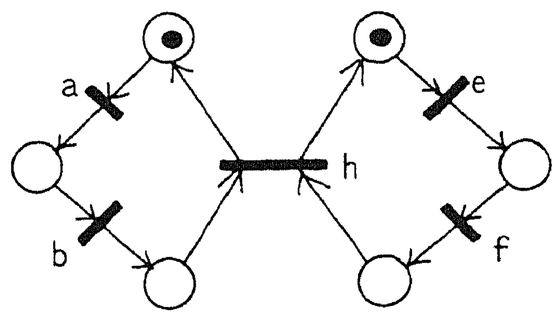


FIGURE 13

Now the process r resulting from the co-operation of p and q would be:

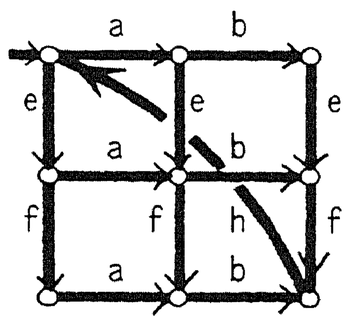


FIGURE 14

That is, $r=([a(e(bf+fb)+bef)+e(a(bf+fb)+fab))\cdot h)^{\omega}$.

The axiom system ACP (Table 3) gives the means to compute the results of such communicating processes in an algebraic way. ACP is an extension of PA (Table 2), but not only in the sense that axioms are added; one axiom from PA (viz. $M1$) is adapted: $x\|y$ is in ACP a sum of *three* terms, namely $x\lfloor\!\lfloor y$, $y\lfloor\!\lfloor x$ and the new summand $x|y$. Here $x\lfloor\!\lfloor y$ is, as in PA, 'like $x\|y$' but taking its first step from x; likewise $y\lfloor\!\lfloor x$; and $x|y$ is like $x\|y$ but requires the first action to be the result of a communication (between a first subatomic 'step' of x and a first subatomic step of y).

This new operator '$|$' is called *communication merge*; on the set A of atoms and subatomic actions it is a binary function, the *communication function,* which is given a priori. It is commutative and associative. The precise choice of the communication function varies with the application of ACP which one has in mind — just as the choice of the alphabet A. Thus ACP is in fact parametrized by A and by the communication function $|:A\times A\to A$.

The difference between what we called 'independent' atoms and 'subatomic actions' needs, fortunately, not to be made explicit in the axiom system. What is atomic and what subatomic follows by an inspection of the communication function '$|$'.

Besides a new operator '$|$', communication merge, there appear two new ingredients in the signature of ACP as compared to that of PA.

The first is a constant δ, which is a 'zero' for $+$ and moreover satisfies the axiom $\delta x=\delta$ (A7). The 'process' δ exhibits some (but not all) of the features of deadlock or rather failure. The main reason for introducing δ is algebraical: by means of δ the unsuccessful communications are eliminated. We will refer to the constant δ as *'deadlock'* (without claiming that δ models all of the deadlock phenomenon). An intuitive view of δ which 'explains' the axioms A6, 7 in Table 3 is: δ is the action in which the process acknowledges the fact that it cannot further execute actions. So, whenever the process has another option, it will not perform this acknowledgement of stagnation: $x+\delta=x$.

The second new ingredient is formed by the *encapsulation operators* ∂_H where $H\subseteq A$. Putting ∂_H in front of at process expression p, result $\partial_H(p)$, means that the subatomic actions mentioned in H and occurring in p, cannot anymore communicate with an 'external' process — they have had their chance inside p.

Summarizing, we have the following signature for ACP:

$x+y$	alternative composition (sum)
$x\cdot y$	sequential composition (product)
$x\|y$	parallel composition (merge)
$x\lfloor\!\lfloor y$	left merge
$x\|y$	communication merge
$\|:A\times A\to A$	communication function
$\partial_H(x)$	encapsulation
δ	deadlock

Note that ACP is an extension of PA in the following sense: let the communication function be trivial, i.e. $a|b=\delta$ for all $a,b\in A$. Then the models A_ω, $\mathbf{A}^\infty$, A^∞ for PA (with signature $+,\cdot,\|,\lfloor\!\lfloor$) are just *reducts* in the modeltheoretic sense of the models A_ω, $\mathbf{A}^\infty$, A^∞ for ACP which we will construct below and which have signature $+,\cdot,\|,\lfloor\!\lfloor,|,\partial_H,\delta$.

ACP

$x+y = y+x$	A1
$x+(y+z) = (x+y)+z$	A2
$x+x = x$	A3
$(x+y)\cdot z = x\cdot z+y\cdot z$	A4
$(x\cdot y)\cdot z = x\cdot(y\cdot z)$	A5
$x+\delta = x$	A6
$\delta\cdot x = \delta$	A7
$a\|b = b\|a$	C1
$(a\|b)\|c = a\|(b\|c)$	C2
$\delta\|a = \delta$	C3
$x\|\|y = x\lfloor\!\lfloor y+y\lfloor\!\lfloor x+x\|y$	CM1
$a\lfloor\!\lfloor x = a\cdot x$	CM2
$(ax)\lfloor\!\lfloor y = a(x\|\|y)$	CM3
$(x+y)\lfloor\!\lfloor z = x\lfloor\!\lfloor z+y\lfloor\!\lfloor z$	CM4
$(ax)\|b = (a\|b)\cdot x$	CM5
$a\|(bx) = (a\|b)\cdot x$	CM6
$(ax)\|(by) = (a\|b)\cdot(x\|\|y)$	CM7
$(x+y)\|z = x\|z+y\|z$	CM8
$x\|(y+z) = x\|y+x\|z$	CM9
$\partial_H(a) = a$ if $a\notin H$	D1
$\partial_H(a) = \delta$ if $a\in H$	D2
$\partial_H(x+y) = \partial_H(x)+\partial_H(y)$	D3
$\partial_H(x\cdot y) = \partial_H(x)\cdot\partial_H(y)$	D4

TABLE 3

2.1. Process algebras for ACP

The development of models for ACP is analogous to that for PA, so we will be much shorter now in its description. Again we introduce:

(1) A_ω, *the initial algebra of ACP,*
(2) $\mathbf{A}^\infty$, *the process graph model of ACP,*
(3) A^∞, *the standard model of ACP.*

Here some confusion may arise as to which signature, that of PA or that of ACP, is meant when speaking about A_ω, $\mathbf{A}^\infty$, A^∞. When this confusion is not solved by the context, we will mention the intended signature explicitly, as in $A_\omega(+,\cdot,\|,\lfloor\!\lfloor)$ vs. $A_\omega(+,\cdot,\|,\lfloor\!\lfloor,|,\partial_H,\delta)$.

2.1.1. The initial algebra A_ω of ACP. Before building A_ω, we have fixed the alphabet A, a communication function $|:A\times A\to A$, and a subset $H\subseteq A$ (hence an encapsulation operator ∂_H).

Now A_ω contains as elements: the process expressions (in the signature of ACP) modulo the equality given by ACP. By the following theorem.

THEOREM 2.1 (NORMAL FORM). *For each closed term t there is a closed term t' not containing $\|,\lfloor\!\lfloor,|,\partial_H$ such that $ACP\vdash t=t'$.*

We may think of elements of A_ω as built from $A,+,\cdot$ only (just as in the case of PA), or as the finite process trees encountered in Section 1.

EXAMPLE 2.1. Let $A=\{a,b,c,c^0,d,\delta\}$. Let $|:A\times A\to A$ be given by $c|c=c^0$, and all other communication equal δ (thus $a|b=c|c^0=d|a=\delta|a=\ldots=\delta|\delta=\delta$). Further, let $H=\{c\}$. Then:

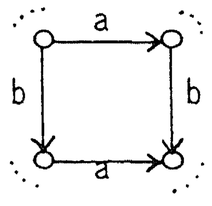
$$\partial_H[(ab+ac)\|cd] =$$
$$\partial_{\{c\}}[ab\lfloor\!\lfloor cd+ac\lfloor\!\lfloor cd+cd\lfloor\!\lfloor(ab+ac)+cd|ab+cd|ac] =$$
$$\partial_{\{c\}}[a(b\|cd)+a(c\|cd)+c(d\|(ab+ac))+(c|a)(d\|b)+(c|a)(d\|c)] =$$
$$\partial_{\{c\}}[a(bcd+c(d\|b)+(b|c)d)+a(ccd+c(d\|c)+(c|c)d)+$$
$$+c(d\|(ab+ac))+\delta(d\|b)+\delta(d\|c)]=$$
$$\partial_{\{c\}}[a(bcd+c(d\|b))+a(ccd+c(d\|c)+c^0d)+c(d\|(ab+ac))] =$$
$$ab\delta+ac^0d.$$

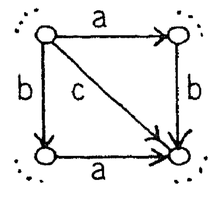
EXAMPLE 2.2. Consider the alphabet $\{a,b,b^0,c,c^0,\delta\}$ with the only proper communications $c|c=c^0$, $b|b=b^0$. Now $a(b+c)$ and $ab+ac$ behave differently in the context $C[\,]=\partial_{\{b,c\}}(\ldots\|c)$; namely:

$$C[a(b+c)] = ac^0,$$
$$C[ab+ac] = a\delta+ac^0.$$

2.1.2. The process graph algebra $\mathbf{A}^\infty$ *for ACP.* The definition of $\mathbf{A}^\infty(+,\cdot,\|,\lfloor\!\lfloor,|,\partial_H,\delta)$ parallels that of $\mathbf{A}^\infty(+,\cdot,\|,\lfloor\!\lfloor)$ for PA, except for two additions.

Let g,h be finitely branching process graphs with acyclic roots. Then the merge $g\|h$ is now the cartesian product graph enriched with 'diagonal' edges $\xrightarrow{a|b}$ in the following situation:

if 

is a subgraph of the cartesian product graph, then the arrow $\circ\xrightarrow{c}\circ$ (where $c=a|b$) is inserted; result: 

The left merge $g\lfloor\!\lfloor h$ and the communication merge yield results which can now be guessed. An example will suffice:

EXAMPLE 2.3. Let $A=\{a,b,c,\delta\}$, $a|b=c$ and all other communications equal δ. Then $ab\|bab$, $ab\lfloor\!\lfloor bab$, $bab\lfloor\!\lfloor ab$ and $ab|bab$ are the following graphs respectively:

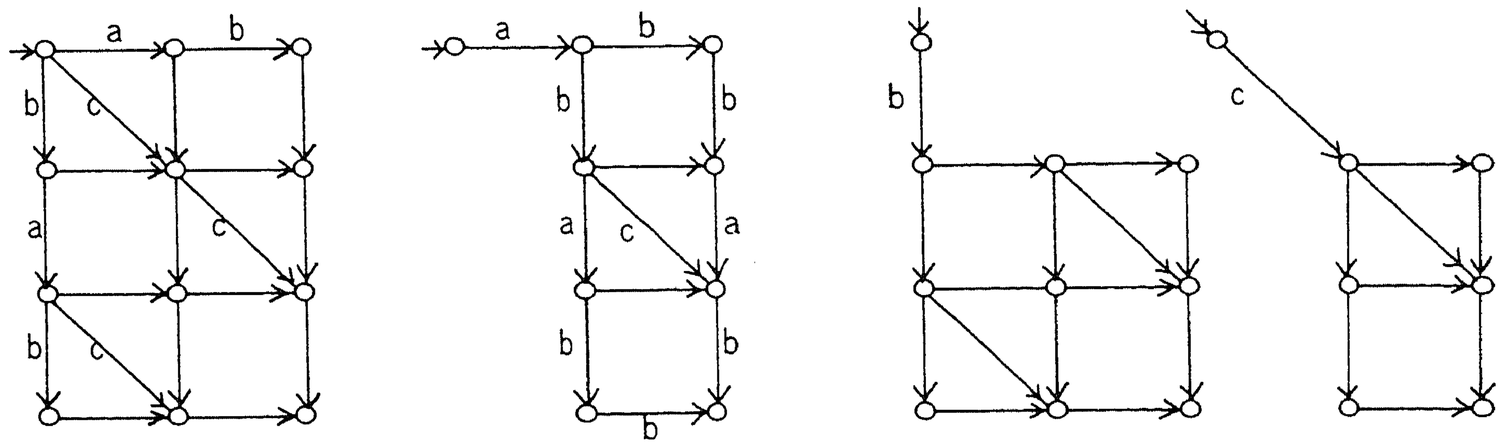


FIGURE 15

Note that we have omitted the diagonal edges labeled with δ, resulting from trivial communications. This brings us to the second addition *bisimulation between process graphs containing* δ*-steps.*

The old concept of bisimulation in Section 1 would not do now, since it would not satisfy the laws $x+\delta=x$ and $\delta x=\delta$. We will choose the following solution: first define the δ*-normal form* of the process graph g as the process graph g' obtained by deleting all δ-steps which have a 'brother' step and creating for the remaining δ-steps if necessary separate end nodes. Afterwards disconnected pieces of the graph are removed.

Now g and h are bisimilar if their δ-normal forms are bisimilar in the old sense.

Finally, the effect of ∂_H on the graph g is simply to replace all $a \in H$ which occur in g, by δ.

The effect of these definitions is that $\mathbf{A}^\infty$ is a model of ACP. Using these graphs, we have an easy way to 'compute' the result of Example 2.1.:

$(ab+ac)\|cd =$

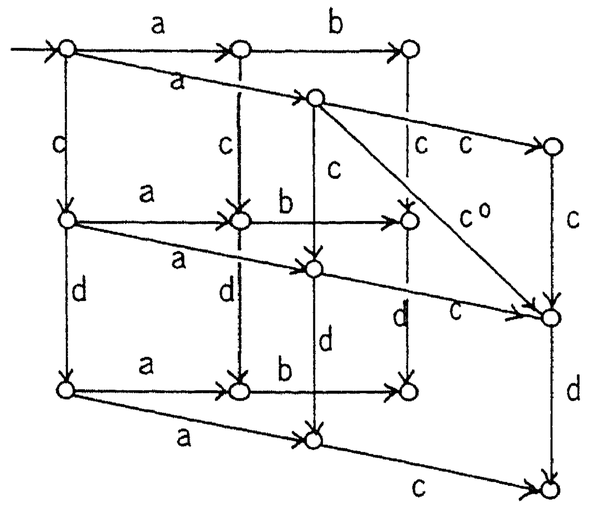


FIGURE 16

$\partial_{\{c\}}[(ab+ac)\|cd] = ab+ac^\circ d =$

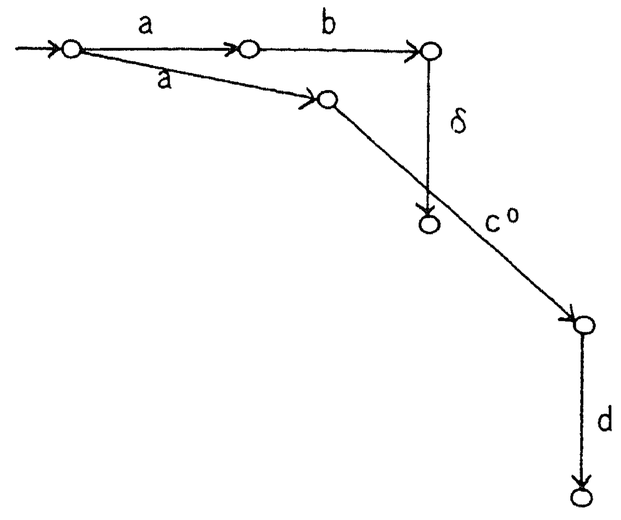


FIGURE 17

2.1.3. The standard model A^∞ for ACP. The standard model A^∞ for ACP is constructed entirely analogous to the corresponding model for PA. An example of a computation in the standard model: Let $A=\{a,b,c,d,\delta\}$, $a/a=d$ the only proper communication. Now let

$$p = (a, ab, aba, abab, \ldots)$$

and

$$q = (a, ac, aca, acac, \ldots).$$

Then

$$\begin{aligned}\partial_{\{a\}}(p\|q) &= \partial_{\{a\}}(a\|a)_1, (ab\|ac)_2, (aba\|aca)_3, \ldots) \\ &= (\partial_{\{a\}}(a\|a)_1, \partial_{\{a\}}(ab\|ac)_2, \ldots) \\ &= (\partial_{\{a\}}(aa+d)_1, \partial_{\{a\}}(\ldots), \ldots) \\ &= (d, d(b+c), d(bc+cb), d(bc+cb)d, \ldots)\end{aligned}$$

2.2. Process algebras with standard concurrency and handshaking

A useful intuition about communicating processes is to postulate that $\|$ is commutative and associative. This does not follow from the axioms of ACP; pathological process algebras with noncommutative and nonassociative $\|$ are possible. But in the process algebras A_ω, $\mathbf{A}^\infty$ and A^∞, $\|$ is indeed commutative and associative. In fact these algebras satisfy the following *axioms of standard concurrency:*

$(x \mathbin{\llcorner} y) \mathbin{\llcorner} z = x \mathbin{\llcorner} (y \| z)$
$(x \mid y) \mathbin{\llcorner} z = x \mid (y \mathbin{\llcorner} z)$
$x \mid y = y \mid x$
$x \| y = y \| x$
$x \mid (y \mid z) = (x \mid y) \mid z$
$x \| (y \| z) = (x \| y) \| z$

TABLE 4

(These axioms are not independent relative to ACP. E.g. commutativity and associativity of $\|$ *are derivable from the other four plus ACP.)*

Moreover, matters are greatly simplified by adopting the handshaking axiom:

$$x \mid y \mid z = \delta$$

which is satisfied by both CSP and CCS. The handshaking axiom implies that *all proper communications are binary.*

Under the hypotheses of standard concurrency and the handshaking axiom we can prove the following fact which is a generalization of the *ACP*-axiom *CM* 1:

THEOREM 2.2 (MILNER). $x_1 \| \ldots \| x_k = \sum_i x_i \mathbin{\llcorner} X_k^i + \sum_{i \neq j} (x_i \mid x_j) \mathbin{\llcorner} X_k^{ij}$.

Here X_k^i is obtained by merging $x_1, \ldots, x_k$ except x_i, and X_k^{ij} is obtained by merging $x_1, \ldots, x_k$ except x_i, x_j ($k \geqslant 3$). Thus, e.g. for $k = 3$:

$$x \| y \| z = x \mathbin{\llcorner} (y \| z) + y \mathbin{\llcorner} (z \| x) + z \mathbin{\llcorner} (x \| y) + (y \mid z) \mathbin{\llcorner} x + (z \mid x) \mathbin{\llcorner} y + (x \mid y) \mathbin{\llcorner} z.$$

2.3. Networks of processes communicating by handshaking

Imagine a process P (figure 18) whose events have a certain spatial position α, β, γ as well as a data content d — so the actions of P are pairs (α, d), (α, d'), (β, d), ..., for simplicity written as $\alpha_d, \alpha_{d'}, \beta_d, \ldots$. E.g. let $\mathfrak{D} = \{0, 1\}$ be the data domain and let P be given by the recursion equation

$$P = \alpha_0 \beta_1 \gamma_0 P.$$

Next, consider a network of such proceses as in figure 19, where the nodes D, M, N, C are given by

$$D = (\alpha_0 \beta_0 \beta_0 + \alpha_1 \beta_1 \beta_1) D$$
$$M = [(\beta_0 + \zeta_0) \gamma_0 + (\beta_1 + \zeta_1) \gamma_1] M$$
$$C = [\gamma_0 (\eta_0 \epsilon_0 + \epsilon_0 \eta_0) + \gamma_1 (\eta_1 \epsilon_1 + \epsilon_1 \eta_1)] C$$
$$N = (\epsilon_0 \eta_1 + \epsilon_1 \eta_0) N.$$

So D is the process which doubles an 'incoming' 0 into 00, likewise for 1; M is the merge process which relays the signals 0, 1 in order of entrance at β or ξ; C is the copy process which relays an incoming signal to both η and ϵ, in either order; and N is the process which inverts an incoming signal.

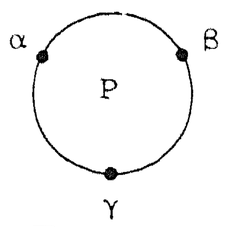


FIGURE 18

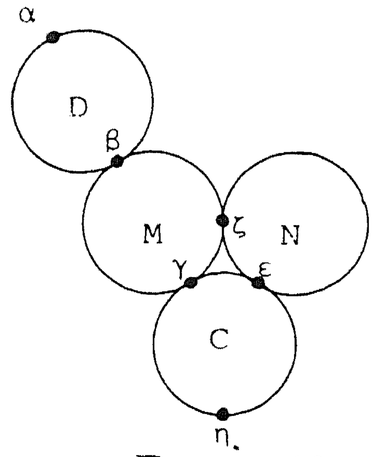


FIGURE 19

The positions $\alpha, \ldots, \zeta$ will be called *ports*; $\beta, \gamma, \epsilon, \zeta$ are *internal* ports. As suggested by figure 19 with its sharing of the internal ports, the processes D, M, C, N cannot operate freely but are constrained by each other: an action β_0 of D is now only an 'intended' action (a subatomic action) needing the same action β_0 of M for the actual passing or 0 along port β. Let us denote this actual event by β°; likewise β_1° denotes passing a 1 at β, etc. (In fact, the word 'passing' is misleading since it suggests a *direction* of flow which, interestingly, disappears at this level of analysis.)

Intuitively, it is clear that the example network has an operational semantics which is a process in A^∞ or $\mathbf{A}^\infty$ over the alphabet

$$A = \{\alpha_d, \beta_d^\circ, \gamma_d^\circ, \epsilon_d^\circ, \xi_d^\circ, \eta_d | d \in \mathcal{D}\}.$$

Now this process can be defined as

$$\partial_H(D \| M \| C \| N)$$

where $H = \{\beta_d, \gamma_d, \epsilon_d, \zeta_d | d \in \mathcal{D}\}$ and where the communication function is defined by: $a|a = a^\circ$ for all $a \in H$ and these are the only proper communications. The operational semantics of the network can now be computed using ACP to any desired depth. This computation can be speeded up by using the Milner Expansion Theorem 2.2. (In fact, for this example the resulting process is regular, that is: given by a finite process graph.)

Before discussing the operational semantics of dataflow networks through networks with *channels* (which were not considered above; there processes are 'directly' connected), we will make some remarks on the present definition of the operational semantics of networks communicating by handshaking.

2.3.1. Handshaking. Handshaking, implicitly introduced above by the example network, is understood here as follows. A network consisting of nodes $P_1,...,P_n$ communicates by handshaking if each port α of P_i $(i=1,..,n)$ is either external (i.e. not connected to any other port) or connected to precisely one port of another process. Here 'α is connected to β' means that α_d only communicates properly with β_d (so if $\alpha_d|\gamma_e \neq \delta$, then $\gamma=\beta$ and $e=d$).

2.3.2. Symmetrical handshaking. Symmetrical handshaking was used in the example above; here a port α is either external or connected to α. By the handshaking convention, a port α can be shared by two processes at most.

The example network can just as well be treated using *asymmetrical* handshaking, as in

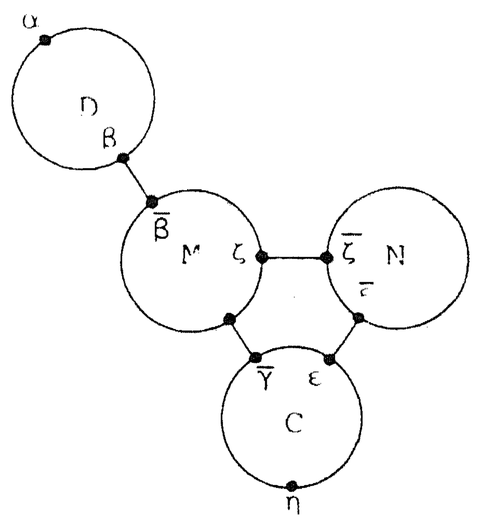


FIGURE 20

where $\beta \neq \bar{\beta}$, etc., and communication is given by $\beta_d|\bar{\beta}_d = \beta_d^\circ$, etc. This is the format used in MILNER [14], where many examples of networks communicating by handshaking are given. One can prove an adequacy theorem for asymmetrical communication, in the sense that communication by handshaking can always be taken to be 1-1 and asymmetrical without loss of defining power. This statement will be made more precise in subsection 3.8.

Our example network was phrased in terms of symmetrical handshaking, to minimize the notational overhead. For *regular* processes (the property 'regular' is the subject of the next section), as all the nodes D,M,C,N in the example are, this works perfectly well. If the nodes are not regular and given by recursion equations containing $\|$, then asymmetrical communication must be chosen; otherwise undesired 'auto-communications' may occur when evaluating the recursive definition.

The condition in our definition of handshaking is a bit severe. One can safely allow a port to be shared by more than two processes, still requiring proper communications to be binary.

EXAMPLE 2.4. Let $\mathcal{D}=\{0\}$, $I_{\alpha\beta}=\alpha_0\beta_0 I_{\alpha\beta}$, likewise $I_{\beta\gamma}$, $I_{\gamma\alpha}$. Let $T=\alpha_0$. Let communication be given by $a|a=a^\circ$ for $a\in H=\{\alpha_0,\beta_0,\gamma_0\}$.

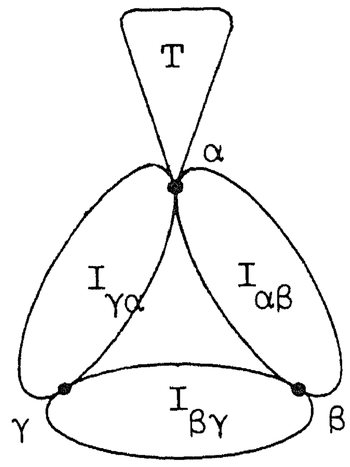


FIGURE 21

In the resulting total process $\partial_H(T\|I_{\alpha\beta}\|I_{\beta\gamma}\|I_{\gamma\alpha})$ the datum 0 is inserted by T and then cycles clockwise through the ring of processors (which are buffers with capacity 1).

A more interesting and fundamental deviation of the handshaking requirements is introduced by MILNER [15].

2.3.3. Synchronous versus asynchronous processes. Process co-operation as described above is *asynchronous*, in the sense of MILNER [15] where a study is made of synchronicity vs. asynchronicity, and where it is argued that synchronous co-operation is the more fundamental of the two.

A *synchronous* network of processes is one where at the pulses of an (imaginary) universal clock all ports exhibit activity simultaneously. As an example consider the following network consisting of two *NOR* circuits; the example is from MILNER [15] with a slight adaptation and serves to demonstrate our claim that synchronous networks can be treated to a large extent within *ACP*. The *NOR* circuit (figure 22) is defined by

$$NOR(k) = \sum_{i,j\in\{0,1\}} (\alpha_i|\beta_j|\gamma_k)NOR(i\downarrow j) \quad (k=0,1)$$

Here $i\downarrow j=1 \Leftrightarrow i=j=0$, and the $\alpha_i|\beta_j|\gamma_k$ are actions which can be perceived simultaneously at the ports α,β,γ. E.g. $\alpha_0|\beta_0|\gamma_1$ is the simultaneous passing (or rather, occurrence) of 0 at α, 0 at β and 1 at γ.

Now consider the network as in figure 23, where NOR' is a copy of NOR obtained by renaming the indicated ports. So

$$NOR'(k) = \sum_{i,j\in\{0,1\}} (\beta_k|\overline{\gamma}_i|\lambda_j)NOR'(i\downarrow j).$$

Communication is given by $(\alpha_i|\beta_j|\gamma_k)\underline{|}(\overline{\beta}_j|\overline{\gamma}_k|\lambda_l)=\alpha_i|\gamma_k^\circ|\beta_j^\circ|\lambda_l$; all other communications result in δ. Further, $H=\{\alpha_i|\beta_j|\gamma_k, \overline{\gamma}_i|\beta_j|\lambda_k \mid i,j,k\in\{0,1\}\}$.

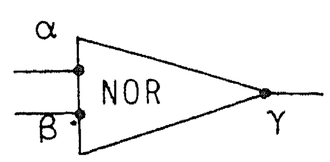


FIGURE 22

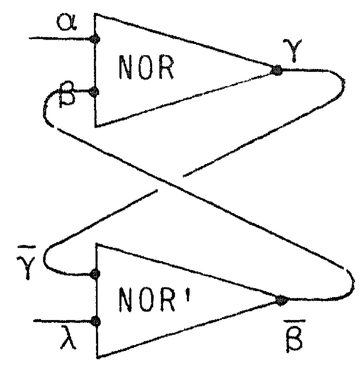


FIGURE 23

Then the network of Figure 23 has as semantics: $\partial_H(NOR(k)\|NOR'(l))$, in the initial state k,l. Abbreviating this expression by $X(k,l)$ we compute using the axioms of ACP:

$$\begin{aligned}
X(k,l) &= \partial_H(NOR(k)\between NOR'(l))+\partial_H(NOR'(l)\between NOR(k))\\
&+ \partial_H(NOR(k)|NOR'(l))=\delta+\delta+\partial_H(NOR(k)|NOR'(l))\\
&= \partial_H(\sum_{i,j}(\alpha_i|\beta_j|\gamma_k)NOR(i\downarrow j) \mid \sum_{i,j}(\bar{\beta}_l|\bar{\gamma}_i|\lambda_j)NOR'(i\downarrow j))\\
&= \partial_H(\sum_{i,j}(\alpha_i|\beta_l^\circ|\gamma_k^\circ|\lambda_j)[NOR(i\downarrow l)\|NOR'(k\downarrow j)]\\
&= \sum_{i,j}(\alpha_i|\beta_l^\circ|\gamma_k^\circ|\lambda_j)X(i\downarrow l,\ k\downarrow j)
\end{aligned}$$

which is a system of four recursion equations, describing the intuitively expected process. The difference with Milner's approach via SCCS (see [15]) is the use of δ: not only does it serve to remove the undesired interleaving results, also it is used to express that certain composite actions are incompatible.

A more direct axiomatization of synchronous processes, related to Milner's SCCS, can be given by omitting the interleaving part of ACP, that is: replace $CM1$ by $x\|y=x|y$, and erase CM2-4. We will not study this axiomatization here, however.

2.4. Dataflow networks. We will return now to the case of networks communicating by handshaking. Above, the connections between ports were directionless and thought of as relaying the data instantaneously. These port connections are *not* channels as used in dataflow networks; e.g. a channel like *Queue* does not relay its messages instantaneously. So let us consider networks such as the one in figure 24, where the arrow-shaped figures denote channels. We will consider as channels: *Queue, Bag* and *Stack*. Now an important realization is that channels and nodes are in fact the same type of entities: both are processes.

Hence this simple form of dataflow is nothing more than a network communicating by handshaking as treated above. The only difficulty is that the processes Queue, Bag and Stack are rather complicated: they are not regular. In the next section we will consider recursion equations within ACP (in fact, even within PA) for Bag and Stack, and discuss some of their properties.

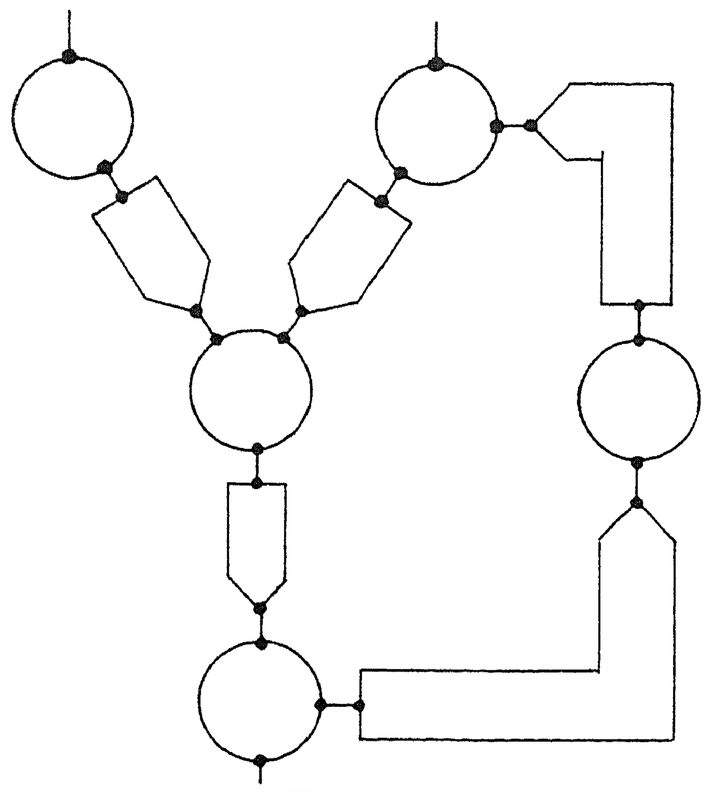


FIGURE 24

For *Queue* the situation is essentially more complicated. If one admits infinite systems of equations, Queue can be defined in *PA* as the first component of the solution of such an infinite system of equations.

One can prove (see [10]) that Queue cannot be defined recursively by a finite system of recursion equations over *PA*.

If one allows extensions of the *PA* formalism, there are two ways of specifying Queue. The first method is via auxiliary operators $\between$ and $\triangle$, that can be axiomatized by finitely many equations (like $\llcorner$ is finitely axiomatized). Then Queue can be recursively defined over *PA* extended with these new operators. (See [10].) The second method uses process graphs defined by means of abstract data types; see [7].

3. Recursively defined processes

In the previous sections we have used, occasionally, some processes which were defined as the solutions of recursion equations; namely, the iteration p^ω of p (as the solution of $X=pX$) and the ω-merge $p^{\overset{\omega}{=}}$ of p (as the solution of $X=p\llcorner X$; see 1.8.).

In this section we will consider this important specification method for processes in a more systematic way. This will produce some criteria as to which processes in A^∞ can be defined recursively; also it will give us some other process algebras.

In the course of these considerations the concept of a *finitely generated* process algebra will prove to be an important concept. Likewise, the concept of a *regular* process plays a prominent role: this is a process corresponding to a finite transition diagram (i.e. having a finite canonical process graph), possibly with cycles. First we need two technical concepts.

3.1. Linear terms and guarded terms

Let $X_1,...,X_n$, be variables ranging over processes. Given the signature of PA or that of ACP, two kinds of terms containing variables $X_1,...,X_n$ are of particular importance:

(i) *Linear terms.* Linear terms are inductively defined as follows:
- atoms a, δ and variables X_i are linear terms,
- if T_1 and T_2 are linear terms then so are T_1+T_2 and aT_1 (for $a \in A$).

An equation $T_1=T_2$ is called linear if T_1, T_2 are linear.

(ii) *Guarded terms.* The *unguarded* terms are inductively defined as follows:
- X_i is unguarded,
- if T is unguarded then so are $T+T'$, $T \cdot T'$, $\partial_H(T)$, $T \| T'$, $T \between T'$, $T | T'$ (for every T').

A term T is guarded if it is not unguarded. Note that we introduced 'formal' variables $X_1,...,X_n$; they are meant as the 'unknowns' in recursion equations. The formal variables should not be confused with the metavariables $x,y,...$ which occur in the axioms of PA and ACP.

Mostly, we will be interested in *finite* systems E of equations. In this section we will always require that E is a *guarded* system of equations. (I.e. the RHS's of the equations in E are guarded.) We will first consider the case of *linear* E, which gives us the regular processes.

3.2. Regular processes

As defined in Section 1, an element $p \in A^\infty$ has a canonical process graph, with the subprocesses as nonterminal nodes and 'o' as terminal node. Now we define:

(i) $p \in A^\infty$ is *regular* if $Sub(p)$ is finite;

(ii) $r(A^\infty)$ is the collection of the regular processes in A^∞.

The next fact is immediate.

THEOREM 3.1. *The following statements are equivalent:*

(i) *p is regular*

(ii) *$Sub(p)$ is finite*

(iii) *$G(p)$ is finite*

(iv) *p is the first component of the solution vector of a finite, guarded, linear system of equations.*

Moreover, $r(A^\infty)$ is closed under all operations (in the signature of PA as well as that of ACP); it is a process algebra whose position relative to the previous ones is as follows: $A_\omega \subseteq r(A^\infty) \subseteq \mathbf{A}^\infty \subseteq A^\infty$.

EXAMPLE 3.1.

(1) Let X be the solution of $X=a+bX$. Then $G(X)$ is as in figure 25, with a tree representation as in figure 26. Note that $Sub(X)=\{X\}$. As a projective sequence, $X=(a+b, a+b(a+b), a=b(a+b(a+b)),...)$. X is a regular process.

FIGURE 25

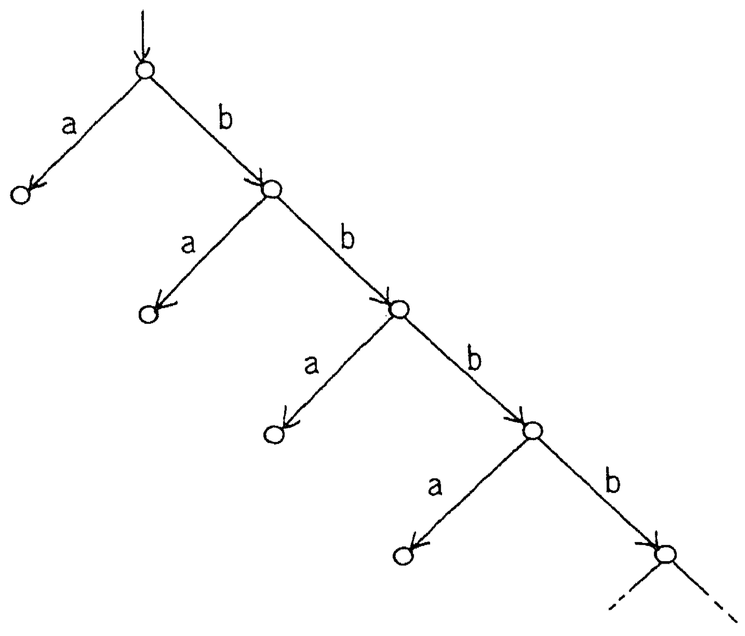


FIGURE 26

(2) Let $E_{X,Y}$ be $\begin{cases} X = aY+c \\ Y = bX+dY+e \end{cases}$

Then the regular solution $\underline{X}$ has the canonical process graph

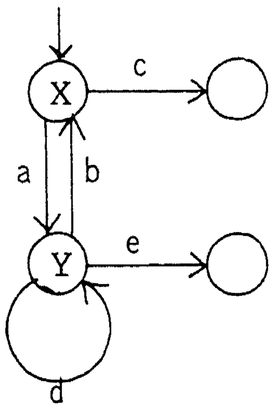


FIGURE 27

(3) The following process $\underline{X}$ is not regular.

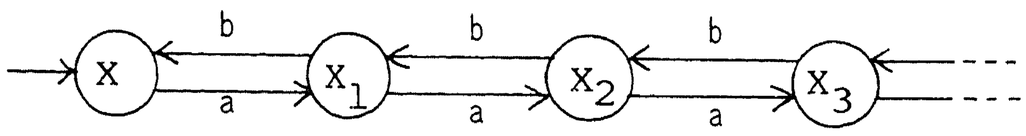


FIGURE 28

It is the first component of the solution vector of the *infinite* system of linear equations

$$\begin{cases} X = X_0 = aX_1 \\ X_{n+1} = aX_{n+2} + bX_n \quad (n \geqslant 0) \end{cases}$$

That X is indeed not regular follows from the realization that there are infinitely many subprocesses (all the $\underline{X}_n$, $n \geqslant 0$, are pairwise different).

3.3. Recursively defined processes

We now define in full generality the concept of a *recursively defined process.* Let $X=\{X_1,...,X_n\}$ be a set of process names (formal variables). We will consider terms over X composed from atoms $a \in A$ and the operators in the signature of PA or that of ACP.

A sytem E_X of *guarded fixed point equations* (or *guarded recursion equations*) for X is a set of n equations $\{X_i = T_i(X_1,...,X_n) | i = 1,...,n\}$ with $T_i(X)$ a guarded term. There is the standard result:

THEOREM 3.2. *Each system E_X of guarded fixed point equations has a unique solution in $(A^\infty)^n$.*

We define $p \in A^\infty$ to be *recursively definable* if there exists a system E_X of guarded fixed point equations over X with solution $(p, q_1,..., q_{n-1})$. With $R(A^\infty)$ we denote the *subalgebra of recursively defined processes.* The relative position of this second new process algebra $R(A^\infty)$ is as follows:

$$A_\omega \subseteq r(A^\infty) \subseteq R(A^\infty) \subseteq \mathbf{A}^\infty \subseteq A^\infty$$

both for PA and ACP. All inclusions are proper.

There is an algebra of some interest which is strictly intermediate between $r(A^\infty)$ and $R(A^\infty)$: it is the process algebra of *uniformly finitely branching* processes. These are processes having canonical process graphs where all the nodes have a uniformly bounded outdegree.

EXAMPLE 3.2. The following is a system of nonlinear guarded recursion equations:

$$\begin{cases} X = aX(X+b) \\ Y = bYXY \end{cases}$$

Likewise $X = a(b \| X)$ is a nonlinear equation. (The process graph of the solution $\underline{X}$ is the one in figure 28.)

A useful fact is the following. Call a process *perpetual* if all its traces are infinite. Then:

THEOREM 3.3. *Let E_X be a system of guarded recursion equations using only $+$ and $\cdot$. Suppose the solutions X are perpetual. Then they are regular.*

An example suggests the simple proof:

$$E_{X,Y,Z} \begin{cases} X = aYX(X+Y)+bYYZ \\ Y = bYY+cZX \\ Z = c(ZX+dZX) \end{cases}$$

Now a short inspection of $E_{X,Y,Z}$ reveals that the solutions X, Y, Z are perpetual. So in products in $E_{X,Y,Z}$ one can erase every factor following X, Y, Z (since for all p, $\underline{X}\,p = \underline{X}$, etc.) I.e. the system of equations

$$E'_{X,Y,Z}\begin{cases} X = aY + bY \\ Y = bY + cZ \\ Z = c(Z + dZ) \end{cases}$$

has the same solutions $\underline{X}$, $\underline{Y}$, $\underline{Z}$. But since $E'_{X,Y,Z}$ is a *linear* system, these solutions are regular.

We will now consider recursion equations for the processes corresponding to Bag, Stack and Counter.

3.4. Bag

Let α -▭- β be a bag with input port α and output port β. (Here 'bag' is considered as a channel which does not preserve, like Queue does, the order of the incoming data. So the contents of Bag can be imagined as a multiset or bag.) Consider a finite data domain D. Then the actions to be performed by Bag are, in our earlier notation, α_d and β_d ($d \in D$). For notational convenience we write d instead of α_d and $\underline{d}$ instead of β_d.

Let B be the initial state of Bag: the empty bag. Now let action d be executed, that is: d is added to the bag. The result is a bag *with the commitment of eventually giving d as output,* i.e. performing action $\underline{d}$. We claim on intuitive grounds that this bag-with-commitment-$\underline{d}$ is $\underline{d} \,\|\, B$. This leads to the equation for B:

$$B = \sum_{d \in D} d(\underline{d} \,\|\, B).$$

Alternatively: consider the process $\Sigma_d d\underline{d}$. Then it is (again intuitively) clear that B is the ω-merge:

$$B = \sum_d d\underline{d} \,\|\, \sum_d d\underline{d} \,\|\, \sum_d d\underline{d} \,\|\, \ldots = (\sum_d d\underline{d})^{\overline{\omega}}.$$

So

$$B = (\sum_{d \in D} d\underline{d}) \between B$$

which indeed is equivalent to the first recursion equation for B, by using the axioms of PA for $\between$.

A third definition:

$$\begin{cases} B_d = d\underline{d} \between B_d = d(\underline{d} \,\|\, B_d) \\ B = \|_d B_d \quad (d \in D) \end{cases}$$

How can one verify that these equations for B indeed describe the intended Bag?

(a) By computing the corresponding canonical process graph and 'validating this against the intuition';
(b) by the more rigorous method employed in [7], which consists of giving a specification of B in terms of abstract data types and *proving* the equation given here correct w.r.t. that specification. Here we will not discuss that method.

We proceed with (a). First, consider the singleton data domain $D=\{d\}$. Then $B=d(\underline{d}\,\|B)$, and now writing

$$B_0 = B,\ B_{n+1}=\underline{d}\,\|B_n=\underline{d}^n\|B$$

one proves immediately

$$\begin{cases} B_0 = dB_1 \\ B_{n+1} = dB_{n+2}+\underline{d}\,B_n \quad (n\geqslant 0). \end{cases}$$

(PROOF:

$$\begin{aligned} B_{n+1} &= \underline{d}\,\|B_n=\underline{d}\,\llcorner B_n + B_n \llcorner \underline{d}=\underline{d}\,B_n+(dB_{n+1}+\underline{d}\,B_{n-1})\llcorner \underline{d} \\ &= \underline{d}\,B_n+dB_{n+1}\llcorner \underline{d}+\underline{d}\,B_{n-1}\llcorner d=\underline{d}\,B_n+d(B_{n+1}\|\underline{d})+\underline{d}\,(B_{n-1}\|\underline{d}) \\ &= \underline{d}\,B_n+dB_{n+2}+\underline{d}\,B_n.) \end{aligned}$$

This yields the canonical process graph

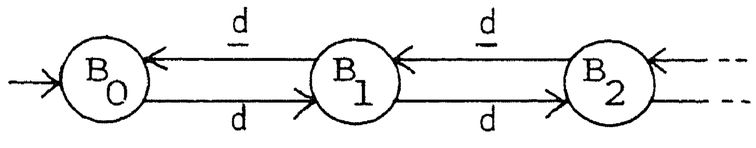


FIGURE 29

The general case $B=\Sigma_d d(d\,\|B)$ is, as process graph, obtained by merging these 'singleton-bags' B_d. So if $D=\{a,b\}$, the canonical process graph of $B=a(\underline{a}\,\|B)+b(\underline{b}\,\|B)$ is:

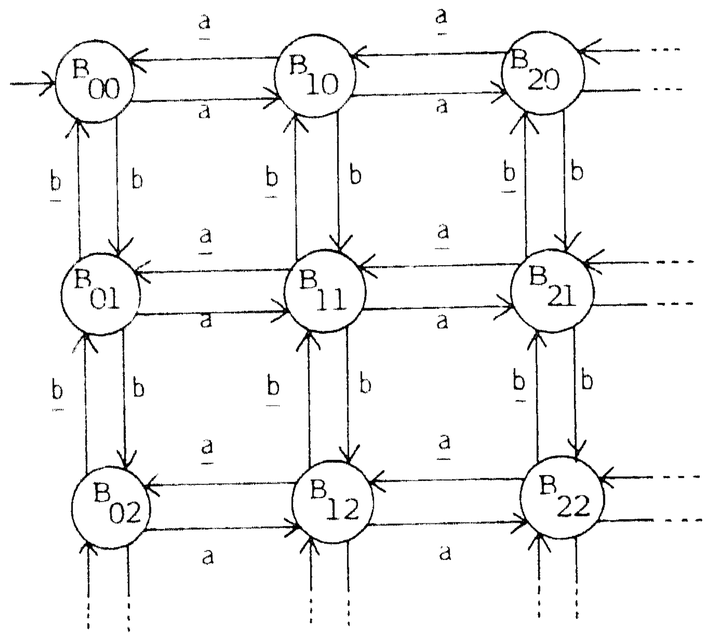


FIGURE 30

We will return to Bag later in this section.

3.5. Stack

For convenience, let $D=\{a,b\}$. As with Bag, 'a' denotes the event of pushing 'a' on the stack, '$\underline{a}$' of popping 'a' from the stack; likewise for b. Now Stack can be defined thus:

$$\begin{cases} S = TS \\ T = aT_a + bT_b \\ T_a = \underline{a} + TT_a \\ T_b = \underline{b} + TT_b \,. \end{cases}$$

Here T is *Terminating Stack,* which must terminate as soon as it is again empty. Further, T_a is T containing an 'a', T_b likewise. S is the iteration T^ω of T; so S is the intended perpetual process Stack. Essentially, this recursive definition of Stack occurs also in HOARE [12].

The recursive definition of Stack above involves the definition of a nonperpetual process, in casu T. This is essential: *Stack S cannot be derived recursively (over $+$ and $\cdot$) without a non-perpetual auxiliary process.* For, if it could, then Theorem 3.3 would entail that S is regular, an obvious contradiction. A consequence is that S cannot be defined recursively (over $+$ and $\cdot$) in one equation.

The canonical process graphs of S and T are as in figure 31 and 32.

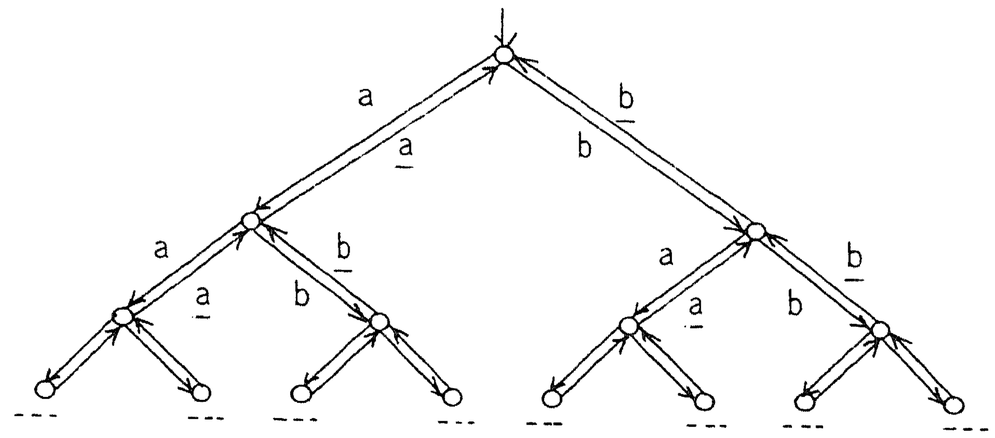


FIGURE 31

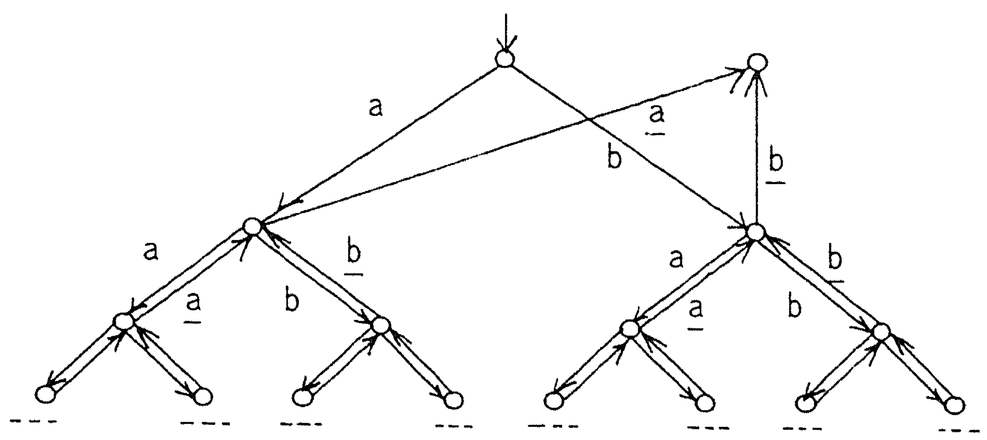


FIGURE 32

3.6. Counter

We will consider a simple counter C without test for zero. The equation for C is obtained by the one for Stack, with $D=\{a\}$:

$$\begin{cases} C = TC \\ T = aT_a \\ T_a = \underline{a} + TT_a \end{cases}$$

or, after eliminating T and writing $D=T_a$:

$$\begin{cases} C = aDC \\ D = \underline{a} + aDD. \end{cases}$$

$G(C)$ is determined as follows: writing $C_n = D^n C$ one easily computes

$$\begin{cases} C_0 = aC_1 \\ C_{n+1} = \underline{a}\, C_n + aC_{n+2} \end{cases}$$

which determines the same process graph as for the singleton-bag above. So we have the interesting fact that C is also the solution of

$$C = a(\underline{a} \,\|\, C).$$

This leads to the question whether it is also possible in the case of the general bag (over an arbitrary but finite data domain D) to eliminate $\|$ in its recursive definition in favour of $+,\cdot$ (and possibly using more equations). The answer is no, if D contains at least two elements. For the lengthy proof see [8].

3.7. Criteria for recursive definability

THEOREM 3.4. *A process which is recursively defined only with* $+$ *and* $\cdot$, *and which has an infinite branch, must have an eventually periodic infinite branch.*

EXAMPLE 3.3. The process *babaabaaabaaaabaaaaab...* cannot recursively be defined over $+$ and $\cdot$.

THEOREM 3.5.

(i) *If* $p \in R(A^\infty)(+,\cdot,\|,\perp\!\!\!\perp)$, *that is: p can recursively be defined in the signature of PA, then Sub(p) is finitely generated (in the usual algebraic sense) over* $+,\cdot,\|,\perp\!\!\!\perp$.

(ii) *Likewise for the reduced signature* $+,\cdot$.

The last fact (ii) can be used to prove that Bag over a non-singleton domain cannot be recursively defined by $+$ and alone; one must prove that Sub(Bag) (i.e. the collection B_{mn} in figure 30, if $D=\{a,b\}$) cannot be finitely generated using $+$, only.

3.8. 1-1 communication

We conclude this section with a theorem stating that binary communication may always be supposed to have a certain simple form.

Consider the alphabet $A=E\cup H$ where H is the set of communication actions, so $H=\{a\in A\mid \exists b\, a|b\neq\delta\}$. Let communication be binary: $a|b|c=\delta$ for all $a,b,c\in A$.

We claim that without loss of defining power (on the external processes, where 'external' refers to E^∞), the communication mechanism $H,|$ can be replaced by a 1-1 communication mechanism $H^*,|^*$. This means: there is a map $-:H^*\to H^*$, such that $\overline{\overline{a}}=a$ and such that all proper communications have the form $a|\bar{a}=b$.

Let us be more precise about the phrase 'without loss of defining power on external processes'. The situation is as in figure 33:

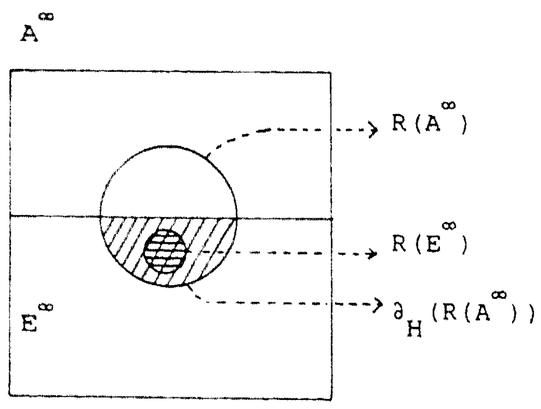


FIGURE 33

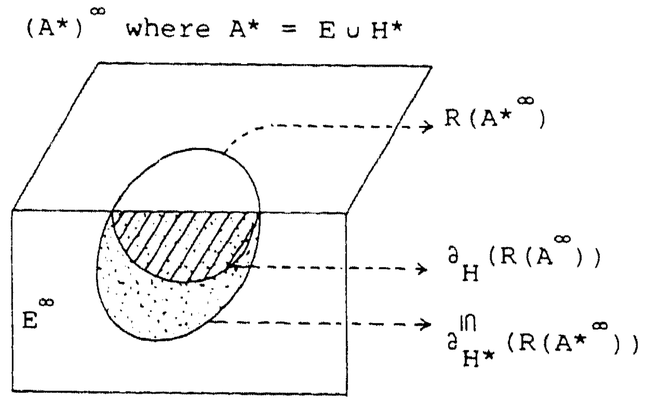


FIGURE 34

In the original setting with H and $|$ (see figure 33), the communication mechanism is able to define the 'external' processes (i.e. in E^∞) contained in $\partial_H(R(A^\infty))$. This $\partial_H(R(A^\infty))$ is a subalgebra of A^∞; it contains a subalgebra $R(E^\infty)$, the external processes recursively definable without communication. Here the difference $\partial_H(R(A^\infty))-R(E^\infty)$ is nonempty; i.e. communication yields more expressive power.

Now it is possible to replace $H,|$ by $H^*,|^*$ such that the situation in figure 34 is obtained. That is, the new communication mechanism given by H^*, $|^*$ recursively defines at least as many processes as the old mechanism.

It is not hard to obtain as a next step an 1-1 *asymmetrical* communication function without impairing the expressive power. (A communication function is asymmetrical if for all a: $a|a=\delta$.) In fact, this is the communication format chosen in MILNER [14].

4. HIDING INTERNAL STEPS IN FINITE PROCESES

In this last section we will discuss the very fundamental problem of abstraction of internal steps ('hiding'). In a process one may wish to distinguish internal and external behaviour and to abstract from the former; obviously the availability of adequate abstraction mechanisms is of crucial importance for a hierarchical construction of systems.

In *trace semantics,* which may be viewed as the theory of ACP augmented with the axiom $x(y+z)=xy+xz$, the abstraction problem seems easy: abstracting from the internal (or silent) steps τ (in Milner's notation) from a trace such as $ab\tau ac\tau\tau a$ results simply in *abaca.*

Also for *synchronous processes* as described in MILNER [15] abstraction from internal steps is easy: in a composite action (i.e. a simultaneous action of *all* ports, internal and external, of the network in consideration), say $e_1|e_2|e_3|i_1|i_2$ where i_1,i_2 are internal, the result after 'hiding' the internal steps is $e_1|e_2|e_3$. (The point here is that each composite action has a nonempty external part, so that hiding does not hide the whole action — therefore the choice structure is left intact.)

However, trace semantics does not respect and reflect deadlock behaviour; and synchronous process co-operation is in our view a special case of the more general mechanism of asynchronous process co-operation, cf. subsection 2.3.3. (MILNER [15] argues the reverse point of view, though.)

For asynchronous processes the initial temptation to treat internal or silent steps τ as above, like the unit element in group theory, that is via equations $x\tau=\tau x=x$, leads at once to difficulties in the presence of communication. Namely, the processes $a(\tau b+c)$ and $a(b+c)$ have different deadlock behaviour: let c,c' be communication atoms such that $c|c'=c^\circ$ is the only proper communication (so $a|c'=\tau|c'=\ldots=\delta$). Then for the context

$$C[\ ]=\partial_{\{c,c'\}}[\ldots\|c']$$

we have

$$C[a(\tau b+c)] = a(\tau\delta+c^\circ)$$

$$C[a(b+c)] = ac^\circ.$$

In this section we will treat abstraction of internal steps for asynchronous processes. We will deal only with *finite* processes; here the theory exhibits some clarity. For infinite processes the situation is at present much less clear — for some comments see our 'concluding remarks' (4.3) at the end of this section.

4.1. Hiding internal steps in finite processes without communication: PA_τ

4.1.1. Bisimulation modulo internal steps. From now on, we consider the alphabet $A \cup \{\tau\}$, where τ is the silent or invisible step. A *trace* σ is a possibly empty finite string over $A \cup \{\tau\}$ (thus $\sigma \in (A \cup \{\tau\}^*)$. With $e(\sigma)$ we denote the trace σ where all τ-steps are erased.

Consider a finite acyclic process graph g over $A \cup \{\tau\}$. A path $\pi: s_0 \longrightarrow\!\!\!\!\!\rightarrow s_k$ in g is a sequence of the form

$$s_0 \xrightarrow[h_0]{l_0} s_1 \xrightarrow[h_1]{l_1} \ldots \xrightarrow[h_{k-1}]{l_{k-1}} s_k$$

($k \geqslant 0$) where the s_i are nodes, the h_i are edges between s_i and s_{i+1}, and each l_i is the label of edge h_i. (The h_i are needed because we work with multigraphs.) The trace *trace*(π) associated to this path is just $l_0 l_1 \ldots l_{k-1}$.

DEFINITION 4.1. A *bisimulation modulo* τ between two finite acyclic process graphs g_1 and g_2 is a relation R on NODES(g_1) $\times$ NODES(g_2) satisfying the following conditions:

(i) (ROOT(g_1), ROOT(g_2)) $\in R$,

(ii) Domain (R) = NODES(g_1) and Codomain(R) = NODES(g_2),

(iii) For each pair $(s_1, s_2) \in R$ and for each path $\pi_1: s_1 \longrightarrow\!\!\!\!\!\rightarrow t_1$ in g_1 there is a path $\pi_2: s_2 \longrightarrow\!\!\!\!\!\rightarrow t_2$ in g_2 such that $(t_1, t_2) \in R$ and $e(trace(\pi_1)) = e(trace(\pi_2))$. (See figure 35.)

(iv) Likewise for each pair $(s_1, s_2) \in R$ and for each path $\pi_2: s_2 \longrightarrow\!\!\!\!\!\rightarrow t_2$ in g_2 there is a path $\pi_1: s_1 \longrightarrow\!\!\!\!\!\rightarrow t_1$ in g_1 such that $(t_1, t_2) \in R$ and $e(trace(\pi_1)) = e(trace(\pi_2))$. (See figure 36.)

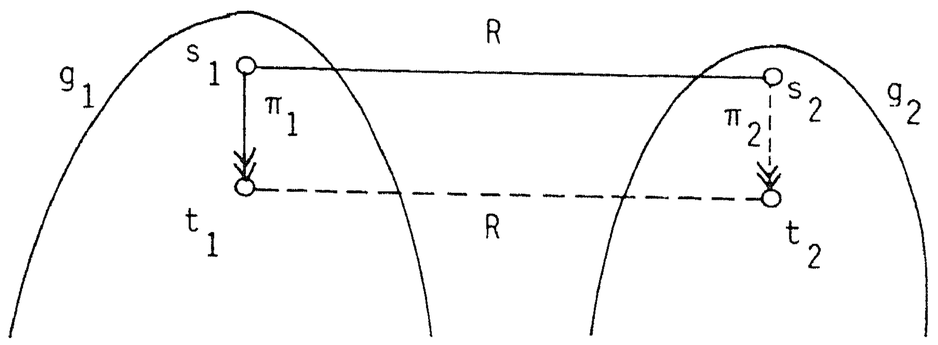


FIGURE 35

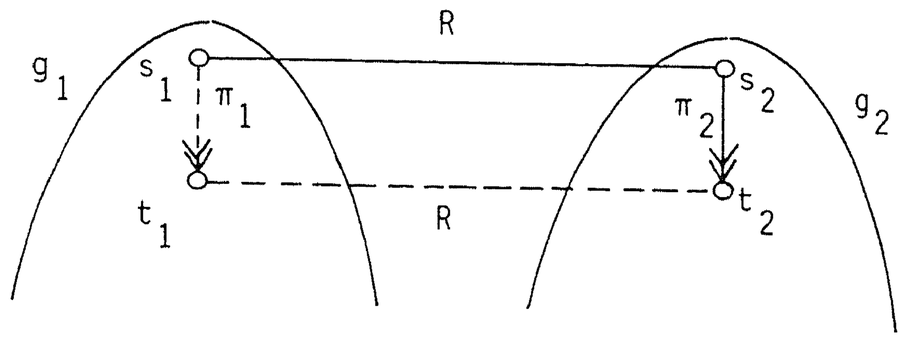


FIGURE 36

Process graphs g_1, g_2 are bisimilar modulo τ if there is a bisimulation modulo τ between g_1, g_2. Notation: $g_1 \leftrightarrows_\tau g_2$.

The notion of bisimulation modulo τ specializes to the notion of bisimulation $\leftrightarrows$ introduced in Section 1, where τ is not around. For technical reasons it is convenient to work with *rooted* bisimulation modulo τ: here a root cannot be related to a nonroot node. If g_1, g_2 are bisimilar in this sense, we write $g_1 \leftrightarrows_{r,\tau} g_2$. Also this notion of bisimulation specializes to $\leftrightarrows$ in Section 1 (see 1.2.2.1).

EXAMPLES 4.1. $a\tau b \leftrightarrows_{r,\tau} ab$ (see figure 37); $ab \leftrightarrows_{r,\tau} a\tau(\tau b + \tau\tau b)$ (see figure 38); $a(\tau b + b) \leftrightarrows_{r,\tau} ab$ (see figure 39); $c(a+b) \leftrightarrows_{r,\tau} c(\tau(a+b)+a)$ (see figure 40).

A negative example: see figure 41. This was the example in the introduction to this section. The heavy line denotes where it is not possible to continue a construction of the bisimulation.

Another negative example: $a(\tau b + c) \not\leftrightarrows_{r,\tau} a(b+c) + ab$.

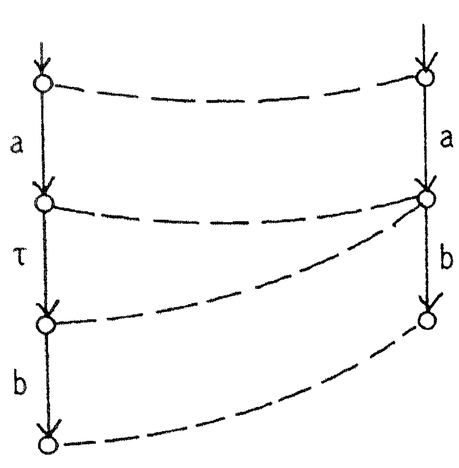


FIGURE 37

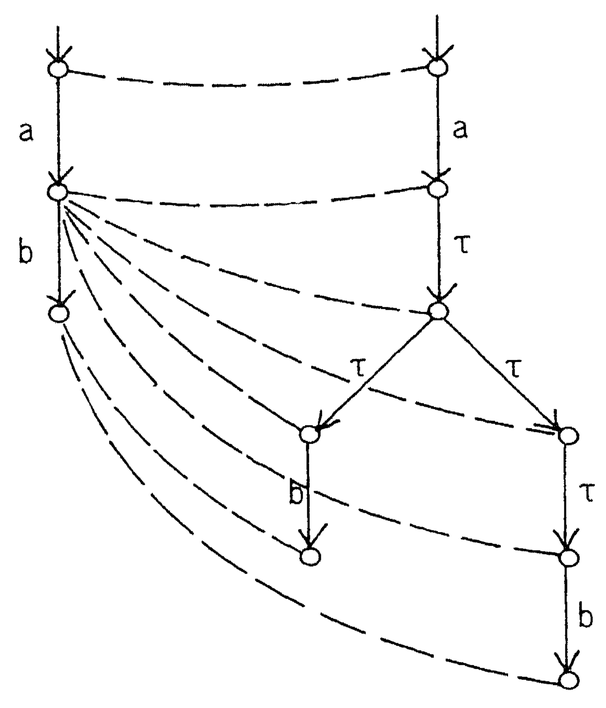


FIGURE 38

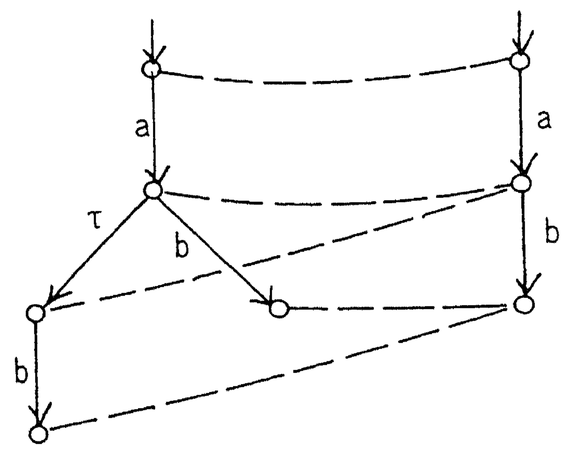


FIGURE 39

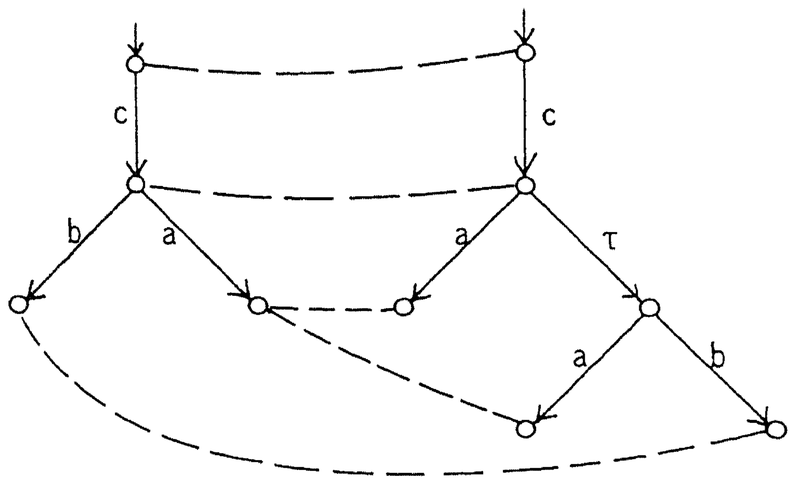


FIGURE 40

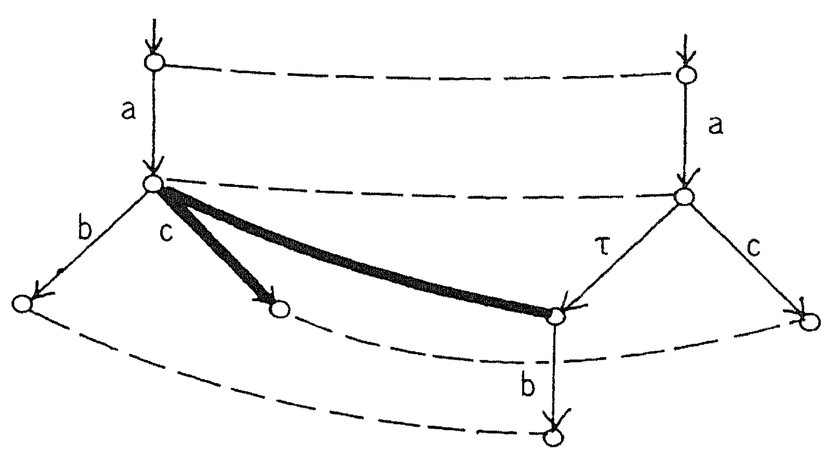


FIGURE 41

THEOREM 4.1. *Rooted bisimulation modulo* τ *is preserved by the operators* $+, \cdot, \between, \|$ *on finite acyclic process graphs.*

(This would not be true for bisimulation modulo τ without the 'rooted' condition. E.g. $a \leftrightarrows_\tau a$, $\tau b \leftrightarrows_\tau b$ but $a + \tau b \not\leftrightarrows_\tau a + b$. Note that $\tau b \not\leftrightarrows_{r,\tau} b$.)

COROLLARY 4.1. *The relation 'bisimilar modulo* τ*' is a congruence on* $A_\omega(+, \cdot, \|, \between)$.

4.1.2 Axioms for abstraction. A beautiful result in MILNER [14] is that the semantical notion of $\leftrightarrows_{r,\tau}$ congruence on finite processes *can be treated algebraically,* namely by three simple equations: Milner's τ-laws T1, T2, T3. Added to PA we obtain PA_τ as in Table 5, where the abstraction operator τ_I serves to 'internalize' steps. (Here $a \in A_\tau$.)

PA_τ

$x + y = y + x$	A1	$x\tau = x$	T1
$x + (y + z) = (x + y) + z$	A2	$\tau x + x = \tau x$	T2
$x + x = x$	A3	$a(\tau x + y) = a(\tau x + y) + ax$	T3
$(x + y)z = xz + yz$	A4		
$(xy)z = x(yz)$	A5		
		$\tau_I(a) = a$ if $a \notin I$	TI1
$x \| y = x \between y + y \between x$	M1	$\tau_I(a) = \tau$ if $a \in I$	TI2
$a \between x = ax$	M2	$\tau_I(x + y) = \tau_I(x) + \tau_I(y)$	TI3
$(ax) \between y = a(x \| y)$	M3	$\tau_I(xy) = \tau_I(x) \cdot \tau_I(y)$	TI4
$(x + y) \between z = x \between z + y \between z$	M4		

TABLE 5

THEOREM 4.2.
(i) *PA_τ is conservative over PA (the latter with actions from A).*
(ii) *The initial algebra of PA_τ is iomorphic to $A_\omega/\leftrightarrows_{r,\tau}$ (the initial algebra of PA modulo the congruence of rooted bisimulation modulo τ).*

Stated differently: the τ-laws T1-3 are a complete axiomatization of rooted bisimulation modulo τ.

Part (i) of the theorem states that PA_τ does not identify processes not containing τ which differ w.r.t. *PA*.

EXAMPLE 4.2. If the τ-laws constitute a congruence, then since $PA_\tau \vdash a\tau = a$ we must also have $PA_\tau \vdash a\tau \between b = a \between b$. Indeed:

$$a\tau \between b = a(\tau \| b) = a(\tau b + b\tau) = a(\tau b + b) = a\tau b = ab = a \between b.$$

The following derivable identity is often useful:

PROPOSITION 4.1. $PA_\tau \vdash \tau(x+y)+x = \tau(x+y)$.

PROOF. $\tau(x+y) = \tau(x+y)+x+y = \tau(x+y)+x+y+x = \tau(x+y)+x$.

In [6] a proof of Theorem 4.2. is given along the following line. On the set of finite acyclic process graphs a reduction procedure is defined which simplifies the graph (lessens its number of edges and nodes) and which is *sound* for $\leftrightarrows_{r,\tau}$. A *normal* proces graph is one in which no further reduction steps are possible. A *rigid* process graph is one which admits only the *trivial* rooted bisimulation *with itself.* (E.g. $a\tau b + ab$ is not rigid since it admits the nontrivial 'auto-bisimulation' as in figure 42.)

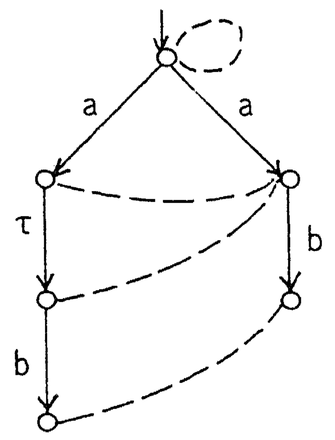


FIGURE 42

Now one can prove that (i) normal process graphs are rigid and (ii) rigid bisimilar process graphs must be identical. This together with the soundness yields the confluency property for the graph reduction procedure (the explicit confluency proof in [6] is in fact superfluous), which in turn implies the completeness of the graph reduction procedure w.r.t. $\leftrightarrows_{r,\tau}$.

An example to see how the graph reduction procedure translates into the

axioms T1-3: one of the reduction steps consists of replacing in a graph a part as in figure 43 (i) by the part in figure 43 (ii) (i.e. deleting an a-step).

FIGURE 43 (i) (ii)

In terms of terms this amounts to $a(\tau x+y)+ax=a(\tau x+y)$.

We remark that the confluency result mentioned above only holds for the graph reduction procedure; when T1-3 are viewed as reduction rules (in whatever direction), together with a restatement of PA as a rewrite system (i.e. choosing the direction left to right in all but the axioms for commutativity and associativity) confluency does *not* hold.

Before extending the PA_τ-formalism with communication, we mention the following curious fact (which is significant for some choices in the development of the present theory):

PROPOSITION 4.2. *The equation $X=a+\tau X$ has infinitely many solutions in the initial model of PA_τ.*

PROOF. If p is a solution, then also $\tau(p+q)$ is a solution for arbitrary q:

$$\begin{aligned} a+\tau\tau(p+q) &= a+\tau(p+q)=a+p+\tau(p+q) \\ &= a+a+\tau p+\tau(p+q) \\ &= a+\tau p+\tau(p+q) = p+\tau(p+q)=\tau(p+q). \end{aligned}$$

Therefore, since τa is a solution (by T1 and T2), $\tau(\tau a+q)$ solves the equation for arbitrary q. This proves the proposition.

Although we do not treat infinite processes here, we note as a corollary from this proposition that recursion equations, guarded by atoms from $A\cup\{\tau\}$, are no longer an adequate specification mechanism for infinite processes as they do not have unique solutions.

4.2. Hiding internal steps in finite processes with communication: ACP_τ

The virtue of the τ-laws T1-3 is not yet fully realized in PA_τ; it is more realized in the presence of communication — indeed the motivation for rejecting some alternative to the τ-laws as in the example in the introduction to this section was stated in terms of communication behaviour. Therefore we want to combine ACP with the τ-laws; the result is the axiom system ACP_τ in Table 6.

It turns out that (apart from the τ-laws) the atom τ must also in the axioms concerning '$|$' be treated differently from the $a \in A$; otherwise some desirable congruence properties are lost. Namely, a term as $\tau\tau a|\tau\tau b$ will be evaluated in ACP_τ as $a|b$ (and not as $(\tau|\tau)\,(\tau a \| \tau\tau b)$ as ACP would prescribe).

ACP_τ

$x+y = y+x$	A1	$x\tau = x$	T1
$x+(y+z) = (x+y)+z$	A2	$\tau x+x = \tau x$	T2
$x+x = x$	A3	$a(\tau x+y) = a(\tau x+y)+ax$	T3
$(x+y)z = xz+yz$	A4		
$(xy)z = x(yz)$	A5		
$x+\delta = x$	A6		
$\delta x = \delta$	A7		
$a\|b = b\|a$	C1		
$(a\|b)\|c = a\|(b\|c)$	C2		
$\delta\|a = \delta$	C3		
$x\|y = x\lfloor y+y\lfloor x+x\|y$	CM1		
$a\lfloor x = ax$	CM2	$\tau\lfloor x = \tau x$	TM1
$(ax)\lfloor y = a(x\|y)$	CM3	$(\tau x)\lfloor y = \tau(x\|y)$	TM2
$(x+y)\lfloor z = x\lfloor z+y\lfloor z$	CM4	$\tau\|x = \delta$	TC1
$(ax)\|b = (a\|b)x$	CM5	$x\|\tau = \delta$	TC2
$a\|(bx) = (a\|b)x$	CM6	$(\tau x)\|y = x\|y$	TC3
$(ax)\|(by) = (a\|b)(x\|y)$	CM7	$x\|(\tau y) = x\|y$	TC4
$(x+y)\|z = x\|z+y\|z$	CM8		
$x\|(y+z) = x\|y+x\|z$	CM9		
		$\partial_H(\tau)=\tau$	DT
		$\tau_I(\tau) = \tau$	TI1
$\partial_H(a) = a$ if $a \notin H \subseteq A$	D1	$\tau_I(a) = a$ if $a \notin I \subseteq A-\{\delta\}$	TI2
$\partial_H(a) = \delta$ if $a \in H$	D2	$\tau_I(a) = \tau$ if $a \in I$	TI3
$\partial_H(x+y) = \partial_H(x)+\partial_H(y)$	D3	$\tau_I(x+y) = \tau_I(x)+\tau_I(y)$	TI4
$\partial_H(xy) = \partial_H(x).\,\partial_H(y)$	D4	$\tau_I(xy) = \tau_I(x).\,\tau_I(y)$	TI5

TABLE 6

Here the alphabet is $A \cup \{\tau\}$; and a,b in Table 6 vary over A only. In the renaming operators ∂_H, τ_I we require $\tau \notin H$ and $\delta \notin I$, since these constants

should not be renamed.

In order to discuss some properties of ACP_τ, we begin with establishing a graph model for ACP_τ.

4.2.1. The model of finite acyclic process graphs for ACP_τ. Consider, as in 4.1.1, the collection $\mathfrak{X}$ of finite acyclic process graphs over $A \cup \{\tau\}$. In Theorem 4.2 (ii) it was stated that $\mathfrak{X}/\leftrightarrows_{r,\tau}$, i.e. the collection of finite acyclic graphs modulo rooted τ-bisimulation, is (isomorphic to) the initial algebra of PA_τ. (In fact, we used a loose formulation there by not distinguishing $\mathfrak{X}$ from A_ω.) We will now do the same in the context of ACP_τ.

The operations $\|, \lfloor\!\lfloor, |, \partial_H, \tau_I$ on $\mathfrak{X}$ are defined as follows. The definition of ∂_H and τ_I is clear — their effect is merely renaming some atoms (labels at the edges) into δ resp. τ. The definition of $\|$ and $\lfloor\!\lfloor$ is also easy: it is analogous to that for ACP (see 2.1.2) with the additional communication $\tau|a = \delta$ for all $a \in A$ and $\tau|\tau = \delta$. The communication merge $g_1|g_2$ is different now:

$$g_1|g_2 = \sum\{(s \longrightarrow s')\,(g_1\|g_2)_{s'} | s \longrightarrow s' \text{ is a maximal com. step in } g_1\|g_2\}.$$

Here $(g)_s$ denotes the subgraph of g with root s ($\in$ NODES (g)) and 'maximal' refers to the accessibility ordering on EDGES(g) (i.e. $s_1 \longrightarrow s_2$ is greater in this ordering than $s_3 \longrightarrow s_4$ if s_3 can be reached from s_2). A 'communication step' in $g_1\|g_2$ is one obtained as a 'diagonal' edge $\xrightarrow{a|b}$, resulting from the communication of $\xrightarrow{a}$ and $\xrightarrow{b}$

The structure $X = \mathfrak{X}(+, \cdot, \|, \lfloor\!\lfloor, |, \partial_H, \tau_I, \delta, \tau)$ is not yet a model of ACP_τ. It has a homomorphic image which is a model of ACP_τ, and which is obtained by dividing out $\leftrightarrows_{r,\tau}$, rooted τ-bisimulation. To define $\leftrightarrows_{r,\tau}$ on the elements of $\mathfrak{X}$, we must extend the definition of $\leftrightarrows_{r,\tau}$, given before, such that the presence of δ's in graphs is taken into account: this is done as above in 2.1.2, so that in effect we work with 'δ-normal graphs'.

Now one can prove the important fact:

LEMMA 4.1.
(i) *Rooted τ-bisimulation is a congruence on* X.
(ii) $\mathfrak{X}/\leftrightarrows_{r,\tau} \models ACP_\tau$.

To prove this, we use results in [6] stating that $\leftrightarrows_{r,\tau}$ can be analyzed into some elementary graph reductions which have the confluency property. Denoting the subset of axioms $A\,1-7$, $T\,1-3$ of ACP_τ by AT, we have, also essentially from [6], the following proposition.

PROPOSITION 4.2. *Let t,s be terms built from $A \cup \{\tau\}$ by $+$ and only. Then:*

$$\mathfrak{X}/\leftrightarrows_{r,\tau} \models t = s \Rightarrow AT \vdash t = s.$$

Now consider $\Sigma = ACP_\tau - AT$, the set of axioms of ACP_τ minus AT. This set of axioms gives rise to a rewrite system (in fact on equivalence classes of terms

modulo the associativity and commutativity axioms, $A\,1,2,5$) by choosing in every axiom the direction from left to right. Let $\xrightarrow{\Sigma}$be one step reduction, and $\xrightarrow{\Sigma}\!\!\!\!\!\rightarrow$ be the transitive reflexive closure of $\xrightarrow{\Sigma}$. The reductions in Σ are confluent and terminating. Let $\xrightarrow{\Sigma}$denote reduction to the unique normal form. (Note that these normal forms are built by $+,\cdot$ only.) Then, applying Proposition 4.2 on t_3,t_4 in the diagram of figure 44, (together with Lemma 4.1 (ii)) we have immediately:

LEMMA 4.2.

(i) *I.e. if* $ACP_\tau \vdash t_1 = t_2$, *then* t_1 *and* t_2 *can be reduced by means of the rewrite rules (from left to right) associated to the axioms in* $ACP_\tau - AT$ *to normal forms* t_3,t_4 *which are convertible via the AT-axioms.*

(ii) *Every term t can be proved equal in* ACP_τ *to a term t′ built from* $A \cup \{\tau\}$ *by* $+$ *and* $\cdot$ *only; moreover, t′ is unique modulo* $\leftrightarrows_{r,\tau}$.

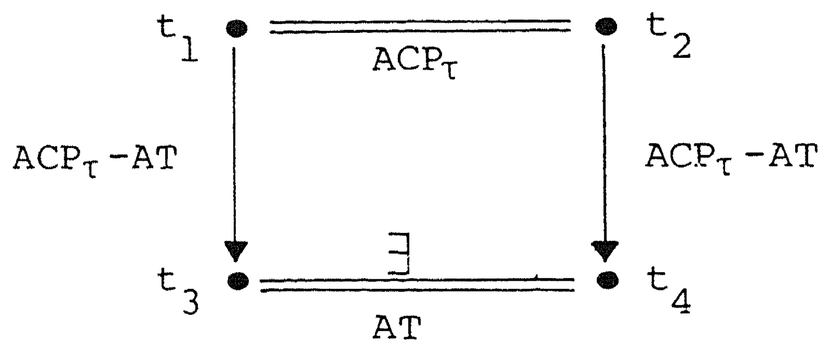
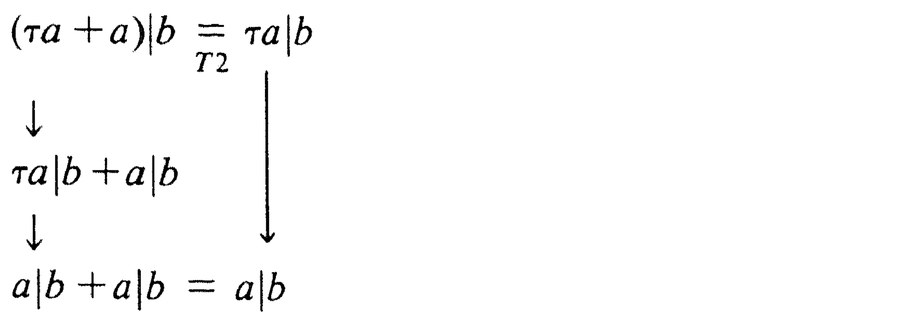
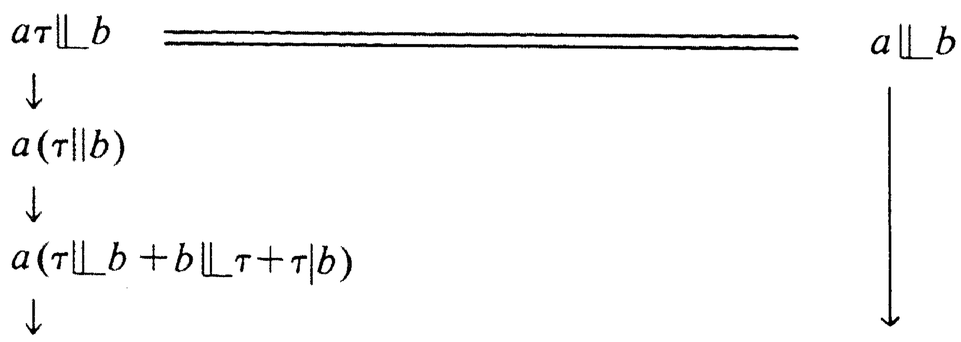


FIGURE 44

EXAMPLE 4.3. The following examples illustrate Lemma 4.2 (i):



(i)



(ii)

$$
\begin{array}{lcl}
(\tau a + a)\between b & = & \tau a \between b \\
\downarrow & & \downarrow \\
\tau a \between b + a \between b & & \tau(a \| b) \\
\downarrow & & \Big\downarrow \\
\tau(a \| b) + a \between b & & \\
\downarrow & & \\
\tau(a \between b + b \between a + a|b) + a \between b & & \\
\downarrow & & \\
\tau(ab + ba + a|b) + ab & \underset{(*)}{=} & \tau(ab + ba + a|b)
\end{array}
\qquad \text{(iii)}
$$

Here (*) is an instance of the (from AT) derivable rule $\tau(x+y)+x=\tau(x+y)$ as in Proposition 4.1.

A further corollary of Lemma 4.1 and 4.2 is:

THEOREM 4.3.
(i) *$\mathcal{X}/\leftrightarrows_{r,\tau}$ is isomorphic to $I(ACP_\tau)$, the initial algebra of ACP_τ.*
(ii) *ACP_τ is conservative over ACP (the latter over the alphabet A). I.e., for τ-less terms t_1, t_2: $ACP_\tau \vdash t_1 = t_2 \Rightarrow ACP \vdash t_1 = t_2$.*

A corollary of Theorem 4.3 (i) and the fact that $\|$ in ACP_τ behaves like $\|$ in ACP is the associativity of $\|$:

PROPOSITION 4.3. $I(ACP_\tau) \models x\|(y\|z) = (x\|y)\|z$

In fact, $I(ACP_\tau)$ satisfies all 'axioms of standard concurrency' as in 2.2 (Table 4) except the second one. Although this second axiom $(x|y)\between z = x|(y \between z)$ does not hold in $I(ACP_\tau)$, as can be seen by evaluating $(a|\tau b)\between c$ to $(a|b)c$ and $a|(\tau b \between c)$ to $(a|b)c + (a|c)b + a|b|c$, a restricted form does hold in $I(ACP_\tau)$, namely:

$$(x|ay)\between z = x|(ay \between z).$$

In view of the linearity of $|$ and $\between$ this can be rephrased as follows: $I(ACP_\tau) \models (x|y)\between z = x|(y \between z)$ for *stable* y. Here y is stable, in the terminology of MILNER [14], if y admits no τ-step as a first step.

Some other useful identities in $I(ACP_\tau)$ are:

$$x\|\tau y = \tau x\|y = \tau(x\|y)$$

$$x \between \tau y = x \between y, \quad x \between \tau = x.$$

For a binary communication mechanism (so that the handshaking axiom $x|y|z=\delta$ holds) we have analogous to the Milner Expansion Theorem 2.2:

THEOREM 4.4 (EXPANSION THEOREM FOR ACP_τ). *Let $a|b|c=\delta$ for all $a,b,c\in A$. Then, in the notation of Theorem 2.2:*

$$I(ACP_\tau)\vdash x_1\|\ldots\|x_k = \sum_{1\leqslant i\leqslant k} x_i\llcorner X_k^i + \sum_{1\leqslant i<j\leqslant k}(x_i|x_j)\llcorner X_k^{i,j}.$$

This is not a straightforward generalization of Theorem 2.2, since our proof of that theorem employed the axioms of standard concurrency (in Table 4) of which, as remarked above, the second one does not hold in $I(ACP_\tau)$.

The diagram in figure 45 gives an impression of the modular construction of ACP_τ. Here $\Sigma_1 \trianglelefteq \Sigma_2$ means that Σ_2 is a conservative extension of Σ_1; for each axiom system part of the signature (viz. the alphabet) is mentioned.

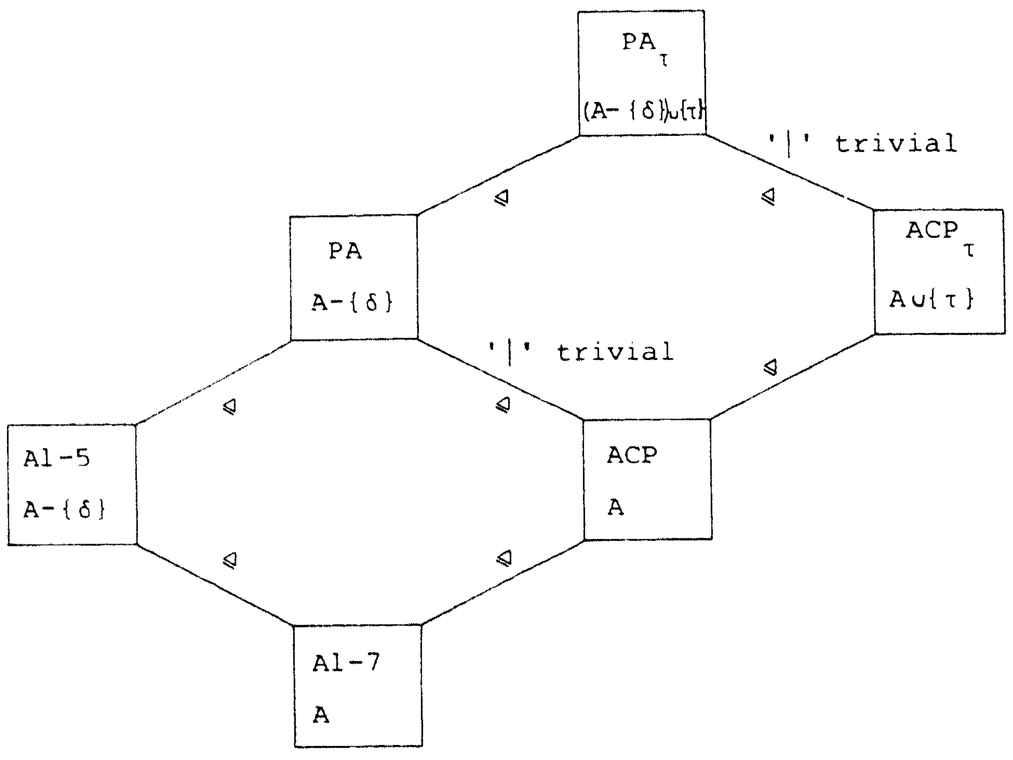


FIGURE 45

4.3. Concluding remarks

In [6] we have described an abstraction mechanism that is at least able to deal with the following situation: suppose two channels (say, bags B_1, B_2) are connected in series:

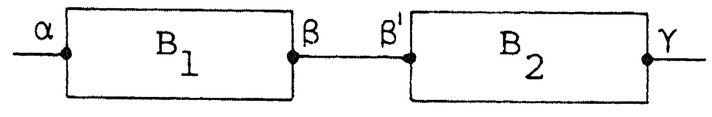


FIGURE 46

The result, clearly, is again a bag B; however in B there are internal steps visible, viz. the passings of the data through the port connection $\beta-\beta'$. Now a minimal requirement for an adequate abstraction mechanism is that it can deal with such a simple situation: the mechanism should be able to hide the

internal data transmissions and allow a proof that the connection of B_1, B_2 yields again a bag.

It is hard to find the 'canonical' extension of the above algebraic framework for finite processes with internal steps to infinite processes. This has to do with the possible presence of infinitely long traces of internal steps. E.g. the notion of bisimulation can be extended to the infinite case in several nonequivalent ways whose consequences are by no means immediately clear. One possibility, which is (formally) the straightforward generalization of $\leftrightarrows_{r,\tau}$, admits the possibility of collapsing infinite τ-traces; thus equating 'the' solution X of the recursion equation $X = a + \tau X$ (which one would expect to be as in figure 47) with the finite process $'\tau a'$.

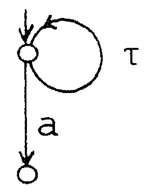


FIGURE 47

Two difficulties arise here, one technical, one conceptual. The technical problem was mentioned in the remark after Proposition 4.2: X is *not* uniquely determined by $X = a + \tau X$. The conceptual problem is that equating X with $'\tau a'$ implies a certain fairness assumption, viz. that X will not always take the option τ. Interestingly, this built in fairness assumption can be used to attack the problem of protocol verification (where the fairness assumption is that a defective channel will not always be defective), as was pointed out to us by C.J. KOOMEN [13].

It is also possible to extend $\leftrightarrows_{r,\tau}$ in another way, such that infinite τ-traces cannot be collapsed. In that case $'\tau a'$ and X are different. It might be that here is a bifurcation point in the development of the theory.

However, for many purposes such as the one explained above (proving that composing two bags yields again a bag), one can work within the restricted algebra of finitely branching processes which are bounded in the sense of not having infinite τ-traces. Here all 'reasonable' extensions of the concept of bisimulation coincide. In [6] an abstraction mechanism was worked out which essentially resides in this algebra of bounded processes.

Even though, maybe, the real interest is for infinite processes with invisible steps, it is certainly safe to say that an adequate algebraic framework to deal with them presupposes a clear understanding of such a framework for the finite case; and that was the subject of this last section.

References

1. J.W. de Bakker, J.I. Zucker (1982). Denotational semantics of concurrency. *Proc. 14th ACM Symp. on Theory of Computing,* p. 153-158.
2. J.W. de Bakker, J.I. Zucker (1982). Processes and the denotational semantics of concurrency. *Information and Control, Vol. 54, No. 1/2,* 70-120.
3. J.A. Bergstra, J.W. Klop (1982). *Fixed Point Semantics in Process Algebras,* Department of Computer Science Technical Report IW 206/82, Mathematisch Centrum, Amsterdam.
4. J.A. Bergstra, J.W. Klop (1984). Process algebra for synchronous communication. *Information and Control 60 1-3,* 109-137.
5. J.A. Bergstra, J.W. Klop (1983). *A Process Algebra for the Operational Semantics of Static Data Flow Networks,* Department of Computer Science Technical Report IW 222/83, Mathematisch Centrum, Amsterdam.
6. J.A. Bergstra, J.W. Klop (1983). *An Abstraction Mechanism for Process Algebras,* Department of Computer Science Technical Report IW 231/83, Mathematisch Centrum, Amsterdam.
7. J.A. Bergstra, J.W. Klop (1983). *An Algebraic Specification Method for Processes over a Finite Action Set,* Department of Computer Science Technical Report IW 232/83, Mathematisch Centrum, Amsterdam.
8. J.A. Bergstra, J.W. Klop (1984). The algebra of recursively defined processes and the algebra of regular processes. J. Paredaens (ed.). *Proc. 11th ICALP Antwerpen,* Springer LNCS 172, 82-94.
9. J.A. Bergstra, J.W. Klop, J.V. Tucker (1983). Algebraic tools for system construction. E. Clarke, D. Kozen (eds.). *Logic of Programs, Proc. 1983,* Springer LNCS 164, 34-44.
10. J.A. Bergstra, J. Tiuryn (1983). *Process Algebra Semantics for Queues,* Department of Computer Science Technical Report IW 241/83, Mathematisch Centrum, Amsterdam.
11. M. Hennessy (1981). A term model for synchronous processes. *Information and Control, Vol. 51, No. 1,* 58-75.
12. C.A.R. Hoare (1980). A model for communicating sequential processes. R.M. McKeag, A.M. McNaghton (eds.). *On the construction of programs,* Cambridge University Press, 229-243.
13. C.J. Koomen (1983). Personal communication.
14. R. Milner (1980). *A Calculus for Communicating Systems,* Springer LNCS 92.
15. R. Milner (1983). Calculi for synchrony and asynchrony. *Theoretical Computer Science 25,* 267-310.
16. D.M.R. Park (1981). Concurrency and automata on infinite sequences. *Proc. 5th GI (Gesellschaft für Informatik) Conference,* Springer LNCS 104.

Relaxation Times for Queueing Systems

J.P.C. Blanc
Technical University Delft
P.O. Box 356, 2600 AJ Delft, The Netherlands

E.A. van Doorn
Technical University Twente
P.O. Box 217, 7500 AE Enschede-Drienerlo, The Netherlands

When a stochastic queueing model is used for performance analysis of, e.g., a computer or communication system, the steady-state situation is usually assumed to prevail. Since many systems exist where the validity of this assumption is questionable, while determination of the time-dependent behaviour of the system is difficult or even impossible, some simple means to characterize the speed with which system performance measures tend to their steady-state values is called for. In this paper the concept of relaxation time is put forward to provide such a characterization.We give a survey of results pertaining to relaxation times for a variety of queueing models. Also, some conjectures and open problems will be mentioned.

1. Introduction

Queueing models are widely used for modeling and performance evaluation of computer and communication systems, as well as in many other fields. Suppose we want to predict the behaviour of a system for various values of the system parameters on the basis of such a model. The usual procedure is then to determine, either analytically or by simulation, steady-state characteristics of the model and to draw conclusions from the results thus found. Although acceptable in a majority of cases, many examples can be found where this procedure leads to conclusions which are unacceptably inaccurate, because no account is taken of the fact that the system starts working at a certain time t_0, say, under initial conditions which are known not to reflect steady-state behaviour. As a consequence, system behaviour at a time t, $t_0<t<\infty$, might deviate considerably from what steady-state results predict. A typical example occurs in a simulation context when one starts at t_0 with an empty system and draws conclusions from observations obtained during the period $[t_0,t_1]$ (see [26] for other examples).

The analytical approach to remedy this difficulty would be to determine the system's time-dependent behaviour under certain initial conditions, but this is notoriously difficult and, even if it is successful, does not often lead to results which are easy to interpret. This being so, it would be helpful to have at least some criterion for deciding whether the use of steady-state results is justified after some time $t-t_0$ has elapsed. Indeed, this is precisely what we need when

studying system behaviour by simulation. What we seek, therefore, are some means to characterize the speed with which system characteristics tend to their steady-state values.

Since most time-dependent results available indicate that tendency to steady-state is exponential, we are led to the concept of *relaxation time*, which we define, for a function $f(t)$ tending to a finite limit $f(\infty)$, as

$$T(f) = \inf\{T \mid f(t) - f(\infty) = O(e^{-t/T}) \ (t \to \infty)\} \tag{1.1}$$

(In some contexts $T^{-1}(f)$ is called the *decay parameter* of f.) Since we can think of numerous functions related to a particular queueing system (e.g., virtual waiting time, average queue length, probability of the system being empty), it is not a priori clear how to relate the concept of relaxation time to a queueing system, indeed, whether it is possible to do so in a sensible way. However, there are good reasons (about which later) to define the *relaxation time T of a queueing system* as

$$T = T(p_{00}), \tag{1.2}$$

where $p_{00} = p_{00}(t)$ is the probability that the system is empty at time $t > 0$, given that there are no customers in the system at time 0, and that 0 is an arbitrary point of time with respect to the arrival process to the queueing system.

We remark that MORSE [23] seems to have been the first to use the term 'relaxation time' in a queueing context, but his definition differs from ours. COHEN'S [6] definition of relaxation time for a queueing system, although not explicitly related to p_{00}, coincides with ours (cf. also KINGMAN [18]).

We were motivated to choose definition (1.2) by the following considerations. A description of the state of a system at t_0 — the initial conditions — requires (probabilistic) specification of at least

- the number of customers in the system;
- the distribution of the residual interarrival time;
- the distribution of the residual service times if there are customers being served.

Any deviation at t_0 from the steady-state situation in any of these factors leads to time-dependent phenomena. Since we cannot hope to be able to deal with all these effects simultaneously (cf. subsection 3.1), we will concentrate on deviations from the equilibrium distribution of the number of customers in the system. Thus we shall mostly assume that t_0 is an arbitrary point of time with respect to the arrival process and that residual service times are distributed as they are in equilibrium. Then, there are indications, such as KINGMAN'S [19], [20] solidarity theorems for transition probabilities in a Markovian system, and certain results for specific models, that a large number of functions related to a queueing system have a common relaxation time $T(p_{00})$.

In the following sections we shall determine relaxation times for several queueing models, and we shall show that these are the relaxation times for functions other than $p_{00}(t)$ as well, whereby we restrict ourselves to functions

related to the number of customers in the system. It should be noted that knowledge of a relaxation time is not sufficient to answer such concrete questions as: how much time does it take for $p_{00}(t)$ to be within 5% of its steady-state value? For, by definition,

$$p_{00}(t)-p_{00}(\infty) = e^{-t/T}g(t),$$

where $g(t)=O(e^{\epsilon t})$ $(t\to\infty)$ for all $\epsilon>0$, so that we need information on $g(t)$ as well. In many cases, however, such information can be given. Birth-death queueing models will be discussed in Section 2, and the $GI/G/1$ queue in Section 3. In Section 4 we present some results and a conjecture for Jackson queueing networks.

We conclude this introduction with some remarks pertaining to the computation of relaxation times. In many instances we can obtain an explicit representation for the Laplace transform

$$\int_0^\infty e^{-\phi t}(p_{00}(t)-p_0)dt, \quad \mathrm{Re}\phi>0 \tag{1.3}$$

(here, and in what follows, $p_0=p_{00}(\infty)$), which is analytic in the half plane $\mathrm{Re}\phi\leqslant 0$ as well, apart from a finite number of isolated algebraic singularities. In that case the asymptotic behaviour of $p_{00}(t)-p_0$ is determined by the singularity ϕ^* which is closest to the imaginary axis; in fact, it is readily seen from [28], that

$$T(p_{00}) = -\{\mathrm{Re}\phi^*\}^{-1}. \tag{1.4}$$

If $p_0>0$ and, instead of the continuation of (1.3), we consider the continuation of the Laplace transform

$$\Omega(\phi) = \int_o^\infty e^{-\phi t}p_{00}(t)dt, \quad \mathrm{Re}\phi>0, \tag{1.5}$$

then there will be an additional pole at $\phi=0$ which, of course, has no influence on $T(p_{00})$.

2. Birth-death queueing models

In this section we will discuss single server queueing systems for which the queue length process $\{N(t),\ 0\leqslant t<\infty\}$ is a birth-death process. The birth (arrival) rates and death (service) rates will be denoted by λ_n and μ_n, respectively, where n, $n=0,1,\ldots,K$, is the queue length (including the customer in service) and $K-1$ $(1\leqslant K\leqslant\infty)$ the size of the waiting room. We clearly have $\mu_0=0$ and, if $K<\infty$, $\lambda_K=0$. The parameters λ_n and μ_{n+1} for $n=0,1,\ldots,K-1$ are assumed to be positive. The model will be referred to as $M_{(n)}/M_{(n)}/1/K-1$. With appropriate interpretation of the service rates this model encompasses any Markovian multiserver delay and/or loss system with state-dependent arrival and service rates.

It will be convenient to treat the cases $K<\infty$ and $K=\infty$ separately. Before starting off with the finite case we remark that the exponential distributions

involved make that of the three factors mentioned in the introduction only the initial queue length distribution influences the time-dependent behaviour of the pertinent model.

2.1. The $M_{(n)}/M_{(n)}/1/K-1$ queueing system ($K<\infty$)

From KARLIN and MCGREGOR [17] (see also [9] and references there) we know that the transition probabilities

$$p_{ij}(t) = Pr\{N(t)=j \mid N(0)=i\}, \quad i,j=0,1, \ldots, K,$$

for the queue length process of the $M_{(n)}/M_{(n)}/1/K-1$ queue can be represented as

$$p_{ij}(t) = p_j + \pi_j \sum_{n=1}^{K} \exp(-x_n t) Q_i(x_n) Q_j(x_n) \sigma(x_n), \tag{2.1}$$

where the π_n are constants defined by

$$\pi_0 = 1, \quad \pi_n = \frac{\lambda_0 \lambda_1 \ldots \lambda_{n-1}}{\mu_1 \mu_2 \ldots \mu_n} \quad (n>0), \tag{2.2}$$

and the p_n are the steady-state probabilities of having n customers in the system, which satisfy

$$p_n = \pi_n / \sum_{m=0}^{K} \pi_m. \tag{2.3}$$

Further, the Q_n are polynomials defined by the recurrence relation

$$\lambda_n Q_{n+1}(x) = (\lambda_n + \mu_n - x) Q_n(x) - \mu_n Q_{n-1}(x) \tag{2.4}$$

$$Q_{-1}(x)=0, \quad Q_0(x)=1,$$

and the x_n are the (distinct and positive) zeros of

$$S(x) = \{(\mu_K - x) Q_K(x) - \mu_K Q_{K-1}(x)\}/x \tag{2.5}$$

(note that $Q_n(0)=1$ for all $n \geqslant 0$, so that $S(x)$ is a polynomial of degree K). Finally, $\sigma(x)$ is given by

$$\sigma^{-1}(x) = \sum_{m=0}^{K} \pi_m Q_m^2(x) \tag{2.6}$$

(note that $\sigma(x)<1$). Assuming that the zeros x_n are numbered in increasing order of magnitude, we have in particular

$$p_{00}(t) - p_0 = \exp(-x_1 t)\{\sigma(x_1) + o(1)\} \quad (t \to \infty), \tag{2.7}$$

from which it follows that the relaxation time of the $M_{(n)}/M_{(n)}/1/K-1$ queue is given by

$$T = x_1^{-1}. \tag{2.8}$$

Thus determination of the relaxation time of the $M_{(n)}/M_{(n)}/1/K-1$ queue amounts to determination of the smallest zero of a polynomial of degree K. In

most cases this will have to be done numerically (cf. [22]), but it may happen that an explicit expression for T exists, as in the example of § 2.1.1.

It is evident from (2.1) that if $Q_n(x_1)\neq 0$ for $n=0,1\ldots, K$, then T is also the relaxation time for $p_{ij}(t)$ ($i>0$ or $j>0$), and consequently for $p_{\omega j}(t)$ as well, ω indicating any initial queue length distribution which is not the equilibrium distribution. If $Q_n(x_1)=0$ for some n, which is exceptional but not impossible (see § 2.2.1), then T is larger than the relaxation time of $p_{ij}(t)$ ($i=n$ or $j=n$). It can be shown that there is at most one value of n for which $Q_n(x_1)=0$.

Obviously, the state space being finite, T is also the relaxation time for the mean queue length and any higher order moment, irrespective of the initial state distribution.

2.1.1. The $M/M/1/K-1$ queueing system. Our first example illustrating the results of the previous section is the $M/M/1$ queue with finite waiting room of size $K-1$. The arrival and service rates are now independent of the queue length: $\lambda_n=\lambda$, $\mu_n=\mu$, except $\mu_0=\lambda_K=0$. From [16] we obtain the following expression for the polynomials Q_n of (2.4):

$$Q_{n+1}(x) = \left[\frac{\mu}{\lambda}\right]^{n/2}\left\{\left[1-\frac{x}{\lambda}\right]U_n(\xi(x))-\left[\frac{\mu}{\lambda}\right]^{1/2}U_{n-1}(\xi(x))\right\},\quad n\geqslant 0, \tag{2.9}$$

where

$$\xi(x) = \tfrac{1}{2}(\lambda\mu)^{-1/2}(\lambda+\mu-x), \tag{2.10}$$

and the U_n are Chebysev polynomials of the second kind, which satisfy

$$U_n(\cos\theta) = \sin(n+1)\theta/\sin\theta. \tag{2.11}$$

Some simple algebra involving the recurrence formula

$$\begin{aligned} &2\xi U_n(\xi) = U_{n-1}(\xi)+U_{n+1}(\xi)\\ &U_{-1}(\xi)=0,\quad U_0(\xi)=1 \end{aligned} \tag{2.12}$$

for these polynomials, gives us

$$S(x) = -\left[\frac{\mu}{\lambda}\right]^{K/2}U_K(\xi(x)). \tag{2.13}$$

We see from (2.11) that the zeros of $U_K(\xi)$ are $\cos n\pi/(K+1)$, $n=1,2,\ldots,K$, so that, by (2.10) and (2.13), the zeros of $S(x)$ are

$$x_n = \lambda+\mu-2(\lambda\mu)^{1/2}\cos n\pi/(K+1),\quad n=1,2,\ldots,K. \tag{2.14}$$

It follows that

$$T = x_1^{-1}=\{\lambda+\mu-2(\lambda\mu)^{1/2}\cos\pi/(K+1)\}^{-1}. \tag{2.15}$$

This result is also implicitly contained in [24, p. 65] and [30, p. 13].

We finally note the following. Since $Q_1(x)=1-x/\lambda$, we have $Q_1(\lambda)=0$. Now, if λ and μ are such that

$$\mu = 4\lambda(\cos \pi/(K+1))^2,$$

then

$$x_1 = \lambda+\mu-2(\lambda\mu)^{1/2}\cos \pi/(K+1) = \lambda,$$

so that $Q_1(x_1)=0$. This is an example of the exceptional situation referred to in the previous section, since the relaxation time of, e.g., $p_{10}(t)$ is smaller than T.

2.1.2. The $M/M/K/0$ loss system. The loss system $M/M/K/0$ $(K<\infty)$ can be viewed as an $M/M_{(n)}/1/K-1$ queue where $\lambda_n=\lambda$ and $\mu_n=n\mu$, $n=0,1,\ldots,K$, except $\lambda_K=0$. Hence the technique of subsection 2.1 can be applied to find the relaxation time of this system. As observed in [16], the polynomials Q_n of (2.4) can now be identified in terms of Charlier polynomials $c_n(x,a)$, viz.

$$Q_n(x) = c_n\left[\frac{x}{\mu}, \frac{\lambda}{\mu}\right], \tag{2.16}$$

where

$$c_n(x,a) = \sum_{m=0}^{n} m! \begin{bmatrix} n \\ m \end{bmatrix} \begin{bmatrix} x \\ m \end{bmatrix} (-a)^{-m}. \tag{2.17}$$

Using the recurrence relation for Charlier polynomials [16], it can subsequently be shown that

$$S(x) = \frac{\lambda}{x}\left\{c_{K+1}\left[\frac{x}{\mu}, \frac{\lambda}{\mu}\right] - c_K\left[\frac{x}{\mu}, \frac{\lambda}{\mu}\right]\right\}. \tag{2.18}$$

Unfortunately, no explicit expressions for the zeros of $S(x)$ seem to exist in general, so that x_1 has to be evaluated numerically.

2.2. The $M_{(n)}/M_{(n)}/1/\infty$ queueing system

Let us assume that the arrival and service rates λ_n and μ_n of the $M_{(n)}/M_{(n)}/1/\infty$ queue satisfy the condition

$$\sum_{n=0}^{\infty} (\lambda_n \pi_n)^{-1} \sum_{m=0}^{n} \pi_m = \infty, \tag{2.19}$$

where the π_n are given by (2.2). This condition, which is fulfilled in all practical applications, ensures that there is a unique set $\{p_{ij}(t)\}$ of transition probabilities satisfying the usual requirements (see, e.g., [27]).

KARLIN and MCGREGOR [14] have shown that there is an analogue to (2.1) which reads

$$p_{ij}(t) = p_j + \pi_j \int_{0+}^{\infty} \exp(-xt) Q_i(x) Q_j(x) d\psi(x), \tag{2.20}$$

where the Q_n are the polynomials of (2.4), and the p_n, the steady-state

probabilities, are now given by

$$p_n = \pi_n / \sum_{m=0}^{\infty} \pi_m, \tag{2.21}$$

which is to be interpreted as zero if $\Sigma\pi_m = \infty$. Finally, $d\psi$ is the unique mass distribution on $[0,\infty)$ of total mass 1 with respect to which the polynomials $\{Q_n\}_{n=0}^{\infty}$ are orthogonal:

$$\int_0^{\infty} Q_i(x)Q_j(x)d\psi(x) = \pi_j^{-1}\delta_{ij}. \tag{2.22}$$

In particular we have

$$p_{00}(t) - p_0 = \int_{0+}^{\infty} \exp(-xt)d\psi(x). \tag{2.23}$$

By $S(d\psi)$ we denote the support of $d\psi$, i.e.,

$$S(d\psi) = \{x \mid \int_{-\epsilon}^{+\epsilon} d\psi(x) > 0 \quad \text{for all} \quad \epsilon > 0\},$$

and we let

$$\gamma(d\psi) = inf\{x > 0 \mid x \in S(d\psi)\}. \tag{2.24}$$

CALLAERT [3], [4] (see [11] for a simpler proof) has shown that

$$\gamma(d\psi) = sup\{\eta \mid p_{00}(t) - p_0 = O(e^{-\eta t})\ (t \to \infty)\}, \tag{2.25}$$

so that the relaxation time of the $M_{(n)}/M_{(n)}/1/\infty$ queue satisfies

$$T = \{\gamma(d\psi)\}^{-1}. \tag{2.26}$$

When it comes to calculation of T for specific parameters several possibilities present themselves.The first (and most convenient) possibility is that $\{Q_n\}$ constitutes a system of orthogonal polynomials for which the associated mass distribution is known. One such example will be presented in § 2.2.1.

The second possibility is to follow a suggestion by KARLIN and MCGREGOR [15] to try to find the Stieltjes transform

$$\Psi(\phi) = \int_{0+}^{\infty} \frac{d\psi(x)}{x + \phi}, \quad |arg\phi| < \pi,\ |\phi| > 0, \tag{2.27}$$

and to calculate the largest singularity of the analytic continuation of $\Psi(\phi)$. This singularity equals $-\gamma(d\psi)$, which is evident from our remark on Laplace transforms in the introduction and the fact that

$$\int_0^{\infty} e^{-\phi t}(p_{00}(t) - p_0)dt = \Psi(\phi), \quad \mathrm{Re}\,\phi > 0, \tag{2.28}$$

as can be seen by substituting the representation (2.23) for $p_{00}(t) - p_0$ (cf. [31, Section VIII.4]). The example of § 2.2.3 was analyzed in this manner in [16].

A third possibility to calculate T is of course to forget about the representation (2.23) and to try to find the Laplace transform of $p_{00}(t)-p_0$ by some other method. The model of § 2.2.2 was originally studied in this way; some of the results presented in § 2.2.3 were thus obtained as well.

If none of these approaches leads to a result which is computationally expedient, one has to content oneself with an approximation for T (cf. [11]).

We next address the question of whether T is the relaxation time of time-dependent functions other than $p_{00}(t)$. As regards the transition probabilities $p_{ij}(t)$ ($i>0$ or $j>0$), it can be shown (see [3], [4]) that this is true indeed, except perhaps when $Q_i(\gamma)=0$ or $Q_j(\gamma)=0$ ($\gamma=\gamma(d\psi)$), in which case the relaxation time can be smaller than T. Setting aside this exceptional case (there is at most one value of n for which $Q_n(\gamma)=0$), it follows that the probabilities $p_{\omega j}(t)$, ω denoting any initial distribution with finite support, have relaxation time T. If ω has infinite support, it may happen that the relaxation time of $p_{\omega j}(t)$ is larger than T. An example of this phenomenon will be given in § 2.2.3.

It seems likely that the relaxation time for the mean queue length (and higher order moments) equals T, provided the initial distribution does not have the disturbing effects described above. A thorough investigation of this question on the basis of (2.20) requires more insight into the behaviour of the polynomials $\{Q_n\}$ than is present as yet.

2.2.1. The system $M/M/\infty$. The 'queue length' process in the system $M/M/\infty$ is probabilistically equivalent to that of an $M/M_{(n)}/1/\infty$ queue where $\lambda_n=\lambda$ and $\mu_n=n\mu$, $n=0,1,\ldots$, so that the representation (2.20) is valid. The polynomials Q_n corresponding to this queue can be identified in terms of Charlier polynomials as in (2.16). Since the Charlier polynomials $c_n(x,a)$ are orthogonal with respect to a distribution $d\psi$ which consists of masses $e^{-a}a^x/x!$ at the points $x=0,1,\ldots$, it follows that

$$T = \mu^{-1} \tag{2.29}$$

(cf. [16]). This result can also be obtained from Takács' findings for the system $M/G/\infty$ [30, p. 160].

2.2.2. A queueing model where potential customers are discouraged by queue length. We next consider a system of the type $M_{(n)}/M/1/\infty$, where $\mu_{n+1}=\mu$ and $\lambda_n=\lambda/(n+1)$, $n\geqslant 0$, which is intended to model decreasing willingness of a customer to join the queue as the queue length increases (see [25], [8] and references there; a more general model is studied in [5]).

Either directly from [5] or via the expression for the Laplace transform of $p_{00}(t)$ as given in [25], it follows that

$$T = 4\{(\lambda+4\mu)^{1/2}-\lambda^{1/2}\}^{-2}. \tag{2.30}$$

It is interesting to compare this result with the relaxation time for the $M/M/1/K-1$ queue of § 2.1.1, which can also be said to model customer reluctance to join a long queue. It is seen that for fixed μ the relaxation time

of (2.30) tends to infinity, whereas the relaxation time of (2.15) tends to zero as λ goes to infinity.

2.2.3. The $M/M/s$ queueing system ($s<\infty$). The distribution of the number of customers in the queue $M/M/s$ ($0<s<\infty$) can be studied in terms of an $M/M_{(n)}/1/\infty$ model where $\lambda_n=\lambda$ and

$$\mu_n = \begin{cases} n\mu & n=0,1,\ldots,s \\ s\mu & n=s+1,\ s+2,\ \ldots \end{cases}$$

For this model KARLIN and MCGREGOR [16] have explicitly determined the Stieltjes transform (2.27). On the basis of this result it was shown in [10, Ch. 6] that the relaxation time of the $M/M/s$ queue satisfies

$$T = (\lambda^{1/2}-(s\mu)^{1/2})^{-2}, \tag{2.31}$$

provided the traffic intensity $\rho=\lambda/s\mu$ is not smaller than some critical value ρ^*. If $\rho<\rho^*$, then T is larger than the right hand side of (2.31); actually, T^{-1} equals λ times the smallest positive root of the equation

$$R_s(x,\rho) = C(x), \tag{2.32}$$

where

$$C(x) = \tfrac{1}{2}\{1-x+\rho^{-1}-\{(1-x+\rho^{-1})^2-4\rho^{-1}\}^{1/2}\}, \tag{2.33}$$

and $R_s(x,y)$ is determined by the recurrence relations

$$\begin{aligned} R_{n+1}(x,y) &= 1-x+\frac{n}{sy}\{1-R_n^{-1}(x,y)\} \quad n=1,2,\ldots,s-1, \\ R_1(x,y) &= 1-x. \end{aligned} \tag{2.34}$$

The critical value ρ^* is the largest root <1 of the equation

$$R_s(1-x^{-1/2},x) = x^{-1/2}, \tag{2.35}$$

if $s>1$, and $\rho^*=0$ if $s=1$. Some values of ρ^* are given in Table 2.1.

EXAMPLE. For $s=2$ and $\rho<\rho^*=\frac{1}{9}$ an explicit expression can be obtained from the above calculation scheme, viz.,

$$T = 2\{2\lambda+\mu+(\mu^2-4\lambda\mu)^{1/2}\}^{-1}, \tag{2.36}$$

in agreement with [4].

TABLE 2.1. Critical values for the traffic intensity in the system $M/M/s$

s	ρ^*
1	0
2	0.111
3	0.211
4	0.284
5	0.340
10	0.498

There is an interesting difference between the cases $\rho<\rho^*$ and $\rho\geqslant\rho^*$ as regards the asymptotic behaviour of $p_{00}(t)-p_0$ (cf. [10, Ch. 6]). Namely, if $\rho<\rho^*$ then the distribution $d\psi$ contains an isolated point mass m, say, in the point $\gamma(d\psi)=T^{-1}$, and we have

$$p_{00}(t)-p_0 = \exp(-t/T)\{m+O(1)\} \quad (t\to\infty). \tag{2.37}$$

If $\rho\geqslant\rho^*$, however, the distribution $d\psi$ has zero mass concentrated at $\gamma(d\psi)$, and we have

$$p_{00}(t)-p_0 = \exp(-t/T)\{O(1)\} \quad (t\to\infty). \tag{2.38}$$

The order term in (2.38), although not exponential, can enlarge considerably the speed with which $p_{00}(t)-p_0$ tends to zero as t goes to infinity. This phenomenon will also present itself when one considers functions like the transition probabilities $p_{ij}(t)$ ($i>0$ or $j>0$) and the mean queue length, and explains the fact that ODONI and ROTH [26] found the relaxation time to be a conservative measure for the speed to steady state in the $M/M/1$ queue (where $\rho^*=0$). Indeed, for this system a more precise result than (2.38) can be obtained from [6, p. 84] to the effect that

$$p_{00}(t)-p_0 = \exp(-t/T)(t/T)^{-3/2}\left\{\frac{(1-\rho)\rho^{1/4}}{2\pi^{1/2}(1+\rho^{1/2})}+O(t^{-1})\right\} \quad (t\to\infty), \tag{2.39}$$

provided $\rho=(\lambda/\mu)\neq 1$. It can be shown (cf. [29]) that also for $s>1$ a factor $t^{-3/2}$ dominates the order term in (2.38), provided $\rho>\rho^*$ (and $\rho\neq 1$).

We remark that COHEN [6, p. 180] has obtained asymptotic expressions for the expected queue length in the system $M/M/1$, which show that the relaxation time for this function is given by (2.31) (with $s=1$) if the initial distribution has finite support.

The $M/M/1$ queue will now provide us with an example of the deteriorating effect an initial distribution with infinite support can have on the relaxation time of $Pr\{N(t)=0\}$. From [6, p. 80] it follows that for $i=0,1,\ldots$,

$$\int_0^\infty e^{-\phi t}p_{i0}(t)dt = \frac{\{C(-\phi/\lambda)\}^{i+1}}{\mu\{1-C(-\phi/\lambda)\}}, \quad \mathrm{Re}\,\phi>0, \tag{2.40}$$

where $C(\cdot)$ is the function defined in (2.33). Since $C(-\phi/\lambda)$ is bounded in absolute value by 1 for $\mathrm{Re}\phi>0$, this implies

$$\sum_{i=0}^{\infty}(1-r)r^i\int_0^{\infty}e^{-\phi t}p_{i0}(t)dt = \frac{(1-r)C(-\phi/\lambda)}{\mu\{1-C(-\phi/\lambda)\}\{1-rC(-\phi/\lambda)\}}, \qquad (2.41)$$
$$\mathrm{Re}\,\phi>0, \quad 0<r<1.$$

Apart from the pole at $\phi=0$ ($C(0)=1$), which accounts for the steady-state probability p_0, the largest singularities of the right hand side of (2.41) are the branch point $\phi=-(\lambda^{1/2}-\mu^{1/2})^2$ of the function $C(-\phi/\lambda)$ and the pole at $\phi=-(1-r)(\mu-\lambda/r)$ where $C(-\phi/\lambda)=1/r$. It is readily verified that these singularities coincide for $r=\rho^{1/2}$, and that for $\rho^{1/2}<r<1$ the pole is larger than the branch point. Hence, the relaxation time T_ω of $Pr\{N(t)=0\}$ when the initial distribution ω satisfies $Pr\{N(0)=i\}=(1-r)r^i$, $i=0,1,\ldots$, and $\rho^{1/2}<r<1$, is given by

$$T_\omega = \{(1-r)(\mu-\lambda/r)\}^{-1}. \qquad (2.42)$$

Note that $T_\omega\to\infty$ as $r\uparrow 1$ so that any relaxation time between $(\lambda^{1/2}-\mu^{1/2})^{-2}$ and infinity can be realized by a proper choice of r.

For completeness' sake we remark that if the initial distribution is equal to the steady-state distribution, i.e., $r=\rho$, then $p_{00}(t)=1-\rho$ for all $t\geqslant 0$, so that the relaxation time of $Pr\{N(t)=0\}$ is equal to zero.

3. The GI/G/1 queueing system

In this section we will discuss relaxation times for $GI/G/1$ queueing systems. Customers arrive at the service facility according to a renewal process with interarrival time distribution $A(x)$, $x\geqslant 0$, and mean interarrival time α. The service times form a sequence of independent random variables with a common distribution $B(x)$, $x\geqslant 0$, and with mean service time β. Unless stated otherwise, the traffic intensity $\rho=\beta/\alpha$ is assumed to be smaller than 1. We let

$$\alpha^*(\theta) = \int_0^{\infty}e^{-\theta x}dA(x), \quad \mathrm{Re}\,\theta\geqslant 0, \qquad (3.1)$$

$$\beta^*(\theta) = \int_0^{\infty}e^{-\theta x}dB(x), \quad \mathrm{Re}\,\theta\geqslant 0.$$

The relaxation time for the $GI/G/1$ queueing system was extensively studied in [6, § III.7.3]. Because explicit expressions for the probability $p_{00}(t)$, cf. Section 1, are only available in a few special cases, the asymptotic behaviour of $p_{00}(t)$ as $t\to\infty$ was studied on the basis of its Laplace transform $\Omega(\phi)$, cf. (1.5). The discussion is restricted to queueing systems with service time distributions having Laplace-Stieltjes transforms $\beta^*(\theta)$ with abscissas of convergence $\theta_b<0$, while

$$\beta^*(\theta)\uparrow\infty, \quad \text{as} \quad \theta\downarrow\theta_b. \qquad (3.2)$$

It turned out that in this case the Laplace transform $\Omega(\phi)$ possesses an analytic

continuation into a part of the left half plane, and the relaxation time of the system is determined by the singularity of $\Omega(\phi)$ with the largest real part (apart from a pole at $\phi=0$). Before stating the general result concerning the relaxation time of the $GI/G/1$ system the queueing systems $GI/M/1$ and $M/G/1$ will be discussed in more detail.

3.1. The GI/M/1 queueing system

For a general, but not lattice, interarrival time distribution $A(x)$ the Laplace transform of $p_{00}^{+}(t)$ is given by, cf. [6, § II.3.4],

$$\Omega^{+}(\phi) = \frac{1}{\phi} - \frac{\zeta(\phi)-\phi}{1-\alpha^{*}(\phi)} \frac{\beta}{\zeta(\phi)}, \quad \mathrm{Re}\phi>0; \tag{3.3}$$

here the superscript + indicates that the first customer arives at $t=0$ (i.e. $N(0+)=1$), and $\zeta=\zeta(\phi)$, $\mathrm{Re}\phi>0$, is the unique root of the equation

$$\alpha^{*}(\zeta) = 1+\beta\phi-\beta\zeta, \quad \mathrm{Re}\zeta>0. \tag{3.4}$$

If the assumption that the first customer arrives at $t=0$ is replaced by the assumption that the first customer arrives at a random instant $t=t_1\geqslant 0$, then it is easily verified, that the Laplace transform $\Omega^{\gamma}(\phi)$ of $p_{00}^{\gamma}(t)$ is given by

$$\Omega^{\gamma}(\phi) = \frac{1}{\phi} - \gamma(\phi)\frac{\zeta(\phi)-\phi}{1-\alpha^{*}(\phi)} \frac{\beta}{\zeta(\phi)}, \quad \mathrm{Re}\phi>0; \tag{3.5}$$

here $\gamma(\phi)=E\{\exp(-\phi t_1)\}$. In particular, if t_1 is distributed as the residual interarrival time at an arbitrary instant, so that $t=0$ is an arbitrary instant in the arrival process — in agreement with our definition of relaxation time, cf. Section 1 — , then we find

$$\Omega(\phi) = \frac{1}{\phi} - \frac{\zeta(\phi)-\phi}{\alpha\phi} \frac{\beta}{\zeta(\phi)}, \quad \mathrm{Re}\phi>0. \tag{3.6}$$

Let ζ_0 be the root of $\alpha^{*'}(\zeta)=-\beta$ with the largest real part (ζ_o is the unique positive root of $\alpha^{*'}(\zeta)=-\beta$ in the case $\rho<1$). Then $\zeta(\phi)$ has a branch point at $\phi_0=\zeta_0-[1-\alpha^{*}(\zeta_0)]/\beta$. The Laplace transform $\Omega(\phi)$ is regular in the domain $\mathrm{Re}\phi>\phi_0$, apart from a pole at $\phi=0$. This implies that the relaxation time of the $GI/M/1$ system is equal to

$$T = -1/\phi_0. \tag{3.7}$$

It should be noted however, that if $t=0$ is not an arbitrary instant in the arrival process, it can happen that the transient effects due to the arrival process dominate those due to the queueing mechanism, cf. (3.3), (3.5). To be more precise, the Laplace transform $\Omega^{+}(\phi)$ possesses poles in the domain $\mathrm{Re}\phi>\phi_0$ at points ϕ for which $\alpha^{*}(\phi)=1$ (the same statement holds for $\Omega^{\gamma}(\phi)$ as far as zeros of $1-\alpha^{*}(\phi)$ are not compensated by zeros of $\gamma(\phi)$). Let ϕ_a be the real part of the root(s) of $\alpha^{*}(\phi)=1$, $\phi\neq 0$, with the largest real part(s). Then the relaxation time of $p_{00}^{+}(t)$ is equal to $\max\{-1/\phi_0, -1/\phi_a\}$. As an example consider the $E_m/M/1$ queueing system. For this system ϕ_0 and ϕ_a can be obtained explicitly:

$$\phi_0 = -[1+m\rho-(m+1)\rho^{m/(m+1)}]/\beta, \tag{3.8}$$
$$\phi_a = -m\rho[1-\cos(2\pi/m)]/\beta.$$

Let β be fixed. Because $\phi_a\uparrow 0$ as $\rho\downarrow 0$ and $\phi_0\uparrow 0$ as $\rho\uparrow 1$, it is clear that ϕ_a dominates for small values of ρ and that ϕ_0 determines the relaxation time of $p_{00}^{+}(t)$ for ρ close to unity (see Table 3.2 for the values of ρ for which $\phi_0=\phi_a$ for some $E_m/M/1$ systems). For ρ fixed $\phi_a\uparrow 0$ as $m\to\infty$. This is in agreement with the fact that the limit of $p_{00}^{+}(t)$ as $t\to\infty$ does not exist in the $D/M/1$ system.

REMARK. The zeros of $1-\alpha^*(\phi)$ are not cancelled by zeros of $\zeta(\phi)-\phi$ in (3.3) and (3.5), cf. (3.4), because $\mathrm{Re}\,\zeta(\phi)\geqslant\zeta_0>0$ for $\mathrm{Re}\,\phi\geqslant\phi_0$, while zeros of $1-\alpha^*(\phi)$ have a non-positive real part.

3.2. The M/G/1 queueing system

For the $M/G/1$ queueing system the Laplace transform $\Omega(\phi)$ of $p_{00}(t)$ is given by, cf. [6, § II.4.3],

$$\Omega(\phi) = 1/\xi(\phi), \quad \mathrm{Re}\,\phi>0; \tag{3.9}$$

here $\xi=\xi(\phi)$, $\mathrm{Re}\,\phi>0$, is the unique root of the equation

$$\beta^*(\xi) = 1+\alpha\phi-\alpha\xi, \quad \mathrm{Re}\,\xi>0. \tag{3.10}$$

Because of the assumption (3.2) the equation $\beta^{*\prime}(\xi)=-\alpha$ has a unique root ξ_0 on the interval $(\theta_b,0)$. Therefore, the function $\xi(\phi)$ possesses a branch point at $\phi_0=\xi_0+[\beta^*(\xi_0)-1]/\alpha$. In [6, p. 603] it is shown that $\xi(\phi)$ possesses an analytic continuation into the domain $\mathrm{Re}\,\phi>\phi_0$, and that it has exactly one zero in $\mathrm{Re}\,\phi>\phi_0$, viz. $\phi=0$. Hence ϕ_0 is the singularity with the largest real part of the Laplace transform $\Omega(\phi)$ - apart from a pole at $\phi=0$ -, which implies that the relaxation time of the $M/G/1$ queueing system is equal to

$$T = -1/\phi_0. \tag{3.11}$$

In Table 3.1 relaxation times have been tabulated as a function of ρ for some $M/G/1$ systems in which the service time distributions have rational Laplace-Stieltjes transforms with denominators of degree 1 or 2. Such a Laplace-Stieltjes transform is completely determined by its mean β, its coefficient of variation C_s and its largest pole θ_b (both poles are necessarily real). Note that $T=T(\rho)\downarrow -1/\theta_b$ as $\rho\downarrow 0$ and that $(1-\rho)^2T(\rho)\to 2(C_s^2+1)$ as $\rho\uparrow 1$, cf. (3.19), (3.18). Further note that in the examples the relaxation time is an increasing function of ρ, C_s and θ_b.

TABLE 3.1. The relaxation time for some $M/G/1$ systems ($\beta=1$); C_s denotes the coefficient of variation of the service time distribution and θ_b the largest pole of $\beta^*(\theta)$.

	$M/E_2/1$	$M/M/1$	$M/H_2/1$	$M/H_2/1$	$M/H_2/1$	$M/H_2/1$
C_s^2	0.5	1	3	3	7	7
θ_b	-2	-1	-0.4	-0.2	-0.2	-0.1
ρ						
$\downarrow$ 0.0	0.500	1.000	2.500	5.000	5.000	10.00
0.1	1.413	2.139	4.962	8.064	9.973	16.59
0.2	2.244	3.273	7.371	10.98	14.83	22.76
0.3	3.428	4.889	10.77	14.99	21.66	31.18
0.4	5.275	7.403	16.01	20.99	32.19	43.74
0.5	8.411	11.66	24.80	30.83	49.86	64.19
0.6	14.35	19.68	41.28	48.84	82.92	101.4
0.7	27.56	37.48	77.58	87.64	155.7	181.0
0.8	66.46	89.72	183.5	198.4	367.8	406.7
0.9	283.1	379.7	767.7	797.3	1537	1617
$\uparrow$ 1.0						
$(1-\rho)^2 T(\rho)$:	3	4	8	8	16	16

From [6, p. 24] it follows that for $\mathrm{Re}\phi>0$,

$$\int_0^\infty e^{-\phi t} E\{N(t)|N(0)=0\}dt = \frac{1}{\alpha\phi^2} - \frac{\xi(\phi)-\phi}{\phi\xi(\phi)} \frac{\beta^*(\phi)}{1-\beta^*(\phi)}. \tag{3.12}$$

It seems possible that zeros of $1-\beta^*(\phi)$ rather than the branch point ϕ_0 of $\xi(\phi)$ play a dominating role in the asymptotic behaviour of $E\{N(t)|N(0)=0\}$ as $t\to\infty$. In the following discussion let ϕ_s be the real part of the zeros of $1-\beta^*(\phi)$, $\phi\neq 0$, with the largest real part. For the $M/E_k/1$ system ($k=2,3,..$) it is readily obtained that

$$\phi_0 = -[k+\rho-(k+1)\rho^{1/(k+1)}]/\beta, \tag{3.13}$$

$$\phi_s = -k[1-\cos(2\pi/k)]/\beta.$$

Let β fixed. The value of ϕ_s does not depend on ρ, while the branch point ϕ_0 is an increasing function of ρ for $0<\rho<1$. For $k=2,3,4$ we have $\phi_0\geqslant\phi_s$ as $\rho\downarrow 0$, so that the relaxation time of $E\{N(t)|N(0)=0\}$ is given by (3.11) for every ρ, $\rho<1$. But for $k>4$ we have $\phi_0<\phi_s$ for ρ small enough. However, it is possible that the zeros of $1-\beta^*(\phi)$ are compensated by zeros of $\xi(\phi)-\phi$ in (3.12), cf. (3.10). This requires further investigation. Incidentally, it can be shown that for ρ large enough ($\rho>1$) the difference between $E\{N(t)|N(0)=0\}$ and its asymptote

$$(a-1)t/\beta + \tfrac{1}{2}[1-C_s^2] + [\beta\xi(0)]^{-1},$$

is $O(\exp(\phi_s t))$ as $t\to\infty$, in contrast with $p_{00}(t)$ which is still $O(\exp(\phi_0 t))$ as $t\to\infty$.

3.3. General results for the GI/G/1 queueing system

In [6, § III.7.3] a theorem on the asymptotic behaviour of $p_{00}^{+}(t)$ (i.e. given $N(0+)=1$, cf. subsection 3.1) as $t\to\infty$ was proved for the $GI/G/1$ system. This result can be adapted to the case that $t=0$ is an arbitrary instant in the arrival process in a way similar to that of subsection 3.1:

$$\Omega(\phi) = \frac{1}{\phi} + \frac{1-\alpha^*(\phi)}{\alpha\phi}\,[\Omega^+(\phi) - \frac{1}{\phi}], \quad \mathrm{Re}\,\phi > 0. \tag{3.14}$$

From [6, p. 601] and (3.14) it follows that for $\theta_b < \mathrm{Re}\,\theta < 0 < \mathrm{Re}\,\phi$,

$$\Omega(\phi) = \frac{1}{\phi} - \frac{1-\beta^*(\phi)}{\alpha\phi^2} \times \tag{3.15}$$

$$\exp\left\{\frac{-1}{2\pi i}\int_{L_\theta}\left[\frac{1}{\phi-\theta}+\frac{1}{\theta}\right]\log[1-\beta^*(\theta)\alpha^*(\phi-\theta)]\,d\theta\right\},$$

here L_θ is a line parallel to the imaginary axis. By using the same arguments as in [6, § III.7.3] relation (3.15) leads to the following result.

For $\rho<1$ there exists a unique real value $\phi=\phi_0$, $\theta_b<\phi_0<0$, for which the equation in θ,

$$\beta^*(\theta)\alpha^*(\phi-\theta)=1, \quad \theta \text{ real}, \quad \theta_b<\theta<\phi, \tag{3.16}$$

has a double root. The relaxation time T of the $GI/G/1$ system is equal to $-1/\phi_0$, and

$$p_{00}(t) - p_0 = O((t/T)^{-3/2}\exp(-t/T)), \quad t\to\infty, \tag{3.17}$$

if $A(x)$ and $B(x)$ are not lattice distributions.

REMARK. If $A(x)$ or $B(x)$ is a lattice distribution, then the Laplace transform $\Omega(\phi)$ has beside the branch point $\phi=\phi_0$ other branch points with $\mathrm{Re}\,\phi=\phi_0$. Therefore, the relaxation time for such systems is still $T=-1/\phi_0$, but the behaviour of $p_{00}(t)-p_0$ as $t\to\infty$ is different from (3.17) because of the contributions of the other poles on the line $\mathrm{Re}\,\phi=\phi_0$.

It should be noted, that if $p_{00}^{+}(t)$ or $p_{00}^{\gamma}(t)$ is considered (as for the $GI/M/1$ system in subsection 3.1) then the Laplace transform possesses poles at zeros ϕ, $\phi\neq 0$, of the function $1-\alpha^*(\phi)$. In Table 3.2 the traffic intensity ρ_a for which the largest real part ϕ_a of these poles coincides with ϕ_0 is shown for several models. For $\rho<\rho_a$ the relaxation time of $p_{00}^{+}(t)$ is equal to $-1/\phi_a$, for $\rho>\rho_a$ to $-1/\phi_0$.

TABLE 3.2. The traffic intensity ρ_0 for which $\phi_0 = \phi_a$

System	ρ_a
$E_2/M/1$	0.125
$E_2/E_2/1$	0.172
$E_2/E_4/1$	0.212
$E_2/D/1$	0.278
$E_4/M/1$	0.134
$E_4/E_2/1$	0.192
$E_4/E_4/1$	0.250
$E_4/D/1$	0.368

In [6, p. 612] the following heavy traffic limit was derived for the relaxation time $T = T(\rho) = -1/\phi_0$. Let the Laplace-Stieltjes transforms $\beta^*(\theta)$ and $\alpha^*(\theta/\alpha)$ be fixed (thus only the mean interarrival time α varies). Then

$$\lim_{\rho \uparrow 1} (1-\rho)^2 T(\rho) = \lim_{\rho \uparrow 1} 4(1-\sqrt{\rho})^2 T(\rho) = 2\beta\{C_s^2 + C_a^2\}, \tag{3.18}$$

here $C_s(C_a)$ is the coefficient of variation of the service (interarrival) time distribution. On the other hand it is not difficult to see that for light traffic the following limit holds

$$\lim_{\rho \downarrow 0} T(\rho) = -1/\theta_b. \tag{3.19}$$

In the following cases explicit expressions were found for the branch point ϕ_0 of $\Omega(\phi)$, cf. [6, p. 613, 614]. For the $E_m/E_k/1$ system ($m,k = 1,2, \ldots$)

$$\phi_0 = -[k + m\rho - (m+k)\rho^{m/(m+k)}]/\beta; \tag{3.20}$$

for the $E_m/D/1$ system ($m = 1,2, \ldots$),

$$\phi_0 = -m[\rho - 1 - \log \rho]/\beta. \tag{3.21}$$

These expressions have been used to obtain the relaxation times listed in Table 3.3.

TABLE 3.3. The relaxation times of some $GI/G/1$ systems as a function of the traffic intensity ρ ($\beta = 1$)

ρ	$M/M/1$	$M/E_4/1$	$E_4/M/1$	$E_4/E_4/1$	$M/D/1$	$E_4/D/1$
$\downarrow$ 0.0	1.000	0.250	1.000	0.2500	0.0000	0.0000
0.1	2.139	1.058	1.646	0.5347	0.7130	0.1782
0.2	3.273	1.736	2.379	0.8181	1.235	0.3089
0.3	4.889	2.703	3.429	1.222	1.984	0.4961
0.4	7.403	4.215	5.057	1.851	3.162	0.7904
0.5	11.66	6.791	7.797	2.914	5.177	1.294
0.6	19.68	11.68	12.94	4.921	9.023	2.256
0.7	37.48	22.60	24.27	9.370	17.64	4.411
0.8	89.72	54.83	57.33	22.43	43.21	10.80
0.9	379.7	234.8	239.8	94.93	186.5	46.64
$\uparrow$ 1.0 $(1-\rho)^2 T(\rho)$:	4.0	2.5	2.5	1.0	2.0	0.5

4. QUEUEING NETWORKS

In the last decade there has been a growing interest in the analysis of queueing networks, particularly because of their frequent occurrence in the modeling of computer systems, cf. [21]. Until recently little was known about the time-dependent behaviour of queueing networks. The concept of relaxation time has been discussed on the basis of diffusion approximations, but this will lead at most to heavy traffic limits for the relaxation times. Lately, however, some exact results were obtained for the relaxation times of networks of the type described by JACKSON [12]. In subsection 4.1 the relaxation time of an open network with infinitely many servers at each node will be discussed. For such a network the relaxation time is determined by the eigenvalue with the smallest real part of a matrix which depends on the transition matrix and the mean service times at the nodes. In subsection 4.2 the relaxation time of an open network with two single server nodes will be presented. This relaxation time is equal to the maximum of two values which can be assigned each to a different node, and which are equal to the relaxation times of the two nodes when considered separately as $M/M/1$ queueing systems. This result will be used in subsection 4.3 to formulate a conjecture on the relaxation time of an open Jackson network with an arbitrary number of single server nodes.

In this section the following notation will be used. The network consists of N nodes ($N = 2,3, ..$). The external arrivals form a Poisson process with constant mean interarrival time α. Each node has an infinite capacity (for service or waiting). Each time a customer visits a node he awaits his turn (if necessary) and receives service during an exponentially distributed amount of time. At each node the mean serivce time is fixed. We define for $j,k = 1, ..., N$,

c_j : the probability that a customer enters the network at node j;
β_j : the mean service time at node j;
q_{kj} : the probability that a customer goes to node j when his service at node k has been completed;
λ_j : the arrival rate (external and internal) at node j;
ρ_j : the traffic intensity at node j.

The network is assumed to be stable. Then the arrival rates λ_j satisfy the set of equations, cf. [12],

$$\lambda_j = c_j/\alpha + \sum_{k=1}^{n} q_{kj}\lambda_k, \quad j=1, .., N. \tag{4.1}$$

The network with infinite server nodes is stable for all values of the parameters. The network with single server nodes is stable iff

$$\rho_j = \lambda_j\beta_j < 1, \quad j=1, .. ,N. \tag{4.2}$$

The transition matrix $Q=(q_{kj})$ is assumed to be such that every customer leaves the system after having received a finite number of services, i.e., the network is open. Finally, the trivial case $\lambda_j=0$ for some j, $j=1,...,N$, is excluded.

4.1. Jackson networks with infinite server nodes

In [13] a method was developed for the solution of the functional equation for the generating function of the joint queue length distribution for a queueing system with a Poisson arrival stream and N infinite exponential server stations in series. With the aid of this result it is not difficult to derive that the relaxation time of this system is equal to

$$T = \max_{1\leqslant j\leqslant N}\{\beta_j\}, \tag{4.3}$$

i.e., to the maximum of the relaxation times of these service systems at each of the nodes when considered separately as $M/M/\infty$ systems, cf. § 2.2.1. Further, it can be seen that

$$p_{00}(t)-p_0 = O(P_{n-1}(t/T)e^{-t/T}), \quad t\to\infty; \tag{4.4}$$

here $P_n(\cdot)$ is a polynomial of degree n, and n, $1\leqslant n\leqslant N$, is equal to the number of nodes j for which $\beta_j=T$.This system provides us with an example in which the coefficient of $\exp(-t/T)$ in the expansion of $p_{00}(t)$ as $t\to\infty$ is not bounded. However, calculations have shown that for $m=1,2$ the function $P_m(t/T)\exp(-t/T)$ is decreasing for $t>0$.

The relaxation time of a network with a finite number of infinite exponential server stations — with general transition matrix $Q=(q_{kj})$ and arbitrary probabilities c_j, $j=1, ...,N$ — can be obtained by noting that such a network is equivalent to a (one node) $M/G/\infty$ system when only the total number of customers in the system is considered. In this $M/G/\infty$ system the service time

distribution $B(x)$ is equal to the distribution of the total sojourn time of a customer in the original network. TAKÁCS [30, p. 160] showed that for a $M/G/\infty$ system

$$p_{00}(t) = \exp\{-\frac{1}{\alpha}\int_0^t [1-B(x)]dx\}. \tag{4.5}$$

By determining the distribution $B(x)$ it follows that the relaxation time T of a network with N infinite exponential server stations (and that of the related $M/G/\infty$ system) is equal to $-1/\mathrm{Re}\gamma_{\min}$, where $\gamma_{\min}$ is the eigenvalue with the smallest real part of the matrix

$$\begin{pmatrix} (1-q_{11})/\beta_1 & -q_{12}/\beta_1 & \cdots & -q_{1N}/\beta_1 \\ -q_{21}/\beta_2 & (1-q_{22})/\beta_2 & \cdots & -q_{2N}/\beta_2 \\ \vdots & \vdots & & \vdots \\ -q_{N1}/\beta_N & -q_{N2}/\beta_N & \cdots & (1-q_{NN})/\beta_N \end{pmatrix}. \tag{4.6}$$

Note that T is independent of the mean interarrival time α and of the probabilities c_j, $j=1, \ldots, N$. In the special case $N=2$, $q_{11}=q_{22}=0$, it is found that

$$T = 2\beta_1\beta_2[\beta_1+\beta_2-\sqrt{(\beta_1+\beta_2)^2-4\beta_1\beta_2(1-q_{12}q_{21})}]^{-1}. \tag{4.7}$$

4.2. Jackson networks with two single server nodes

With the aid of a recently developed method for the solution of functional equations for the generating functions of the joint time-dependent queue length distributions for queueing systems with two waiting lines, cf. [7], the following results have been obtained. In [1] it was shown that for the special network with $N=2$, $c_1=1$, $c_2=0$, $q_{12}=0$, $q_{11}=q_{22}=0$, i.e. two single server stations in series, the relaxation time is equal to

$$T = \max_{j=1,2}\{\beta_j/(1-\sqrt{\rho_j})^2\}, \tag{4.8}$$

i.e., T is equal to the maximum of the relaxation times of the service systems at the two nodes when considered separately as $M/M/1$ queueing systems with arrival rate $1/\alpha$, cf. § 2.2.3. Moreover, it was found that in the asymmetric case $\rho_1 \neq \rho_2$,

$$p_{00}(t)-p_0 = O((t/T)^{-3/2}e^{-t/T}), \quad \text{as} \quad t\to\infty, \tag{4.9}$$

cf. (2.39), while in the symmetric case $\rho_1=\rho_2$,

$$p_{00}(t)-p_0 = O((t/T)^{-1/2}e^{-t/T}), \quad \text{as} \quad t\to\infty. \tag{4.10}$$

Comparing this result with that of subsection 4.1 leads one to conjecture that the relaxation time of a system with an arbitrary number of single server nodes in series is equal to the maximum of the relaxation times of the nodes when considered separately, and that

$$p_{00}(t)-p_0 = O(P_{n-1}(t/T)(t/T)^{-3/2}e^{-t/T}), \quad t\to\infty,$$

where n is the number of nodes with relaxation time equal to T. Recently, the relaxation time of the general Jackson network with two single server nodes was obtained by the first author. It was found to be equal to

$$T = \max_{j=1,2} \{\beta_j m_j/(1-\sqrt{\rho_j})^2\}, \tag{4.11}$$

here m_j, $j=1,2$, is the mean number of visits of a fixed customer to node j given that this customer visits node j at least once:

$$m_j = (1-q_{jj})/[(1-q_{11})(1-q_{22})-q_{12}q_{21}]. \tag{4.12}$$

The following gives an interpretation of (4.11). The relaxation time of a single $M/M/1$ system with feedback is equal to

$$T = \beta(1-p)^{-1}/(1-\sqrt{\rho})^2, \tag{4.13}$$

where β is the mean service time, ρ the traffic intensity and p the probability that a customer returns to the queue after completion of one of his services. The result (4.13) is clear if one notes that the Laplace-Stieltjes transform of the distribution of the total service time which a customer receives is given by

$$\sum_{m=1}^{\infty} (1-p)p^{m-1}(1+\beta\theta)^{-m} = [1+\beta\theta/(1-p)]^{-1}. \tag{4.14}$$

In (4.13) the factor $(1-p)^{-1}$ can be interpreted as the mean number of services received by a fixed customer. Hence, from (4.11) it is seen that the relaxation time of a Jackson network with two single server nodes is equal to the maximum of the relaxation times of the service systems of the two nodes when considered separately as $M/M/1$ queues with feedback. Note the different effects which the transition matrix Q has on the relaxation time of networks with infinite server nodes, cf. (4.7), and on that of networks with single server nodes, cf. (4.11), (4.12).

We finally note that a result similar to (4.3) and (4.8) was found for quite a different model. In [2] the time-dependent behaviour of an $M/G/1$ system with two types of customers and a paired service discipline (i.e. a pair of customers of different type is served if possible, if the customer population consists of one type only, a single customer is served) was studied. Let α be the mean interarrival time, c_j, $j=1,2$, the probability that a customer is of type j, and $B(x)$ the service time distribution. Then the relaxation time of this system is equal to the maximum of the relaxation times of two ordinary $M/G/1$ systems with mean interarrival time α/c_j, $j=1,2$, and service time distribution $B(x)$, cf. subsection 3.2.

4.3. A conjecture on the relaxation time of an open Jackson network

Unfortunately, the method with which the results in subsection 4.2 were obtained is at present not applicable in the analysis of queueing systems with more than two waiting lines. However, the form of the relaxation time (4.11) for a Jackson network with two nodes lends itself to a conjecture on the relaxation time of a Jackson network with an arbitrary number of single server

nodes. In fact we conjecture that the relaxation time of such a network is equal to

$$T = \max_{1 \leq j \leq N} \{\beta_j m_j / (1 - \sqrt{\rho_j})^2\}; \tag{4.15}$$

here m_j, $j = 1, \ldots, N$, has the same meaning as in subsection 4.2. By interpreting the transition process as a Markov chain, m_j is readily found to be

$$m_j = |I - Q|_{jj} / |I - Q|, \quad j = 1, \ldots, N; \tag{4.16}$$

here I is the $N \times N$ identity matrix, $Q = (q_{kj})_{1 \leq k,j \leq N}$, $|I - Q|$ is the determinant of $I - Q$, and $|I - Q|_{jj}$ is the determinant of the submatrix of $I - Q$ obtained by deleting the j^{th} row and column. Note that $|I - Q| \neq 0$, because the network is open.

REMARK. Probably the relaxation time of a Jackson network with an arbitrary number of servers at each node will also have a form like (4.15) - with a proper modification, cf. (2.31) - provided that the traffic intensities ρ_j are not too small, cf. subsection 2.2.3. On the other hand it is not easy to generalize (4.15) to a network with more than one class of customers, because the relaxation time of a single $M/M/1$ system with K customer classes is equal to that of a $M/H_K/1$ system, and the latter has a form different from (2.31), cf. subsection 3.2.

Finally, the main implication of the hypothesis (4.15) will be discussed. Consider for example the following network with three nodes ($N = 3$). Let $\beta_1 = \beta_2 = \beta_3 = \beta$, $c_1 = 1$, $c_2 = c_3 = 0$, and

$$Q = \begin{bmatrix} 0 & \epsilon & \epsilon \\ 0 & 0 & 1-\epsilon \\ 0 & 1-\epsilon & 0 \end{bmatrix}, \quad 0 < \epsilon \leq \tfrac{1}{2},$$

see figure 4.1. It is readily verified, that for this network the traffic intensities at the three nodes are the same, and equal to $\rho = \beta/\alpha$. However, $m_1 = 1$ and $m_2 = m_3 = [\epsilon(2-\epsilon)]^{-1} \geq 4/3$, cf. (4.16), so that the hypothesis (4.15) implies

$$T = \frac{1}{\epsilon(2-\epsilon)} \frac{\beta}{(1-\sqrt{\rho})^2}. \tag{4.17}$$

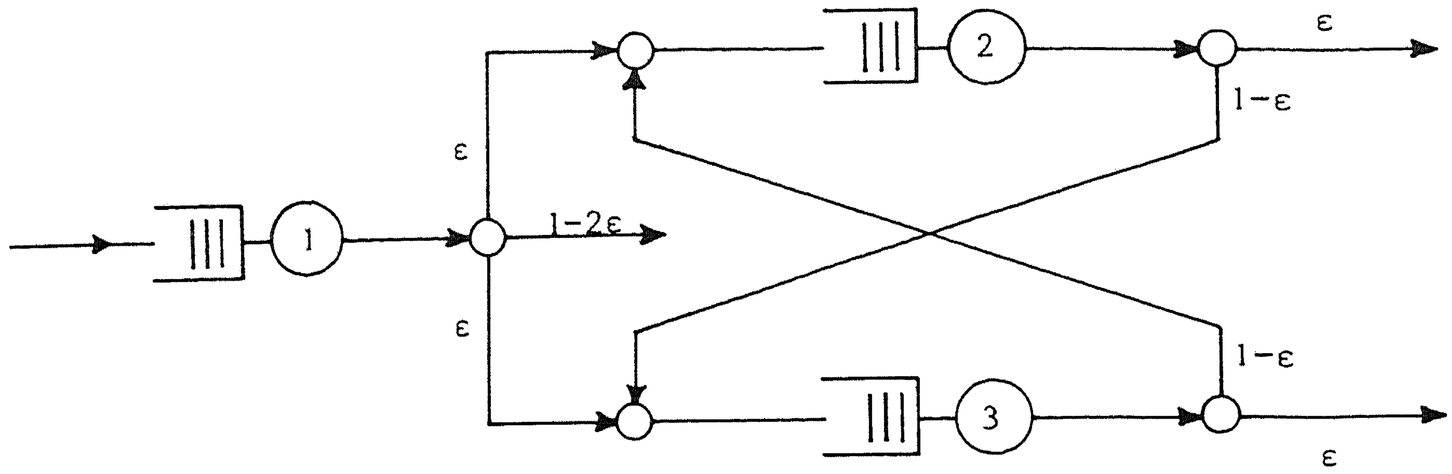


FIGURE 4.1. A three-node network

Note that the right hand side of (4.17) tends to infinity as $\epsilon \downarrow 0$ for fixed traffic intensity ρ. In general, the hypothesis (4.15) implies that for nodes with the same traffic intensity the relaxation time of a node (when considered separately) is larger when a few customers are served many times than when many customers are served a few times. In the above example the mean number of nodes visited by a fixed customer is equal to 3, but given that the customer reaches node 2 or 3, it is equal to $2+(1-\epsilon)/\epsilon$. Hence, for ρ fixed and ϵ small enough there are customers with an arbitrarily long expected sojourn time in the system. It is likely that this fact influences the relaxation time of the network. Therefore it supports the conjecture (4.15).

REMARK. If the same network as above is considered, but with infinite server nodes, then the eigenvalues of the matrix (4.6) are $1/\beta$, ϵ/β and $(2-\epsilon)/\beta$. Hence, the relaxation time of this network is

$$T = \beta/\epsilon.$$

This relaxation time T also tends to infinity as $\epsilon \downarrow 0$.

REFERENCES

1. J.P.C. BLANC (1985). The relaxation time of two queueing systems in series. *Commun. Statist.-Stochastic Models 1,* 1-16.
2. J.P.C. BLANC (1984). Asymptotic analysis of a queueing system with a two-dimensional state space. *J. Appl. Probab. 21,* 870-886.
3. H. CALLAERT (1971). *Exponentiële Ergodiciteit voor Geboorte- en Sterfteprocessen,* Ph. D. thesis, University of Louvain.
4. H. CALLAERT (1974). On the rate of convergence in birth-and-death processes. *Bull. Soc. Math. Belg. 26,* 173-184.

5. T.S. CHIHARA, M.E.H. ISMAIL (1982). Orthogonal polynomials suggested by a queueing model. *Adv. in Appl. Math. 3,* 441-462.
6. J.W. COHEN (1982). *The Single Server Queue, Rev. ed.,* North-Holland, Amsterdam.
7. J.W. COHEN, O.J. BOXMA (1983). *Boundary Value Problems in Queueing System Analysis,* North-Holland, Amsterdam.
8. E.A. VAN DOORN (1981). The transient state probabilities for a queueing model where potential customers are discouraged by queue length. *J. Appl. Probab. 18,* 499-506.
9 E.A. VAN DOORN (1981). On the time dependent behaviour of the truncated birth-death process. *Stochastic Process. Appl. 11,* 261-271.
10. E.A. VAN DOORN (1981). *Stochastic Monotonicity and Queueing Applications of Birth-Death Processes,* Lecture Notes in Statistics 4, Springer-Verlag, New York.
11. E.A. VAN DOORN (1982). *Conditions for Exponential Ergodicity and Bounds for the Decay Parameter of a Birth-death Process.* Report BW 174/82, Centre for Mathematics and Computer Science, Amsterdam.
12. J.R. JACKSON (1957). Networks of waiting lines. *Oper. Res. 5,* 518-521.
13. R.R.P. JACKSON, P. ASPDEN (1980). A transient solution to the multistage Poisson queueing system with infinite servers. *Oper. Res. 28,* 618-622.
14. S. KARLIN, J.L. MCGREGOR (1957). The differential equations of birth-and-death processes and the Stieltjes moment problem. *Trans. Amer. Math. Soc. 85,* 489-546.
15. S. KARLIN, J.L. MCGREGOR (1957). The classification of birth- and-death processes. *Trans. Amer. Math. Soc. 86,* 366-400.
16. S. KARLIN, J.L. MCGREGOR (1958). Many server queueing processes with Poisson input and exponential service times. *Pacific J. Math. 8,* 87-118.
17. S. KARLIN, J.L. MCGREGOR (1965). Ehrenfest urn models. *J. Appl. Probab. 2,* 352-376.
18. J.F.C. KINGMAN (1962). On queues in which customers are served in random order. *Proc. Cambridge Philos. Soc. 58,* 79-91.
19. C. KINGMAN (1963). The exponential decay of Markov transition probabilities. *Proc. London Math. Soc. 13,* 337-358.
20. J.F.C. KINGMAN (1963). Ergodic properties of continuous-time Markov processes and their discrete skeletons. *Proc. London Math. Soc. 13,* 593-604.
21. L. KLEINROCK (1976). *Queueing Systems II,* John Wiley, New York.
22. F. MACHIHARA (1983). On the property of eigenvalues of some infinitesimal generator. *Oper. Res. Lett. 2,* 123-126.
23. P.M. MORSE (1955). Stochastic properties of waiting lines. *Oper. Res. 3,* 255-261.
24. P.M. MORSE (1958). *Queues, Inventories and Maintenance,* John Wiley, New York.
25. B. NATVIG (1974). On the transient state probabilities for a queueing model where potential customers are discouraged by queue length. *J. Appl. Probab. 11,* 345-354.

26. A.R. Odoni, E. Roth (1983). An empirical investigation of the transient behaviour of stationary queueing systems. *Oper. Res. 31,* 432-455.
27. G.E.H. Reuter (1957). Denumerable Markov processes and the associated contraction semigroups on *l*. *Acta Math. 97,* 1-46.
28. J.P. Schouten (1961). *Operatorenrechnung,* Springer-Verlag, Berlin.
29. J.H.A. de Smit (1972). The time dependent behaviour of the queue length process in the system *M/M/s*. *CORE discussion paper no. 7217,* University of Louvain.
30. L. Takács (1962). *Introduction to the Theory of Queues,* Oxford University Press, New York.
31. D.V. Widder (1946). *The Laplace Transform,* Princeton University Press, Princeton.

Some Current Developments in Density Estimation

Piet Groeneboom
University of Amsterdam
P.O. Box 19268, 1000 GG Amsterdam, The Netherlands

Some recent results on the minimax risk of nonparametric density estimators, and, in particular, on the relation between minimax risk and metric entropy, are reviewed. We also discuss results on the distribution theory for the maximum likelihood estimator of a decreasing density and the connection of these results with properties of Brownian motion.

1. Introduction

Every mathematician is probably familiar with the situation where he (she) is asked to describe to nonmathematicians the research he (she) is doing and to explain why this is an interesting and worthwhile endeavor. A very realistic description of what happens in such a case is given in the book by Davis and Hersh [12], where on p. 37-39 a 'researcher on the decision problem for non-Riemannian hypersquares' is interviewed by a public information officer on the occasion of a renewal of his research grant.

A statistician who has to explain to a general mathematical public the kind of problems he (she) is interested in finds himself (herself) in a similar situation. In the following notes I will try to explain some current developments in the theory of (probability) density estimation in such a way that every mathematician should be able to understand it, and I apologize to statistical readers for the triviality of some of the remarks I will make. Furthermore, I have chosen for the approach of treating some typical examples in depth (with proofs), rather than covering a large area without really entering into the mathematics of the subject.

In statistical consultation, one is often confronted with the following problem. Someone (the client) shows graphs of a certain observed frequency distribution and asks 'what theoretical probability distribution would fit this observed distribution?'.

Figure 1.1 shows an example of such a graph. The graph is based on a sample of 1000 observations, generated by the STATAL random number generator from a decreasing density on [0,1] (to be specified later). The number of observations in each interval (of length 0.05), [0,.05), [.05,.1), etc. has been determined, and the graph connects linearly the values of these fractions (which are

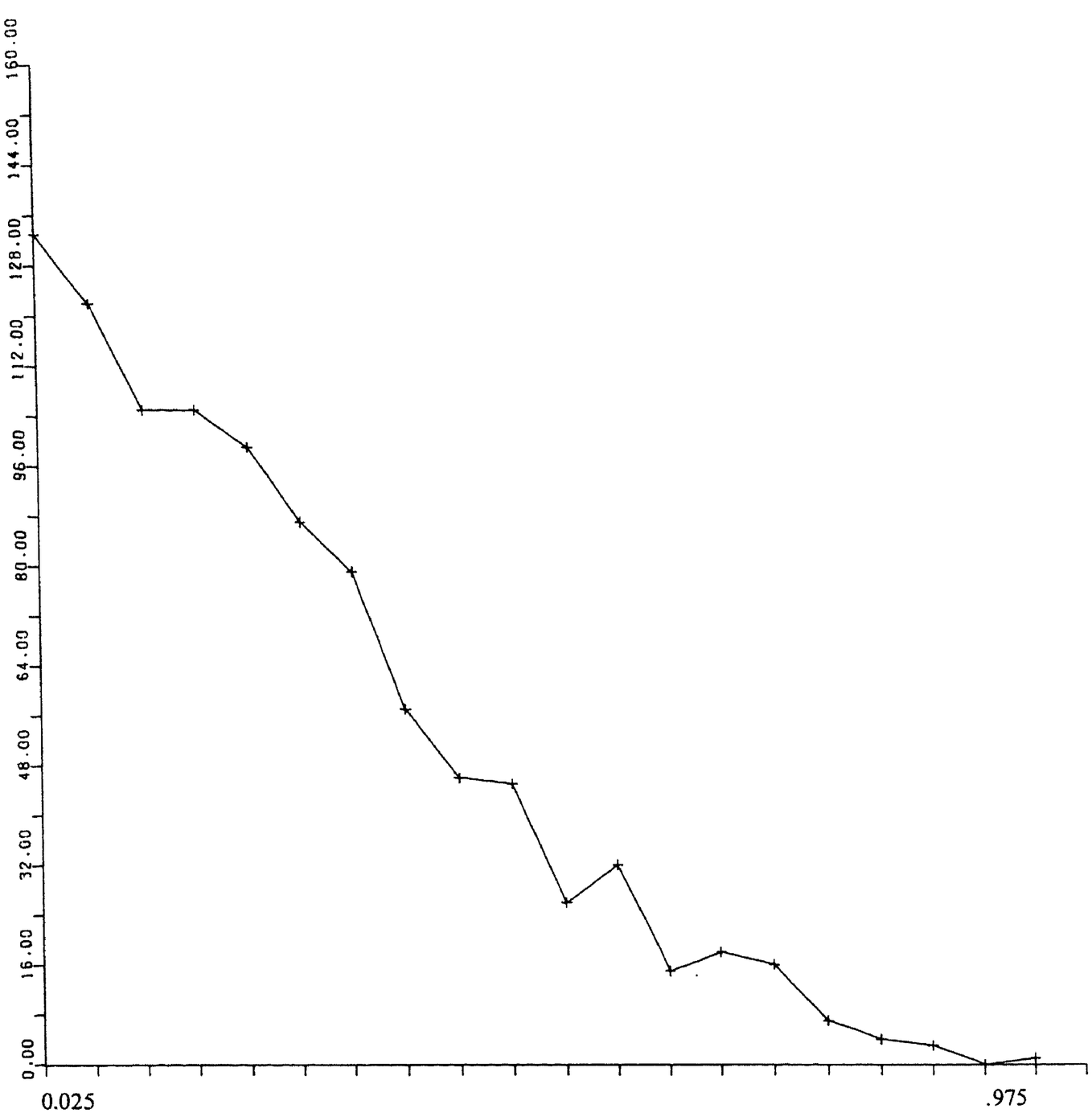


FIGURE 1.1. Frequency polygon, based on 1000 observations generated by a decreasing density on [0,1]

assigned to the midpoints of the intervals). This type of graph is called a *frequency polygon* and is familiar to everyone from the cartoons about worried businessmen looking at decreasing frequency polygons of sales figures.

Another method of summarizing these data is given by the *histogram* of figure 1.2. In this case one represents the fractions (or numbers) of observations in the intervals [0,.05) etc. by blocks where the height of the block indicates the fraction of observations in the interval.

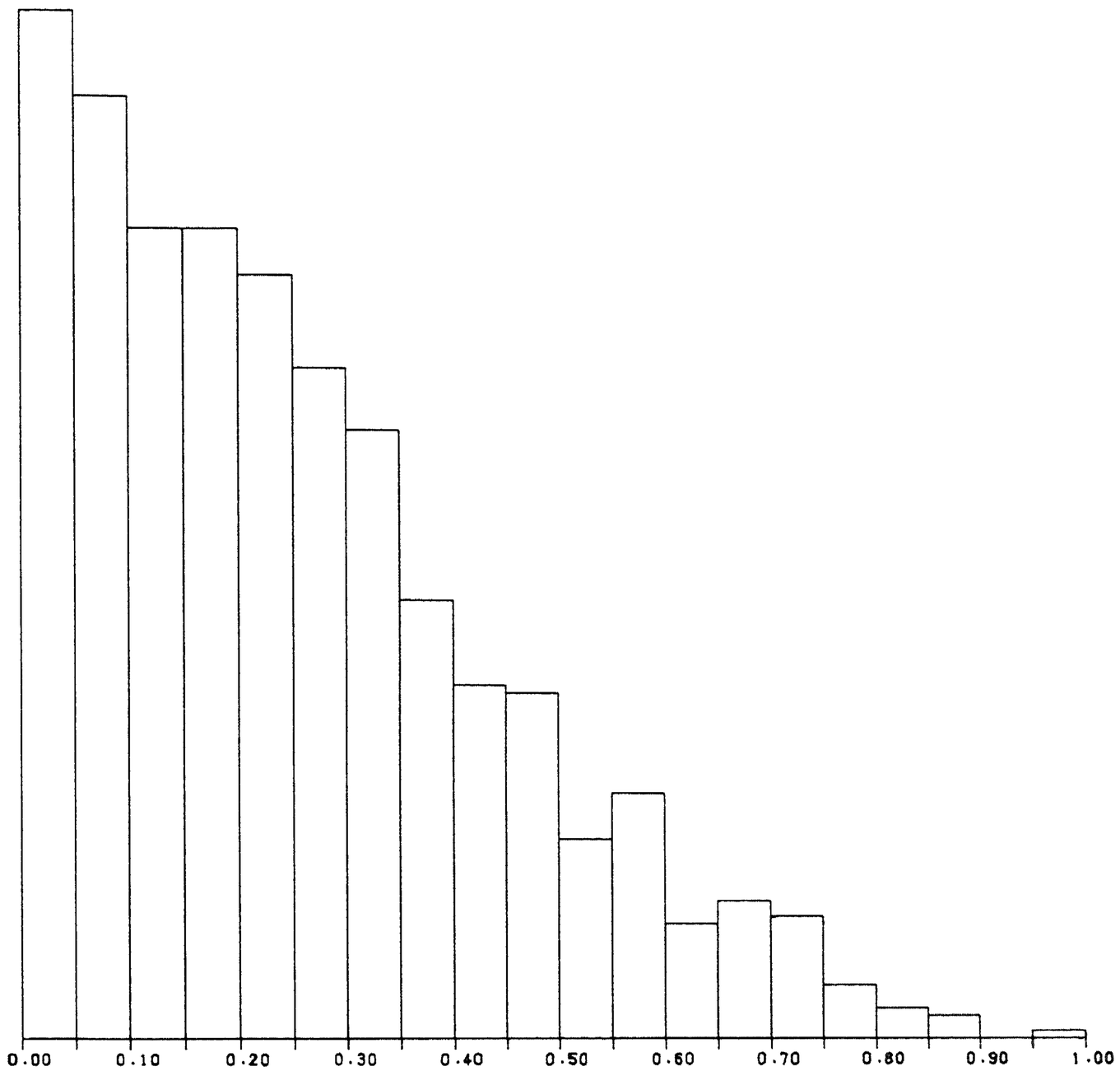


FIGURE 1.2. Histogram, based on 1000 observations generated by a decreasing density on [0,1]

A third method of representing the observed distribution is given by *kernel estimators.* A kernel estimator of an unknown density f on [0,1], based on a random sample $X_1,\ldots,X_n$ generated by the density f, is a function $f_{h,n}:[0,1]\to\mathbb{R}$ defined by

$$f_{h,n}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K((x-X_i)/h), \tag{1.1}$$

where the *kernel* K is a probability density (usually a Gaussian or 'normal' density) and h is the *window size,* representing the degree of smoothing. Figure 1.3 shows a graph of a kernel estimate f_n, based on the same sample of $n = 1000$ observations that was used in figures 1.1 and 1.2. The density $f(t) = 3(1-t)^2$, $t\in[0,1]$, from which the observations were generated, is also

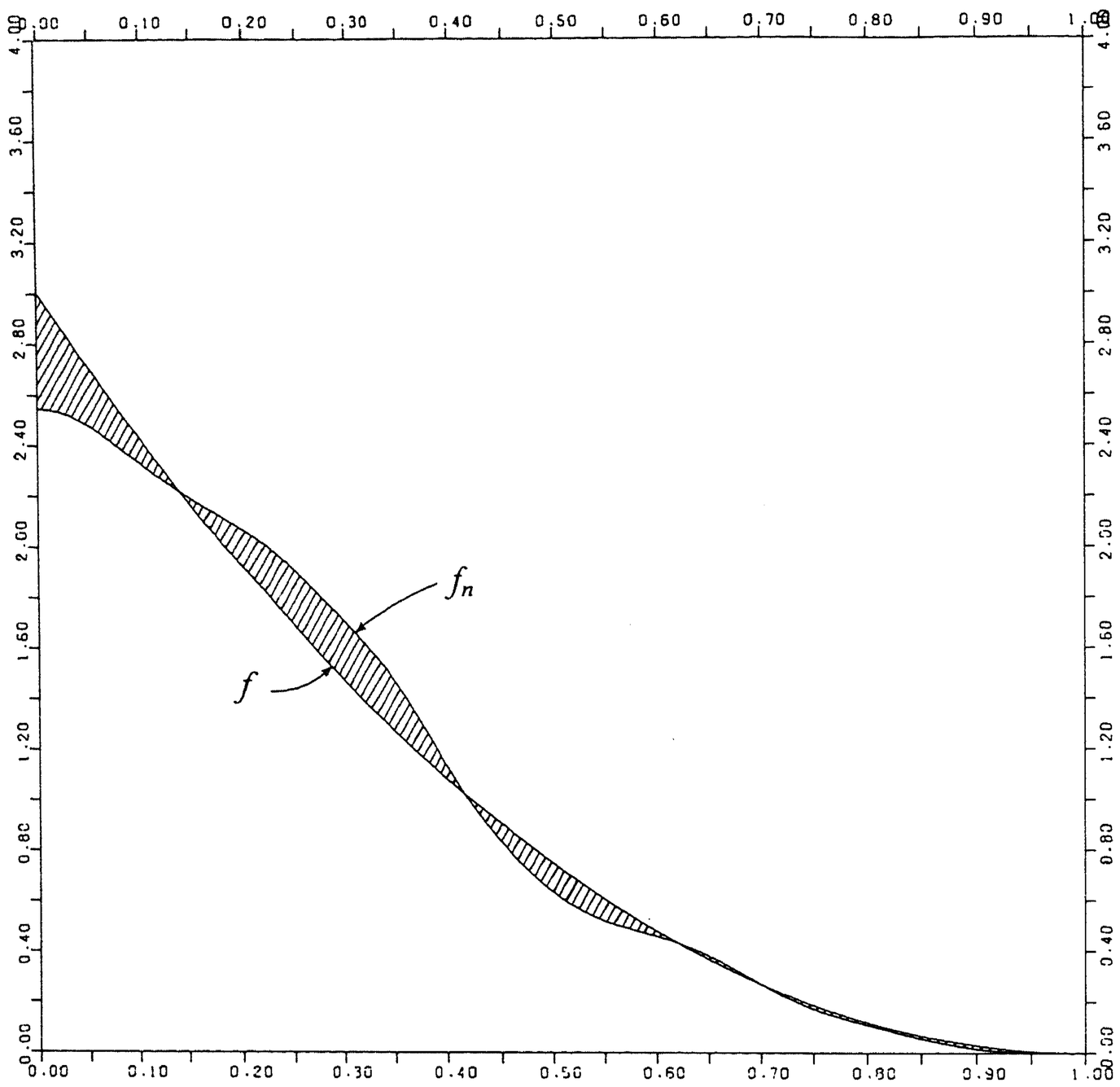


FIGURE 1.3. Kernel estimate, $n=1000$, $f(t)=3(1-t)^2$, $t\in[0,1]$, $K(x)=(2\pi)^{-\frac{1}{2}}\exp(-\frac{1}{2}x^2)$

shown in the graph. The area of the shaded region equals the L_1-distance between the kernel estimator and the density f. Returning to the general question 'What theoretical probability distribution would fit this empirical distribution (provided by the client)?', we can remark first of all that this question is meaningless if one does not specify beforehand:

(i) the family $\mathcal{F}$ of densities one wishes to consider;

(ii) a *loss function* (usually a distance, such as the L_1-distance on $\mathcal{F}$), measuring the deviation between the real density and the estimator of the density.

In the same way, questions like 'how big should the window size of my kernel estimator be?' or 'how should I choose the intervals of my histogram?' are meaningless if (i) and (ii) above have not been specified.

In the old days, one only considered certain standard families of curves in the fitting problem, for example the Gaussian densities

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma^2}(x-\mu)^2\}, \quad x \in \mathbb{R} \tag{1.2}$$

which are completely specified by the two parameters μ and σ. But during the last two decades there has been an explosive development of techniques that are meant for the more general situation where one does not restrict the family of possible densities to a family parametrized by a subset of $\mathbb{R}^m$ $(m<\infty)$, but instead considers infinite dimensional families. These techniques fall under the heading of *nonparametric density estimation* and have been greatly stimulated by developments in computer graphics.

In Sections 2 and 3 we will give a discussion of the best performance one can expect from nonparametric density estimators according to the criterion of minimax risk. We will restrict ourselves to the choice of the L_1-distance as our loss function for densities on $\mathbb{R}^d (d<\infty)$. See for example figure 1.3, where $d=1$. This is a very natural loss function, since it corresponds to the total variation distance

$$D(P,Q) = \sup_{B\in\mathcal{B}} |P(B)-Q(B)| \tag{1.3}$$

between probability measures P and Q on $\mathbb{R}^d$, where $\mathcal{B}$ is the collection of Borel sets of $\mathbb{R}^d$. If P and Q are absolutely continuous with respect to Lebesque measure, with densities p and q respectively, we have

$$\int_{\mathbb{R}^d} |p-q| = 2D(P,Q), \tag{1.4}$$

and unlike the L_2-distance for example, the L_1-distance is always well-defined and *invariant under monotone transformations of the coordinate axes* (a lucid account of the L_1-theory is given in the book by L. DEVROYE and L. GYÖRfi *Nonparametric Density Estimation: the L_1-view*).

The minimax risk is defined as follows. Let $X_1,\ldots,X_n$ be a random sample of n d-dimensional vectors, generated by a density f belonging to a class of densities $\mathcal{F}$ on $\mathbb{R}^d$. The *risk* under f of an estimator $\hat{f}_n = \hat{f}_n(.|X_1,\ldots,X_n)$ of f, based on a sample $X_1,\ldots,X_n$ from f, is the *expected value* of the L_1-distance between $\hat{f}_n$ and f:

$$E_f d_1(\hat{f}_n,f) = \int_{\mathbb{R}^{nd}} \ldots \int d_1(\hat{f}_n(.|x_1,\ldots,x_n),f) f(x_1)\ldots f(x_n)dx_1\ldots dx_n \tag{1.5}$$

where d_1 denotes the L_1-distance and $x_i \in \mathbb{R}^d$, $1 \leqslant i \leqslant n$. The *minimax risk* for the class $\mathcal{F}$, corresponding to samples of size n and loss function d_1, is now defined by

$$R_M(d_1,n) = \inf_{\hat{f}_n} \sup_{f\in\mathcal{F}} E_f d_1(\hat{f}_n,f) \tag{1.6}$$

where the infinum is taken over all possible density estimators $\hat{f}_n$ based on a random sample of n observations generated by a density in the family $\mathcal{F}$.

Thus, a minimax estimator (if it exists), would minimize the maximum risk over all density estimators.

If one wants to estimate a parameter θ belonging to a finite-dimensional parameter set $\Theta \subset \mathbb{R}^m$ by an estimator $\hat{\theta}_n$ based on a sample of n observations, one usually has convergence of $\sqrt{n}(\hat{\theta}_n - \theta)$ to a limiting Gaussian distribution under the probability distribution specified by θ, as the sample size n tends to infinity. This means that the Euclidean distance between $\hat{\theta}_n$ and θ is of the order of $n^{-\frac{1}{2}}$ (the so-called '$\sqrt{n}$ law'). In nonparametric density estimation, the situation is radically different. The L_1-distance between a density f and its estimator $\hat{f}_n$, based on a sample of n observations generated by f, is typically of an order $n^{-\alpha}$, with $\alpha < \frac{1}{2}$.

In Section 2 we will discuss the relation between the metric entropy (for the definition, see Section 2) of the set of densities one wishes to consider and the rate of convergence to zero of the minimax risk. To our knowledge, this relation has for the first time been clearly pointed out by L. BIRGÉ in his dissertation [4] (see also BIRGÉ [5]). Once this relation has been established, one can use results from approximation theory to give bounds and rates of convergence for the minimax risk. Roughly speaking, the bigger the metric entropy, the bigger the minimax risk (this relation can be exactly specified in certain cases, see Theorem 2.1). This is not surprising, since the metric entropy measures the 'massivity' of a set, and the identifiability of a density within a set of densities will depend on the massivity of this set. Generally, (uniform) smoothness assumptions for the densities are reflected by the metric entropy of the set of densities: the smoother the densities are, the smaller the metric entropy will be (but we will give an example of a situation where things can go badly wrong, even for a class of very smooth densities). Completely different types of restrictions can be put on the class of densities; for example, we may consider a class of decreasing densities on the interval [0,1], without any smoothness restrictions. The metric entropy of this set will again give us the rate of convergence to zero of the minimax risk. Hence, using the entropy concept, we can treat smoothness restrictions and order restrictions in a similar way.

In Section 3 we give a fundamental lemma (Assouad's lemma), providing a lower bound for the minimax risk. We will also briefly discuss the concept of local asymptotic minimax risk, and give a local minimax result for the estimation of a monotone density (Theorem 3.1). Apart from this, the treatment of the minimax risk in Sections 2 and 3 mainly uses the elegant techniques of BIRGÉ [5], [6], [7], with some simplifications which were made possible by the fact that we look at more special situations and do not try to obtain the best constants.

In Section 4 we will discuss the behavior of a particular density estimator. We will also take a quick look at some distribution theory and the connection with Brownian motion.

2. Metric entropy and upper bounds for the minimax risk

We first recall some definitions (see Kolmogorov and Tikhomirov [29]). Suppose S is a subset of a metric space with metric d and let $\epsilon>0$. An $\epsilon-$*net* or $\epsilon-$*covering* of S is a subset $N\subset S$ such that

$$\forall s\in S \quad \exists n\in N: \quad d(n,s)\leqslant\epsilon \tag{2.1}$$

(often the ϵ in (2.1) is replaced by 2ϵ). A subset $A\subset S$ is called $\epsilon-$*separated* (or $\epsilon-$*distinguishable*) if

$$x,y\in A, \quad x\neq y \;\Rightarrow\; d(x,y)\geqslant\epsilon \tag{2.2}$$

Suppose S is totally bounded. Then, for each $\epsilon>0$, the (metric) ϵ-entropy $H_\epsilon(S)$ of S is the logarithm of the *minimum* number of elements of an $\epsilon-$*net* of S. The $\epsilon-$*capacity* $C_\epsilon(S)$ of S is the logarithm of the *maximum* number of elements of an $\epsilon-$*separated* subset of S. The ϵ-entropy and ϵ-capacity satisfy the set of inequalities

$$C_{2\epsilon}(S)\leqslant H_{2\epsilon}(S)\leqslant C_\epsilon(S) \tag{2.3}$$

Suppose $\mathcal{F}$ is a set of probability densities on a compact set $S\subset\mathbb{R}^d$, metrized by the L_1-distance. Here and in the following, 'density' will always mean 'probability density with respect to Lebesgue measure'. The following theorem specifies a relation between the behavior of the ϵ-entropy, as $\epsilon\downarrow 0$, and the rate of convergence to zero of the minimax risk as the sample size increases.

Theorem 2.1. *Let $\mathcal{F}$ be a set of probability densities on a compact set $S\subset\mathbb{R}^d$, metrized by the L_1-distance d_1. Suppose that there exist numbers $\delta>0$ and $C_1>0$ such that for all $\epsilon>0$ the ϵ-entropy satisfies*

$$H_\epsilon(\mathcal{F})\leqslant C_1\epsilon^{-\delta}. \tag{2.4}$$

Then there exist a constant $C_2>0$ such that

$$R_M(d_1,n)\leqslant C_2 n^{-1/(2+\delta)}, \quad n\in\mathbb{N}, \tag{2.5}$$

where $R_M(d_1,n)$ is the minimax risk for the class $\mathcal{F}$, corresponding to samples of size n and loss function d_1, defined by (1.6).

Remark 2.1. The following result is (a part of) Theorem 1, Section 4 of Devroye and Györfi [14].

Let G be the set of densities on $[0,1]$, bounded by $2+\delta$ (some $\delta>0$), and infinitely many times continuously differentiable on $(0,1)$. Then we have for *any* sequence of density estimators $(\hat{f}_n)_n\in\mathbb{N}$, where $\hat{f}_n$ is based on a sample of size n,

(i) $$\inf_n \sup_{f\in G} E_f\int |\hat{f}_n-f| \geqslant 1$$

(ii) for any sequence $(a_n)_n\in\mathbb{N}$ of positive numbers a_n tending to 0,

$$\sup_{f\in G} \limsup_{n\to\infty} a_n^{-1} E_f\int |\hat{f}_n-f| = \infty.$$

This result shows that conditions like compact support and smoothness are not sufficient to ensure a reasonable identifiability of the unknown density, but that we need a condition like (2.4) on the metric entropy of the class of densities, to have the risk of our estimators tend to zero uniformly for all densities in the class. No matter how sophisticated or 'adaptive' our estimators f_n are, there will always be some densities in the class G which will be estimated rather poorly by this estimator.

Before giving the proof of Theorem 2.1, we give two examples of its use.

EXAMPLE 2.1 (Smooth densities). Suppose that $\mathcal{F}$ is the class of densities on [0,1] such that, for some $\alpha \in (0,1]$,

$$|f^{(p)}(x) - f^{(p)}(y)| \leqslant C|x-y|^{\alpha}, \quad x,y \in (0,1), \tag{2.6}$$

where $p \in \mathbb{N} \cup \{0\}$ and $C>0$ is a constant independent of f (i.e. condition (2.6) holds *uniformly* for $f \in \mathcal{F}$). Then there exists a constant $C_1>0$ such that the ϵ-entropy $H_\epsilon(\mathcal{F})$ satisfies

$$H_\epsilon(\mathcal{F}) \leqslant C_1 \epsilon^{-1/(p+\alpha)}, \tag{2.7}$$

and hence, by Theorem 2.1,

$$R_m(d_1,n) \leqslant C_2 n^{-(p+\alpha)/(1+2p+2\alpha)}, \tag{2.8}$$

for some constant $C_2>0$ and all $n \in \mathbb{N}$. Results like (2.7) can be found in papers on approximation theory, see e.g. KOLMOGOROV and TIKHOMIROV [29] and LORENTZ [32].

By the techniques that we will discuss in Section 3, it can be shown that there also exists a constant $C_3>0$ such that

$$R_M(d_1,n) \geqslant C_3 n^{-(p+\alpha)/(1+2p+2\alpha)} \tag{2.9}$$

for all $n \in \mathbb{N}$. Hence the 'speed of estimation' in this density estimation problem is $n^{-(p+\alpha)/(1+2p+2\alpha)}$.

We now sketch the construction of an ϵ-net for the case $p=0$, i.e.

$$|f(x)-f(y)| \leqslant C|x-y|^{\alpha}, \quad x,y \in (0,1),\ \alpha \in (0,1], \tag{2.10}$$

for $f \in \mathcal{F}$ (f is *uniformly α-Hölder continuous*).

Fix $\epsilon>0$, let $\eta = 1/\{1+[(\epsilon/C)^{-1/\alpha}]\}$, where $[x]$ is the largest integer $\leqslant x$, and let $\mathcal{Q}$ be the set of functions ϕ on [0,1] such that

$$\begin{cases} \phi(i\eta) = j\epsilon, \quad i,j \in \mathbb{N} \cup \{0\}, \quad i \leqslant \eta^{-1} \\ \phi((i+1)\eta) = \phi(i\eta) + k\epsilon, \quad k = -1,0, \quad \text{or} \quad 1 \\ \phi \quad \text{is linear on the intervals} \quad [i\eta,(i+1)\eta] \end{cases}$$

Figure 2.1 shows a picture of such a function ϕ on a part of the interval [0,1].

For each $f \in \mathcal{F}$ there exists $\phi \in \mathcal{Q}$ such that $d_1(f,\phi) \leqslant 2\epsilon$. The set N of functions

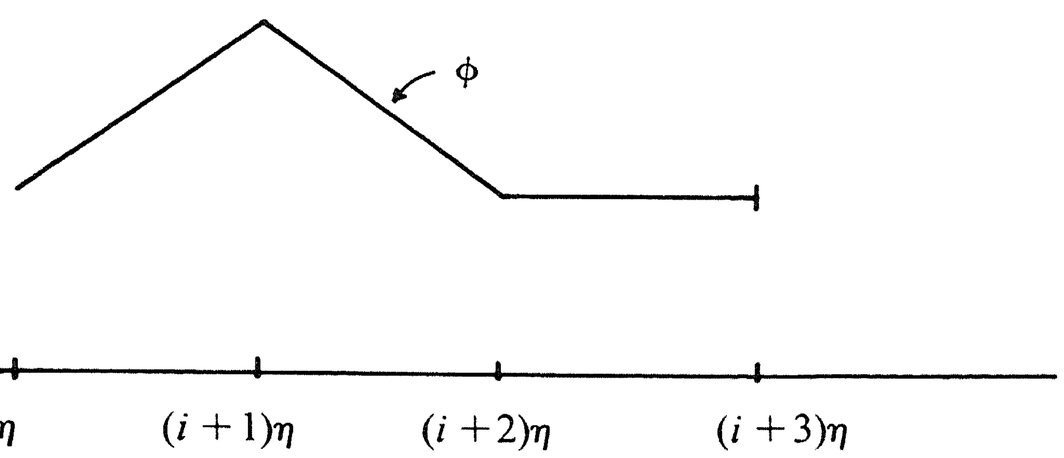


FIGURE 2.1.

$\phi\in\mathcal{Q}$ such that $d_1(f,\phi)\leqslant 2\epsilon$ for some $f\in\mathcal{F}$ is contained in the set of functions $\phi\in\mathcal{Q}$ such that $1-\epsilon\leqslant\phi(i\eta)\leqslant 1+\epsilon$ for at least one index i (since $|f(i\eta)-1|\leqslant\epsilon$ for at least one index i, if $f\in\mathcal{F}$, using the fact that f is a probability density).

Hence $\mathrm{Card}(N)\leqslant(\eta+1)^{-1}3^{\eta^{-1}}$. Picking one density $f\in\mathcal{F}$ in each L_1-ball $B(\phi;2\epsilon)$ of radius 2ϵ around a ϕ such that $B(\phi;2\epsilon)\cap\mathcal{F}\neq\varnothing$ provides us with a 4ϵ-net $N_{4\epsilon}$ of $\mathcal{F}$ such that

$$\log\{\mathrm{Card}(N_{4\epsilon})\} \leqslant C_1\eta^{-1}=C'_1\epsilon^{-\frac{1}{\alpha}}$$

Thus there exists a constant $C'_1>0$ such that, for all $\epsilon>0$,

$$H_\epsilon(\mathcal{F}) \leqslant C'_1\epsilon^{-1/\alpha}$$

For example, if $\alpha=1$ (a uniform Lipschitz condition on $\mathcal{F}$), we get $H_\epsilon(\mathcal{F})\leqslant C'_1\epsilon^{-1}$, and hence the speed of estimation is of order $n^{-1/3}$. We will meet the same speed of estimation in the next example on decreasing (but possibly discontinuous) densities on [0,1].

EXAMPLE 2.2 (Decreasing densities, BIRGÉ [7]). Suppose $\mathcal{F}$ is the family of decreasing (i.e. non-increasing) densities f on [0,1] such that $f\leqslant M$, for all $f\in\mathcal{F}$. We will show that there exists a $C_1>0$ such that the ϵ-entropy $H_\epsilon(\mathcal{F})$ satisfies

$$H_\epsilon(\mathcal{F}) \leqslant C_1\epsilon^{-1}, \quad \epsilon>0. \tag{2.11}$$

The following construction of a 4ϵ-net for $\mathcal{F}$ is based on BIRGÉ [7], with some simplifications due to the fact that we do not try to obtain the best (or at least a 'very good') constant C_1.

Let $\epsilon\in(0,1)$ and $p\in\mathbb{N}$ satisfy

$$M = (1+\epsilon)^p-1. \tag{2.12}$$

To avoid trivialities, we suppose $M>1$ (otherwise $\mathcal{F}$ only consists of one element: the uniform density on [0,1]). Define, for $0\leqslant i\leqslant p$

$$x_i = M^{-1}\{(1+\epsilon)^i-1\}, \quad y_i = (1+\epsilon)^i-1, \tag{2.13}$$

and for $0<i\leqslant p$,

$$I_i = [x_{i-1}, x_i) \tag{2.14}$$

The length l_i of the i-th interval I_i is $M^{-1}\epsilon(1+\epsilon)^{i-1}$. The 4ϵ-net that we will construct, will be based on the finite set $\mathcal{G}$ of functions g, which are constant on the intervals I_i and take values in the set

$$Y = \{y_0, \ldots, y_p\}. \tag{2.15}$$

Now let f be a decreasing density on [0,1]. We define

$$f_i = f(x_i);\ \bar{f}_i = l_i^{-1} \int_{I_i} f(x)dx. \tag{2.16}$$

Then $\sum_{i=1}^{p} l_i f_i \leqslant \sum_{i=1}^{p} l_i \bar{f}_i = 1$. Suppose

$$\bar{f}_i = \lambda y_{j-1} + (1-\lambda)y_j \qquad \begin{cases} 0\leqslant\lambda\leqslant 1 \\ y_{j-1}, y_j \in Y \end{cases} \tag{2.17}$$

(see figure 2.2).

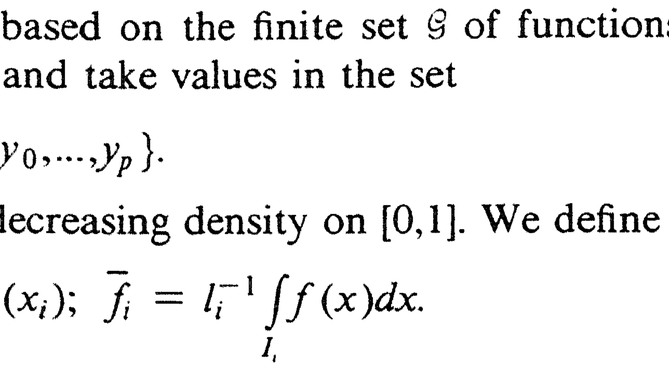


FIGURE 2.2.

Then we define the approximating function g on I_i by

$$g = \begin{cases} y_{j-1}, \text{ if } \lambda > \frac{1}{2} \\ y_j \ , \text{ if } \lambda \leqslant \frac{1}{2} \end{cases} \tag{2.18}$$

The function g is decreasing, nonnegative, and $g\leqslant M$. If $\lambda\leqslant\frac{1}{2}$, we get on the interval I_i

$$|\bar{f}_i - g| = |\bar{f}_i - y_j| \leqslant \tfrac{1}{2}\epsilon(1+\bar{f}_i) \tag{2.19}$$

and similarly $|\bar{f}_i - g| \leqslant \frac{1}{2}\epsilon(1+\bar{f}_i)$, if $\lambda > \frac{1}{2}$. Hence

$$\int_{I_i} |\bar{f}_i - g| dx \leqslant \tfrac{1}{2}\epsilon l_i(1+\bar{f}_i). \tag{2.20}$$

Since

$$\int_{I_i} |f(x)-\bar{f}_i|dx \leqslant \frac{1}{2} l_i(f_{i-1}-f_i)$$

and $l_{i+1} = (1+\epsilon)l_i$, we now get

$$\int_0^1 |f(x)-g(x)|dx \leqslant \sum_{i=1}^{p} \{\int_{I_i} |f(x)-\bar{f}_i|dx + \int_{I_i} |\bar{f}_i-g|dx\}$$

$$\leqslant \frac{1}{2}\{l_1 f_0 + \epsilon \sum_{i=1}^{p-1} l_i f_i\} + \epsilon \leqslant 2\epsilon$$

Thus, for each decreasing density f on [0,1], such that $f \leqslant M$, there exists a decreasing function g, which is constant on the intervals I_i and takes values in the set $Y = \{y_0,\ldots,y_p\}$, satisfying

$$\int_0^1 |f(x)-g(x)|dx \leqslant 2\epsilon \tag{2.21}$$

The number of functions g of this type equals the number of ways one can choose p nonnegative integers k_j such that $\sum_{j=1}^{p} k_j \leqslant p$. This number in turn equals the number of ways we can pick $p+1$ nonnegative integers $k_1,\ldots,k_{p+1}$ such that $\sum_{j=1}^{p+1} k_j = p$. This number is $\binom{2p}{p}$ (see e.g. FELLER [17], Section II. 5)

Choosing one $f \in \mathcal{F}$ for each $g \in \mathcal{G}$ in such a way that $d_1(f,g) \leqslant 2\epsilon$, provides us with a 4ϵ-net $N_{4\epsilon}$ of $\mathcal{F}$ such that

$$\text{Card}(N_{4\epsilon}) \leqslant \binom{2p}{p}$$

Since $\binom{2p}{p} \leqslant \dfrac{2^{2p}}{\sqrt{\pi p}}$, we have

$$H_{4\epsilon}(\mathcal{F}) \leqslant 2p \log 2 = (2\log 2)\cdot \frac{\log(M+1)}{\log(1+\epsilon)}$$

and $1/\log(1+\epsilon) \sim \epsilon^{-1}$, as $\epsilon \downarrow 0$. Thus there exists a constant $C_1 > 0$ such that: for all $\epsilon > 0$,

$$H_\epsilon(\mathcal{F}) \leqslant C_1 \epsilon^{-1}$$

and hence, by Theorem 2.1,

$$R_M(d_1,n) \leqslant C_2 n^{-1/3},$$

for some constant $C_2 > 0$. We will show in Section 3 that there also exists a constant $C_3 > 0$ such that

$$R_M(d_1,n) \geqslant C_3 n^{-1/3}, \quad n \in \mathbb{N},$$

implying that the speed of estimation is of order $n^{-1/3}$ for this estimation problem.

The proof of Theorem 2.1 is based on the felicitous idea, introduced by LE

CAM and further developed in the context of density estimation by BIRGÉ [4], [5], [6], [7], of constructing estimators on the basis of a family of tests of hypotheses. BIRGÉ calls these estimators *'d-estimators'*, where d denotes the distance function, used to define the loss function (in our case the L_1-distance). These estimators are concentrated on a ϵ-net and they give a connection between the ϵ-entropy and the minimax risk.

We now give the construction of the d-estimators based on one observation generated by a probability distribution p_θ, where θ belongs to a parameter set Θ. Suppose Θ is metrized by a metric d and totally bounded for this metric. Let, for $\epsilon>0$, N_ϵ be an ϵ-net of Θ and $\{B_s,\ s\in N_\epsilon\}$ be the family of balls, with radius ϵ and centers $s\in N_\epsilon$, covering Θ. Furthermore, let $\{\phi_{s,t}\}$ be a family of tests $\phi_{s,t}$ between the balls B_s and B_t for parameters $s,t\in N_\epsilon$, $s\neq t$, and let $J_s(X)$ be the set of t's in N_ϵ such that the tests $\phi_{s,t}$ rejects B_s and accepts B_t on the basis of the observation X. We suppose that $\phi_{s,t}$ is a real-valued function, defined on the space of possible observations, such that $0\leqslant\phi_{s,t}\leqslant 1$, and

$$\begin{cases}\phi_{s,t}(X) = 1 \Leftrightarrow B_s \text{ is rejected and } B_t \text{ is accepted}\\ \phi_{s,t}(X) = 0 \Leftrightarrow B_s \text{ is accepted and } B_t \text{ is rejected}\end{cases} \tag{2.22}$$

Define

$$L_s(X) = \begin{cases}\max\limits_{t\in J_s(X)} d(s,t)\\ \\ 0, \text{ if } J_s(X) = \varnothing\end{cases} \tag{2.23}$$

A d-estimator is now defined in the following way:

DEFINITION 2.1. *A d-estimator, based on the family of tests* $\{\phi_{s,t}\}$ *is a point* $\hat\theta(X) = t\in N_\epsilon$ *such that*

$$L_t(X) = \min_{s\in N_\epsilon} L_s(X). \tag{2.24}$$

In other words: a d-estimator is a point $s\in N_\epsilon$ such that the maximal distance $d(s,t)$ to 'preferred' points $t\in N_\epsilon$ (i.e. points t such that $\phi_{s,t}(X) = 1$) is minimized.

Now let $X_1,\ldots,X_n$ be a sample, generated by a density $f\in\mathcal{F}$, where $\mathcal{F}$ is as in Theorem 2.1, metrized by the L_1-distance d_1. A sample $X = (X_1,\ldots,X_n)$ can be considered as *one* observation, generated by the product measure P_f^n, where P_f is the probability measure corresponding to f. We identify $\mathcal{F}$ with the set $\Theta = \{P_f^n\colon f\in\mathcal{F}\}$ and metrize Θ by $d(P_f^n,\ P_g^n) = d_1(f,g)$.

Let $B(f;\epsilon) = \{h\in\mathcal{F};\ d_1(h,f)\leqslant\epsilon\}$ and $B(g;\epsilon)=\{h\in\mathcal{F}\colon d_1(h,g)\leqslant\epsilon\}$ be two closed ϵ balls in $\mathcal{F}$. In the problem of testing a ball $B(f;\epsilon)$ against a ball $B(g;\epsilon)$ we call a type I error, the error of concluding that the observations were generated by a density $h\in B(g;\epsilon)$, whereas they were actually generated by a density $h\in B(f;\epsilon)$, and a type II error, the error of concluding that the

observations were generated by a density $h \in B(f;\epsilon)$ whereas they were actually generated by a density $h \in B(g;\epsilon)$. The following lemma shows that the probabilities of type I and type II errors tend to zero exponentially fast as the sample size increases (if the balls $B(f;\epsilon)$ and $B(g;\epsilon)$ are disjoint).

LEMMA 2.1. *There exists a test $\phi_{f,g}$ between $B(f;\epsilon)$ and $B(g;\epsilon)$, based on a sample of size n, such that the sum of the maximal probabilities of errors of the first and the second kind is dominated by α_n, where*

$$\alpha_n = \exp\{-\tfrac{1}{8}n(d_1(f,g)-2\epsilon)_+^2\} \tag{2.25}$$

Here $x_+ = x$, if $x \geqslant 0$, and 0 otherwise. Otherwise stated:

$$\sup_{h \in B(f;\epsilon)} \int \phi_{f,g} dP_h^n + \sup_{h \in B(g;\epsilon)} \int (1-\phi_{f,g}) dP_h^n \leqslant \alpha_n \tag{2.26}$$

SKETCH OF PROOF. Let $D(P,Q) = \sup_{B \in \mathfrak{B}} |P(B)-Q(B)|$ be the total variation distance between two probability measures on $\mathbb{R}^d$, where $\mathfrak{B}$ is the collection of Borel sets of $\mathbb{R}^d$ (generated by the Euclidean topology). If P and Q have densities p and q w.r.t. Lebesgue measure, we have

$$D(P,Q) = \tfrac{1}{2}d_1(p,q),$$

(see (1.3) and (1.4)).

Let $\mathcal{P}$ be the set of probability measures on $\mathbb{R}^d$ (or, more correctly, on $\mathfrak{B}$) and let

$$\begin{cases} B_1 = \{P \in \mathcal{P} : 2D(P,P_f) \leqslant \epsilon\} \\ B_2 = \{P \in \mathcal{P} : 2D(P,P_g) \leqslant \epsilon\}. \end{cases} \tag{2.27}$$

Then $B(f;\epsilon) \subset B_1$ and $B(g;\epsilon) \subset B_2$ and the distance between the balls B_1 and B_2 is $\frac{1}{2}(d_1(f,g)-2\epsilon)_+$, if we use the total variation distance D. The balls B_1 and B_2 are convex and weakly compact *(unlike the L_1-balls $B(f;\epsilon)$ and $B(g;\epsilon)$)*. It now follows that

$$\nu_0 = \sup\{P\colon P \in B_1\}$$

is a 2-*alternating capacity* (CHOQUET [10], [11]) and that

$$\nu_1 = \inf\{P\colon P \in B_2\}$$

is a 2-*monotone capacity,* see HUBER and STRASSEN [24] and BEDNARSKI [2]. Let $d_1(f,g)-2\epsilon > 0$. It then follows from the results in the last-mentioned papers that there exists a mutually absolutely continuous pair (P,Q), with $P \in B_1$ and $Q \in B_2$, and a test ϕ of the form

$$\phi(x_1,\ldots,x_n) = \begin{cases} 1, & \text{if } \prod_{i=1}^n \frac{dQ}{dP}(x_i) \geqslant 1 \\ 0, & \text{otherwise,} \end{cases} \tag{2.28}$$

such that

$$\int \phi dP^n + \int (1-\phi) dQ^n = \sup_{P_1 \in B_1} \int \phi dP_1^n + \sup_{P_2 \in B_2} \int (1-\phi) dP_2^n$$
$$= 1 - \inf\{D(P_1^n, P_2^n): P_1 \in B_1,\ P_2 \in B_2\}.$$

Such a pair (P,Q) is called a *least favorable pair* for testing B_1 against B_2 (or a *least informative experiment* in the terminology of LE CAM [31] and BEDNARSKI [2]).

Take $\phi_{f,g} = \phi$ and put $\mu = P + Q$. Then it can be shown that

$$\int \phi_{f,g} dP^n + \int (1-\phi_{f,g}) dQ^n = 1 - D(P^n, Q^n)$$
$$\leqslant exp\{-\frac{n}{2}\int \{(\frac{dP}{d\mu})^{1/2} - (\frac{dQ}{d\mu})^{1/2}\}^2 d\mu\}$$
$$\leqslant exp\{-\frac{1}{8} n (d_1(f,g) - 2\epsilon)^2\}. \quad \square$$

REMARK 2.2. We note that the least favorable pair of probability measures (P,Q), introduced in the proof of Lemma 2.1, does not necessarily consist of probability measures which are absolutely continuous w.r.t. Lebesgue measure. Thus the test of $B(f;\epsilon)$ against $B(g;\epsilon)$, satisfying (2.26), may be based on a pair of probability measure *outside these balls.* This is caused by the fact that generally $B(f;\epsilon)$ and $B(g;\epsilon)$ are not weakly compact.

Proof of Theorem 2.1

Fix $\epsilon > 0$, and choose an ϵ-net N_ϵ of $\mathcal{F}$ such that

$$\log\{\text{Card}(N_\epsilon)\} \leqslant C_1 \epsilon^{-\delta} \tag{2.29}$$

(see condition (2.4) of Theorem 2.1). By Lemma 2.1 there exists a family of tests $\{\phi_{f,g}\}$, based on samples of size n, where $f,g \in N_\epsilon$ and $\phi_{f,g}$ is a test between the balls $B(f;\epsilon)$ and $B(g;\epsilon)$ satisfying (2.26).

Let $\hat{\theta}_n = \hat{\theta}_n(X_1,\ldots,X_n)$ be a d-estimator, based on the family of tests $\{\phi_{f,g}\}$ (see Definition 2.1). Fix $f \in \mathcal{F}$ and let $g \in N_\epsilon$ be a density such that $d_1(f,g) \leqslant \epsilon$. Furthermore, let N_i be the number of densities $h \in N_\epsilon$ such that $(i+2)\epsilon \leqslant d_1(h,g) < (i+3)\epsilon$. Then we have, by Lemma 2.1, for $i \geqslant 1$,

$$P_f\{d_1(\hat{\theta}_n, f) \geqslant (3+i)\epsilon\} \leqslant P_f\{d_1(\hat{\theta}_n, g) \geqslant (2+i)\epsilon\}$$
$$\leqslant \sum_{j \geqslant i} N_j exp\{-\frac{1}{8} n j^2 \epsilon^2\}$$

Hence,

$$E_f d_1(\hat{\theta}_n, f) \leqslant 4\epsilon + \epsilon \sum_{i \geqslant 1} P_f\{d_1(\hat{\theta}_n, f) \geqslant (3+i)\epsilon\}$$
$$\leqslant \epsilon(4 + \sum_{i \geqslant 1} \sum_{j \geqslant i} N_j exp\{-\frac{1}{8} n j^2 \epsilon^2\}$$

$$E_f d_1(\hat{\theta}_n, f) = \epsilon(4 + \sum_{i \geq 1} iN_i exp\{-\tfrac{1}{8} n i^2 \epsilon^2\})$$

Let $n \geq 8C_1$, where C_1 is as in (2.29), and choose $\epsilon > 0$ in such a way that $\epsilon^{2+\delta} = 8C_1/n$. Then the function $j \to j\ exp\{-\frac{1}{8} n j^2 \epsilon^2\}$ is decreasing for $j \geq j_0 = 1 + [1/C_1]$, and hence

$$\begin{aligned} E_f d_1(\hat{\theta}_n, f) &\leq \epsilon(4 + \sum_{i \geq 1} iN_i exp\{-\tfrac{1}{8} n i^2 \epsilon^2\}) \\ &\leq 4\epsilon + \epsilon j_0 \mathrm{Card}(N_\epsilon) exp\{-\tfrac{1}{8} n \epsilon^2\} \\ &= (4 + j_0)(8C_1)^{1/(2+\delta)} n^{-1/(2+\delta)} \end{aligned}$$

Put $C = (4 + j_0)(8C_1)^{1/(2+\delta)}$. Then we get $\sup_{f \in \mathfrak{F}} E_f d_1(\hat{\theta}_n, f) \leq C n^{-1/(2+\delta)}$. □

3. Lower bounds for the minimax risk

In obtaining lower bounds for the minimax risk, we compare two kinds of 'distinguishability' of the probability densities:

1) distinguishability in terms of the loss function on the set of densities
2) distinguishability in terms of some 'information measure'.

Usually the distinguishability in terms of an information measure is measured by the *Kullback-Leibler information*

$$K(Q,P) = \begin{cases} \int \log\left[\frac{dQ}{dP}\right] dQ, & \text{if} \quad Q \ll P \\ \\ \infty, & \text{otherwise} \end{cases} \tag{3.1}$$

One then uses an information-theoretic lemma (Fano's Lemma, see e.g. Ibragimov and Hasminskii [26], p. 323-325) to give lower bounds for the minimax risk. This technique is used in e.g. Birgé [4], [5], [6], [7], Bretagnolle and Huber [9] and Ibragimov and Hasminskii [25], [26], [27].

Here we give another Lemma (Assouad's Lemma), where the Kullback-Leibler information $K(Q,P)$ is replaced by the *Hellinger distance* $h(P,Q)$, defined by

$$h(P,Q) = \{\tfrac{1}{2} \int \{\left[\frac{dP}{d\mu}\right]^{1/2} - \left[\frac{dQ}{d\mu}\right]^{1/2}\}^2 d\mu\}^{1/2}, \tag{3.2}$$

where μ is any measure dominating P and Q (for example: $\mu = P + Q$). Roughly speaking, the Hellinger distance can be considered as a local version of the Kullback-Leibler information; it has the advantage of being a *distance*. The Kullback-Leibler information, sometimes called 'Kullback-Leibler distance' is not really a distance (it does not satisfy the triangle inequality).

We now state Assouad's Lemma in a form slightly adapted to our purposes.

LEMMA 3.1 (ASSOUAD, 1982). *Let* $A_r=\{0,1\}^r=\{a:a=(a_1,...,a_r),\ a_i=0 \text{ or } 1\}$ *and let* $\mathfrak{F}$ *be a collection of probability densities on* $\mathbb{R}^d$. *Suppose that* $\phi:a\to f_a$ *is a bijection of* A_r *on a subset* $\mathfrak{F}_r$ *of* $\mathfrak{F}$ *and that* $\{B_1,...,B_r\}$, *is a partition of* $\mathbb{R}^d$ *into measurable sets* $B_1,...,B_r$ *such that, if* $a=(a_1,...,a_{i-1},\ 1,\ a_{i+1},...,a_r)$ *and* $a'=(a_1,...,a_{i-1},\ 0,\ a_{i+1},...,a_r)$

$$\frac{1}{2}\int_{\mathbb{R}^d}(f_a^{1/2}-f_{a'}^{1/2})^2dx\leqslant\beta_i\leqslant 1 \tag{3.3}$$

$$\int_{B_i}|f_a-f_{a'}|dx\geqslant\alpha_i>0. \tag{3.4}$$

Let $\hat{f}_n$ *be* any *density estimator, based on a sample of size n, generated by a density* $f\in\mathfrak{F}$. *Then*

$$\sup_{f\in\mathfrak{F}}E_f\int_{\mathbb{R}^d}|\hat{f}_n-f|dx\geqslant\frac{1}{2}\sum_{i=1}^{r}\alpha_i\max\{1-(2n\beta_i)^{1/2},\quad\frac{1}{2}(1-\beta_i)^{2n}\}.$$

We omit the proof of this lemma, but instead discuss some applications. Usually the α_i's are taken equal to some α and the β_i's taken equal to some β. One then looks for densities which are α-separated in the L_1-distance, but have the smallest possible Hellinger distance. In fact we are then dealing with the (local) α-capacity of the set $\mathfrak{F}$. Hopefully these remarks will become more clear by looking at some examples.

EXAMPLE 3.1 (Continuation of Example 2.1). Suppose that $\mathfrak{F}$ consists of the densities f on $[0,1]$, satisfying

$$|f(x)-f(y)|\leqslant C|x-y|^\alpha,\quad x,y\in[0,1]$$

where C is independent of f. Let $\epsilon\in(0,1)$ and $\eta=\{1+[(\frac{1}{4}\epsilon/C)^{-1/\alpha}]\}^{-1}$. Suppose $b_j=j\eta\leqslant 1$, for some positive integer j and let f_j be the piecewise linear function defined on $[b_{j-1},\ b_j]$ by $f_j(b_{j-1})=0$, $f_j(b_{j-1}+\frac{1}{4}\eta)=\frac{1}{4}\epsilon$, $f_j(b_{j-1}+\frac{1}{2}\eta)=0$, $f_j(b_{j-1}+\frac{3}{4}\eta)=-\frac{1}{4}\epsilon$, $f_j(b_j)=1$, and f_j is linearly interpolated for other values $x\in[b_{j-1},b_j]$ (see figure 3.1). Define $r=\eta^{-1}$ and $\mathfrak{F}_r=\{f:[0,1]\to\mathbb{R}|\ f=1+\sum_{i=1}^r\lambda_i f_i,\ \lambda_i=\pm 1\}$. Let the function $\phi:\{0,1\}^r\to\mathfrak{F}_r$ be defined by

$$\phi(a)=1+\sum_{i=1}^{r}\lambda_i f_i,\quad\begin{cases}\lambda_i=1, & \text{if } a_i=1\\ \lambda_i=-1, & \text{if } a_i=0.\end{cases}$$

Then, if $a=(a_1,...,a_{i-1},\ 1,\ a_{i+1},...,a_r)$ and $a'=(a_1,...,a_{i-1},\ 0,\ a_{i+1},...,a_r)$

$$\int_{b_{i-1}}^{b_i}|f_a-f_{a'}|dx=\frac{1}{4}\epsilon\eta, \tag{3.5}$$

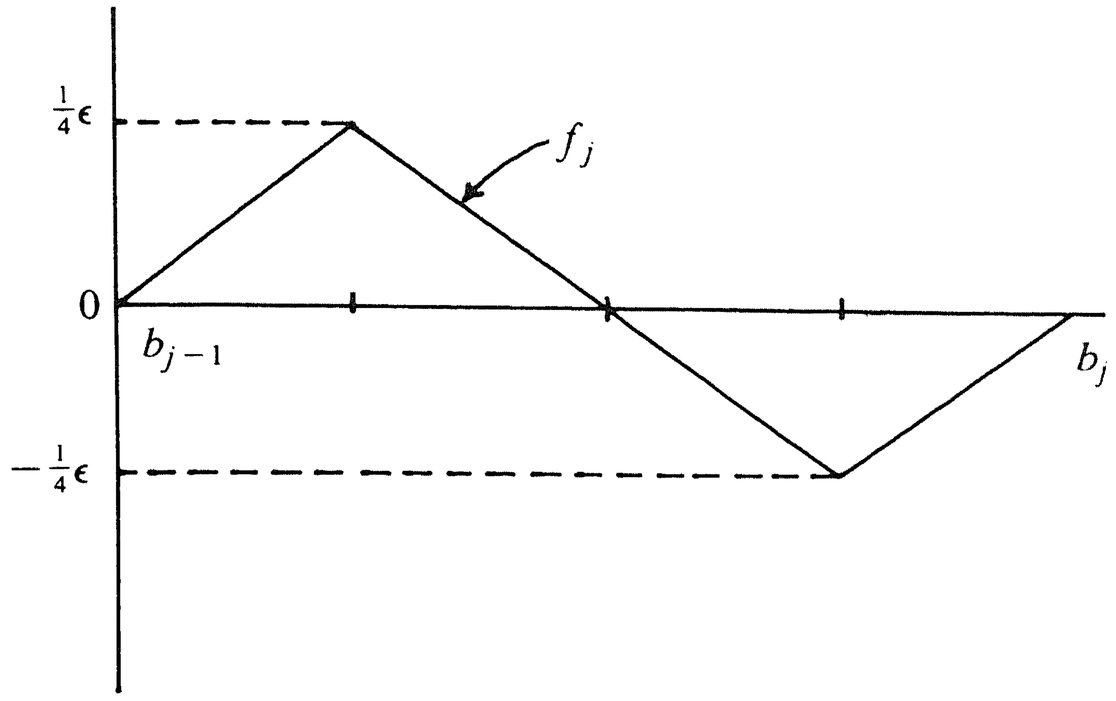


FIGURE 3.1.

and the Hellinger distance satisfies

$$h^2(f_a,f_{a'}) \leqslant \frac{1}{12}\eta\epsilon^2 \tag{3.6}$$

Thus the conditions (3.3) and (3.4) of Assouad's Lemma are satisfied, with $\alpha_i = 1/4\epsilon\eta$ and $\beta_i = (1/12)\eta\epsilon^2$, and we obtain for $n = [1/(\eta\epsilon^2)]$ and *any* density estimator $\hat{f}_n$ based on a sample of size n generated by a density $f \in \mathfrak{F}$,

$$\sup_{f\in\mathfrak{F}} E_f \int_{\mathbb{R}} |\hat{f}_n - f| dx \geqslant \epsilon(1-6^{-\frac{1}{2}})/8 \tag{3.7}$$

Hence the minimax risk $R_M(d_1,n)$ satisfies (for a constant $c>0$)

$$R_M(d_1,n) \geqslant c \,.\, n^{-\alpha/(1+2\alpha)}$$

Since, by Example 2.1 ((2.8) with $p=0$),

$$R_M(d_1,n) \leqslant C' \,.\, n^{-\alpha/(1+2\alpha)}$$

for some $C'(>c)$, the speed of convergence to zero of the minimax risk is of order $n^{-\alpha/(1+2\alpha)}$.

EXAMPLE 3.2 (Continuation of example 2.2). Let $\mathfrak{F}$ be the set of decreasing densities on [0,1], such that $f \leqslant M$ for each $f \in \mathfrak{F}$, with $M>1$. We will show that the minimax risk satisfies

$$R_M(d_1,n) \geqslant Cn^{-\frac{1}{3}}, \quad n \in \mathbb{N} \tag{3.8}$$

Since it was shown in Section 2 that $R_M(d_1,n) \leqslant C' n^{-1/3}$, for all $n \in \mathbb{N}$ and some $C'>0$, the minimax risk tends to zero at the rate $n^{-1/3}$.

Let $\epsilon \in (0, \frac{1}{2})$, $r \in \mathbb{N}$, $u = \{(1+\epsilon)^r - 1\}^{-1}$, $\lambda = (1+\epsilon)/\{ru\epsilon(1+\frac{1}{2}\epsilon)\}$, and $x_i = u\{(1+\epsilon)^i - 1\}$, $0 \leqslant i \leqslant r$.

Define, for $1 \leqslant i \leqslant r$, the intervals I_i by $I_i = [x_{i-1}, x_i)$. The interval I_i has length $l_i = u\epsilon(1+\epsilon)^{i-1}$. Let the functions f_i and g_i be defined on the interval I_i by

$$f_i(x) = \lambda(1+\epsilon)^{-i}(1+\tfrac{1}{2}\epsilon), \quad x \in I_i, \tag{3.9}$$

and

$$g_i(x) = \begin{cases} \lambda(1+\epsilon)^{-i+1}, & \text{first half of } I_i \\ \lambda(1+\epsilon)^{-i}, & \text{second half of } I_i \end{cases} \tag{3.10}$$

(see figure 3.2).

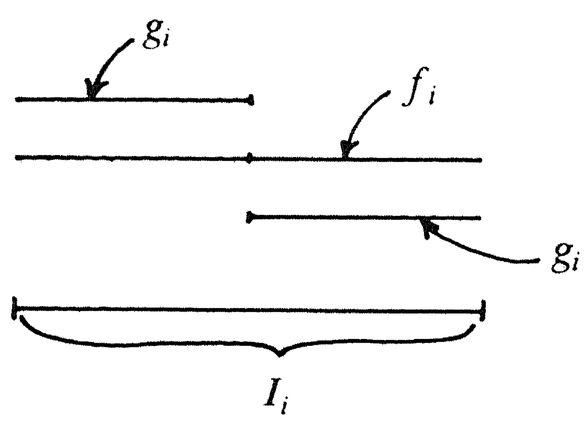


FIGURE 3.2.

Then $\int_{I_i} g\,dx = \int_{I_i} f\,dx = 1/r$, and

$$\int_{I_i} |f_i - g_i|\,dx = \epsilon/\{(2+\epsilon)r\}, \tag{3.11}$$

$$\frac{1}{2}\int_{I_i} \{\sqrt{f_i} - \sqrt{g_i}\}^2 dx < \frac{1}{32}\epsilon^2/r. \tag{3.12}$$

Now let $\mathcal{F}_r$ be the family of 2^r functions, defined on $[0,1)$ by

$$f = \sum_{i=1}^{r} (\lambda_i f_i + (1-\lambda_i) g_i) 1_{I_i}, \tag{3.13}$$

where $\lambda_i = 0$ or 1 and 1_{I_i} is the indicator function of the interval I_i. Suppose that ϵ and r satisfy

$$g_1(0) = \frac{1+\epsilon}{r\epsilon(1+\frac{1}{2}\epsilon)}\{(1+\epsilon)^r - 1\} \leqslant M. \tag{3.14}$$

Since $f \leqslant g_1(0)$, if $f \in \mathcal{F}_r$, we have $\mathcal{F}_r \subset \mathcal{F}$ if (3.14) is satisfied. Hence, by (3.11), (3.12) and Lemma 3.1,

$$R_M(d_1,n) \geqslant \frac{1}{2}\frac{\epsilon}{2+\epsilon}\{1-\sqrt{\frac{2n\epsilon^2}{32r}}\} \tag{3.15}$$

Choose, for each $r\in\mathbb{N}$, the number $\epsilon_r>0$ such that $(1+\epsilon_r)^r=M$. Then

$$(1+\epsilon_r)\{(1+\epsilon_r)^r-1\}/\{r\epsilon_r(1+\tfrac{1}{2}\epsilon_r)\} \sim \frac{M-1}{\log M}, \tag{3.16}$$

as $r\to\infty$. Since $(M-1)/\log M<M$, for $M>1$, there exists r_0 such that the left-hand side of (3.16) is smaller than M, if $r\geqslant r_0$. Hence, by (3.15), $\mathcal{F}_r\subset\mathcal{F}$, if $r\geqslant r_0$. Taking $n=[r/\epsilon_r^2]$ yields

$$R_M(d_1,n) \geqslant \frac{\epsilon_r}{2(2+\epsilon_r)}(1-\tfrac{1}{4}) \sim (3/16)\,(\log M)^{1/3}n^{-1/3},$$

as $r\to\infty$ (and hence $\epsilon_r\downarrow 0$, $n\to\infty$). Thus there exists a constant $C>0$ such that (3.8) holds.

REMARK 3.1. BIRGÉ [7] gives a better constant in the lower bound of the minimax risk (at the cost of more difficult computations).

REMARK 3.2. The restriction to the interval [0,1] in Examples 3.1 and 3.2 is not essential, but the restriction to compact intervals is. For example, if $\mathcal{F}$ is the family of decreasing densities on $[0,\infty)$, we get arbitrarily slow rates of convergence for the minimax risk (like in Remark 2.1), even if $f\leqslant M$, for all $f\in\mathcal{F}$.

If $\mathcal{F}$ is the family of decreasing densities f on $[0,L]$ such that $f\leqslant M$, for all $f\in\mathcal{F}$, we obtain by similar computations as in examples 2.2 and 3.2

$$C_1(\log LM)^{1/3}n^{-1/3} \leqslant R_M(d_1,n) \leqslant C_2(\log LM)^{1/3}n^{-1/3} \tag{3.17}$$

Hence, for fixed n, the minimax risk grows at the rate $(\log LM)^{1/3}$, as the area LM of the rectangle $[0,L]\times[0,M]$ tends to infinity. BIRGÉ has shown that $C_2/C_1\leqslant 40$, which shows that the minimax risk is squeezed in rather tightly by the bounds in (3.17).

It was shown in Examples 2.2 and 3.2 that the minimax risk for the estimation of decreasing densities on [0,1], bounded by some $M>0$ (which is the same for all densities in the class), tends to zero at the rate $n^{-1/3}$, as the sample size $n\to\infty$, if the loss is measured by L_1-distance. This suggests that a more precise picture of what is going on is obtained by looking at neighborhoods around a (decreasing) density f, which shrink at the rate $n^{-1/3}$, and by evaluating the (local) minimax risk of estimators based on a sample of size n over such a neighborhood. This leads to the following definition.

DEFINITION 3.1. *Let $\mathcal{F}$ be a class of densities on $\mathbb{R}^d$ and let $E_f d_1(\hat{f}_n,f)$ be the risk under f of an estimator $\hat{f}_n$ of f based on a sample $X_1,\ldots,X_n$ from f, where d_1 denotes the L_1-distance. Then the local asymptotic minimax risk at a density $f\in\mathcal{F}$ is defined by*

$$R_{LM}(f,d_1) = \sup_{c>0} \lim_{n\to\infty} \inf_{f_n} \sup_{g\in U_{n,c}(f)} n^{1/3} E_g d_1(\hat{f}_n,g) \tag{3.18}$$

where

$$U_{n,c}(f) = \{g\in\mathcal{F}: d_1(g,f)\leqslant c\,.\,n^{-1/3}\}$$

We now have the following result.

THEOREM 3.1. *There exists a constant $c_1>0$ such that for each decreasing density f on [0,1], with a bounded continuous derivative f' such that $f'<0$ on (0,1),*

$$R_{LM}(f,d_1) \geqslant c_1 \int_0^1 |f(t)f'(t)|^{1/3}dt, \tag{3.19}$$

where $R_{LM}(f,d_1)$ is defined by (3.18).

PROOF. We give the proof for the situation where $f'\leqslant a<0$ and $f\geqslant b>0$ on (0,1), but only minor changes are needed to give the proof for the situation where f' (or f) is allowed to tend to zero at the right endpoint of (0,1).

Let $x_0, x_1,\ldots,x_{2m}$ be an increasing sequence of points in [0,1] such that $x_0=0$, and

$$\begin{aligned}\delta_i &= x_{2i-1}-x_{2i-2}=x_{2i}-x_{2i-1} \\ &= \tfrac{1}{2}n^{-1/3}f(x_{2i-1})^{1/3}/\,|f'(x_{2i-1})\,|^{2/3}\end{aligned} \tag{3.20}$$

for $i=1,\ldots,m$. Suppose that $[x_{2m-2},x_{2m-1})$, $[x_{2m-1},x_{2m})$ is the last pair of intervals of this type, contained in [0,1). *Although m, δ_i, and the points $x_1,\ldots,x_{2m}$ depend on n, we suppress this dependence to avoid cumbersome notation.* Furthermore, for ease of notation we put $y_i=x_{2i-1}$. Define the functions f_i and g_i on the interval $J_i=[y_i-\delta_i, y_i+\delta_i)$ by

$$f_i(x) = f(y_i), \quad x\in J_i, \tag{3.21}$$

and

$$g_i(x) = \begin{cases} f(y_i)+\delta_i|f'(y_i)|, & y_i-\delta_i\leqslant x<y_i \\ f(y_i)-\delta_i|f'(y_i)|, & y_i\leqslant x<y_i+\delta_i. \end{cases} \tag{3.22}$$

Let $\tilde{f}_n$ be a probability density on $[0, x_{2m})$ such that $\tilde{f}_n|J_i=k_nf_i$. Then $k_n\to 1$, as $n\to\infty$, implying that the function $\tilde{g}_n$, defined on $[0, x_{2m})$ by $\tilde{g}_n|J_i=k_ng_i$ will be nonnegative and hence a probability density for n sufficiently large (since $\int_0^{x_{2m}}\tilde{g}_n(x)dx=\int_0^{x_{2m}}\tilde{f}_n(x)dx=1$).

As $n\to\infty$, we have

$$\tfrac{1}{2}\int_{J_i}(\tilde{f}_n^{1/2}-\tilde{g}_n^{1/2})^2\sim\tfrac{1}{4}\delta_i^3 f(y_i)^{-1}f'(y_i)^2 \tag{3.23}$$

$$\int_{J_i}|\tilde{f}_n-\tilde{g}_n|\sim 2\delta_i^2\,|f'(y_i)\,| \tag{3.24}$$

Applying Assouad's Lemma we obtain by (3.20), (3.23) and (3.24)

$$R_{LM}(f,d_1) \geqslant \lim_{n\to\infty} n^{1/3}\sum_i \delta_i^2 |f'(y_i)|\{1-\sqrt{\tfrac{1}{2}n\delta_i^3 f'(y_i)^2/f(y_i)}\}$$

$$= \tfrac{3}{4}\lim_{n\to\infty} n^{1/3}\sum_i \delta_i^2 |f'(y_i)|$$

$$= \tfrac{3}{8}\int_0^1 |f(x)f'(x)|^{1/3}dx. \quad \square$$

The Grenander maximum likelihood estimator $\hat{f}_n$, to be discussed in section 4, has the property that for any 'smooth' density f such that $f'<0$ on $(0,1)$,

$$\lim_{n\to\infty} n^{1/3}E_f\int |\hat{f}_n(t)-f(t)|dt = c\cdot\int_0^1 |f(t)f'(t)|^{1/3}dt, \tag{3.25}$$

where $c\approx 0.62$ (see GROENEBOOM [20]). If $f(t)=1$, $t\in[0,1]$ (the *uniform* density on [0,1]), the right-hand side of (3.25) is zero, and it can be shown that in this case

$$\lim_{n\to\infty} n^{1/2}E_f\int |\hat{f}_n(t)-f(t)|dt = \frac{\sqrt{\pi}}{2} \tag{3.26}$$

(see GROENEBOOM [20], Remark 3.2).

The behavior of kernel estimators is rather different. For example, it is proved in DEVROYE and GYÖRFI [14] that for *any* kernel estimator of the form (1.1) with a kernel K with bounded support

$$\hat{g}_n(t) = (nh)^{-1}\sum_{i=1}^n K((t-X_i)/h),$$

based on a sample $X_1,\ldots,X_n$ generated by a density f, we have

$$\liminf_{n\to\infty}\inf_{h>0} n^{1/3}E_f\int |\hat{g}_n(t)-f(t)|dt \geqslant (8/(9\pi))^{1/3} \tag{3.27}$$

if f is the uniform density on [0,1] (Theorem 7, Ch. 5, DEVROYE and GYÖRFI [14]). This shows that these kernel estimators can only achieve a rate of convergence $n^{-1/3}$, whereas the Grenander estimator achieves the rate $n^{-1/2}$.

More generally, it can be shown that the Grenander estimator achieves the rate $n^{-1/2}$ for any density f on [0,1], which only consists of flat parts and a finite number of jumps, whereas kernel estimators would only achieve rate $n^{-1/3}$ in this case.

A comparison of (3.25) and (3.19) indicates that the Grenander estimator has very good properties according to a (suitably defined) criterion of local minimax risk. However, at present it is still an unsolved problem how to choose the collection $\mathcal{F}$ of decreasing densities (and, for that matter, the corresponding neighborhoods $U_{n,c}(f)$ in (3.18)) in order to obtain nontrivial upper bounds for the local minimax risk. This has to do with the somewhat peculiar behavior of the functional $f\to\int_0^1 |f(t)f'(t)|^{1/3}dt$, and the fact that the convergence in (3.25) is certainly not uniform in f.

4. The Grenander maximum likelihood estimator

Distribution theory

At the end of Section 3 it was noticed that a particular density estimator 'the Grenander maximum likelihood estimator' has a better performance in estimating decreasing densities than kernel estimators. We will now describe the construction of the Grenander estimator and we will offer an explanation for its good performance. The general consequences of an analysis of the behavior of the Grenander estimator are rather striking and not limited to the case of decreasing densities.

Suppose $X_1,\ldots,X_n$ is a sample of n independent random variables generated by a density f on $[0,\infty)$. The *empirical distribution function* F_n of the sample is defined by

$$F_n(x) = \frac{1}{n}.\#\{i: X_i \leqslant x\}, \tag{4.1}$$

where $\#A$ denotes the number of elements in the set A. Thus $F_n(x)$ is the fraction of observations less than or equal to x. The *concave majorant* $\hat{F}_n$ of F_n on $[0,\infty)$ is by definition the smallest concave function $\geqslant F_n$ on $[0,\infty)$. Figure 4.1 shows a picture of the empirical distribution function F_n and its concave majorant $\hat{F}_n$ for a sample of $n=100$ observations, generated by the uniform density

$$f(t) = 1, \quad t \in [0,1]. \tag{4.2}$$

Grenander shows (in Grenander [18]) that the *maximum likelihood estimator* (MLE) of a decreasing density, based on a sample generated by this density, is given by the derivative of the concave majorant $\hat{F}_n$ of the empirical distribution F_n of the sample. Since the function $\hat{F}_n$ is piecewise linear with at most n changes of direction, the derivative is meaningful except at (at most) n points. Let f_n denote this derivative, defined at points of discontinuity by taking left-hand limits. This function satisfies

$$\prod_{i=1}^{n} \hat{f}_n(X_i) = \sup_{f \in \mathcal{F}} \prod_{i=1}^{n} f(X_i) \tag{4.3}$$

where $\mathcal{F}$ is the set of decreasing left-continuous densities on $[0,\infty)$. Thus $\hat{f}_n$ is *that* density f in the class $\mathcal{F}$ for which the 'joint' density $\Pi_{i=1}^{n} f(X_i)$ at the observed points $X_1,\ldots,X_n$ attains its highest value, and for this reason $\hat{f}_n$ is called the maximum likelihood estimator. For a picture of $\hat{f}_n$, based on the same sample as used for figure 4.1, see figure 4.2.

The distribution theory of the Grenander estimator is still incomplete. An interesting early result is given in Sparre Andersen [33], where it is proved that the number of jumps N_n of the function $\hat{f}_n$ is of order $\log n$, if the observations are generated by the uniform density defined by (4.2). More precisely, he proved that the distribution of the random variable $(N_n - \log n)/\sqrt{\log n}$ tends to a Gaussian distribution with mean zero and variance 1 (the 'standard

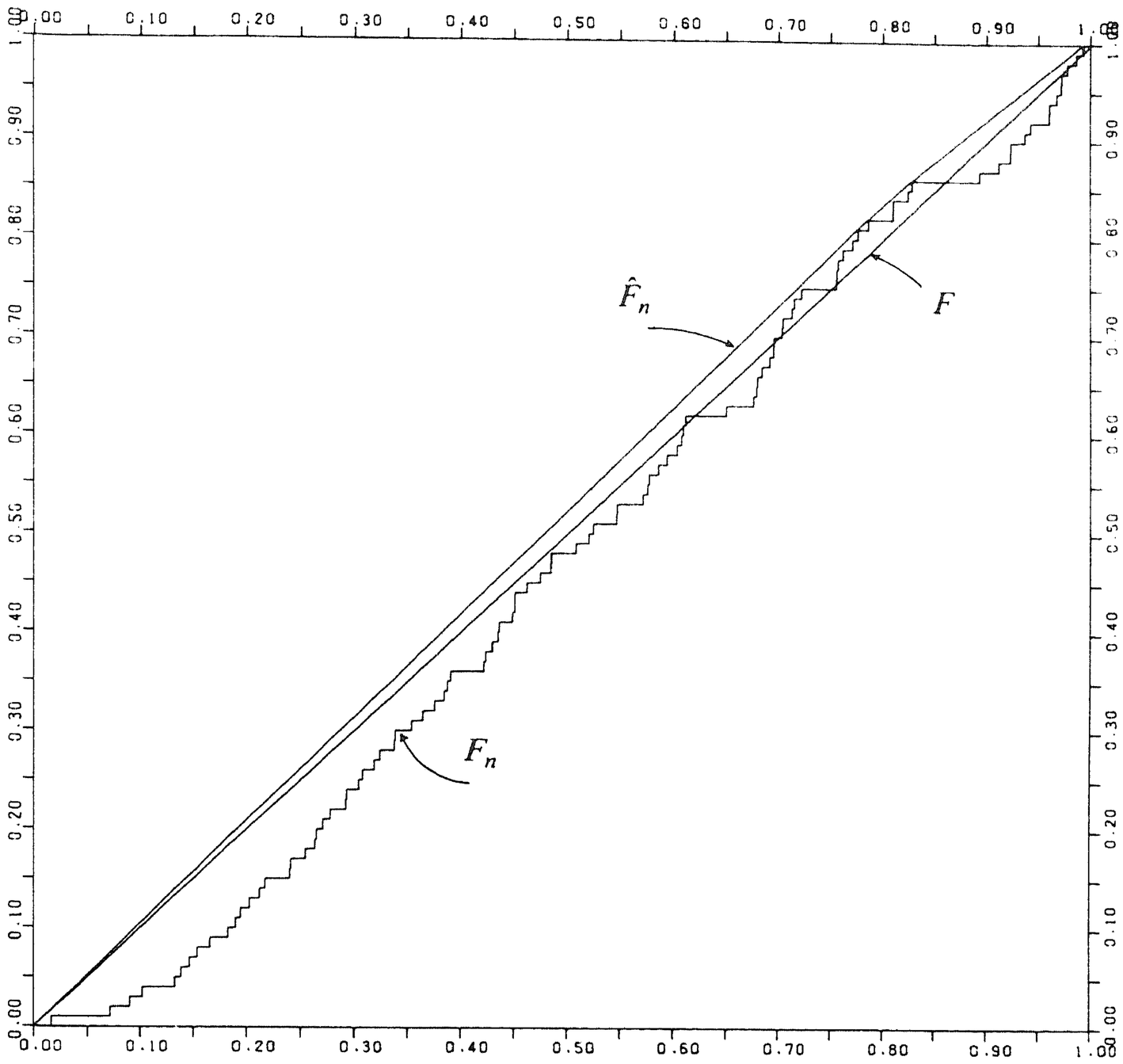


FIGURE 4.1. Concave majorant $\hat{F}_n$, $n=100$, $F(t)=t$, $t\in[0,1]$

normal distribution'), as the sample size n tends to infinity. The proof in SPARRE ANDERSEN [33] is based on rather elaborate enumeration techniques. At present it is possible to give a very quick proof of this result by using some properties of Brownian motion.

Since the further distribution theory of the Grenander estimator (and of density estimators in general) has been developed by using the relation between the empirical distribution function and Brownian motion, we now turn to an informal description of Brownian motion.

Let $X_1, X_2, \ldots$ be an infinite sequence of independent identically distributed random variables such that $P\{X_i=1\}=P\{X_i=-1\}=1/2$ for each i. For example, X_i could represent the outcome of the i-th trial in a fair coin-tossing game, where $X_i=1$ represents 'heads' and $X_i=-1$ represents 'tails'. Corresponding to each infinite sequence $(X_1, X_2, \ldots,)$ we define a function $W_n:[0,\infty)\to\mathbb{R}$ by

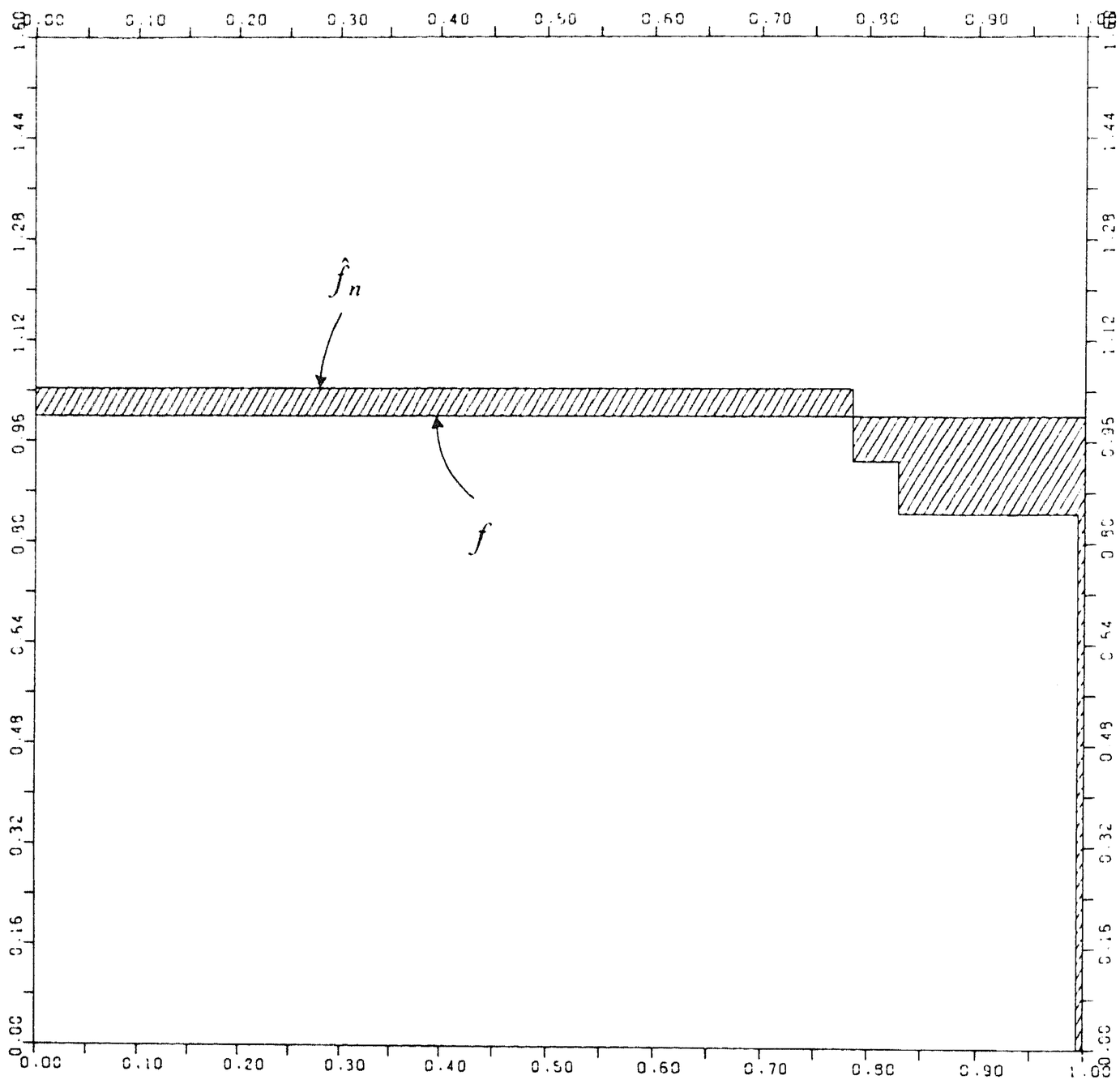


FIGURE 4.2. Grenander estimator, $n = 100$, $t = 1$, $t \in [0, 1]$

$$\begin{cases} W_n(0) = 0 \\ W_n(j/n) = n^{-1/2} \sum_{i=1}^{j} X_i, \quad j \in \mathbb{N} \end{cases} \tag{4.4}$$

and $W_n(t)$ is defined by linear interpolation for other values of $t \in [0, \infty)$. Each such function W_n is a possible realization of a *random walk* of a particle which jumps up or down according to the outcomes of a fair coin tossing game. By the central limit theorem we have that the distribution of $W_n(j/n)$ tends to a Gaussian distribution with mean zero and variance t, as $n \to \infty$ and $j/n \to t > 0$. More generally, it has been shown by Wiener that one can define a limiting process consisting with probability one of continuous (nowhere differentiable) functions (or 'paths') W on $[0, \infty)$ such that $W(t)$ has a Gaussian distribution with mean zero and variance t, for each t, and such that the distribution of $W(t) - W(s)$ is independent of that of $W(s)$ for $t > s$ (the process has *independent increments).* This process is called *Brownian motion* and can be considered

as the limit (as $n \to \infty$) of the random walks W_n, defined by (4.4) on the basis of coin-tossing sequences $(X_1, X_2, \ldots)$.

The *Brownian bridge* on [0,1] is a process of continuous paths $B:[0,1] \to \mathbb{R}$ which are obtained from Brownian motion paths W by the transformation

$$\begin{cases} B(t) = (1-t)W(t/(1-t)), & t \in [0,1) \\ B(1) \overset{def}{=} 0 \end{cases} \tag{4.5}$$

This transformation is called *Doob's transformation*. For a discussion of these concepts see e.g. BILLINGSLEY [3], DOOB [15] and ITÔ and MCKEAN [28].

Brownian motion and the Brownian bridge arise in the context of density estimation in the following way. All the density estimators used in practice are based on the empirical distribution function F_n. Now it is already known for a long time (see e.g. DOOB [15]) that the so-called *empirical process*

$$\sqrt{n}(F_n(t) - \int_0^t f(u)du), \quad t \in [0, \infty), \tag{4.6}$$

where F_n is the empirical distribution function based on a sample of size n generated by the density f on $[0, \infty)$, behaves for large n as a Brownian bridge with a changed time scale. More precisely it has been shown by KOMLÓS, MAJOR and TUSNÁDY [30] that the supremum distance (over t) between the empirical process defined by (4.6) and a Brownian bridge process (with changed time scale)

$$\{B_n(F(t)), \quad t \in [0, \infty)\} \tag{4.7}$$

where $F(t) = \int_0^t f(u)du$, is smaller than $k \,.\, n^{-1/2} \log n$, with a probability tending to one as $n \to \infty$, for some fixed constant $k > 0$. In particular we will have that well-behaving functionals of the empirical process will converge in distribution to the corresponding functional of the Brownian bridge; this is the so-called *invariance principle*. As an example, we have the following result.

THEOREM 4.1. *Let $\hat{f}_n$ be the Grenander density estimator, based on a sample of size n, generated by the uniform density f on [0,1] (see (4.2)). Then we have, as $n \to \infty$,*

$$n^{1/2} \int_0^1 |\hat{f}_n(t) - f(t)| dt \xrightarrow{d} 2 \max_{t \in [0,1]} B(t), \tag{4.8}$$

i.e. the L_1-distance between $\hat{f}_n$ and f, multiplied by $n^{1/2}$, converges in distribution to 2 times the maximum of the Brownian bridge.

SKETCH OF PROOF. Since $\hat{f}_n$ is the slope of the concave majorant $\hat{F}_n$ of the empirical distribution function F_n on [0,1], we have that $n^{1/2}(\hat{f}_n - 1)$ is the slope of $n^{1/2}(\hat{F}_n - F)$, where $F(t) = \int_0^t f(u)du = \int_0^t du = t$, for $t \in [0,1]$. This means that $S_n = n^{1/2}(\hat{f}_n - f)$ is the slope of the concave majorant of the empirical process $n^{1/2}(F_n - F)$ on [0,1].

Applying the invariance principle, we get that the functional $\int_0^1 |S_n(t)|dt$ of the empirical process converges in distribution to the corresponding functional $\int_0^1 |S(t)|dt$ of the Brownian bridge, where $S(t)$ is the slope of the concave majorant of the Brownian bridge at t. But $\int_0^1 |S(t)|dt$ is just 2 times the maximum M of the Brownian bridge, since it is obtained by integrating $S(t)$ from 0 to the location ζ of the maximum and by integrating $-S(t)$ from ζ to 1. See figure 4.3.

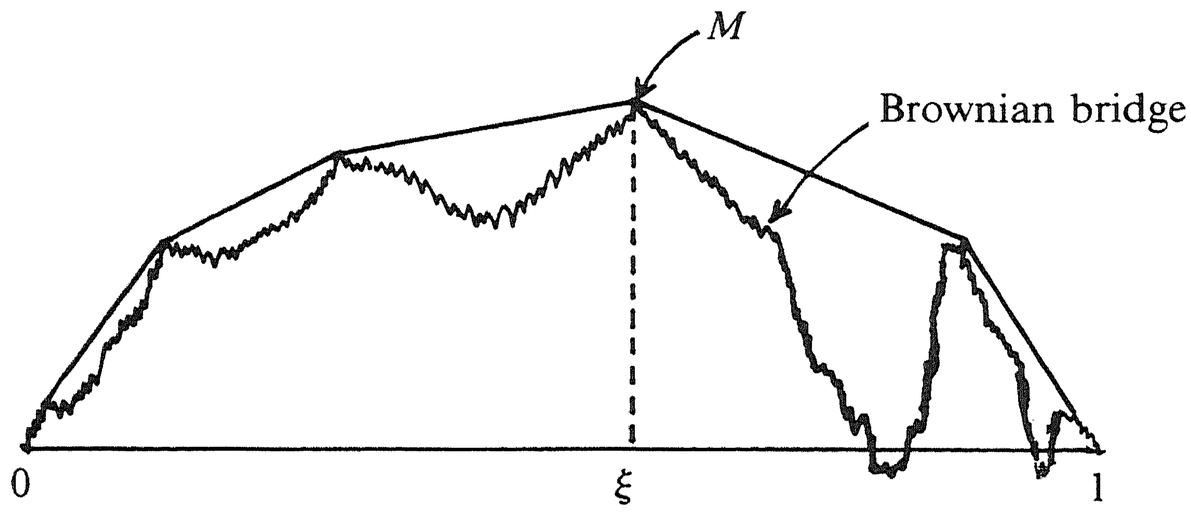


FIGURE 4.3.

(The slopes $S(t)$ will tend to $\infty(-\infty)$ as $t\downarrow 0$ $(t\uparrow 1)$, which cannot be adequately shown in the picture.) □

Since the mean (or *first moment*) of the distribution of $2\max_{t\in[0,1]}B(t)$ equals $\sqrt{\pi/2}$, we obtain relation (3.26) as a corollary of Theorem 4.1. Similarly, by using the relation between the empirical process and the Brownian bridge, one can derive Sparre Andersen's result on the number of jumps of the Grenander estimator $\hat{f}_n$ if the underlying density is uniform (using the techniques of GROENEBOOM [19]).

Theorem 4.1 is typical in the sense that the computation of the distribution of the functional $n^{1/2}\int_0^1|\hat{f}_n - f|dt$ of the empirical process $n^{1/2}(F_n - F)$ is transferred to the computation of the distribution of a corresponding functional of the Brownian bridge, but atypical in the sense that for functionals corresponding to density estimators we usually have to make a much closer (local) comparison between the behavior of the functionals for the empirical process and the Brownian bridge, using the results of KOMLÓS, MAJOR and TUSNÁDY [30] (see the paragraph preceding Theorem 4.1). Also, the uniform density is a very 'atypical' decreasing density, and the results are completely different if the density is strictly decreasing. In this case the 'risk' $E_f\int|\hat{f}_n - f|dt$ decreases at a rate $n^{-1/3}$ (instead of $n^{-1/2}$). More precise information is given in the following theorem (Theorem 3.1 in GROENEBOOM [20]).

THEOREM 4.2. *Let f be a decreasing density, concentrated on a bounded interval $[0,B]$, with a bounded second derivative, and such that $f'(t)\neq 0$, for $t\in(0,B)$. Then there exists a constant $C=C(f)$ such that the distribution of the standardized L_1-distance*

$$n^{1/6}\{n^{1/3}\int_0^B |\hat{f}_n(t)-f(t)|dt - C\} \tag{4.9}$$

converges to a Gaussian distribution with mean zero.

The precise form of the constant C and the limiting Gaussian distribution cannot be given here, and the proof of this theorem is also omitted. However, we will try to describe informally the rather striking difference in behavior of the Grenander estimator $\hat{f}_n$ under the conditions of Theorem 4.1, resp. Theorem 4.2. Figure 4.4 shows a picture of the Grenander estimator $\hat{f}_n$ based on a sample of 1000 observations from the density f on [0,1], defined by

$$f(t) = 3(1-t)^2, \quad t\in[0,1], \tag{4.10}$$

which satisfies the conditions of Theorem 4.2.

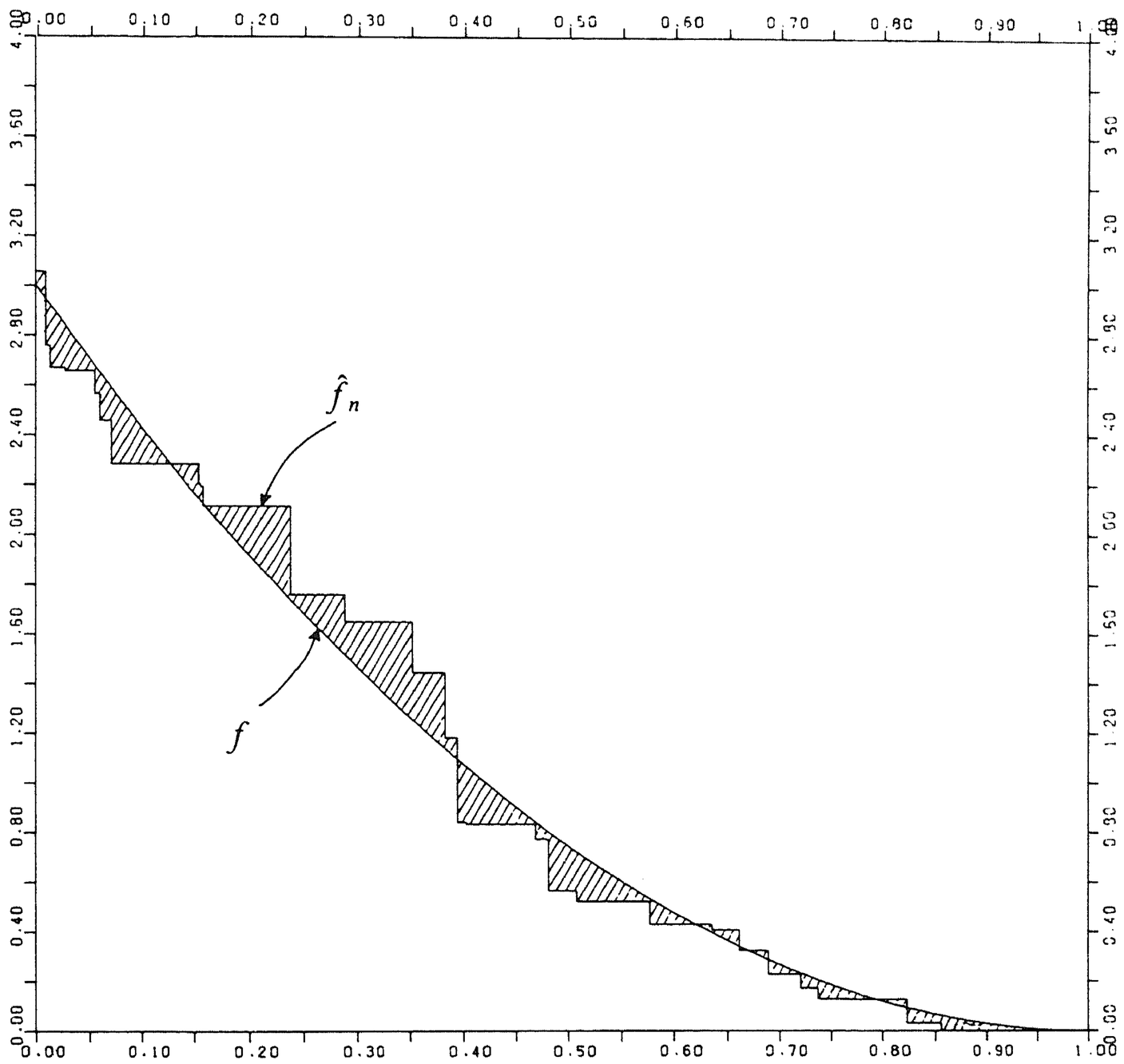


FIGURE 4.4. Grenander estimator, $n = 100$, $f(t)=3(1-t)^2$, $t\in[0,1]$

In this case the number of jumps is of order $n^{1/3}$ (instead of order $\log n$, as in the case of the uniform density), and it can be shown that the number of jumps of $\hat{f}_n$ in an arbitrary interval $(c,d) \subset (0,1)$ will tend to infinity with probability one, as $n \to \infty$. In contrast to this, the number of jumps of $\hat{f}_n$ in each interval $(\epsilon, 1-\epsilon)$, $\epsilon > 0$, will remain *bounded* (in probability) as $n \to \infty$, if the underlying density f is uniform, and in this case the only cluster points will be 0 and 1 (for a picture, see figure 4.2). In the case of the density f, defined by (4.10), the curvature of the distribution function $F(t) = \int_0^t f(u)du, t \in [0,1]$, forces the concave majorant $\hat{F}_n$ of the empirical distribution function F_n to have many changes of direction, and as $n \to \infty$, the distributions of the derivatives $S_n(t)$ and $S_n(u)$ at two different points t and u of the interval (0,1) will become more and more independent. For the uniform density, there will be dependence over the whole interval, even as $n \to \infty$.

Thus the Grenander estimator 'adapts' itself to the curvature of the underlying distribution whereas the usual kernel estimators don't have this property. This explains the better behavior of the Grenander estimator. Recently, there have been attempts to make the kernel estimators more 'adaptive' (see HABBEMA et al. [22], DUIN [16], BREIMAN et al. [8], CHOW et al. [12], HALL [23]). For example, with the (kernel) density estimators proposed by HABBEMA et al. [22] the window size is determined 'adaptively', according to a criterion applied on the data set (the method of 'cross-validation'). However, it seems clear that none of these adaptive kernel estimators can detect jumps of a density, whereas the Grenander estimator actually adapts itself both to jumps and to flat parts of a density. Also, the foregoing considerations apply to a much more general situation than the estimation of a monotone density, since, essentially, the discussed properties were based on *local* monotonicity of the density. So, although in the case of the estimation of nonmonotone densities the Grenander estimator would no longer be applicable, we still are dealing *locally* with the random process on which the Grenander estimator is based *globally* in the case of a decreasing density. This process is a jump process of locations of maxima of Brownian motion with respect to a family of parabolas (the shape of which is determined by the underlying density; the structure of this process is determined in Section 4 of GROENEBOOM [21]). We will discuss the relevance of this process for the estimation of densities and distribution functions in a forthcoming paper.

ACKNOWLEDGMENT

I want to thank LUCIEN BIRGÉ for inspiring conversations on the subject matter of the present paper.

REFERENCES

1. P. ASSOUAD (1982). *Classes de Vapnik-Çervonenkis et Vitesse D'estimation,* Preprint Université Paris à Orsay.
2. T. BEDNARSKI (1982). Binary experiments, minimax tests and 2-alternating capacities. *Ann. Statist. 10,* 226-232.

3. P. BILLINGSLEY (1968). *Weak Convergence of Probability Measures,* Wiley, New York.
4. L. BIRGÉ (1980). *Thèse, 3e partie,* Université Paris VII.
5. L. BIRGÉ (1983). Approximation dans les espaces métriques et théorie de l'estimation. *Z. Wahrsch. Verw. Gebiete 65,* 181-237.
6. L. BIRGÉ (1983). *On Estimating a Density Using Hellinger Distance and Some Other Strange Facts,* MSRI-Preprint 45-83, Berkeley.
7. L. BIRGÉ (1983). *Estimating a Density under Order Restrictions. Non-asymptotic minimax risk,* Preprint Université Paris X - Nanterre.
8. L. BREIMAN, W. MEISEL, E. PURCELL (1977). Variable kernel estimates of multivariate densities. *Technometrics 19,* 119-137.
9. J. BRETAGNOLLE, C. HUBER (1979). Estimation des densités: risque minimax. *Z. Wahrsch. Verw. Gebiete 47,* 119-137.
10. G. CHOQUET (1953-1954). Theory of capacities. *Ann. Inst. Fourier 5,* 131-292.
11. G. CHOQUET (1959). Forme abstraite du théorème de capacibilité. *Ann. Inst. Fourier 9,* 83-89.
12. Y.S. CHOW, S. GEMAN, L.D. WU (1983). Consistent cross-validated density estimation. *Ann. Statist. 11,* 25-38.
13. P.J. DAVIS, R. HERSH (1981). *The Mathematical Experience,* Birkhäuser, Boston.
14. L. DEVROYE, L. GYÖRfi (1985). *Nonparametric Density Estimation, the L_1 View,* Wiley, New York.
15. J.L. DOOB (1949). Heuristic approach to the Kolmogorov-Smirnov theorems. *Ann. Math. Statist. 20,* 393-403.
16. R.P.W. DUIN (1976). On the choice of smoothing parameters for Parzen estimators of probability density functions. *IEEE Trans. Comput. C-25,* 1175-1179.
17. W.F. FELLER (1968). *An Introduction to Probability Theory and its Applications, Vol. I (3rd ed.),* Wiley, New York.
18. U. GRENANDER (1956). On the theory of mortality measurement, Part II. *Skand. Akt. 39,* 125-153.
19. P. GROENEBOOM (1983). The concave majorant of Brownian motion. *Ann. Probab. 11,* 1016-1027.
20. P. GROENEBOOM (1984). *Estimating a Monotone Density,* Report MS-R8403, Centre for Mathematics and Computer Science, Amsterdam. To appear in: LE CAM (ed.). *Proceedings of the Neyman-Kiefer Conference,* Berkeley, June-July 1983.
21. P. GROENEBOOM (1984). *Brownian Motion with a Parabolic Drift and Airy Functions,* Report MS-R8413, Centre for Mathematics and Computer Science, Amsterdam.
22. J.D.F. HABBEMA, J. HERMANS, K. VAN DEN BROEK (1974). A stepwise discriminant analysis program using density estimation. G. Bruckmann (ed.). *Compstat 1974,* 101-110, Vienna, Physica Verlag.
23. P. HALL (1983). Large sample optimality of least squares cross-validation in density estimation. *Ann. Statist. 11,* 1156-1174.

24. P.J. HUBER, V. STRASSEN (1973). Minimax tests and the Neyman-Pearson lemma for capacities. *Ann. Statist. 1,* 251-263.
25. I.A. IBRAGIMOV, R.Z. HASMINSKII (1980). Estimation of a distribution density (Russian). *Zap. Nauchn. Semin. LOMI 98,* 61-85. English translation in *Journ. Sov. Math. 21,* (1983).
26. I.A. IBRAGIMOV, R.Z. HASMINSKII (1981). On nonparametric density estimates (Russian). *Zap. Nauchn. Semin. LOMI 108,* 73-89. English translation to appear in *Journ. Sov. Math.*
27. I.A. IBRAGIMOV, R.Z. HASMINSKII (1981). *Statistical Estimation, Asymptotic Theory,* Springer, Berlin.
28. K. ITÔ, H.P. MCKEAN, JR.(1974). *Diffusion Processes and their Sample Paths., 2nd. ed.,* Springer, Berlin.
29. A.N. KOLMOGOROV, V.M. TIKHOMIROV (1961). ϵ-Entropy and ϵ-capacity of sets in function spaces. *Amer. Math. Soc. Transl. (2) 17,* 277-364.
30. J. KOMLÓS, P. MAJOR, G. TUSNÁDY (1975). An approximation of partial sums of independent r.v.'s and the sample d.f. *Z. Wahrsch. Verw. Gebiete 32,* 111-131.
31. L. LE CAM (1972). Limits of experiments. *Proc. Sixth Berkeley Symp. on Math. Statist. and Probab. Vol. 1 Theory of Statistics,* 245-261, Univ. of California press, Berkeley.
32. G.G. LORENTZ (1966). Metric entropy and approximation. *Bull. Amer. Math. Society 72,* 903-937.
33. E. SPARRE ANDERSEN (1954). On the fluctuation of sums of random variables II. *Math. Scand. 2,* 195-223.

Experimental Mathematics

Michiel Hazewinkel
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Experimental mathematics in this paper is understood to mean the use of a computer for doing mathematical experiments. For instance experiments designed to get the first glimmer of an idea how to tackle a given set of problems or experiments to indicate where to look for counterexamples or to precisize conjectures. This is rapidly becoming a major area of research and may well develop into a semi-separate discipline like computational fluid dynamics or statistics.

1. Introduction

The subject of description and discussion in this note is *experimental mathematics*. With this phrase I mean — more or less — using a computer as a mathematical laboratory, in which there can be done experiments for gaining insight and intuition for understanding (mathematical) problems and which can serve to generate ideas for conjectures. Or experiments which can suggest where to find, or how to construct, a counterexample. Or experiments designed to illustrate and modify certain potential routes for proving a conjecture and calculations to test or refine certain, as yet quite vague, conjectures. In brief I intend to discuss a branch of mathematics which relates to more established mathematical thinking roughly as experimental physics to theoretical physics.

It is a simple fact of observation that computational results may — and very often do — lead to the development of new mathematics, i.e., also conceptual advances; just as observational and experimental results have always done since the time of Archimedes, both in the physical sciences and in mathematics.

Of course experimental mathematics in this sense is not purely a modern phenomenon. It is well known that Gauss did masses of calculations (examples) and derived insights from the results and e.g. the Littlewood-Richardson rule[1] in the representation theory of the symmetric groups and general linear groups was first observed empirically in 1934, [77], later proved and since has led to a minor industry in combinatorics and representation theory.

However computers have certainly added a new dimension to the enterprise of experimental mathematics, as if our mathematical laboratory suddenly obtained a new batch of instruments for measuring and exploring a new range of phenomena; also it may well be that in many fields of mathematics a natural limit for 'hand' calculations had been reached. In any case, the last 20

years or so have seen (the beginning of) a remarkable flowering of experimental mathematics often in the hands of investigators with a physical or engineering background.

My interest here, in this talk, is in experimental mathematics as a tool of discovery. That means that I shall not really talk about scientific computing in so far as that activity is aimed at obtaining numerical answers for problems which are well understood (in principle) and solved, but where actually doing all the calculations is beyond the capacities of a modern human calculator (and even of one of a number of generations ago), irrespective of how much ingenuity and talent is needed to do the job numerically. However, there is no sharp boundary between scientific computing and experimental mathematics for several reasons. It may, for instance, very well happen that a computational scheme will suggest conceptual advances (cf. subsection 3.4 below), or be so successful that a mathematical challenge arises: is this merely an unusually successful numerical trick or do we here have evidence for a previously unrecognized 'truth' about a certain mathematical or physical, or chemical, or ..., problem (cf. especially subsection 3.13 below and also the later half of note 5). Scientific computing is already a multi-billion dollar industry (with computational fluid dynamics taking care of most of the budget) and well on its way to becoming a separate mathematical discipline — much like e.g. statistics — , with a methodology and aesthetics of its own. There certainly is something like a beautiful computation, and in that aspect it becomes very close to experimental mathematics which, in my view, will also become — it probably already is — a discipline in its own right.

There is also a second reason why scientific computing, or even just the availability of enormous computing power, stimulates 'pure' mathematics. The mere existence of computing power has influence on the kinds of theoretical problems which can be considered and investigated[27]. Thus a number of research areas with a fully developed theoretical (or pure, if one wants) component, like e.g. semi-parametric statistics — I have in mind bootstrap methods and the jackknife statistic [37] —, two and more dimensional statistics (with its heavy dependence on computer graphics) [96][4], and computerized (read: applied) tomography would probably not have existed without very substantial computing power [64]. In this connection it is interesting to observe that the theoretical problem at the basis of computerized tomography, inversion of the Radon transform, was solved in 1917 [95]; as a matter of fact the formula seems to have been known (in dimension 3) to the Dutch physicist LORENTZ before 1906 cf. [21], [110], and it has been rediscovered independently a number of times[2]. A Nobel prize was given for applying — more precisely: implementing — this formula and, later, these applications generated, and still generate, whole series of new theoretical problems [64], [53].

I shall also not discus 'computer assisted proofs' such as that of the four colour problem, and I shall certainly not say anything about the philosophical implications and questions thereof [35, page 380-386].

Also, these lines are written from the point of view of a user of experimental mathematics but not a doer, and I shall concentrate on three examples where

doing experiments (of quite moderate size as such things go) resulted in new unexpected insights, sometimes concerning a topic where really nothing interesting was supposed to happen. And where the mathematical experiments gave rise to new concepts, solution methods and even whole new areas of inquiry. The examples which will be discussed briefly and anecdotically below, in general terms, and omitting virtually all hard mathematics are 'the hard hexagon model of lattice statistical mechanics', 'chaos and universality for iterated maps' and 'integrable systems and the soliton revolution'. These three examples are the topics of respectively Sections 4, 5, 6 below. Besides that a number of other examples will be briefly mentioned in Sections 3 and 7.

2. Two contrasting opinions

Here are two rather opposite opinions:

> 'As I see it, within another generation, the mainstream of mathematics will not be analysis, number theory and topology but rather numerical analysis, operations research, and statistics. ... I am not suggesting that the pure areas of mathematics or for that matter the classical topics in applied mathematics such as transform methods, partial differential equations and approximation theory, will disappear. Instead like Newtonian mechanics, they may move permanently from centre stage in mathematics departments.' J.C. Frauenthal [45]

> '... by the judicious use of computers we can penetrate into new areas and discover linkages to diverse areas of mathematics unforeseen by our forebears. With insight obtained from numerous solutions, often displayed naturally by graphs and cinemas, we may be liberated from the prejudices of our conservative and sometimes misguided mathematical intuitions.
> Almost everyone using computers has experienced instances where computational results have sparked new insights. The range covered is large: from uncovering mistakes in formal derivations or calculations to suggestions for combinations of parameters with which to make asymptotic expansions and thereby obtain equations which are analytically tractable; and finally to shining the light of inspiration into areas which have been thought devoid of possible new concepts or new fundamental truths.
> Although several pioneering steps have been taken, we are just at the beginning of a mind augmenting revolution that inexpensive and robust computing will allow the prepared investigator.' N. Zabusky [116]

This is precisely as John von Neumann expected things to develop. Speaking in 1946 he remarked [48]:

> 'The advance of analysis is at this moment stagnant along the entire front of nonlinear problems ... not transient ... we are up against an imported conceptual difficulty.'

And he was counting on the computer to remedy this situation [ibid.]:

> '... we conclude by remarking that really efficient high-speed computing devices may, in the field of nonlinear partial differential equations as well as in many other fields which are now difficult or entirely denied of access, provide us with heuristic hints which are needed in all parts of mathematics for genuine progress ... . This should lead ultimately to important analytical advances.'

Also, one should perhaps reflect that our much vaunted intuition (in mathematics) and feeling for phenomena is perhaps overrated. H. HAHN, [56], once described intuition as 'force of habit rooted in psychological inertia' [3,7], and without fresh experience to feed on, one can easily see how this might become so. If, therefore, as seems to be the case, we have indeed in a number of fields reached something of a limit in computation by hand, experimental mathematics becomes a must. Quoting HAHN, as above, ZABUSKY, loc. cit., speaks in this connection of the enriching possibilities of 'computational synergetics' and mathematical innovations, given a judious use of computer power.

Below in Sections 4, 5, 6 I shall try to describe in more detail how in a few instances experimental mathematics — theoretical and applied mathematics interactions — went, and shall try to point out the synergetic influences. These short descriptions and the section of loose quotes below should suffice to indicate which way things seem to be going.

In addition it seems worth remarking that in all three of the main examples described below there is nice mix of pure and applied mathematics (besides experimental mathematics and physics) and not much seems to remain of the supposed gap between the two. This also makes papers dealing with these topics hard to classify, a more and common phenomenon, which indicates that present day mathematics is far less tree like than would be convenient for bibliographical and information storage and retrieval purposes.

3. QUOTES

As I remarked before, and as ZABUSKY remarked in the quote above, it is a simple fact of experience that doing mathematical experiments on a computer may easily lead to sudden illuminating (true or false, but stimulating) insights. Let me try to illustrate this by quoting from the more recent scientific literature. Let me also stress that I made no especial effort to find such quotes. These are simply the ones I happened to come across since the moment, now about a year ago, when I started thinking about a lecture on experimental mathematics. There are likely to be many many more and it seems clear that the controversy indicated above was in fact already settled long before FRAUENTHAL made his remarks.

3.1. *From computational physics in general*

> 'The goals of computation .... include the discovery of new simplifying physical principles by observing the computed behaviour of the model.' D.R. HAMANN [57][5,6]

To this I would also like to add that computers enable both experimentalists and theorists to explore physical systems in a manner not previously possible (by 'real' experiments). For instance certain parameters can be pushed to unphysical values, or simply to values impossible to realize in an existing laboratory. Also this way experiments can be carried out in sciences where experiments have been said to be impossible; such as economics.

And it is well known that (new) principles often manifest themselves most clearly in some sort of limit, some sort of extreme case. As R. ISAACS remarks in his advice to young applied mathematicians [65]: if you do not understand how something will behave, take an extreme case.

3.2. *Concerning Yang-Mills gauge theories*

For a quantum field theory of strong interactions based on quarks (interacting by exchanging gluons) one wants both 'confinement' wherein an isolated quark would have infinite energy and asymptotic freedom which means that the interactions between quarks become weaker as they move closer together. This seems hard to do, and maybe even counterintuitive. However out of Monte Carlo simulations for studying solutions to interacting quantum fields there came:

> 'The main result is that we now have rather compelling numerical evidence that this theory [Yang-Mills gauge theory] can simultaniously give rise to the phenomena of quark confinement ... and asymptotic freedom ....' M. CREUTZ [32]

3.3. *On food webs*

A food web is a schematic diagram showing the (who eats whom) relationships among species in a community of plants and animals. Omnivores are animals consuming prey from two or more trophic levels. In simulated webs with Lotka-Volterra interactions between species long food chains lead to severe population fluctuations that are inconsistent with long-term persistance. Also numerical studies of the dynamical stability of model webs with Lotka-Volterra interactions predict that the number of omnivores in a real food web is significantly lower than would be found if the connections within the web were made at random. This last fact turned out to be the case, and the first one goes a way towards explaining that in real food webs species tend to interact directly only with a handful, four or five or so, of other species regardless of the size of the ecological community. Sources for these remarks are [93] and [83].

3.4. From computational fluid dynamics (CFD)
Computational fluid dynamics (CFD) is the process of solving problems in fluid dynamics (including aerodynamics) on a computer. That is they basically deal with one particular set of partial differential equations, the Navier-Stokes equations. In spite of that, this is a multi-billion dollar industry which mostly belongs to scientific computing and which is rapidly turning into a discipline of its own (besides applied mathematics, statistics, pure mathematics, experimental mathematics, ...) with its own aesthetics and paradigms. It has however very definite and interesting relations with all three of pure, applied and experimental mathematics. For example:

> 'Some mathematical and CFD developments go hand in hand: Lax's theories of hyperbolic conservation laws and of differencing in conservation form (see [74], [75]) are parts of a single picture.
> A recent example is provided by Glimm's existence proof for non-linear hyperbolic equations [47], which was loosely suggested by Godunov's computing scheme and has in turn given rise to new algorithms (see [29]).' A.J. CHORIN [28]

3.5. On glassy solids and quench echos

> 'When we try to understand atom motion in amorphous solids we face a complicated problem in classical mechanics. ... . Without a periodic crystal lattice to simplify the calculations, we must look for other properties that make things tractable. A phenomenon recently observed in computer models of many-body systems gives us such a simplification, at least in the calculation of a number of properties of glassy solids.
> In spite of their seemingly random motion, atoms in computer-simulated glasses 'remember' the time interval between a pair of freezings, simplifying certain many-body calculations.' S.R. NAGEL a.o. [89]

3.6. From geology
One use of simulation or computer modeling is to find out whether certain accepted axioms of dogmas are indeed tenable. Just as mathematics has often been concerned with the question of whether a certain set of axioms is compatible. Cf. also note 7. From palaeo geomagnetics we have e.g.:

> 'Computer models, designed to synthesize palaeosecular variations of the geomagnetic field, cast doubt on some widely accepted palaeo magnetic dogmas.' K.M. CREER [31]

3.7. A chaotic quote

Chaos, in the setting of iterated maps of an interval into itself, will be briefly discussed in Section 5 below. Period doubling bifurcations play an important role there. From thermosolutal convection (convection in the presence of a stabilizing concentration of a solute):

> 'Numerical experiments on two-dimensional convection reveal a transition from periodic oscillations through a sequence of period-doubling bifurcations. ... . This is the first example of period-doubling in solutions of partial differential equations.' D.R. MOORE a.o. [87]

3.8. From catalytic chemistry

The properties of single atoms (from a chemical point of view) have been known for a long time and also those of bulk substances, but not those of clusters of say 2-200 atoms. Especially in connection with catalysis.

> 'Some preliminary computational studies and complementary model experiments, ..., suggested that some really exciting chemistry could exist in this domain and provided a strong incentive to learn how to make the clusters.' TH.H. MAUGH II [82]

3.9. Re-phase transitions and the van der Waals picture of liquids

> 'A remarkable revival of the van der Waals picture of liquids occurred during the last two decades. This renaissance was spurred by the discovery [1], [2], [115] from computer simulations that a system of hard spheres (impenetrable 'billiard balls') has a first order fluid-solid transition that is intimatedly related to the freezing and melting transitions of real materials ... .' D. CHANDLER a.o. [27]

3.10. Re-planet formation

One possible model for the formation of the planets of our solar systems involves the idea of lots of small pieces which when they collide may under the right conditions adhese to one another. This idea was computer-simulation tested by G.W. WETHERILL with spectacular results as the pictures below will testify[36]. The first picture refers to the initial state with a hundred planetesimals, the second depicts the situation after a long time interval with about 20 'small planets' and the third depicts the result a really long time later with just five planets left.

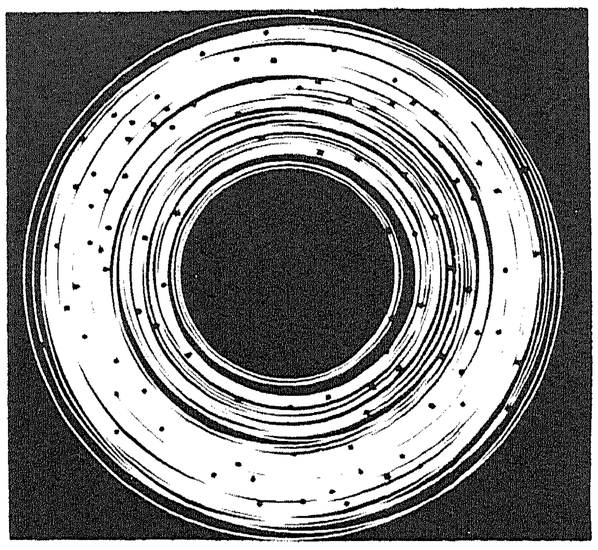


FIGURE 1, from [112]

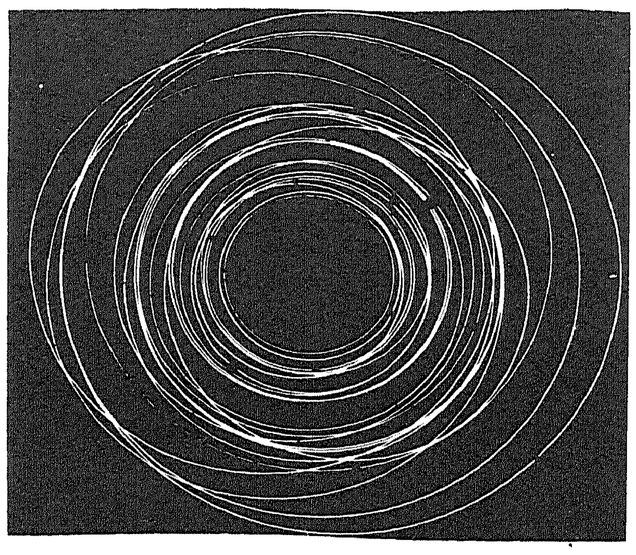


FIGURE 2, from [112]

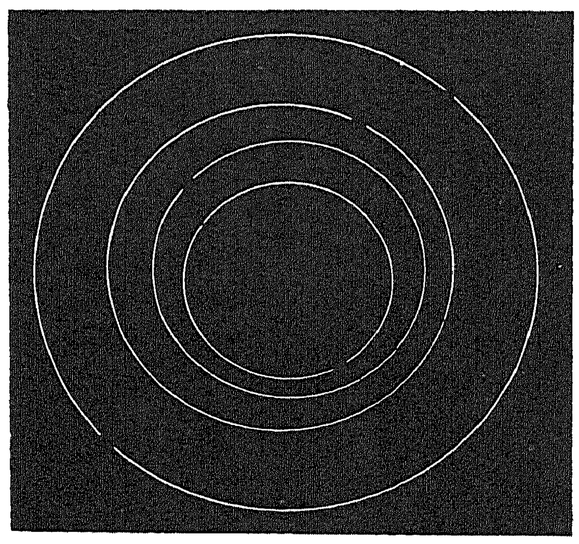


FIGURE 3, from [112]

3.11. From cosmology

> 'Take a mixture of gas and dust, cook it appropriately with the aid of a large computer and a galaxy may emerge. That, at least, is the dream of astronomers who study the most remote galaxies.' J. SILK [103]

Besides that it has become clear that the universe contains very large, indeed unusually, large voids; it is not at all homogeneous with galaxies or clusters of galaxies, or superclusters randomly distributed. Instead it is very clumpy. It thus becomes interesting to test whether various candidate cosmogenies predict (or admit) such clumpiness. Computer studies concerning this have indeed been carried out and some of these are reported on in a beautiful report [50] in the National Geographic Magazine. An artist's impression of the resulting filimentary structure (caused by clumping of neutrinos) is shown in figure 4[8].

FIGURE 4, from [50]

3.12. From fluid dynamics (out of equilibrium)

'Progress (in fluid dynamics) through the years has been uncertain however, with periods of success amid long periods of frustration and fragmentation of effort. But today we are in an upswing. In particular it seems that we may be close to understanding quantitatively why a fluid out of equilibrium can behave as it does - long an intractable problem. Two tools especially have contributed: the laser and computer simulation. These tools, the one experimental, the other theoretical, yield unambiguous results that allow one to test theories (some of which were proposed long ago) and that suggest paths for father study.' H.J.M. HANLEY 1984. *Physics Today,* p. 25

'Computer simulations indicate that simple liquids can display a surprising range of exotic nonequilibrium phenomena, more commonly seen in systems of macromolecules.' D.J. EVANS a.o. [38]

'However computers are prompting important changes within mechanics itself ... . We will see that the effort to model real systems forces us to pay close attention to constraints, in particular,

> to nonholonomic constraints, which we do not often encounter in textbook problems in classical mechanics'. W.G. HOOVER [62]

3.13. From quantum field theory

Finite element methods are well known in partial differential equations. Basically one selects a number of functions (often monomials in the variables) and attempts to the PDE by taking linear combinations of these functions. To this end divide the region into nonoverlapping patches, impose the PDE at one point in every patch and impose conditions of matching (with the functions on a neighboring patch or boundary conditions, as the case may be) at the boundaries of each patch. This gives algebraic conditions for the coefficients.

In principle one can also take operator valued coefficients and try to do similar things for the equations of quantum field theory, by using say a lattice. There arises the extra difficulty of seeing to it that the equal time commutation relations hold (at all times). This turns out to be possible [17], [18]. In other words the resulting operator difference equations preserve equal-time commutation relations. When the same idea is applied to a free fermion theory it turns out that the resulting difference equations are consistent with equal-time anti commutation relations, and other nice properties, and, quite surprisingly it turns out that the oft-encountered problem of so-called fermion doubling is avoided. This last fact was a totally unexpected bonus and is remarkable in that there are general theoretical results [69] showing that fermion doubling when taking lattice approximation is difficult to avoid. As CARL BENDER recently remarked in a telephone conversation with me:

> 'It is as if Nature intended us to use finite element methods' C.M. BENDER Dec. 1983 [9,10]

4. THE HARD HEXAGON MODEL OF LATTICE STATISTICAL MECHANICS[13]

In lattice statistical mechanics models one works with a lattice in d-space for example a square lattice in two space as depicted in figure 5. Atoms are supposed to be located at some or all of the sites. Each atom can be in several states. To each configuration c there is assigned an energy $E(c)$. For a large chunk of N sites of the lattice now write down the so-called partition function

$$Z_N = \sum_c \exp(-E(c)/kT) \tag{4.1}$$

(where k stands for the Boltzmann constant and T for the temperature. This is *the* basic object of statistical mechanics and from it one calculates various thermodynamically interesting quantities such as the free energy $F = -kT\ln Z_N$, the probability of the system being in state c, the free energy per site in the large N limit $f(t) = -kT \lim_{N\to\infty} N^{-1} \ln Z_N(T)$ (one expects this limit to exist), the internal energy per site $u(T) = -T^2 \frac{\partial}{\partial T}(T^{-1} f(T))$, the specific heat per site $c(T) = \frac{\partial}{\partial T} u(T)$, ... (also all kinds of average and

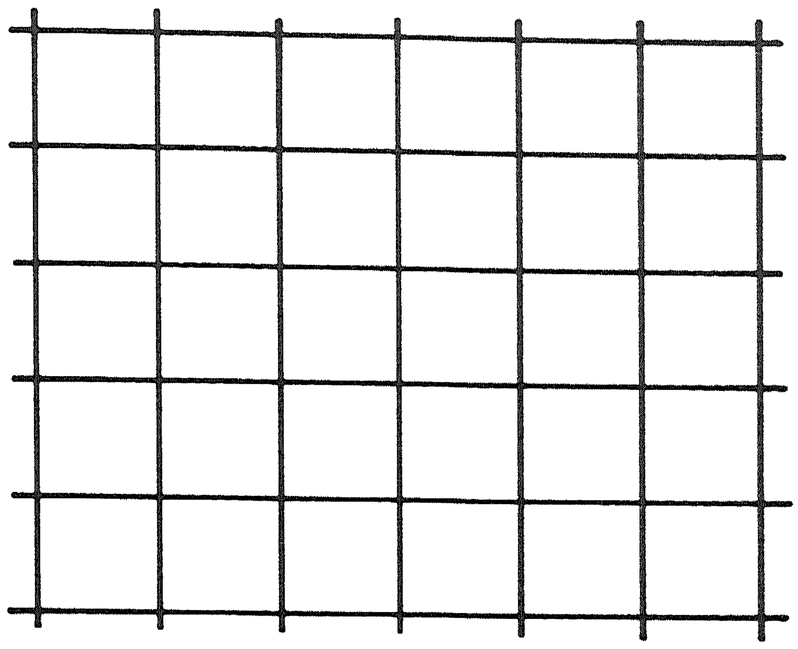


FIGURE 5

expected values, such as correlations), the partition function per site

$$\kappa = \kappa(T) = \lim_{N\to\infty} Z(T)^{1/N} \tag{4.2}$$

and one is in particular interested in finding out whether these functions $f(T)$, $u(T)$, $c(T)$, ... have singularities at certain values of T (phase transitions). For instance for the square lattice depicted above one could be interested in the model where all sites all are occupied with an atom at each site i with spin either up ($\sigma_i = 1$) or down ($\sigma_i = -1$) and nearest-neighbour-only interaction resulting in an energy function (Hamiltonian)

$$E(\sigma) = -J\sum_{(i,j)}\sigma_i\sigma_j + K\sum_i \sigma_i \tag{4.3}$$

where the first sum is over all pairs of adjacent sites (i,j) and the second one over all sites i. This is the well known nearest neighbour Ising model, and is not the subject of this section. In the case of the hard hexagon model one considers a triangular lattice as shown in figure 6. The possible states at each site are 1 (atom present) and 0 (empty). The energy function (Hamiltonian) is such that the partition function takes the form of the generating function

$$Z(z,N) = \sum_p g(p,N)z^p = 1 + Nz + \frac{N(N-7)}{2}z^2 + \ldots \tag{4.4}$$

where $g(p,N)$ is the number of ways in which p atoms can be distributed over the lattice of N sites such that no two coincide and no two neighbouring sites are occupied. Thus if a given site is occupied a whole hexagon of sites is forbidden (cf. figure 6), as if we were dealing with a gas of impenetrable hexagonal atoms. Whence the name hard hexagon model [12].

The parameter z in (4.4) has much to do with T in (4.1) and plays the same role. It is called the activity.

Now if there are only a few atoms, say 1, each site has equal probability of

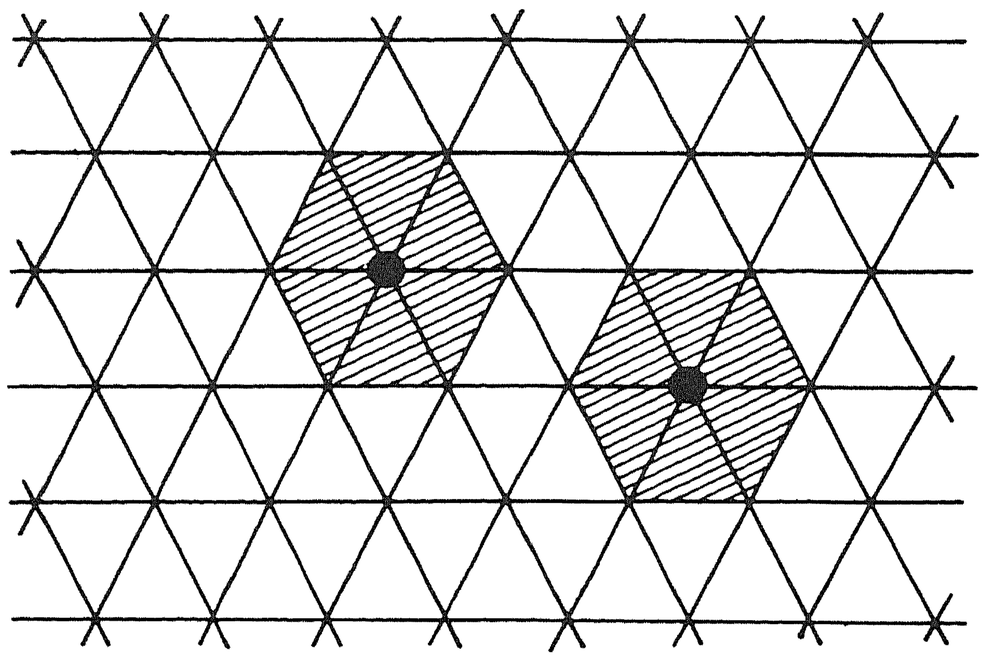


FIGURE 6

being occupied. So for small z one expects the full triangular symmetry to be present. There are three ways of packing very large densities of atoms on the triangular lattice (cf. figure 7): either all the rectangular sites are occupied (and no others), or all the circular ones or all the triangular ones. There is loss of symmetry, indicating a phase transition, which is of course just the sort of thing one is looking for when constructing such models. Let $\rho_S =$ density on square sites, $\rho_T =$ density on triangular sites and $\rho_C =$ density on circular sites. Suppose that as z increases the square sites are preferred, then $\rho_S \to 1$, $\rho_T \to 0$, $\rho_C \to 0$ as $z \to \infty$ and if $R = \rho_S - \rho_T$, say, the graph of R as a function of z would look something like in figure 8. I.e. there must be a critical point z_c where R first becomes nonzero. By various numerical calculations (maximum eigenvalue estimates, series expansions in z and z^{-1}) estimates for z_c can be obtained. One such by J. GAUNT in 1967 gave $z_c = 11.05 \pm 0.15$. There is also a nonphysical critical point z_n for which GAUNT obtained $z_n = -0.0900 \pm 0.0003$. If one is in an experimental mood one can calculate sum and product of z_c and z_n to find $z_c + z_n = 10.96 \pm 0.15$, $z_c z_n = -0.995 \pm 0.014$ and observe that these are practically integers. This would result in

$$z_c = \tfrac{1}{2}(11 + 5\sqrt{5}) = [\tfrac{1}{2}(1 + \sqrt{5})]^5 \tag{4.5}$$

All this was observed by GAUNT but he did not include the conjecture (4.5) in his paper. Other calculations resulted in a value for $\kappa(1)$ (cf. (4.2) above) of $\ln \kappa(1) = 0.3333 \pm 0.0001$ by METCALF and YANG in 1978 and they did publish the conjecture that $\ln \kappa(1) = 1/3$.

Around this time RODNEY J. BAXTER, Canberra, Australia, decided to take up the challenge, convinced that he had devised a class of methods which would yield far more precise numerical results. This method is based on so called transfer matrices, in this case corner transfer matrices, and it results in

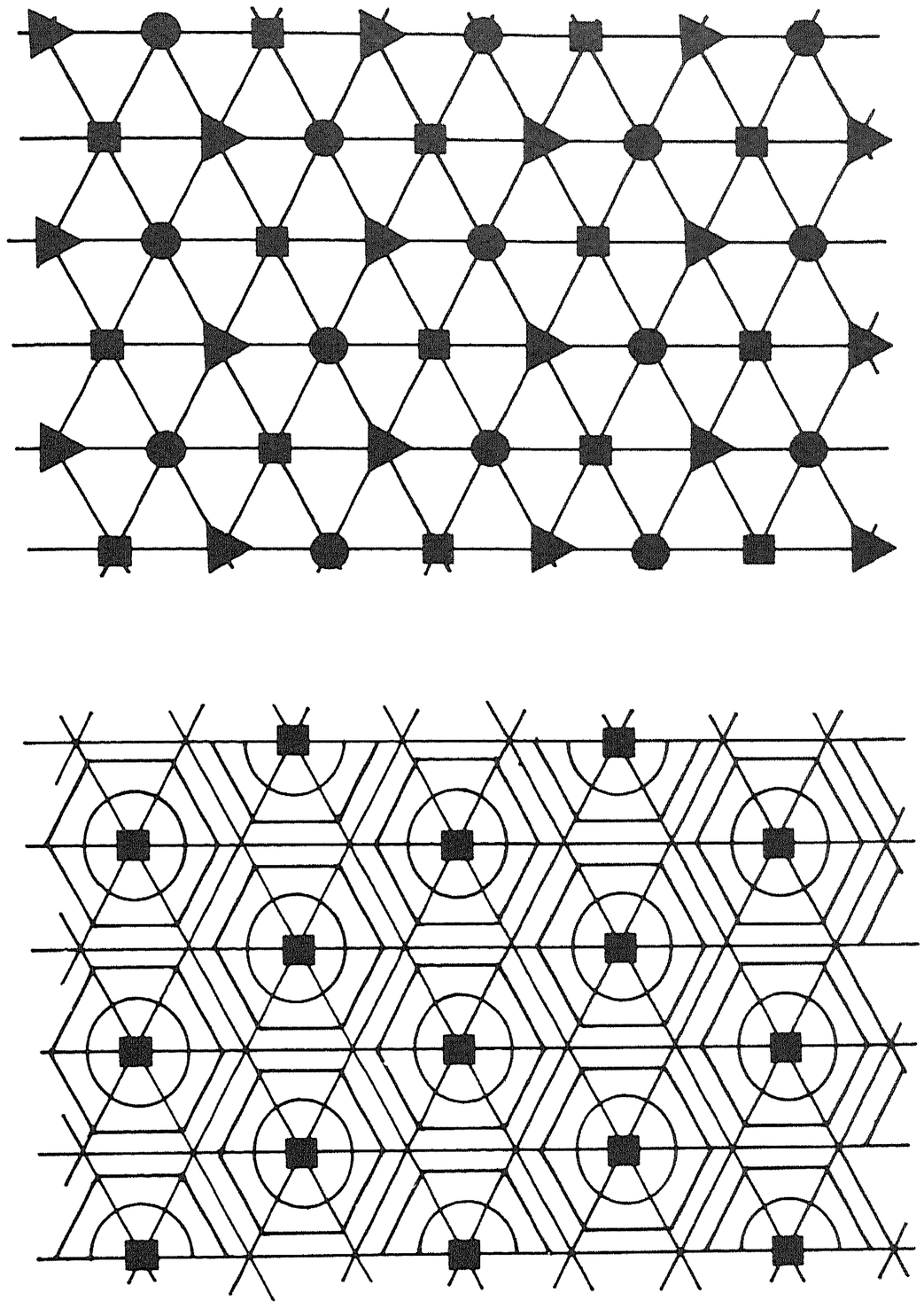


FIGURE 7

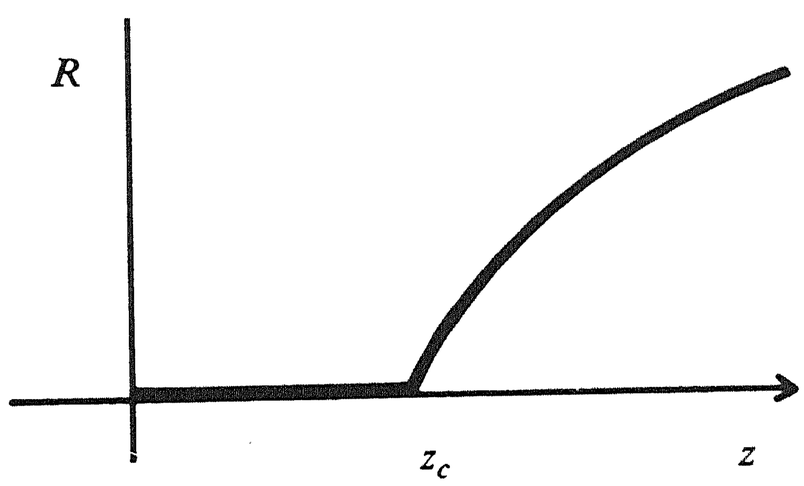


FIGURE 8

the partition function $Z(z,N)$ being written as a trace of the sixth power of an (in principle infinite) matrix

$$Z = Trace\ A^6 \tag{4.6}$$

The power 6 here is important from the numerical point of view, leading, with a bit of luck, to rapid convergence of the series expansion

$$Z = \lambda_1^6 + \lambda_2^6 + \lambda_3^6 + \ldots$$

where $\lambda_1, \lambda_2, \lambda_3, \ldots$ are the eigenvalues of A in descending magnitude. How to actually calculate $\lambda_1, \lambda_2, \ldots$ requires more clever ideas (cf. [13]), but some of the results are given in table 1.

TABLE 1

approximating matrix size	ln $\kappa(1)$	error
2×2	0.333 050	1.9×10^{-4}
3×3	0.333 242 657	6.5×10^{-8}
5×5	0.333 242 721 958	1.8×10^{-11}
7×7	0.333 242 721 976 1	4.7×10^{-15}

so that, obviously, $\ln \kappa(1)$ is not 1/3.

Of course the λ_i are functions of z, and knowing a small z expansion of $Z(z)$ and $A(z)$ one can write down the leading terms of λ_i in the small z expansion

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}
1	1	$-z$	z^2	z^2	$-z^3$	$-z^3$	z^4	z^4	z^4

and test various monomials in the λ_i suggested by these leading terms in a search for some kind of regularity. BAXTER did just that, and found (for the 7×7 approximation):

$\lambda_4\lambda_3^{-2}$	0.999 999 853
$\lambda_5\lambda_2^{-1}\lambda_3^{-2}$	0.999 999 539
$\lambda_6\lambda_3^{-3}$	0.999 757 797
$\lambda_7\lambda_2^{-1}\lambda_3^{-3}$	0.999 730 684

Thus it seemed that $\lambda_j = \lambda_2^s x^{n_j}$, $s \in \{0,1\}$, $x = \lambda_3$. Now BAXTER had encountered some such situation before. Namely when he solved the eight vertex model, and in that case theta functions and elliptic functions had played a fundamental role. So he programmed the computer to calculate the exponents in

a product expansion

$$z = -x\prod_{n=1}^{\infty}(1-x^n)^{c_n}$$

(one of the sorts of thing one naturally thinks of if one has theta functions in mind) and BAXTER found 5,-5,-5,5,0, 5,-5,-5,5,0, 5,-5,-5,5,0, 5,-5,-5,5,0, 5,-5,-5,5,0, 5,-5,-5,5,0, ... A most stimulating result. This then provided the starting point for solving the hard hexagon model exactly [14] including that indeed $z_c = \frac{1}{2}(11+5\sqrt{5})$.

The story does not stop here. Far from it. BAXTER found that he could make good use of certain (formal) identities of the type

$$\sum_{n=0}^{\infty}\frac{q^{n^2}}{(1-q)(1-q^2)\ldots(1-q^n)} = \prod_{n=1}^{\infty}\frac{1}{(1-q^{5n-4})(1-q^{5n-1})}$$

known as Rogers-Ramanujan identities [10], [11], [12]. These 'belong' to the world of theta functions. More precisely it turned out that there are four regimes for the generalized hard hexagon model. For three of these the identities that BAXTER found and could use turned out to be known. For the fourth one he could conjecture (and verify to degree 80) one which turned out to be new, again using computer support. This one was shortly after proved by G.E. ANDREWS [5].

Still the story is not finished. One can consider the 'decorated hard hexagon' model in which instead of two possible states 0 and 1 one has k possible states at each site. This has also been considered by BAXTER and ANDREWS and turns out to involve generalized Roger-Ramanujan type identities in which the magic number 5 is replaced by $2k+1$. And things go on ... .

All in all there now is a flourishing interdisciplinary area of research between combinatorics and lattice statistical mechanics which arose to a large extent from Baxter's work on the hard-hexagon model [11,14].

5. CHAOS AND UNIVERSALITY FOR ITERATED MAPS OF AN INTERVAL INTO 5. ITSELF

We are interested in a map of an interval into itself. For instance

$$\begin{aligned} f_\mu(x) &= 1-\mu x^2\,, \quad [0,1]\to[0,1] \\ f_\mu(x) &= \mu x[1-x]\,, \quad [0,1]\to[0,1] \\ f_\mu(x) &= \mu\sin\pi x\,, \quad [-1,1]\to[-1,1] \end{aligned} \tag{5.1}$$

And we are especially interested in what happens if the mapping is iterated a large number of times and how this 'limit behaviour' changes as the parameter μ changes.

Before I say anything about the phenomenology let me quote something from [73] about the history of the topic:

> 'The methods used to study smooth transformations of intervals are by and large elementary and the theory could have been

> developed long ago *if anyone had suspected that there was anything worth studying.* In actual fact, the main phenomena were discovered through numerical experimentation and the theory has been developed to account for the observations. In this respect, computers have played a crucial role in its development.' O.E. LANFORD [73]

A quote which certainly supports the point of view of VON NEUMAN and ZABUSKY rather than that of FRAUENTHAL[16].

Here is something of the phenomenology observed. For small μ ($\mu<0.75$ for the second of the maps of (5.1)), there is a unique attracting point x_0; that is for almost all x (in fact all x except $x=0$) the sequence

$$x, \quad f_\mu(x), \quad f_\mu^{(2)}(x) = f_\mu(f_\mu(x)), \quad f_\mu^{(3)}(x), \ \ldots$$

converges to x_0. Then as μ becomes larger x_0 splits into an attracting orbit of period 2, that is there are two points, x_1 and x_2, say, such that $f_\mu(x_1)=x_2$, $f_\mu(x_2)=x_1$ and for almost all x, $f_\mu^{(n)}(x)$ comes arbitrarily close to x_1 or x_2 and hops back and forth between the two with each new iteration. For still larger μ (at $\mu_2=1.25...$) an attracting orbit of period 4 appears which in turn splits into one of period 8 at $\mu_3=1.368...$ etc. It turns out (numerically) that these μ_n have a limit and that

$$\mu_\infty - \mu_0 \sim const.(4.6692...)^{-n} \quad \text{as} \quad n\to\infty \tag{5.2}$$

This number 4.6692... now turns out to be a universal constant meaning that the same constant appears for all kinds of different maps, a numerical discovery of M. FEIGENBAUM [40] and COULLET-TRESSER [30]. It is now often known as the Feigenbaum number. There is more. If one plots the position of the attractors of period 1, 2, 4, 8, 16, ... as they are about the fission one obtains something like the following picture (figure 9).

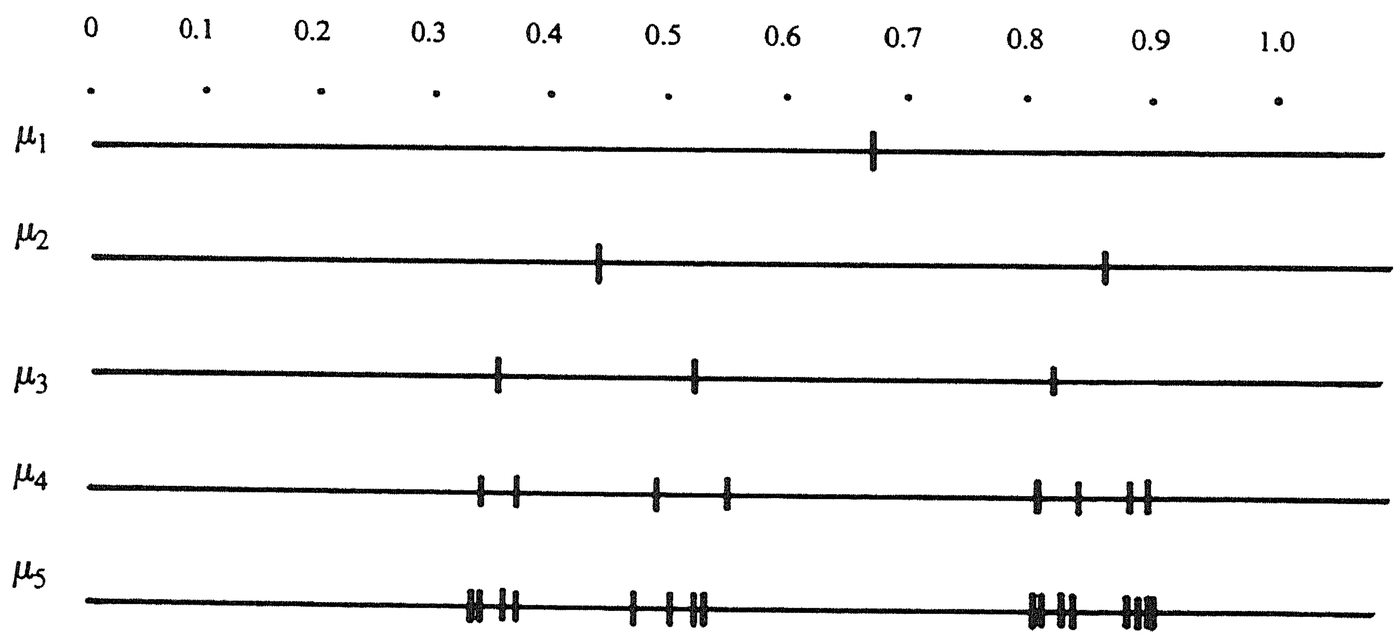


FIGURE 9

One observes that the left half of each line is the mirror image of the line immediately above scaled down by a factor of about 2.5. The precise factor (in the limit) turns out to be $a=2.5092078$ .... and again it turns out to be a universal constant.

These numerical observations or discoveries of course simply cry out for an explanation and great progress has been made in the theoretical understanding of why such things happen. Mathematically the clue lies in the consideration of the nonlinear mapping (of functions into functions) $T:f\to a^{-1}(f\circ f)(ax)$ and to search for a scaling constant a for which this mapping has a fixed point. The fixed point is hyperbolic with one eigenvalue (of its linearization at the fixed point) greater than 1. This eigenvalue is equal to 4.6692... .

There also remain lots of open questions. For instance there is very little known of the solvability of functional equations like $f(x)=a^{-1}(f\circ f)(x)$ and of the properties of the solutions. E.g. does there exist a smooth solution? Another open question concerns the order in which various periodic orbits appear as μ increases. Omitting the periods of order $\geqslant 8$ this sequence is

$$1,2,4,6,7,5,7,3,6,7,5,7,6,7,4,7,6,7,5,7,6,7$$

and it also appears to be of a universal nature [86]. This is as yet unexplained and ununderstood.

As μ_n reaches its limiting value 1,401... and goes past it the motion of a point becomes chaotic[17] meaning that it is virtually impossible to predict the position of the n-th iterate $f^{(n)}(x)$ for a starting point x; in other words small differences in starting position rapidly (exponentially fast) become very large differences in the higher iterates. For still larger μ a measure of more ordered motion may reappear etc... We are also as yet quite far from understanding this pattern of reappearance and disappearance of more ordered motion.

Turning to the more dimensional case, there also appear to be universality phenomena for both conservative and dissipative mappings of pieces of planes into themselves which still are ununderstood and provide a fruitful hunting ground for experimental mathematicians [23], [41], [51], [91], [113].

Deterministic chaos theory has become a thriving business [18] and has made significant contact with other areas of investigation such as scaling and renormalization (group) theory in physics and theories of turbulence in fluid dynamics.

6. Integrable systems and the soliton revolution

Probably the first mathematical experiment on a computer was done in Los Alamos, at the time that the MANIAC, the Los Alamos copy of the Princeton Von Neumann machine, was barely finished. Fermi, Ulam and Pasta had deliberatedly selected a problem for which the machine would be much more suitable than a human calculator. Here is Stan Ulam on the topic in his autobiography [111].

> 'As soon as the machines were finished Fermi, with his great common sense and intuition, recognized immediately their importance

> for the study of problems in theoretical physics, astrophysics, and classical physics. We discussed this at length and decided to attempt to formulate a problem simple to state, but such that a solution would require a lengthy computation which could not be done with pencil and paper or with the existing mechanical computers. After deliberating about possible problems we found a typical one requiring long-range prediction and long-time behaviour of a dynamical system. It was the consideration of an elastic string with two fixed ends, subject not only to the usual elastic force proportional to strain, but having, in addition, a physically correct small nonlinear term. The question was to find out how this nonlinearity after very many periods of vibrations would gradually alter the well-known periodic behaviour of back and forth oscillation in one mode; how other modes of the string would become more important; and how, we thought, the entire motion would eventually thermalize, imitating perhaps the behaviour of fluids which are initially laminar and become more and more turbulent and convert their macroscopic motion into heat. ...
> Our problem turned out to have been felicitously chosen[37]. The results were entirely different qualitatively from what even FERMI, with his great knowledge of wave motion, had expected. The original objective had been to see at what rate the energy of the string, initially put into a single sine wave (the note was struck as one tone), would gradually develop higher tones with the harmonics, and how the shape would finally become a 'mess' both in the form of the string and in the way the energy was distributed among higher and higher modes. Nothing of the sort happened. To our surprise the string started playing a game of musical chairs only between several low notes, and perhaps even more amazingly, after what would have been several hundred ordinary up and down vibrations, it came back almost exactly to its original sinusoidal shape. ...
> Another Los Alamos physicist, JIM TUCK, was curious to see if after this near return to the original position, another period started again from this condition and what it would be after a second 'period'. With PASTA and METROPOLIS, he tried it again and, surprisingly, the thing came back, a percent or so less exactly. These continued and, after six or twelve such periods, it started improving again and a sort of superperiod appeared. Again this is most peculiar.' S. ULAM [111]

Here is a picture of the sort of thing which went on (figure 10). This of course demanded an explanation. It was almost as if there were certain entities which were stable in time and for which some sort of superposition principle would hold.

These entities were found, they are the so-called solutions, a term coined by

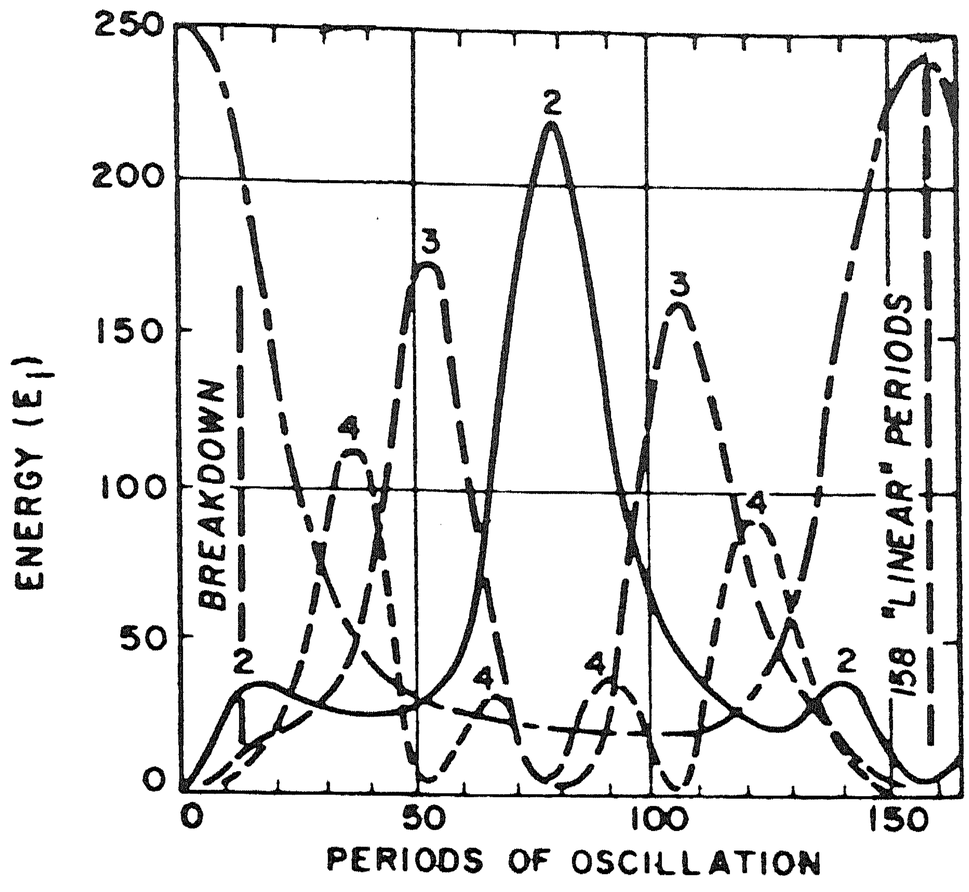


FIGURE 10, from E. Fermi, J.R. Pasta, S.M. Ulam
Los Alamos Report LA-1940, Mag 1955

KRUSKAL, MIURA, GARDNER, GREENE, ZABUSKY to describe a solitary travelling wave which retains its shape while travelling and with the remarkable stability property that when it encounters another soliton both emerge intact from a temporary messy interference pattern (apart from a phase change). A picture illustrating this behaviour of solitons is figure 11. For a lengthy and most thorough account of how the concept of solitons developed initially in the hands of the five persons just named and of how the computer or more precisely mathematical experiments with the help of a computer continued to play an important role cf. the review paper [116] by one of those deeply involved, N. ZABUSKY. Such was the start of the soliton revolution and out of it there came the so-called 'inverse spectral transform' method of solving a number of nonlinear equations such as the Korteweg-de Vries equation $u_t + uu_x + u_{xxx} = 0$, the sine-Gordon equation $\phi_{zz} - \phi_{tt} = \omega_0^2 \sin\phi$, the cubic Schrödinger equation, ... and with it the number of important physical models which can be exactly solved increased from around four to something like thirty. By now the soliton business is booming and both in theory and in applications it accounts for hundreds of papers each year (perhaps more).

Solitons like those depicted in figure 11 can, of course, be small, but this does not mean that we can linearize the KdV equation e.g. to $u_t + u_{xxx} = 0$, or the sine-Gordon equation to $\phi_{zz} - \phi_{tt} = \omega_0^2 \phi$. The solitons than disappear, they are truly nonlinear phenomena. The top picture of figure 12 shows a solution of the linearized sine-Gordon equation (discretized as coupled systems of pendulums). The second picture of figure 12 shows a true soliton solution of the sine-Gordon equation. The pictures of figure 13 also show such solutions to

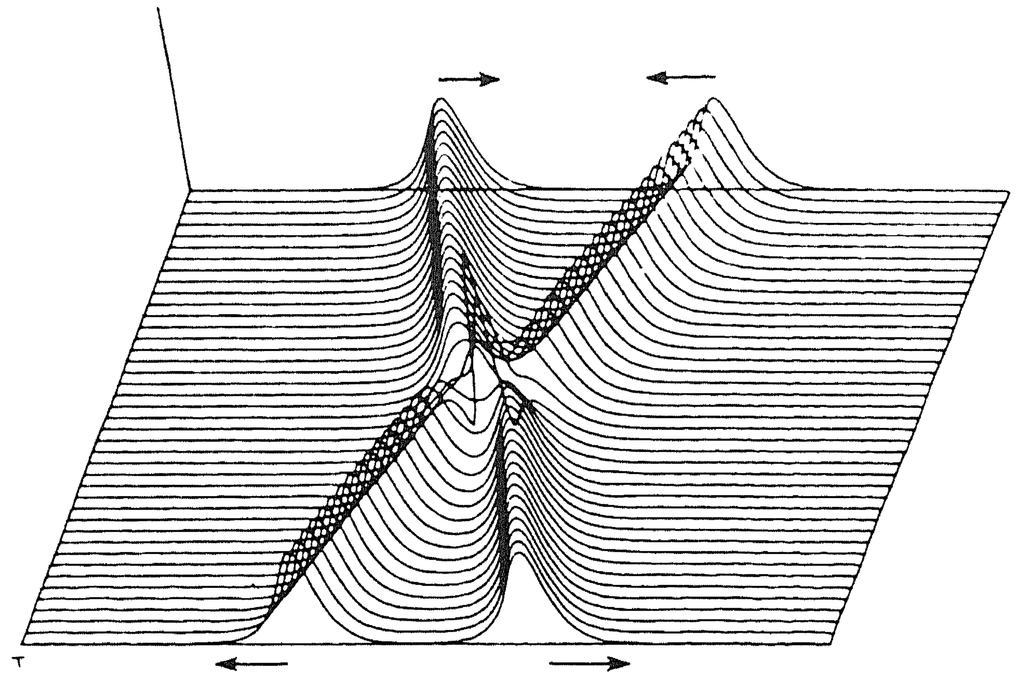


FIGURE 11, from [116]

the sine-Gordon, this time in an application to magnetic systems.

There exists so far no method (algorithm) for determining whether a given system is (completely) integrable which is the mathematical property lying behind the soliton phenomenon. If a system is suspected of being completely integrable the done thing is, also nowadays, to first throw it on a computer [16]. The following two sets of pictures may indicate what one looks for in such cases. The figures 14, 15, and 16 depict the orbits of an unequal mass, respectively equal mass so-called Toda-lattice at higher and higher energies [19]. All the dots in the left sides of pictures 15 and 16 come from a single orbit. The unequal mass Toda lattice of the left exhibits more and more chaotic behaviour with increasing energy: it is not integrable. The equal mass Toda lattice of the right hand side of the preceding three pictures shows much more regular type behaviour. It turned out to be integrable. It is also a historical fact that the integrability of the Toda lattice was thus discovered by computer experiments [27], [44]. The theoretical proof, by H. FLASCHKA, followed some years later [28].

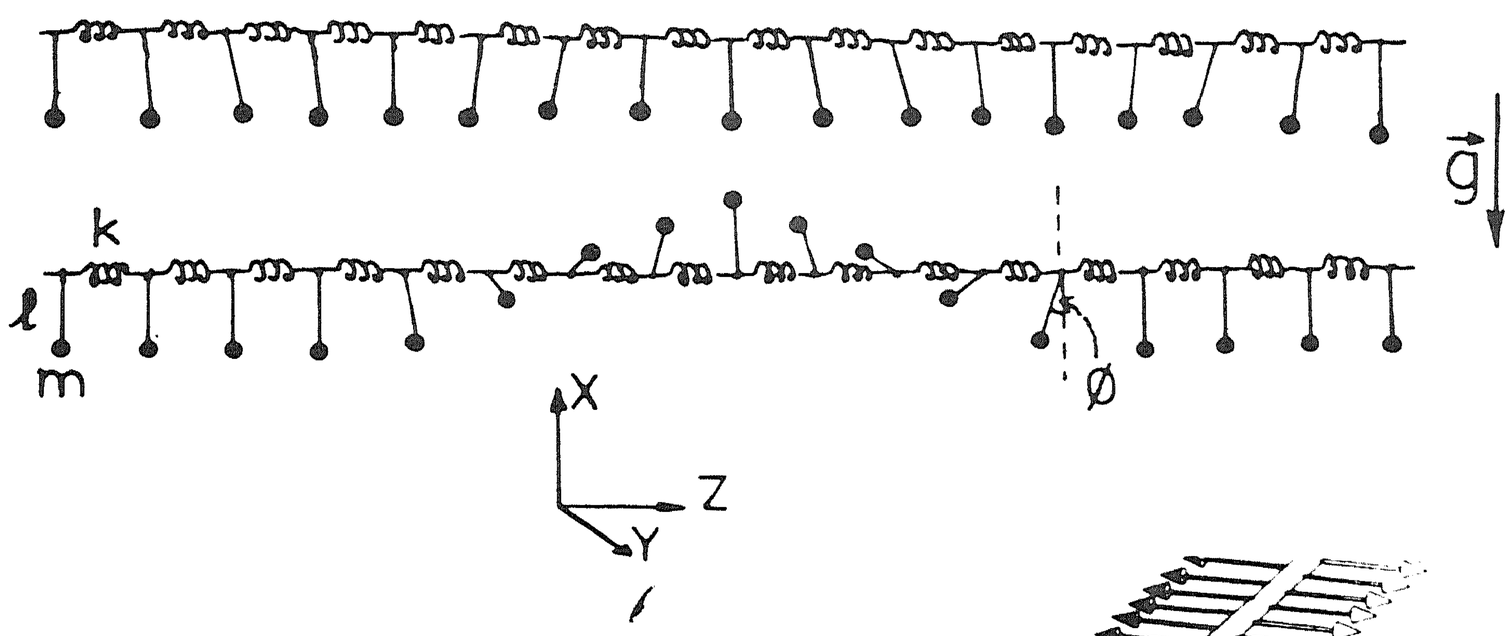


FIGURE 12, from [67]

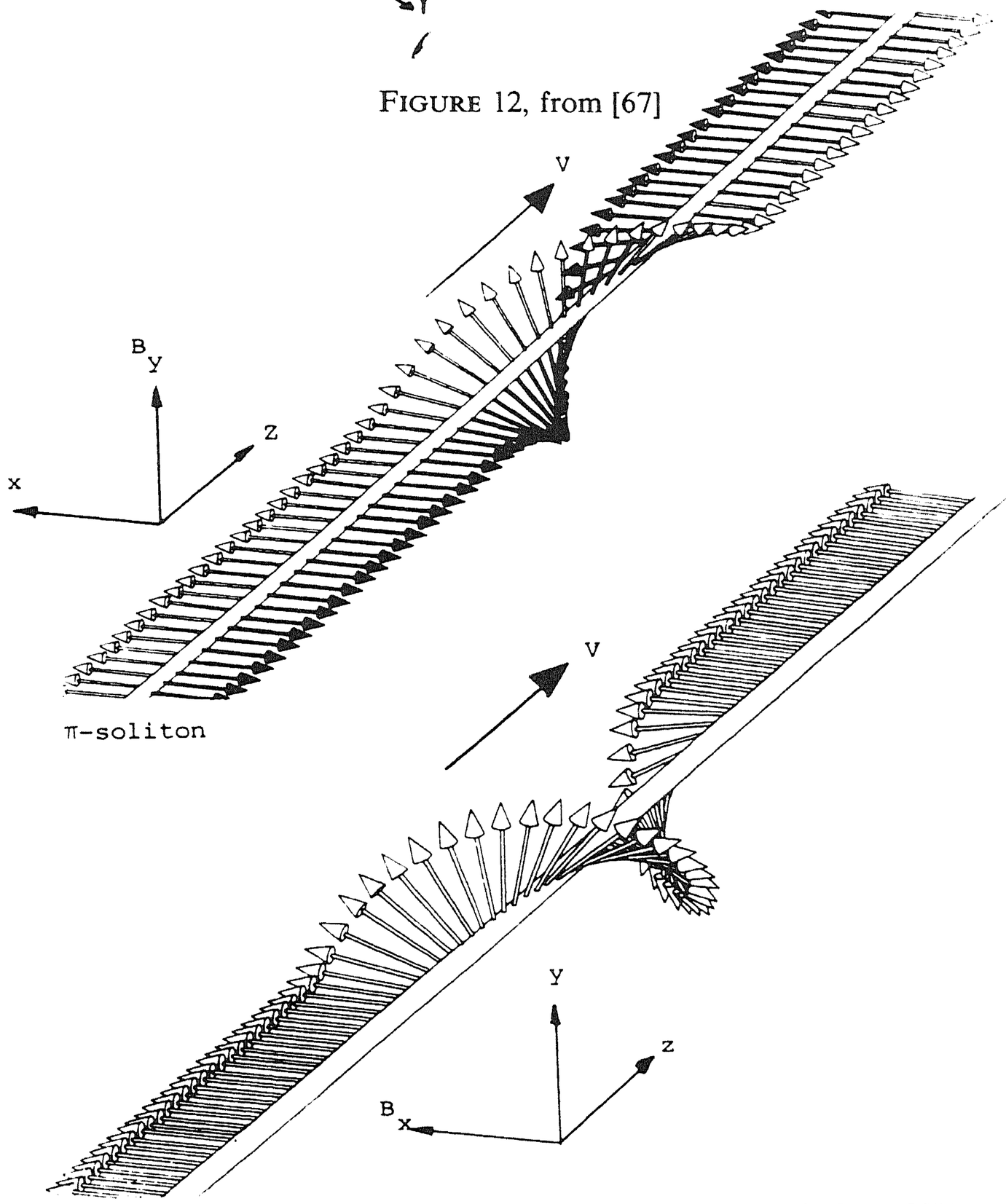


FIGURE 13, from [68]

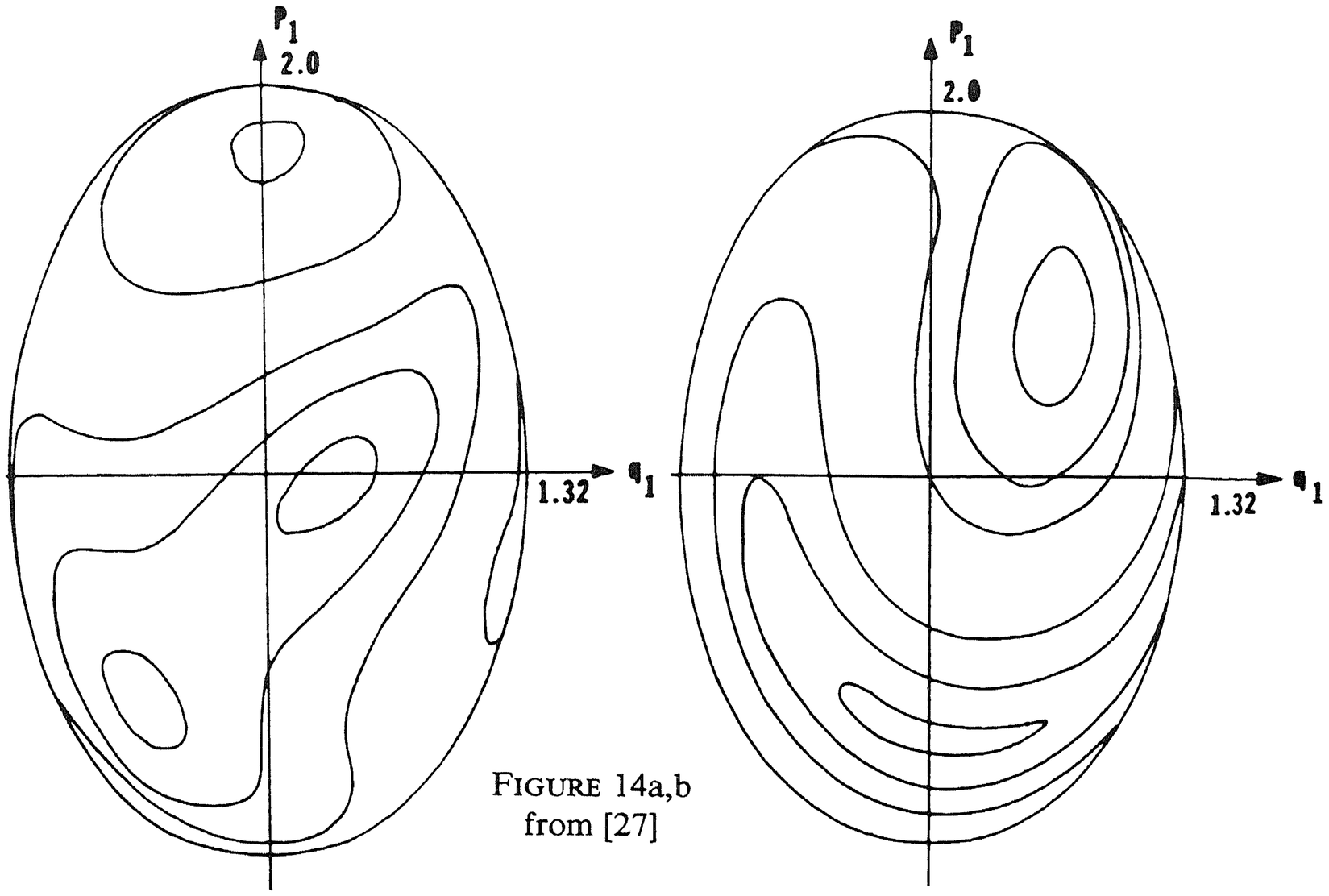


FIGURE 14a,b
from [27]

FIGURE 15a,b
from [27]

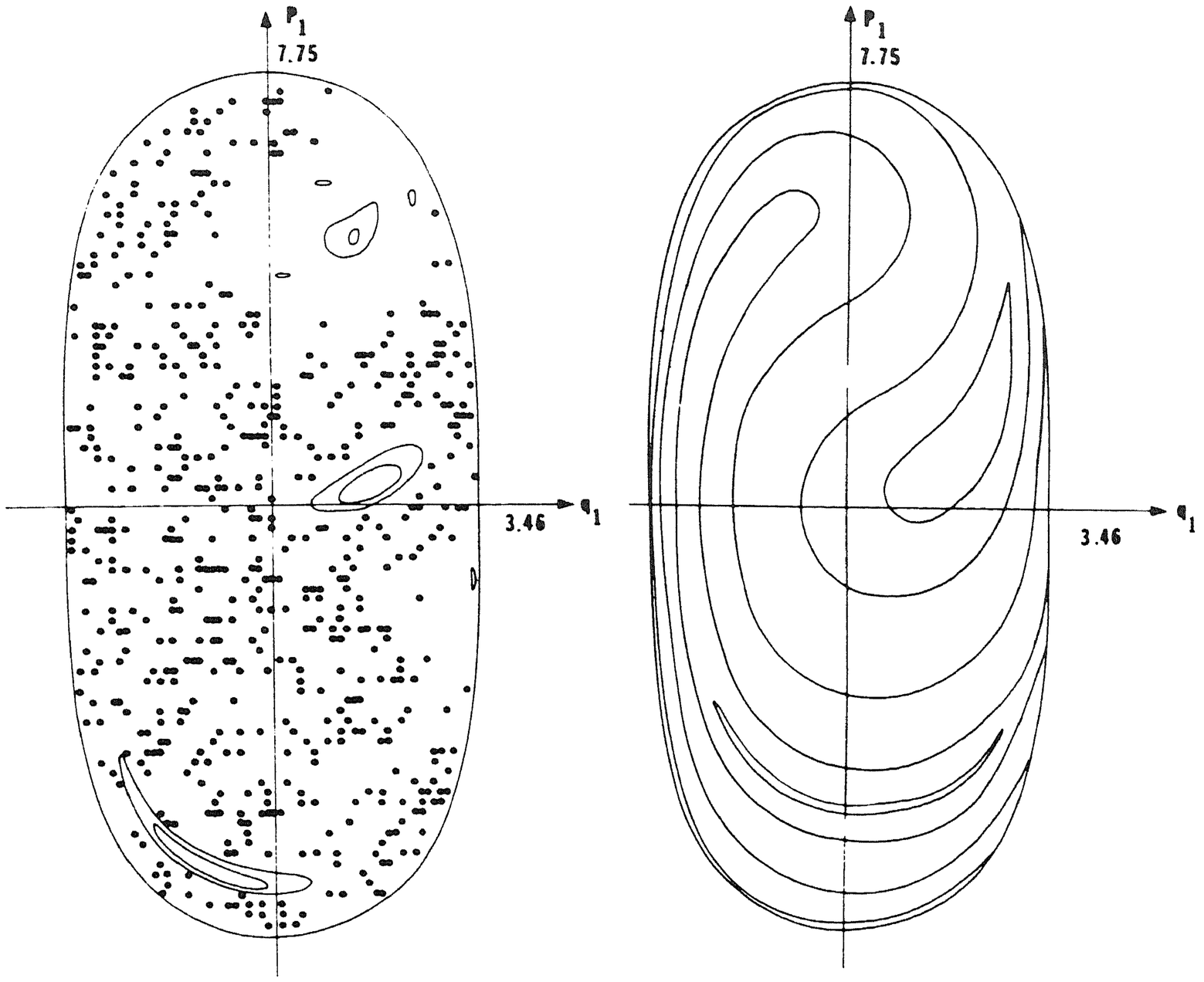


FIGURE 16a,b from [27]

7. SOME MORE EXAMPLES IN BRIEF

The three examples described above are simply three examples, if rather important ones. There are many more. Eleven stimulating computer experiments are described in the uncommonly interesting book [52] and both this work and, so far, this article have totally ignored the role of computer experiments and verifications in number theory [20] and algebraic geometry. As an example of the latter it was a computer which came up with the fact that $27^5+84^5+110^5+133^5=144^5$, thus disproving Eulers assertion (circa 1769) that it is also impossible to find three fourth powers whose sum is a fourth or four fifth powers whose sum is a fifth power.

7.1. The Atkin-Swinnerton-Dyer conjectures

Another example involves the socalled Atkin-Swinnerton-Dyer conjectures. Associated to an elliptic curve over $\mathbb{Z}$ — whatever that is — , there is its Artin L-function — whatever that is —, which can be developed into a power series in a certain way. The coefficients obtained in this way turned out numerically to satisfy certain congruences of the form

$$a_{np} - \alpha(p)a_n + \beta(p)a_{n//p} \equiv 0 \ mod \ p^{\nu_p(n)}, \quad n = 1,2 \ldots$$

Here p is a prime number, $p^{\nu_p(n)}$ is the largest power of p dividing n and $a_{n//p} = a_{n/p}$ if p divides n and $=0$ otherwise. For an account of the first numerical work in this direction cf. [7], cf. also [108] which also discusses more high powered numerical algebraic geometry. Later these congruences were indeed proved; cf. e.g. [58] for a proof.

7.2. Julia sets

Consider a complex polynomial $p(z)$. In 1879 CAYLEY proposed to extend Newton's method for calculating the roots of a polynomial to the complex case. This gives the formula

$$N(z_k) = z_k - p(z_k)/p'(z_k) \tag{7.3}$$

and he posed the problem of determining for each root a of $p(z)$ its set of attraction, $A(a)$, and is boundary $\partial A(a)$. These boundaries are socalled Julia sets and one of their more remarkable properties is e.g. in the case of the cubic $z^3 - 1$, that one has $\partial A(1) = \partial A(-\frac{1}{2} + \frac{1}{2}i\sqrt{3}) = \partial A(-\frac{1}{2} - \frac{1}{2}i\sqrt{3}) = J$.

To see what happens pictorially PEITGEN c.s. [92] defined level sets of equal attraction as follows: let $0 < \epsilon << 1$, $L_0(a) = \{z : |z - a| \leqslant \epsilon\}$, $L_{k+1}(a) = \{z \notin L_0(a) : N(z) \in L_k(a)\}$ and in their various pictures they coloured $z \in L_k(a)$ black if $\mathrm{Im}(N^k(z))$ is positive and white if $\mathrm{Im}(N^k(z))$ is negative[21]. The resulting picture for the polynomial $z^2 - 1$ with roots ± 1 is shown in figure 17. Apparently each point of the Julia set, in this case the imaginary axis, comes so to speak with a binary address. Figure 18 shows part of the picture for the third degree polynomial $z^3 - 1$. In the upper third of this picture one discerns what looks like a curved version of a neighbourhood of the imaginary axis in figure 17. It seems as if the dynamical system for $z^3 - 1$ in this neigbourhood behaves like the system of a quadratic polynomial, instead of a third degree one. This has since been proved.

7.3. Formal groups

A commutative formal group of dimension 1 over a ring R is a formal power series in two variables $F(X, Y)$ which satisfies

$$F(0, Y) = Y, \quad F(X, 0) = X, \quad F(X, Y) = F(Y, X), \tag{7.5}$$
$$F(F(X, Y), Z) = F(X, F(Y, Z)).$$

One way to obtain such a thing if R is an integral domain, e.g. $R = \mathbb{Z} =$ the ring of integers, is to take a power series $f(X)$ over the quotientfield $Q(R)$

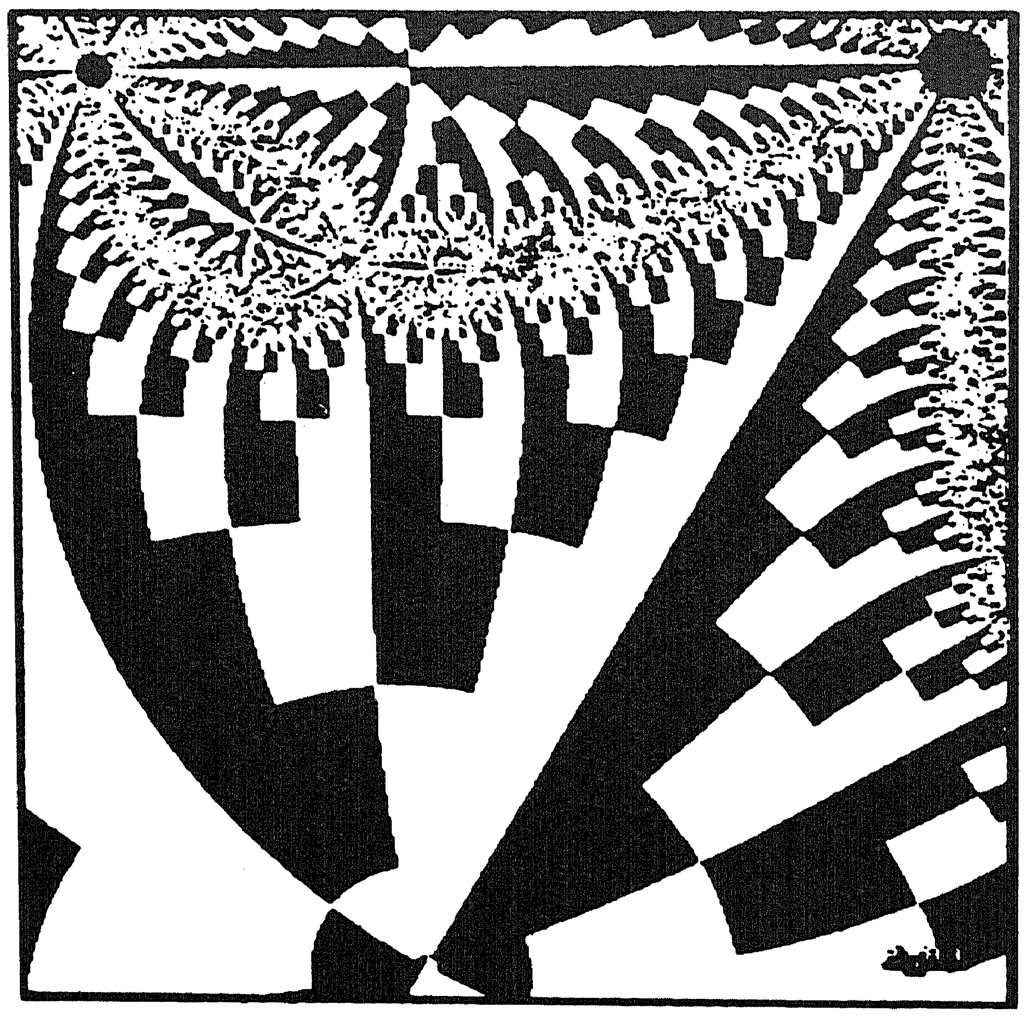


FIGURE 17, from [93]

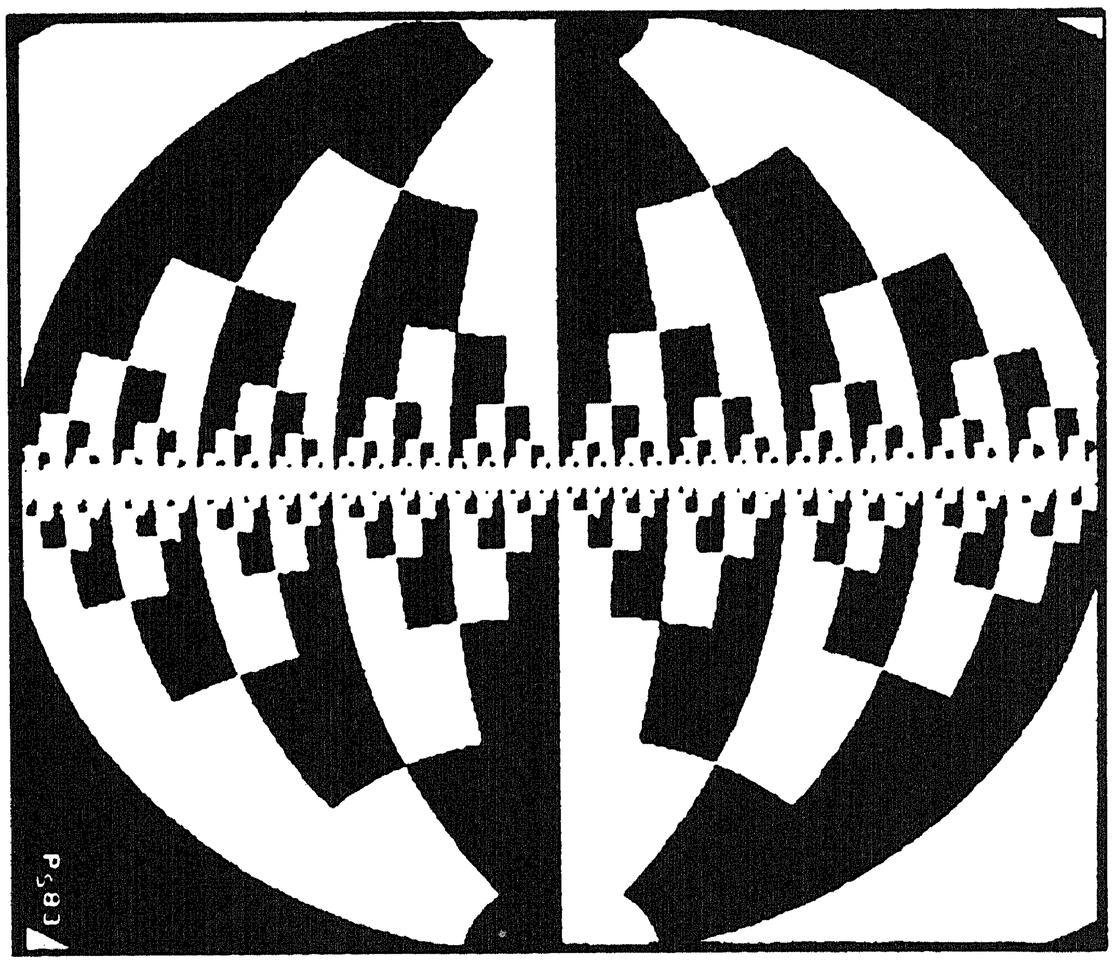


FIGURE 18, from [93]

which looks like $f(X)=X+a_2X^2+\ldots$ and to define $F(X,Y)=f^{-1}(f(X)+f(Y))$ where f^{-1} is the inverse function of $f(X)$, i.e. $f^{-1}(f(X))=X$. For suitable $f(X)$ the coefficients of $F(X,Y)$ then miraculously are in $R\subset Q(R)$ and it is a theorem that every one-dimensional formal group over an integral domain can be obtained in this way.

There exist universal formal groups from which every such animal can be obtained by assigning particular values to parameters $V_1, V_2\ldots$ These universal examples can be recursively calculated. One such universal formal group is given by

$$\begin{aligned}
F_V(X,Y) = {} & X+Y-V_1(XY^2+X^2Y)+V_1^2(XY^4+X^4Y)\\
& +3V_2^2(X^2Y^3+X^3Y^2)-V_1^3(XY^6+X^6Y)\\
& -6V_1^3(X^2Y^5+X^5Y^2)-13V_1^3(X^3Y^4+X^4Y^3)\\
& -3V_2(XY^8+X^8Y)+(6V_1^4-12V_2)\\
& (X^2Y^7+X^7Y^2)+(27V_1^4-28V_2)(X^3Y^6+X^6Y^3)\\
& +(52V_1^4-42V_2)(X^4Y^5+X^5Y^4)\\
& +(6V_1V_2+V_1^5)(XY^{10}+X^{10}Y)+45V_1V_2(X^2Y^9+X^9Y^2)\\
& +(163V_1V_2-27V_1^5)(X^3Y^8+X^8Y^3)\\
& +(362V_1V_2-27V_1^5)(X^3Y^8+X^8Y^3)\\
& +(362V_1V_2-106V_1^5)(X^4Y^7+X^7Y^4)\\
& +(532V_1V_2-192V_1^5)(X^5Y^6+X^6Y^5)+\ \ldots\\
& +(-105\,024\,048V_1^3V_2^2+954\,161\,30V_1^7V_2+213\,396\,72V_1^{11})\\
& (X^{10}Y^{13}+X^{13}Y^{10})+\ \ldots
\end{aligned}$$

and I challenge anyone to see the regularity in this [22].

This shows that playing experimental mathematics games on a computer is fine but will not lead to stimulating results unless a) one has a good idea of what should be calculated and b) the results are presented in a form suitable for the superior human pattern recognition faculties[23].

In this particular case the formal group $F_V(X,Y)$ itself is simply totally the wrong thing to look at. The power series $f_V(X)$ such that $F_V(X,Y)=f_V^{-1}(f_V(X)+f_V(Y))$ looks like

$$f_V(X) = X+\frac{V_1}{3}X^3+\left(\frac{V_1^4}{9}+\frac{V_2}{3}\right)X^9+ \qquad (7.6)$$

$$\left[\frac{V_1^{13}}{27}+\frac{V_1V_2^3}{9}+\frac{V_2V_1^9}{9}+\frac{V_3}{3}\right]X^{27}+\ \ldots$$

and here one can see the hidden regularity; especially when one reflects that we are dealing with the prime number $p=3$ and if one substitutes $3=p$, $9=p^2$,

$27=p^3$, $4=1+p$, $13=1+p+p^2$. And as a matter of historical fact this is (essentially) how the general formula for $f_V(X)$ was discovered. In November 1969 I spent a month calculating $f_V(X)$ up to degree 27 removing by means of suitable isomorphisms all terms that I could get rid off. All this in a vain attempt to find a counter example to something. Formula (7.6) was what I finally found (apart from two sign mistakes). Nowadays such things should be done by machine. Since then the formula has found quite a few applications in various parts of mathematics [58].

This also brings me to another point I wish to stress. For problems with a geometric content colored computer graphics are important for experimental mathematics[23] and for problems with a more algebraic or analytic flavour it will be symbolic computation, formula manipulation computation, which will perhaps be more important than number crunching[24].

7.4. Anti-diffusions

Consider a process with an autocatalytic component, i.e. such that initial disturbances will tend to grow, up to a certain point. One possible model, at first

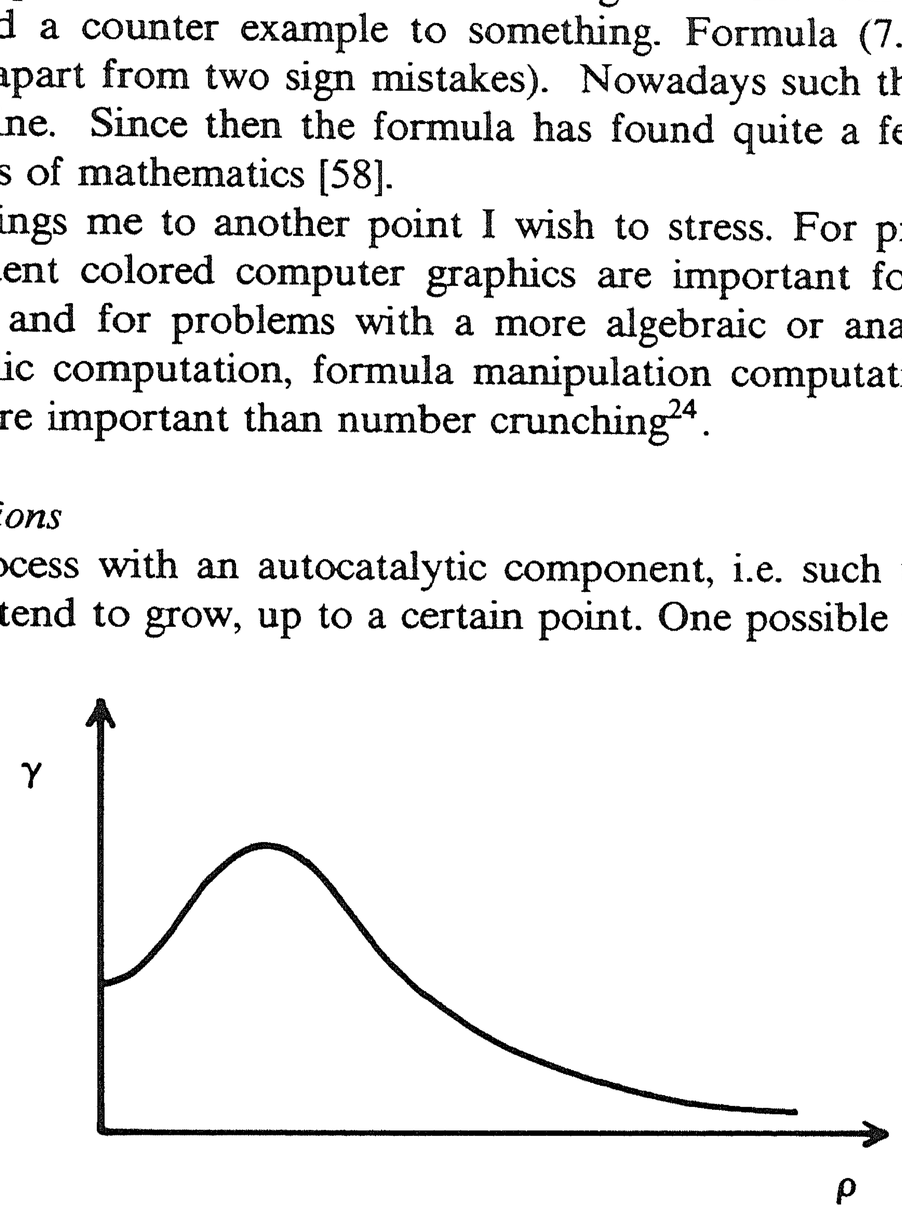


FIGURE 19

sight, for such a thing could be an anti diffusion equation of the form

$$\rho_t = -\frac{\partial^2}{\partial x^2}\phi(\rho) \tag{7.8}$$

where ρ is some sort of density and ϕ is a function of the form shown in figure 19. Our hope was that starting from an initially homogeneous ρ and small initial disturbances, or, better, small stochastic disturbances all the time, this would give rise to stable periodic patterns in space. Analytically, virtually nothing is known about equations like (7.8), beyond the fact that they are highly unstable. So I suggested my student to put it on a (small) computer. One of the sequences of pictures he came up with is figure 20.

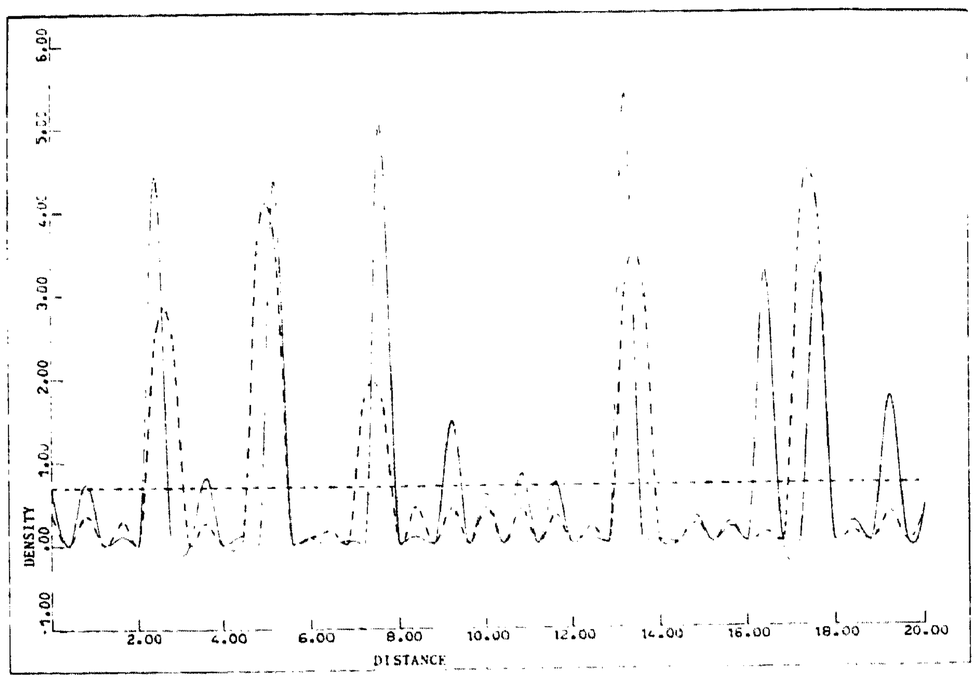


FIGURE 20a from [68]. — horizontal dotted line: starting density; continuous curve: after 1000 periods; dotted curve: after 2000 periods

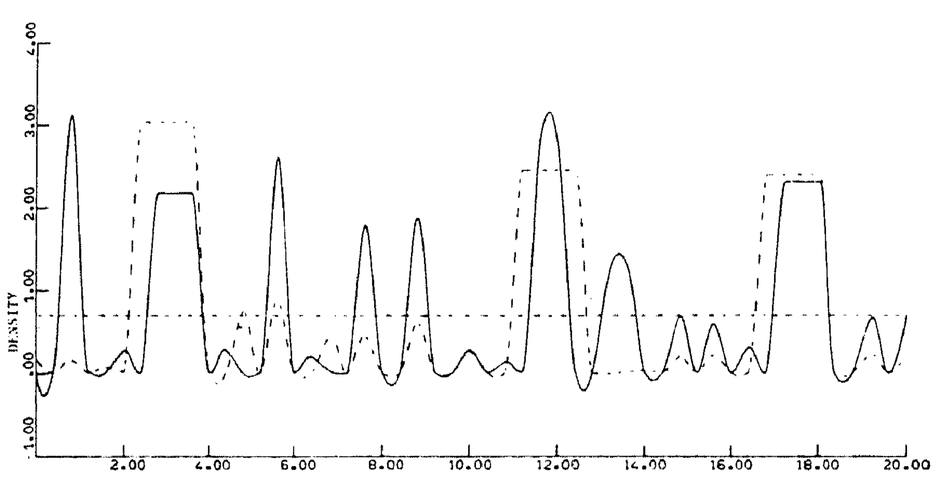


FIGURE 20b as in a) (different parameter value)

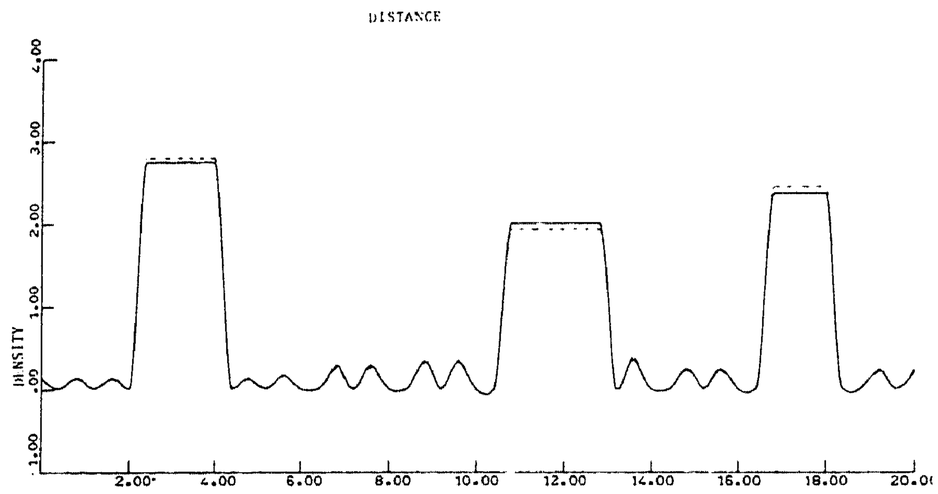


FIGURE 20c continuation of b); continuous curve: after 10000 periods dotted curve: after 15000 periods

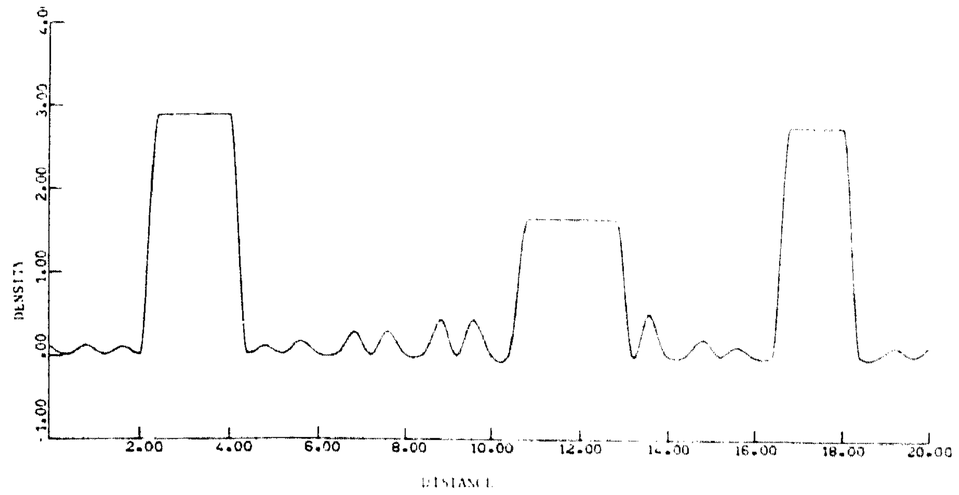


FIGURE 20d continues b) and c): after 25000 periods

Such patterns seem to arise remarkable often in this context and they also persist for long times. They still could be transient phenomena of course and indeed there are reasons to believe so (no proof). Even so they persist for very long times. Similar phenomena occur in [36][25] e.g. and they pose the general problem of how to deal mathematically with such 'patterns' which are semistable in the sense of persisting for very long times (also in the face of disturbances) but eventually disappear, or which persist only in a looser sense, in that there are always the same number of bumps at roughly the same equal distance, but they keep moving and changing shape slightly and never settle down[29].

7.5. Traveling salesman[26]

The traveling salesman problem is the following. Consider n cities, n large, and the distances between them. Find the shortest circuit which passes through each of them once. This can be viewed as a programming problem with decision variables x_{ij}, $x_{ij}=1$ if the stretch from city i to city j is to be included in the circuit and 0 otherwise, and a large number of restrictions to make the path a socalled Hamiltonian one, i.e. one which passes through each vertex precisely once. The convex hull of all admissible integral vectors constitutes a polytope in $\mathbb{R}^{n^2}$, which has not yet been characterized. Early in the game DANTZIG, FULKERSON and JOHNSON developed a quite successful algorithm which approached the problem as a (continuous) linear programming problem with $0 \leqslant x_{ij} \leqslant 1$ and with a smaller set of the restrictions than the set defining the original polytope. They started with the trivial restrictions $\Sigma_i x_{ij}=1$, $\Sigma_j x_{ij}=1$ and then if a 'subcircuit' came out (e.g. $x_{12}=1=x_{21}$) a new restriction (here $x_{12}+x_{21} \leqslant 1$) was added. This approach got neglected when branch and bound became more successful.

In 1953 ALAN HOFFMAN and HAROLD KUHN carried out an experiment[38]. Stand in the middle of the polytope and fire a gun at random in all directions. All shots turned out to pass through a part of the 'wall' defined by facets of the trivial type $x_{ij}=0$. These 'experiments' contributed to new insight in the structure of the traveling salesman polytope and the best algorithms anno 1983

are based on a combination of the Dantzig c.s. 1954 method (initially) followed by branch and bound methods.

8. A FEW FINAL REMARKS

The three main examples of Section 4,5 and 6 above are but a random selection dictated by personal taste. There are of course many more. Indeed it seems clear by now that experimental mathematics is developing very fast and that it is already generating conjectures, results and challenging problems at a higher rate than can be handled by the theoreticians. Here are some more challenges posed by experimental results (besides the ones already mentioned).

— There is a wealth of material, bifurcation pictures and phase diagrams, concerning the socalled Josephson-junction , an equation which probably will play the role of *the* well studied and illustrative example which in the past has been played by the VAN DER POL equation [16], [98], [99], [106]. It is perhaps also interesting to remark that the socalled 'breather solutions' of the Josephson-junction were first discovered numerically[30].

— As a rule, if a Hamiltonian system is not integrable, its behaviour becomes more and more chaotic as energy is increased. No proof is available. Exceptions are of course systems which decouple into integrable subsystems as $E \to \infty$. There are however also systems which do not have this property and still show a return to more regular behaviour as E increases [1], [2].

— There is quite a bit of numerical evidence for various kinds of universal behaviour for iterated maps of more-dimensional objects, e.g. subsets of the plane, which await theoretical elucidation [23], [41], [51], [113].

— There are literally masses of experimental results dealing with percolation through porous media and associated phenomena like clogging of throats of pores and 'fingering'. Both computer generated and as a result of real hydrology experiments. Mostly, again, awaiting analysis and concept formation to bring some order and classification[33].

— Stimulated by a hypothesis of CRICK and MITCHISON [34] to the effect that one of the functions of dream sleep might be an 'unlearning process', HOPFELD a.o. [63] carried out mathematical and computer modelling on networks of neurons. I quote:

> 'Although our model was not motivated by higher nervous function, our system displays behaviours which are strikingly parallel to those needed for the hypothesized role of 'unlearning' in rapid eye movement sleep'.

Here again is a conceptual and mathematical challenge[32].

Before finishing let me stress again that 'user friendly' outputs like colour graphics, movies are likely to be more important in experimental mathematics than rows and rows of numbers. Also symbolic calculation and formula manipulation is likely to grow in relative importance, again because symbolic formulae are better suited to human pattern recognizing abilities than numbers[35].

Also we really need the computer assistance at this point, again because we seem to have in many cases reached a sort of natural limit of what can be done by hand[31].

Let me also remark on the pleasing fact that all three main examples I discussed above have as much to do with classical pure mathematics as with classical applied mathematics and that thus it seems that experimental mathematics is doing much to remove the silly and distressing distinction between the two.

Finally let me close with expressing the hope that what has been said above will have helped to make it clear that experimental mathematics is a vigorous, fast growing subject, synergetically related to its scientific neighbours. Indeed I have the feeling that we are at the beginning of what may well turn out to be a heroic period in mathematics comparable in significance and future influence to the 1920's in physics. In any case I hope to have helped to make it clear that VON NEUMANN appears to have been absolutely right in his predictions of 1946.

NOTES

1. The Littlewood-Richardson rule deals with the question of the multiplicities of the representation $\Lambda^\gamma E$ of $GL(E)$, E a vectorspace, in the direct sum decomposition of $\Lambda^\alpha E \otimes \Lambda^\beta E$. Here α, β and γ are partitions.
2. I owe the information about BOCKWINKEL and LORENTZ to JAAP J. SEIDEL and F. ALBERTO GRÜNBAUM.
3. HAHN uses this phrase in the context of a critique of the Kantian idea that mathematics, especially geometry, is completely based on intuition a priori. To this end he discusses the counterintuitive properties of such things as Peano and Sierpinsky curves and noneuclidean geometry. Such logical constructs are of course equally intuition and mind enriching as computer experiments.
4. Cf. also 'Computer graphics comes to statistics' (GINA KOLATA), *Science 217* (1982), 919-920. By means of three dimensional projections generated by means of computer motion graphics from multi dimensional data sets, combined with human pattern recognition abilities it seems to be possible to detect previously unrecognized interesting phenomena. (Discrepancies in this case).
5. Later in this paper, discussing renormalization-group ideas and 'the new physical principle of scale invariance' the author remarks: 'In this example it was a new physical principle that permitted computation capable of solving a previously intractable set of problems. The initial computational test of the principle played a mayor role in establishing its utility.' This is an aspect of experimental mathematics that I do not stress in this paper, though of course it is similar to experiments - as discussed in 3.10 — to find out whether a given type of model is capable of producing the phenomena it is designed to 'explain'. However, to the remark of DONALD R. HAMANN on renormalization ideas I would like to add that from a lecture of KENNETH G. WILSON in Los Alamos

in 1972 I have the impression that, at least in the case he was discussing (the Kondo problem), the desire to find some computational scheme to handle the problem had a lot to do with the genesis of Wilson's renormalization group ideas.

6. The 'soliton story' and the 'iterated maps and chaos story' which are the subject matter of Sections 6 and 5 of this paper are also briefly mentioned in [57].

7. As I have remarked before [59], unaided intuition or common sense are poor instruments of thought when confronted with cause and effect relations which cannot be linearly ordered, i.e. when there are mutual interactions and/or feedback loops present. From this point of view mathematics is a highly necessary tool for finite human brains. A God would have no need of it. And within mathematics itself experimental mathematics is proving to be an equally necessary tool for helping our mathematical intuition. It also does a similar job in geology, physics, chemistry etc. Examples are e.g. the Phillips stabilization paradox of economics, [9] (dealing with Government spending to stabilize an economy), the fact that monopoly positions can very well be disadvantageous [8], and the Arrow impossibility theorems, see e.g. [100] and [88], (dealing with the design of democratic voting systems). As ERIC T. BELL [15] says: 'One service mathematics has rendered the human race: it has put common sense back where it belongs, on the top shelf next to the dusty canister labelled "discarded nonsense"'.

8. The picture has to do with studies by S. WHITE, M. DAVIS, C. FRANK (Berkeley); other studies were done by S. DJORGOVSKY (Berkeley), J. CENTRELLA, A. MELOTT (Lawrence-Livermore Lab.). The socalled inflasionary cosmogonical model of A. GUT (MIT) is important here.

9. In a short 'News and Views' report on the work of C.M. BENDER and D.H. SHARP, JOHN MADDOX (*Nature 303* (1983), p. 279) comments that the chief value of their method will be to sharpen physical intuition, and that much the same may be true of a new numerical technique of M. CREUTZ [33] for calculating partition functions in statistical physics. In both cases, especially in my view the first, things work so well that one feels to have received a first hint of the presence of some unsuspected physical or mathematical principle.

10. There appear to be even more bonusses coming out of the FEM approach to quantum field theory, (BENDER, MILTON, SHARP, to be published), dealing with finding a gauge invariant FEM model and what happens as a certain dimensionless lattice spacing parameter goes to zero.

11. Further developments from the hard hexagon model involve directed lattice animals, polymers, directed percolation theory etc. Numerical calculations here continue to play a dominant role in finding, formulating and testing conjectures as a good look at the following papers will show: D. DHAR, *Phys. Rev. Lett. 49* (1982), 959-962; V. HAKIM, J.P. NADAL, *J. Phys. A 16* (1983), L213-L218; J.P. NADAL, B. DERRIDA, J.

VANNIMENUS, *J. de Physique 43* (1982), 1561, B. DHAR, M.K. PHANI, M. BARMA, *J. Phys. A 15* (1982), L279-L284; N. BREUER, H.K. JANSSEN, *Z. Phys. B 48,* 347-350; F. FAMILY, *J. Phys. A 15* (1982), L583-L592; J.E. GREEN, M.A. MOORE, *J. Phys. A 15* (1982), L597-L599; A.R. DAY, T.C. LUBENSKY, *J. Phys. A 15* (1982), L285-L290; J.L. CARDY, *J. Phys. A 15* (1982), L593-L595; S. REDNER, A. CONIGLIO, *J. Phys. A 15* (1982), L273-L278.

Also finding the exact results has involved (as in the hard-hexagon case) considerable computer assistance. The original version of the bijection between directed lattice animals and certain kinds of discrete paths which is at the basis of a combinatorial approach to these exact results involved first numerical comparison of the respective numbers of animals and paths respectively and also considerable numerical search in finding the right 'size' parameters for these things (the latter search involved analogues with orthogonal polynomials). These matters will be reported on in G. VIENNOT, Problèmes combinatoires posés par la physique statistique, Sém.Bourbaki, Febr. 1984, Exposé 626.

12. Instead of, say, triangular lattice gas with nearest neighbor exclusion.

13. The story as outlined below is the sort of thing which rarily, if ever, gets published in the official journals. As outlined here it owes very much to a cassette tape and copies of the slides of a lecture that BAXTER gave at King's college in London in July 1980. I am extremely grateful to BAXTER for sending me this material.

14. RODNEY J. BAXTER received the much coveted Boltzmann medal for his work on exactly solvable lattice statistical mechanics.

15. The 10×10 approximation figures are even more spectacular.

16. It also illustrates another point. Interesting systems, phenomena, of a particular kind, ... etc. are (likely to be) rare. For instance (completely) integrable Hamiltonian systems are rare (in the class of all Hamiltonian systems). Another role for the computer in experimental mathematics could be in a searching for interesting unusual phenomena of certain specified kinds. Much as in [72] where is described how in astronomy computers can help in finding interesting stars. However, cf. also note 37.

17. Such deterministic chaos; i.e. chaotic behaviour caused by perfectly deterministic maps may provide another model for modeling certain random phenomena. I.e. models different from stochastic models. One type of noise which frequently appears in (solid state) electronics is socalled $1/f$- noise [84] and it may be possible that deterministic chaos will be fruitful in its study and analysis [101], [54]. The problem of how to distinguish observationally between deterministic chaos noise and stochastic noise is still open.

18. There have, of course, been other inputs than the computer experiments briefly indicated in this section. Notably the invention of 'Strange attractors' (E. LORENZ 1963 [78], D. RUELLE and F. TAKENS, 1971 [97]).

Lorenz's model of a strange attractor is a severely cut down approximation of atmospheric flow and the fact that there is a strange attractor present illustrates some of the notorious difficulties of wheather prediction.

19. More precisely it shows the intersection of these orbits with the $p_1 - q_1$ plane. The Toda lattice of picture 14, 15 and 16 is the one with Hamiltonian $H = \frac{1}{2}(p_1^2 m_1^{-1} + p_2^2 m_2^{-1}) + \exp(-q_2 + q_1) + \exp(q_2) - 3$. The pictures on the left have mass ratio $m_2/m_1 = 0.33$; the ones on the right $m_2/m_1 = 1.0$.
20. The first 300 million or so of the non real zeros of the Riemann zeta function do indeed lie exactly where they should ([79]) and the mathematics developed to prove such a thing certainly would not have developed without the big machines. Another instance of this is the matter of the mathematics of fast prime number tests [76]. Also the high interest in effective upper bounds for the solutions of diophantine equations (Baker-Gelfond theory) is certainly connected with the availability of lots of computing power. All these, however, I consider instances, like the case of semiparametric statistics discussed in the introduction, where the presence of the big machines enlarged the set of problems which we are willing and interested to think about, rather then examples of experimental mathematics.
21. In 'reality' H.-O. PEITGEN c.s. used colors and the resulting pictures are really quite beautiful. Four of them occur in the 1984 Springer Verlag mathematics calender. They have also been the material of an art exhibition in the Sparkasse in Bremen [44] in Jan/Febr. 1984.
22. As a matter of fact the example shown is a p-typical universal formal group, in this case for $p = 3$. These are not truly universal but are universal for a more restricted class. They are however much more regular than a truly universal one can be, essentially because there is so to speak only one prime number to worry about.
23. ZABUSKY [116] stresses this particularly and has repeatedly insisted on the desirability of using computer graphics and movies in this connection. The studies hinted at in 7.2 above also illustrate this point.
24. There may be considerable number crunching behind a computer generated picture of course, and often there is.
25. In this case the phenomenon is definitely transient.
26. This example I owe to JAN KAREL LENSTRA, CWI, Amsterdam.
27. Another example, besides the ones that follow, is [104]. Here there is a criterium for the existence of a closed orbit for systems with strange attractors. This criterium involves estimates which are designed to be verified by computer. Otherwise one would hardly consider them.
28. Cf. [116] for a detailed account of these happenings.
29. One phenomenon to which we hope to apply ideas along these lines is the phenomenon of Liesegang rings in (colloid) chemistry. Another model designed to deal with this phenomenon is described in [39]. The patterns generated by that model are of a similar nature. They also

appear to be transient, but with a very long life. Here also a mathematical analysis predicting these patterns is almost completely absent.

30. By two physicists: IMRY and SCHULMAN. I owe this bit of information to M. LEVI of Boston Univ.

31. A totally different topic, also of high interest, both from a theoretical and practical point of view, coming out of the availability of computer power is the matter of (flexible?) computer design to meet the requirements of certain problems, cf. [118, 105].

32. To illustrate a point let me quote from [22], noting that this is but one example from very many. 'We have performed Monte-Carlo simulations on the Kinetics of ... . The extent of reaction ... increases with decreasing fraction of divinyl monomer, with increasing solvent concentration and with increasing initiator concentration. These predictions, and the observed trends for the dependence of the overall polymerization rate on the same concentrations, are in qualitative agreement with laboratory experiments'.
This type of work is of course most important e.g. in constructing adequate models and in testing tentative principles and formulating theories. But if things stop right here progress will soon cease. A model which is conceptually murky but works numerically well is of very limited value unless the challenge posed by an unusually well working model is taken up.

33. Another area of vigorous interaction between numerical experiment and theoretical and applied (in the more traditional sense) mathematics is the physics and mathematics of disordered media. The key words here are 'fractals', 'percolation', 'random walks (especially nonintersecting)', 'chaos'. A recent workshop on the topic took place at the IMA in Madison, Wisconsin in Febr. 1983. The proceedings will appear in the *Lect. Notes in Math.* series of Springer Verlag. Inevitably perhaps - everything relates (strongly) to everything else-. This topic has quite a bit to do with the topic of Section 4, cf. note 11.

34. Here is another example to illustrate the point. I quote from [20]. 'High-performance computer graphic techniques have been developed in the last year or two, and are now taking the place of conventual model building. ... Sophisticated computer graphics were used to survey the likely active conformations of known inhibitors of the converting enzyme. This survey guided the synthesis of putative inhibitors with functional groups in rigid orientations. It resulted finally in the synthesis of a potent bicyclic inhibitor molecule, and a patent was applied for a few weeks ago'.

35. One area where symbolic computation is becoming increasingly important is in describing and calculating the symmetries of important physical models such as Gauge theories. Cf. [70, 71].

36. The sources of all reproduced figures in this paper are stated in the captions. I am grateful for the permission to reproduce these.

37. All in all it seems that this happens quite often. I.e. that computer

experiments bring something new to ponder. Rather remarkably often perhaps, indicating that there are very many interesting phenomena still awaiting discovery. Not only in experimental mathematics, but in all of mathematics I often have had the feeling 'can one be so lucky'. There really is very often something fascinating going on. This does not contradict note 16.

38. Cf. the remark by Kuhn (page 118) in the discussion of [49]. There are more interesting challenges in this area. E.g. the 'unreasonable effectiveness' of some assignment problem algorithms. Cf. the discussion between EDMONDS and KUHN, loc. cit.

REFERENCES

1. B.J. ALDER, T.E. WAINWRIGHT (1960). Studies in molecular dynamics II. *J. Chem. Phys. 33,* 1439-1451.
2. B.J. ALDER, T.E. WAINWRIGHT (1957). Phase transition for a hard sphere system. *J. Chem. Phys. 27,* 1208-1209.
3. M.K. ALI, R.L. SOMERJAI (1982). Reappearance of ordered motion in non-integrable Hamiltonian systems. *Prog. Theor. Physics 68, 6,* 1854-1863.
4. M.K. ALI, R.L. SOMERJAI (1980). Reappearance of ordered motion in some non-integrable Hamiltonian systems. *Physica 1D,* 383-390.
5. G.E. ANDREWS (1981). Rogers-Ramanujan identities and the hard hexagon model. *Nat. Acad. USA 78:9,* 5290-5292.
6. K.J. ASTRÖM (1983). Computer aided modeling, analysis and design of control systems - a perspective. *Control Systems Magazine.*
7. A.O.L. ATKIN (1968). Congruences for modular forms. R.F. CHURCHHOUSE, J.-C. HERZ (eds.). *Computers in Mathematical Research,* North Holland Publ. Co. 8-19.
8. R.J.A. AUMANN (1973). Disadvantageous monopolies. *J. Ec. Th. 6,* 1-11.
9. W.J. BAUMOL (1965-1966). Informed judgement, rigorous theory and public policy. *South. Ec. J.,* 137-145.
10. R.J. BAXTER (1980). Talk on *'Hard hexagon model'* at Kings College, London (transparancies + cassete tape).
11. R.J. BAXTER (1981). Rogers-Ramanujan identities in the hard hexagon model. *J. of statistical Phys. 26,* 427-452.
12. R.J. BAXTER (1979). *The Hard Hexagon Model in Lattice Statistics and the Roger-Ramanujan Identities* (handwritten notes, 5 Nov.).
13. R.J. BAXTER (1978). *J. Stat. Phys. 19,* 461-478.
14. R.J. BAXTER (1980). Hard hexagons: exact solution. *J. Phys. A 13,* L61-L70.
15. E.T. BELL (1951). *Mathematics: Queen and Servant of Science,* McGraw Hill.

16. V.N. BELYKH, N.F. PETERSON, O.H. SOERENSEN (1977). Shunted-Josephson-junction model I, II. *Phys. Rev. B 16, 11,* 4853-4859; ibid 4860-4871.
17. C.M. BENDER, D.H. SHARP (1983). Solution of operator field equations by the method of finite elements. *Phys. Rev. Lett. 50,* 1535-1538.
18. C.M. BENDER, D.H. SHARP (1983). Consistent formulation of fermions on a Minkovsky lattice. *Phys. Rev. Lett. 51:20,* 1815-1818.
19. G. BIRKHOFF (1983). Numerical fluid dynamics. *SIAM Rev. 25,* 1-34.
20. D. BLOW (1983). Computer cues to combat hypertension. *Nature 304,* 213-214.
21. H.B.A. BOCKWINKEL (1906). Over de voortplanting van licht in een twee - assig kristal rondom een middelpunt van trilling. *Verslagen KNAW 14,* 636-651. (Aangeboden door Lorentz.)
22. R. BONSIL, H.J. HERMAN, D. STAUFFER (1984). Computer simulation of kinetics of relation by addition polymerization in solvent. To appear in *Macromolecules.*
23. T.C. BOUNTIS (1981). Period doubling bifurcations and universality in conservative systems. *Physica 3D,* 577-589.
24. T. BOUNTIS, H. SEGUR, F. VIVALDI (1982). Integrable Hamiltonian systems and the Painlevé property. *Phys. Rev. A 25, 3,* 1257-1264.
25. R.P. BRENT, J. VAN DE LUNE, H.J.J. te Riele, D.T. Winter (1982). The first 200.000.001 zeros of Riemann's zeta function. H.W. LENSTRA, JR, R. TIJDEMAN (eds.). *Computational Methods in Number Theory,* CWI, Amsterdam.
26. G. CASATI, J. FORD (1975). *Phys. Rev. A 12, 4,* 1702-1708.
27. D. CHANDLER, J.D. WEEKS, H.C. ANDERSON (1983). Van der Waals picture of liquids, solids and phase transformations. *Science 220,* 787-794.
28. A.J. CHORIN (1983). Book review of R. Peyret, Th.D. Taylor, Computational methods for fluid flow. *Bull. Amer. Math. Soc. 9:3,* 368-371.
29. A.J. CHORIN (1976). Random choice solution of hyperbolic systems. *J. Comp. Physics 22,* 517-533.
30. P. COULLET, J. TRESSER (1978). Iterations d'endomorphismes et groupe de rénormalisation. *CR Acad. Sci. Paris 287,* 577-580.
31. K.M. CREER (1983). Computer synthesis of geomagnetic palaeosecular variations. *Nature 304,* 695-699.
32. M. CREUTZ (1983). High-energy physics. *Physics Today,* 35-42.
33. M. CREUTZ (1983). *Phys. Rev. Lett. 50,* 1411.
34. F.C. CRICK, G. MITCHISON (1983). *Natura 304,* 111-114.
35. Ph. J. DAVIS, R. HERSH (1981). *The Mathematical Experience,* Harvester Pr..
36. A. De PALMA (1985). Bifurcation and choice behaviour in complex systems. M. HAZEWINKEL a.o. (eds.). *Bifurcation: Analysis, Principles and Synthesis,* Reidel Publ. Co..
37. B. EFRON (1979). Computers and the theory of statistics: thinking the unthinkable. *SIAM Rev. 21,* 460-480. (How the availability of high-

speed computers changes the statistical tests (and theory) one is prepared to use; thus creating new areas for (also) theoretical investigations (nonparametric statistics).)

38. A.J. EVANS, H.J.M. HARLEY, S. HESS (Jan. 1984). Non-newtonian phenomena in simple fluids. *Physics Today,* 6-33.
39. R. FEENEY, S.L. SCHMIDT, P. STICKHOLM, J. CHADAM, P. ORTOLEVA (1982). Periodic precipitation and coarsening solitons. Applications of the particle growth model. *Preprint.*
40. M. FEIGENBAUM (1978). Quantitative universality for a class of nonlinear transformations. *J. Stat. Physics 19,* 25-52.
41. M.J. FEIGENBAUM, L.P. KADANOFF, S.J. SHENKER (1982). Quasi periodicity in dissipative systems: a renormalization group analysis. *Physica 5D,* 370-386.
42. J. FORD (April 1983). How random is a coin toss. *Physics Today,* 40-47. (On the definition of random versus determinate, complexity theory à la Chaitin, Martin-Lof and the question of infinite precision.)
43. J. FORD, S. STODDARD, J. TURNER (1973). *Progr. Theor. Physics 50,* 1547-1557.
44. FORSCHUNGSGRUPPE 'KOMPLEXE DYNAMIK', (16 Jan. -3 Febr.: 1984). *Harmonie in Chaos und Kosmos*; Bilder aus der Theorie dynamische Systeme, Katalog einer Ausstellung in die Sparkasse in Bremen.
45. J.C. FRAUENTHAL (April 1980). Change in applied mathematics is revolutionary. *SIAM News.*
46. T. GEISEL (1982). Chaos, randomness and dimension. *Nature 298,* 322-323. (How to distinguish chaos from stochastically induced irregular behaviour.)
47. J. GLIMM (1965). Solutions in the large for nonlinear hyperbolic systems of equations. *Comm. Pure and Appl. Math. 18,* 697-716.
48. H.H. GOLDSTINE, J. VON NEUMANN (1963). On the principle of large scale computing machines, section 2: Importance to mathematics. *Collected Works of J. von Neumann, Vol. 5,* 1-32, MacMillan.
49. R.E. GOMORY (1966). The traveling salesman problem. *Proc. IBM Scientific Computing Symposium on Combinatorial Problems,* IBM, 93-121.
50. R. GORE a.o. (1983). The once and future universe. *National Geographic 163:6,* 704-749.
51. J.M. GREENE, R.S. MACKAY, F. VIVALDI, M.J. FEIGENBAUM (1981). Universal behaviour in families of area preserving maps. *Physica 3D,* 468-486.
52. U. GRENANDER (1982). *Mathematical Experiments on the Computer,* Acad. Pr..
53. F.A. GRÜNBAUM (1982). The limited angle reconstruction problem. *Proc. Symp. Appl. Math. 27,* 43-61.
54. J. GUCKENHEIMER (1982). Noise in chaotic systems. *Nature 298,* 358-361.
55. I. GUMOVSKI, C. MIRA (1980). *Dynamique Chaotique.* Cepadues Ed..

56. H. HAHN (1956-1976). The crisis in intuition. J.R. NEWMAN (ed.). *The World of Mathematics,* Simon & Schuster, 1956.
57. D.R. HAMANN (May 1983). Computers in physics: an overview. *Physics Today,* 24-33.
58. M. HAZEWINKEL (1978). *Formal Groups and Applications,* Acad. Pr..
59. M. HAZEWINKEL (1973). Wiskunde als laboratorium. *Inaugural Address,* Rotterdam.
60. J.E. HIRSCH, D.J. SCALAPINO (May 1983). Condensed matter physics. *Physics Today,* 44-52.
61. T. HOGG, B.A. HUBERMAN (1983). Quantum dynamics and nonintegrability. *Phys. Rev. A 28:1,* 22-31.
62. W.G. HOOVER (Jan. 1984). Computer simulation of many-body dynamics. *Physics Today, 44-50.*
63. J.J. HOPFIELD, D.I. FEINSTEIN, R.G. PALMER (1983). Unlearning has a stabilizing effect in collective memories. *Nature 304,* 158-159.
64. IEEE (1983). *Proc. IEEE,* special issue on computerized tomography.
65. R. ISAACS (1979). On applied mathematics. *J. Opt. Th. and Appl. 27,* 31-50.
66. L.J. DE JONGH (1983). Solitonen een doorbraak in de niet-lineare physica. *Ned. Tijdschrift v. Natuurkunde A 49 (1),* 8-17.
67. W.J.M. DE JONGE, K. KOPINGA, A.M.C. TINUS (1983). Solitonen in eendimensionale magnetische systemen. *Ned. Tijdschrift v. Natuurkunde 4 A 49 (1),* 25-30.
68. J.F. KAASHOEK (1983). *Thesis,* Erasmus University Rotterdam.
69. L.H. KARSTEN, J. SMITH (1981). *Nuclear Physics B183,* 103.
70. P.H.M. KERSTEN, P.K.H. GRAGERT (1983). *Symbolic Integration of Overdeterminal Systems of Partial Differential Equations* (with applications to the computation of infinitesimal symmetries of nonlinear equations), Preprint 430, Twente Techn. Univ., Dept. Math..
71. P.H.M. KERSTEN (1982). *Infinitesimal Symmetrics and Conserved Currents for Nonlinear Dirac Equations,* Preprint, 398, Twente Techn. Univ., Dept. Math..
72. E. KIBBLEWHITE (1982). Counting the stars by computer. *New Scientist,* 478-482.
73. O.E. LANFORD (1981). Smooth transformations of intervals. *Sem Bourbaki 1980/81, Exp. 563, Lect. Notes Math. 901,* 36-54.
74. P.D. LAX (1954). Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Comm. pure and Appl. Math. 7,* 159-194.
75. P.D. LAX, B. WENDROFF (1960). Systems of conservation laws. *Comm. pure and Appl. Math. 13,* 217-238.
76. H.W. LENSTRA, JR. (1983). Fast prime number tests. *Nieuw Archief v. Wiskunde (4) 1,* 133-144.
77. D.E. LITTLEWOOD, A.R. RICHARDSON (1934). Group characters and algebra. *Philos. Trans. Roy. Soc. London A 233,* 99-142.

78. E. Lorenz (1963). Deterministic nonperiodic flow. *J. Atmosph. Sci. 20,* 130-141.
79. J. v.d. Lune, H. te Riele (1983). *On the Zeros of the Riemann Zeta Function in the Critical Strip III,* Preprint NW 146/83, CWI, Amsterdam.
80. J. Maddox (1983). How to do physics by numbers. *Nature 303,* 279.
81. M. Malek Mansour a.o. (1981). Asymptotic properties of Markovian master equations. *Ann. of Physics 131, 2,* 283-313.
82. Th. H. Maugh II (1983). Catalysis: no longer a black art. *Science 219,* 474-477.
83. R.M. May (1983). Book review of S.L. Pimm, Food webs. *Science,* 295-296.
84. P.H.E. Meijer, R.D. Mountain, R.J. Soulen, Jr. (1981). *Proc. of the 6-th Int. Conf. on Noise in Physical Systems,* Nat. Bureau of Standards, US Government Printing office.
85. H. Meinhardt (1982). *Models of Biological Pattern Formation,* Acad. Pr..
86. M. Metropolis, M.L. Stein, P.R. Stein (1973). On finite limit sets for transformations of the unit interval. *J. Comb. Theory 15,* 25-44.
87. D.R. Moore, J. Toomre, E. Knobloch, N.O. Weiss (1983). Period doubling and chaos in partial differential equations for thermosolutal convection. *Nature 303,* 663-667.
88. Y. Murakami (1968). *Logic and Social Choice,* Routledge & Kegan Paul Ltd..
89. S.R. Nagel, G.S. Crest, A. Rahman (Oct. 1983). Quench Echos. *Physics Today,* 24-32.
90. R.D. Nussbaum, M.-O. Peitgen (1983). *Special and Spurious Solutions of* $x'(t) = -\alpha F(x(t-1))$, Forschungsschwerpunkt Dyn. Systeme, Bremen, Report 91.
91. H.-O. Peitgen (1982). *Phase Transitions in the Homoclinic Regime of Area Preserving Diffeomorphisms,* Forschungsschwerpunkt Dyn. Syst., Bremen, Rep. 68.
92. H.-O. Peitgen, D. Saupe, F. v. Haeseler (1983). *Newton's Method and Julia Sets,* Preprint 104 Forschungsschwerpunkt Dynamische Systeme, Univ. Bremen.
93. S.L. Pimm (1982). *Food Webs,* Chapman and Hall.
94. I. Prigogine, G. Nicolis (1985). Self- organization in nonequilibrium systems: towards a dynamics of complexity. M. Hazewinkel, J.H. C. Paelinck, R. Jurkoviç (eds.). *Bifurcation: Analysis, Principles and Synthesis,* Reidel Publ. Co.
95. J. Radon (1917). Ueber die Bestimmung von Funcktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Sächsische Berichte Akad. der Wiss. 69,* 262-277.
96. B.D. Ripley (1981). *Spatial Statistics,* Wiley.
97. D. Ruelle, F. Takens (1971). On the nature of turbulence. *Comm. Math. Physics 20,* 167-192: ibid. *23,* 343-344.

98. J.A. SANDERS (1982). *The (driven) Josephson Equation. An Exercise in Asymptotics,* Rapport 220, Math. Sem. Vrije Univ. A'dam.
99. W.A. SCHLUP (1974). I-V Characteristics and stationary dynamics of a Josephson-juction including the interference term in the current phase relation. *J. Phys. C. Solid State Phys. Vol. 17,* 736-748.
100. A. SEN (1970). *Collective Choice and Social Welfare.* Holden-Day.
101. R. SHAW (1981). Strange attractors, chaotic behaviour and information flow. *Z. Naturforsch. 36,* 80-112.
102. H.W. SIEGBERG (1983). *Chaotic Behaviour of a Class of Nonlinear Differential Delay Equations,* Fachber. Dyn. Systeme, Report 90, Bremen.
103. J. SILK (1983). Deciphering the cosmic code. *Nature 304,* 304-305.
104. YA. G. SINAI, E.B. VUL (1980). Discovery of closed orbits of dynamical systems with the use of computers. *J. Stat. Phys. 23,* 27-47.
105. D.L. SLOTNICK, A. SANICH (1978). Numerical calculation and computer design. *Comp. & Maths. with Appl. 3,* 201-220.
106. F. SULLIVAN, D. KAHANER, H.A. FOWLER, J. KNAPP-CORDES (1983). *Wave Form Simulations for Josephson-junction Circuits Used for Noise Thermometry.* Nat. Bur. Standards, Report NBSIR 83-2643.
107. F. SULLIVAN (Draft, 19 May 1983). Remarks on algorithm design for particle simulations. (Particles with potential interaction attractive at large distances repelling at small.) No results reported. Methodology interesting (at say 10^3 particles).
108. P. SWINNERTON-DYER (1967). The conjectures of Birch and Swinnerton-Dyer and of Tate. T.A. SPRINGER (ed.). *Proc. of a Conf. in Local Fields,* Springer.
109. J. TOWBER (1982). Book review of G. James, A. Kerber, The representation theory of the symmetric groups. *Bull. Amer. Math. Soc. 8:2,* 357-363.
110. G.E. UHLENBECK (1925). Over een stelling van Lorentz en haar uitbreiding voor meer-dimensionale ruimten. *Physica 5,* 423-428.
111. S.M. ULAM (1976). *Adventures of a Mathematician,* Scribner.
112. G.W. WETHERILL (1981). Formation of the earth from planetisimals. *Sci. Amer.,* 130-141.
113. M. WIDOM, L.P. KADANOFF (1982). Renormalization analysis of bifurcations in area preserving maps. *Physica 5D,* 287-292.
114. S. WILSON (1982). Chemistry by computer. *New Scientist,* 576-579. Ab-initio calculation of the pharmacological properties of molecules and hence the design of pharmacological molecules is a thriving business.
115. W.W. WOOD, J.D. JACOBSON (1957). Preliminary results from a recalculation of the Monte-Carlo equation of state of hard spheres. *J. Chem. Phys. 27,* 1207-1208.
116. N. ZABUSKY (1981). Computational synergetics and mathematical innovation. *J. Comp. Phys. 43,* 195-249.
117. N. ZABUSKY (Febr. 1983). *Letter to Physics Today,* 95-96.

118. E. ZAGER, D. TABAK (1983). Flexible architecture microcomputer design. *Microprocessing and Microprogramming 11,* 313.

Numerical Analysis of the Shallow Water Equations

P.J. van der Houwen, B.P. Sommeijer,
J.G. Verwer, F.W. Wubs
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

1. Introduction

In this contribution we will give an expository account of the numerical analysis of hyperbolic differential equations. Recently, these equations have become of particular interest to the Numerical Mathematics Department of the CWI. Our main purpose is to apply this analysis to the *shallow water equations* (SWEs) and therefore, throughout this paper, we will illustrate the analysis by giving theoretical as well as numerical results for the SWEs. In this introductory section we start with a description of the origin of the SWEs.

A windfield (or tidal forces) perturbing a water surface which is initially at rest will generate two types of water waves: long (or tidal) waves and short waves. In the long waves the wave length is large compared with the height of the water surface and the vertical accelerations are small compared with the horizontal accelerations. In the short waves the wave length is smaller than the depth of the water and the vertical accelerations are no longer insignificant. We will concentrate on *long waves in shallow water* generated by wind forces (or tidal forces).

Due to the movement of the water, three other forces will become active: (i) bottom friction (ii) Coriolis force (iii) gravity. Let $\mathbf{R}$ denote the total resulting horizontal force, then we have the following equation

$$\frac{D\mathbf{u}}{Dt} := \frac{\partial \mathbf{u}}{\partial t} + \left[u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} \right]\mathbf{u} = \mathbf{R}, \tag{1.1a}$$

where $\mathbf{u}=(u,v)^T$ denotes the depth-averaged velocity of the water and (x,y) represents an orthogonal coordinate system. In addition to this equation of motion we have the continuity equation (e.g. [7, p. 179])

$$\frac{\partial h}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0 \tag{1.1b}$$

where h denotes the depth. Combining these equations and deriving expressions for the forces due to gravity and Coriolis (see e.g. [7, p. 190] we arrive at the SWEs:

$$\mathbf{w}_t = A(\mathbf{w})\mathbf{w}_x + B(\mathbf{w})\mathbf{w}_y + C(y)\mathbf{w} + \mathbf{r}(\mathbf{w}) \tag{1.2}$$

where $\mathbf{w}=(u,v,h)^T$ and where the matrices A,B and C are defined by

$$A(\mathbf{w}) = -\begin{bmatrix} u & 0 & g \\ 0 & u & 0 \\ h & 0 & u \end{bmatrix}, \quad B(\mathbf{w}) = -\begin{bmatrix} v & 0 & 0 \\ 0 & v & g \\ 0 & h & v \end{bmatrix}, \quad C(y) = \begin{bmatrix} 0 & f & 0 \\ -f & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The vector $\mathbf{r}(\mathbf{w})$ represents the bottom friction, forces due to bottom irregularities, to the wind field and other atmospheric forces (for a discussion of this term we refer to [7]); f and g denote the Coriolis parameter and the acceleration due to gravity, respectively.

Omitting the external forces represented by $\mathbf{r}(\mathbf{w})$ we obtain a system of *hyperbolic* equations *in two space dimensions* (notice that the matrices A and B have the distinct, *real* eigenvalues $\{-u\pm\sqrt{gh},-u\}$ and $\{-v\pm\sqrt{gh},-v\}$, respectively; these eigenvalues correspond to the characteristic directions of the SWEs).

A particularly difficult problem in deriving a mathematical model for a shallow sea is the formulation of the boundary conditions along the 'non-coastal' boundaries of the sea (along 'coastal' boundaries one usually imposes the 'rigid wall' condition which requires that the velocity component normal to the coast vanishes). For a discussion of boundary conditions we refer to [7]. In our examples we will use periodic boundary conditions along the non-coastal boundaries.

2. The space-discretization

A flexible approach in the numerical solution of evolutionary problems in PDEs is obtained by applying the so-called *method of lines.* Herewith the numerical solution process is considered as to consist of two parts, viz. *space-discretization* and *time-integration.*

In the space-discretization the PDE is converted into a system of ODEs by discretizing the space variables, while the time variable is left continuous. Usually, the space-discretization is performed, either by the finite difference method [32], or by the finite element method [42]. Spectral methods can also be applied, however [10]. In this paper we restrict our attention to the finite difference method since this method is easier to present to the nonspecialist. Moreover, in the field of hyperbolic PDEs the finite difference method is still most widely used. We note that Praagman [35] has implemented the finite element method for the shallow water equations (1.2).

In the time-integration the resulting system of ODEs, often called the semi-discrete problem, is integrated by one of the many existing integration formulas which is most appropriate for the problem at hand. This part of the discretization process will be the subject of Section 3.

2.1. *Two simplified models*

Throughout this contribution we will discuss examples and numerical experiments with the aim of illustrating the various aspects and difficulties which are encountered in the numerical solution of hyperbolic problems such as (1.2). For that purpose we will resort to two simplified models which we give first.

Model 1. A conservative shallow water equation

Following [12] we consider the nonlinear hyperbolic system

$$\mathbf{w}_t = A(\mathbf{w})\mathbf{w}_x + B(\mathbf{w})\mathbf{w}_y + C(y)\mathbf{w}, \quad (x,y)\in\Omega, \quad t\geqslant 0, \tag{2.1.1}$$

for the dependent vector variable $\mathbf{w}=[u,v,\phi]^T$, where u and v have the same meaning as in (1.2) and $\phi=2\sqrt{gh}$. Further

$$A(\mathbf{w})=-\begin{bmatrix} u & 0 & \frac{1}{2}\phi \\ 0 & u & 0 \\ \frac{1}{2}\phi & 0 & u \end{bmatrix}, \quad B(\mathbf{w})=-\begin{bmatrix} v & 0 & 0 \\ 0 & v & \frac{1}{2}\phi \\ 0 & \frac{1}{2}\phi & v \end{bmatrix}, \quad C(y)=\begin{bmatrix} 0 & f & 0 \\ -f & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

It can be verified, using a simple transformation, that (2.1.1) can be obtained from (1.2), provided all external forces except the Coriolis force f are neglected. We prefer to use the same notation $\mathbf{w}$ for the dependent variable, although its third component has a different meaning than in (1.2). Any confusion is precluded. We observe that in the numerical solution process the treatment of the external forces is relatively simple. Hence, unless otherwise stated, we neglect these forces in our examples and experiments.

An important tool in the analysis of evolutionary problems in PDEs, analytically as well as numerically, is the total energy integral. For (2.1.1) the total energy can be expressed as

$$E(t) = \frac{1}{2g}\int\int_{\Omega}(u^2+v^2+\tfrac{1}{4}\phi^2)\tfrac{1}{4}\phi^2\,dxdy. \tag{2.1.2}$$

The origin of the name energy integral is that in many applications the physical energy of the physical system underlying to the partial differential equations can be expressed by an integral expression. This expression, in turn, is a convenient tool for examining well-posedness questions [36]. For example, if the physical energy is conserved, E must remain constant in time. If energy dissipates, E should monotonically decrease in time. E is also useful for finding sensible boundary conditions.

Following the aforementioned authors, we define the rectangle

$$\Omega = \{(x,y):0\leqslant x\leqslant L, \quad 0\leqslant y\leqslant D\}. \tag{2.1.3}$$

Then, using the boundary conditions,

$$\mathbf{w}(x,y,t) = \mathbf{w}(x+L,y,t), \quad v(x,0,t)=v(x,D,t)=0, \tag{2.1.4}$$

a straightforward computation reveals that $\dot{E}(t)=0$, i.e., these boundary conditions imply that (2.1.1) conserves the total energy $E(t)$. The conservation of

energy property should be accounted for in the discretization of (2.1.1). Observe that (2.1.4) implies periodicity in the x-direction and that no boundary conditions are necessary for u and ϕ at $y=0$, D. Other boundary conditions may also lead to well-posedness of (2.1.1) on $\Omega\times(t>0)$. It is also of interest to observe that if we add bottom friction to (2.1.1), i.e., if we replace the matrix C by

$$C = \begin{bmatrix} -\lambda & f & 0 \\ -f & -\lambda & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \lambda=\lambda(x,y,w)>0,$$

it follows that $\dot{E}(t)<0$, which means energy dissipation. Finally, for the general problem (1.2) it may happen that E will increase in time due to the influence of the source term or energy transport through boundaries.

For future reference we list some specific problem parameters for the just described initial-boundary value problem [11]:

Coriolis force $f=\hat{f}+\beta(y-D/2)$
Initial height function

$$h(x,y) = H_0+H_1\tanh(9(D/2-y)/2D)+H_2\operatorname{sech}^2(9(D/2-y)/D)\sin(2\pi x/L)$$

Initial velocities $u=-gf^{-1}\partial h/\partial y$, $v=gf^{-1}\partial h/\partial x$

$L=6.0\,10^6$ m, $D=4.4\,10^6$ m, $\hat{f}=10^{-4}$ sec^{-1}, $\beta=1.5\,10^{-11}$ sec$^{-1}m^{-1}$, $g=10$ m sec^{-2}, $H_0=2000$ m, $H_1=220$ m, $H_2=133$ m.

Model 2. A one-dimensional incompressible flow equation
Consider the one-space dimensional hyperbolic system

$$\begin{aligned} \mathbf{w}_t &= B(\mathbf{w})\mathbf{w}_y, \quad y\in\Omega=[0,D], \quad t\geqslant 0, \\ B(\mathbf{w}) &= -\begin{bmatrix} v & \frac{1}{2}\phi \\ \frac{1}{2}\phi & v \end{bmatrix}, \quad \mathbf{w}=[v,\phi]^T \end{aligned} \tag{2.1.5}$$

which is obtained from equation (2.1.1) by dropping the x-dependent term $A(\mathbf{w})\mathbf{w}_x$ and the term $C(y)\mathbf{w}$. We again impose the boundary condition (cf. (2.1.4))

$$v(0,t) = v(D,t)=0, \tag{2.1.6}$$

while no condition for ϕ is prescribed. Hence, for ϕ the boundaries $y=0,D$ are open. The total energy integral $E(t)$ now reads (apart from a constant)

$$E(t) = \int_0^D (v^2+\tfrac{1}{4}\phi^2)\tfrac{1}{4}\phi^2\,dy, \tag{2.1.7}$$

and again we have conservation of total energy, i.e., $\dot{E}(t)=0$ for $t\geqslant 0$. Numer-

ically this one-space dimensional flow problem has similar properties as problem (2.1.1)-(2.1.4).

2.2. Finite difference space-discretization

Let

$$\mathbf{w}_t = \mathfrak{F}(x,y,t,\frac{\partial}{\partial x},\frac{\partial}{\partial y})\mathbf{w}, \quad (x,y)\in\Omega, \quad t\geqslant 0, \tag{2.2.1}$$

or, in case of one-space dimension,

$$\mathbf{w}_t = \mathfrak{F}(x,t,\frac{\partial}{\partial x})\mathbf{w}, \quad x\in\Omega, \quad t\geqslant 0, \tag{2.2.1'}$$

formally represent a system of hyperbolic equations, given on the space domain Ω. Suppose that on the boundary $\partial\Omega$ of Ω correct boundary conditions

$$B(\mathbf{w},t) = 0, \quad \text{on} \quad \partial\Omega \tag{2.2.2}$$

are defined. The space-discretization of this problem essentially consists of three steps: (i) A grid Ω_Δ must be defined covering $\Omega\cup\partial\Omega$. We use the symbol Δ as a formal notation for the grid distance which, of course, may vary over Ω_Δ. (ii) Appropriate finite difference replacements for the operators $\partial/\partial x$, $\partial/\partial y$ must be selected at all points of Ω_Δ. (iii) The boundary conditions must be taken into account.

EXAMPLE 2.2.1. The standard finite difference space-discretization of our second model (2.1.5) proceeds as follows. The interval $[0,D]$ is divided into N_y subintervals of equal length Δy, thus defining the grid

$$\{y_k : y_k = k\Delta y \quad \text{for} \quad k=0(1)N_y\}.$$

On this grid we introduce the so-called grid function

$$\mathbf{W} = [\mathbf{W}_0, \ldots, \mathbf{W}_{N_y}]^T, \quad \mathbf{W}_k = [V_k,\Phi_k]^T,$$

where each component vector $\mathbf{W}_k(t)=[V_k(t),\Phi_k(t)]^T$ depends on time t and is meant to approximate $\mathbf{w}(y,t)$, the exact solution of problem (2.1.5)-(2.1.6), at the gridpoint y_k. Hence $\mathbf{W}$ is still time continuous. The approximation is defined by the choice of the finite difference formulas for approximating the space derivative $\mathbf{w}_y$. At this place we have to face our first difficulty, i.e. the open boundary for ϕ which forces us to approximate v_y in a different way at the points $y_0=0$, $y_{N_y}=D$. We consider the standard second order difference formula

$$\mathbf{w}_y(y_k,t) \simeq \frac{1}{2\Delta y}(\mathbf{W}_{k+1}(t)-\mathbf{W}_{k-1}(t)) \tag{2.2.3}$$

for $k=1(1)N_y-1$. For $k=0$, N_y define $V_k(t)=0$ according to the boundary conditions (2.1.6) and use one-sided first order differences for v_y at these points, i.e.,

$$v_y(0,t) \simeq \frac{1}{\Delta y}\,(V_1(t)-V_0(t)),$$
$$v_y(D,t) \simeq \frac{1}{\Delta y}\,(V_{N_y}(t)-V_{N_y-1}(t)). \tag{2.2.4}$$

After replacing w_y on $\{y_k\}$ by these finite difference quotients, the system of ODEs

$$\dot{\mathbf{W}}_0 = B(\mathbf{W}_0)\frac{\mathbf{W}_1-\mathbf{W}_0}{\Delta y}, \quad V_0(t)=0,$$
$$\dot{\mathbf{W}}_k = B(\mathbf{W}_k)\frac{\mathbf{W}_{k+1}-\mathbf{W}_{k-1}}{2\Delta y}, \quad k=1(1)N_y-1, \tag{2.2.5}$$
$$\dot{\mathbf{W}}_{N_y} = B(\mathbf{W}_{N_y})\frac{\mathbf{W}_{N_y}-\mathbf{W}_{N_y-1}}{\Delta y}, \quad V_{N_y}(t)=0,$$

results. This system is a time continuous, semi-discrete version of the original initial-boundary value problem (2.1.5)-(2.1.6). □

The above example illustrates that the process of space-discretization converts an initial-boundary value problem for a PDE into an initial value problem for a system of ODEs with t as independent variable. Henceforth we will denote this latter system by

$$\dot{\mathbf{W}} = \mathbf{F}(t,\mathbf{W}), \quad t\geqslant 0, \quad \mathbf{W}(0) \text{ prescribed.} \tag{2.2.6}$$

This system is usually called the time-continuous, semi-discrete system. Obviously, there is an intimate relationship with the grid Ω_Δ. The vector function $\mathbf{F}$ is always parameterized with the grid distance Δ. $\mathbf{F}$ approximates the hyperbolic operator $\mathcal{F}$ on the grid Ω_Δ. The length of the vector $\mathbf{W}$, the gridfunction which approximates w on Ω_Δ, depends on Δ too. Occasionally, if this clarifies the discussion, we will therefore use the notation

$$\dot{\mathbf{W}}_\Delta = \mathbf{F}_\Delta(t,\mathbf{W}_\Delta) \tag{2.2.6′}$$

instead of (2.2.6). Further, we shall mostly use the autonomous notation $\dot{\mathbf{W}}=\mathbf{F}(\mathbf{W})$ as our two example models are autonomous.

As a further illustration we describe the space-discretization of our first model (2.1.1)-(2.1.4). Because Ω is a rectangle the derivation is nearly the same as in Example 2.2.1.

EXAMPLE 2.2.2. Divide the x-interval and y-interval into N_x and N_y subintervals of length Δx and Δy, respectively. On the grid

$$\{(x_j,y_k) : x_j=j\Delta x,\ j=1(1)N_x \text{ and } y_k=k\Delta y,\ k=0(1)N_y\},$$

we define $\mathbf{W}_{jk}=[U_{jk},V_{jk},\Phi_{jk}]^T$ as the time continuous approximation for $\mathbf{w}(x_j,y_k,t)$ which results from the application of second order symmetrical differences at all interior points and first order one-sided differences at the boundary points (x_j,y_k), $k=0,N_y$. In the x-direction symmetrical differencing

is possible everywhere because of the periodicity, i.e., $\mathbf{W}_{0k} = \mathbf{W}_{N_x,k}$ and $\mathbf{W}_{N_x+1,k} = \mathbf{W}_{1k}$. Note that $V_{j0} = V_{jN_y} = 0$ due to (2.1.4). □

Grid staggering. Grid staggering, originally introduced by HANSEN [13], is often applied in space-discretization. By this technique u, v and ϕ are calculated at different grid points. Herewith, it is possible to decrease the storage requirements by a factor four without loss of accuracy with respect to the main terms of the SWEs. We will show the idea using the one-dimensional equation (2.1.5) of which the main part is described by:

$$\begin{aligned} v_t &= -\tfrac{1}{2}\phi_0\phi_y \\ \phi_t &= -\tfrac{1}{2}\phi_0 v_y, \end{aligned} \tag{2.2.7}$$

where we have frozen the coefficients of ϕ_y and v_y. If this system is semi-discretized in the usual way, we obtain

$$\begin{aligned} (V_t)_i &= -\tfrac{1}{2}\Phi_0(\Phi_{i+1} - \Phi_{i-1})/2\Delta y, \\ (\Phi_t)_j &= -\tfrac{1}{2}\Phi_0(V_{j+1} - V_{j-1})/2\Delta y. \end{aligned} \tag{2.2.8}$$

Observe that in the case where i is running through even values and j through odd values, the set of equations is independent of its complement $\{i\ odd,\ j\ even\}$. Thus we may omit one of these sets, without loss of accuracy, thereby reducing the number of equations by a factor two. Applying the same technique in the y-direction will lead to a final reduction by a factor four. A part of the resulting grid is depicted in Fig. 2.2.1.

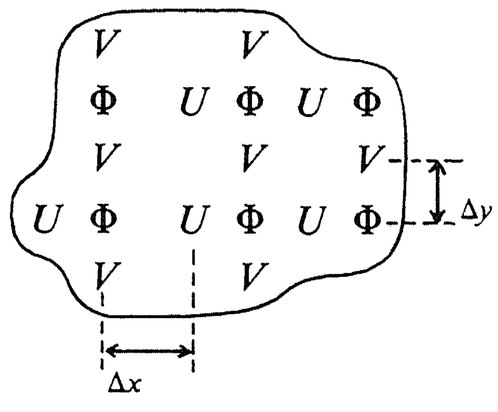


FIGURE 2.2.1

The neglected terms have to be composed by use of the variables of the reduced set, implying a small loss of accuracy due to averaging techniques.

2.3. Some fundamental topics

The choice of a difference method for discretizing hyperbolic equations, such as the shallow water equations, depends on many factors. These may vary from theoretical to practical and will always depend on the specific application area (see e.g. [33], p. 718). In this subsection we will briefly discuss some basic topics concerning the space-discretization. These include *consistency, stability, conservation laws, open boundaries, curved boundaries.* Our purpose is to give some insight in the choice and analysis of finite difference models. It is emphasized that the field is so diverse that completeness is impossible in the present paper. The above topics, however, play a role in a lot of investigations and are of a fundamental nature.

Consistency. The approximation is useful only if it is consistent, i.e., upon grid refinement the approximation should converge to the continuous problem. Normally there is no difficulty in setting up a consistent approximation. A difficulty may lie in finding approximations which converge sufficiently fast if $\Delta \to 0$. Further, an always returning and important question is, how accurate is the numerical solution computed on a certain grid? We will briefly consider these matters for the space-discretization error.

Consider a general initial-boundary value problem (2.2.1)-(2.2.2) and a corresponding semi-discrete approximation (2.2.6′). Let $\mathbf{w}_\Delta$ denote the restriction of the fully continuous function $\mathbf{w}$ to the space grid. Hence, in the case of Example 2.2.1, we have

$$\mathbf{w}_\Delta(t) = [\mathbf{w}(0,t), \ldots, \mathbf{w}(k\Delta y,t), \ldots, \mathbf{w}(N_y\Delta y,t)]^T.$$

Further, let α_Δ denote the *space-approximation error*

$$\alpha_\Delta = \mathbf{F}_\Delta(\mathbf{w}_\Delta) - \dot{\mathbf{w}}_\Delta. \tag{2.3.1}$$

This error is obtained by substituting the exact solution $\mathbf{w}$ into the semi-discrete problem. It measures how much the semi-discrete operator deviates from the partial differential operator including the boundary conditions. Next, let η_Δ denote the *space-discretization error*

$$\eta_\Delta = \mathbf{W}_\Delta - \mathbf{w}_\Delta. \tag{2.3.2}$$

It follows that η_Δ is a solution of the ordinary differential system

$$\dot{\eta} = \mathbf{F}_\Delta(\mathbf{w}_\Delta + \eta_\Delta) - \mathbf{F}_\Delta(\mathbf{w}_\Delta) + \alpha_\Delta,$$

which can be rewritten to

$$\dot{\eta}_\Delta = M_\Delta(t)\eta_\Delta + \alpha_\Delta, \tag{2.3.3}$$

$$M(t) = \int_0^1 F'_\Delta(\mathbf{w}_\Delta + \theta\eta_\Delta)\,d\theta.$$

Here F'_Δ denotes the Jacobian matrix of the vector function $\mathbf{F}_\Delta$ which is assumed to exist. Note that we have used the mean value theorem for vector functions [28, p. 71].

The above derivation reveals three properties of the space error η_Δ which are worth mentioning. (i) Though η_Δ comes into existence only by discretizing space variables, this error is really time-dependent, even when $\alpha_\Delta \neq \mathbf{0}$ is constant. (ii) The space error depends on the stability behaviour of the ordinary differential equation (2.3.3) when proceeding in time. Evidently, this equation should possess similar stability properties as the underlying partial differential equation. (iii) The smaller the approximation error α_Δ, the smaller the space error η_Δ, certainly if (2.3.3) is a stable system. Hence, if the approximation is consistent, i.e., $\alpha_\Delta \to \mathbf{0}$ if $\Delta \to 0$, the space error η_Δ will converge to zero, for all t, upon grid refinement.

Consistency and stability. To clarify the aspect of stability in the above reasoning we will give a typical stability estimate for η_Δ. This stability estimate gives insight in the dependence of the space error η_Δ on the approximation error (2.3.1).

Let $\|\cdot\|$ be some norm on the finite dimensional solution space of the system $\dot{\mathbf{W}}_\Delta = \mathbf{F}_\Delta(\mathbf{W}_\Delta)$, e.g., a known l^p-norm. Then, according to [2, p. 13], it follows that

$$\|\eta_\Delta(t)\| \leqslant e^{\mu_{\max} t}\|\eta_\Delta(0)\| + \int_0^t e^{\mu_{\max}(t-\tau)}\|\alpha_\Delta(\tau)\| d\tau,$$

where

$$\mu_{\max} = \max_{\mathbf{W}} \mu[F'_\Delta(\mathbf{W})],$$

μ being the logarithmic matrix norm belonging to $\|\cdot\|$ (for specific details about this result and the use and meaning of the logarithmic norm, the reader may also consult [6]). Let us assume a zero space error at the initial time $t = 0$. Then

$$\|\eta_\Delta(t)\| \leqslant \int_0^t e^{\mu_{\max}(t-\tau)}\|\alpha_\Delta(\tau)\| d\tau. \tag{2.3.4}$$

In many instances the quantity $\mu_{\max}$ can be proved to be independent of the grid distance Δ. In that case, this worst case estimate proves that

$$\|\eta_\Delta(t)\| \leqslant c(t) \max_t \|\alpha_\Delta(t)\|,$$

$c(t)$ being independent of Δ. Consequently, if the finite difference formula in all gridpoints is consistent of order q, i.e., in a formal notation,

$$\alpha_\Delta = O(\Delta^q), \quad \Delta \to 0, \tag{2.3.5}$$

it follows that $\eta_\Delta(t) = O(\Delta^q)$ as $\Delta \to 0$, establishing q-th order convergence for the semi-discrete solution $\mathbf{W}(t)$. Apparently, in the above derivation time t was kept fixed, i.e., (2.3.5) applies for all t but the constant involved still depends on t[1].

1. More details concerning the actual application of the above derivation can be found in 'Convergence of Methods of Lines Approximations to PDEs', J.G. Verwer and J.M. Sanz-Serna, CWI Report NM-R8404.

EXAMPLE 2.3.1. Let us examine the approximation error α_Δ for the semi-discrete one-dimensional incompressible flow equation (2.2.5). We will denote the k-th component of α_Δ by $\alpha_{\Delta,k}$. If $\mathbf{w}$ is at least two times differentiable, a straightforward Taylor expansion of $\mathbf{w}(y_k \pm \Delta y, t)$ at all grid points y_k shows that

$$\alpha_{\Delta,O} = O(\Delta y),\ \alpha_{\Delta,k} = O((\Delta y)^2)\ (1 \leqslant k \leqslant N_y - 1),\ \alpha_{\Delta,N_y} = O(\Delta y)$$

as $\Delta y \to 0$. Consequently, due to the first order approximations at the boundary, $q = 1$ in relation (2.3.5) instead of $q = 2$.

A decrease of accuracy at a boundary may be reduced by using a higher-order difference formula. This may, however, destroy the stability of the space-discretization. In our terminology this means that the error equation (2.3.3) becomes unstable. We will illustrate this later. First we proceed with the topic conservation laws which provides us further means for examining stability. □

Conservation laws and stability. Let us once more consider the stability estimate (2.3.4) for the differential system (2.3.3) which determines the space error η_Δ. Obviously, if it is required to solve the initial value problem over a large time interval it is highly desirable that the semi-discrete system itself is stable. Stability corresponds to a nonpositive logarithmic matrix norm, so the worst case estimate then reads

$$\|\eta_\Delta(t)\| \leqslant \int_0^t \|\alpha_\Delta(\tau)\| d\tau \leqslant t \max_{0 \leqslant \tau \leqslant t} \|\alpha_\Delta(\tau)\|. \tag{2.3.6}$$

This estimate still allows a linear growth of the space error, but should be considered as rather pessimistic. If system (2.3.3) is stable, it is to be expected that an eventual growth of η_Δ is less than the linear growth of the above estimate. Certainly this is true if $M_\Delta(t)$ is a constant matrix, i.e. if F'_Δ is constant.

One must reckon with a much more serious situation if the semi-discrete system is unstable, which corresponds to a positive $\mu_{\max}$ in the estimate (2.3.4). Then the worst case estimate allows an exponential growth of the space error which may be fatal. From practical experiences we know that exponential growth, also called 'blow up', really occurs. The next example serves to illustrate this.

EXAMPLE 2.3.2. [6]. Consider the ODE system described in Example 2.2.2 which is a semi-discrete approximation to Model 1 of subsection 2.1. On the space grid Ω_Δ we approximate the total energy $E(t)$, given by (2.1.2), by the trapezoidal approximation

$$E_\Delta = \frac{\Delta x \Delta y}{2g} \sum_{j=1}^{N_x} \{ \sum_{k=1}^{N_y - 1} \hat{\mathbf{W}}_{jk}^T \hat{\mathbf{W}}_{jk} + \tfrac{1}{2}(\hat{\mathbf{W}}_{j0}^T \hat{\mathbf{W}}_{j0} + \hat{\mathbf{W}}_{jN_y}^T \hat{\mathbf{W}}_{jN_y}) \}, \tag{2.3.7}$$

where

$$\hat{\mathbf{W}}_{jk} := [\tfrac{1}{2}U_{jk}\Phi_{jk},\ \tfrac{1}{2}V_{jk}\Phi_{jk},\tfrac{1}{4}\Phi_{jk}^2]^T,$$

U_{jk}, V_{jk} and Φ_{jk} being the components of $\mathbf{W}_{jk}$ defined in Example 2.2.2. Because we have conservation of total energy in Model 1 a legitimate requirement is that the semi-discrete model conserves the semi-discrete total energy (2.3.7), i.e., $\dot{E}_\Delta(t)=0$. It turns out that this requirement is not fulfilled. For $\Delta x=\Delta y=200$ km we have computed E_Δ over a relatively large time interval of approximately 17 days by means of a highly accurate, stable numerical integration method. Figure 2.3.1 shows a plot of E_Δ. One can see that after approximately 17 days a sudden 'blow up', or energy explosion, occurs. Of course, this explosion completely ruins all results of the numerical computation. □

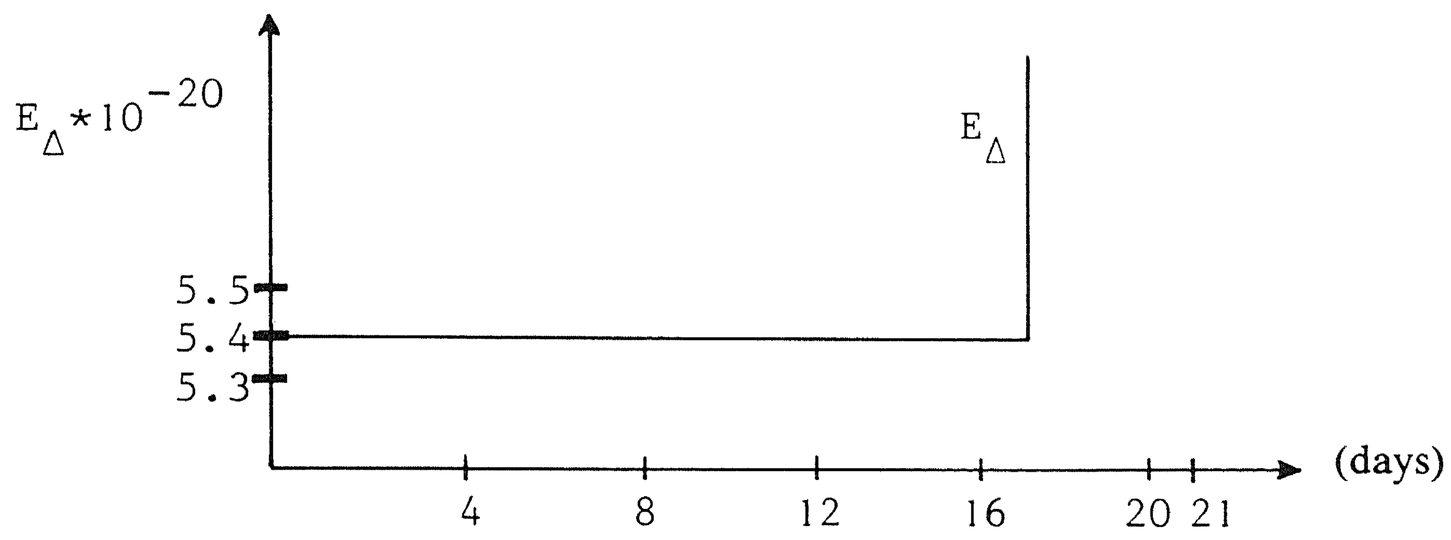


FIGURE 2.3.1. Explosion of semi-discrete total energy

From the stability estimate (2.3.4) one can deduce that the 'blow up' will be delayed if one refines Ω_Δ. In practice this is much too costly, however, and instead one imposes artificial damping or one tries to obey the conservation laws. The (ad hoc) technique of artificial damping is widely known. Alternative names are artificial dissipation or artificial viscosity. The basic idea is to add small terms to the original PDE such that the semi-discrete approximation becomes stable. Observe that bottom friction in our Model 1 also has a stabilizing influence. The inherent difficulty of this technique is that one has to make a compromise between stability and accuracy, for one changes the original PDE and thus solves a different problem. Fortunately, in many applications one is satisfied with a rough accuracy and then the technique of artificial dissipation performs quite satisfactorily.

The instability illustrated in Figure 2.3.1 is essentially due to the fact that in the space-discretization described in Example 2.2.2 the conservation of total energy $E(t)$ has not been taken into account. If the PDE conserves physical quantities such as mass, total energy, momentum, it is sensible to transfer these

properties to the finite difference approximation in order to improve it. In this connection the conservation law of total energy is an important tool for the stability analysis due to the fact that E and E_Δ can always be written as quadratic functionals. Hence if E_Δ is constant in time 'blow up' simply cannot happen. In numerical literature one has introduced the name energy method for stability analyses along the lines of energy conservation laws [36]. The energy method is of great use for examining the stability of particularly nonlinear models, since here the standard classical approach of Fourier analysis cannot be applied.

EXAMPLE 2.3.3. Following [6] we will briefly illustrate the energy method for the one-dimensional model (2.1.5) which we prefer for reasons of presentation. All results go through for the related two-dimensional model (2.1.1). Let $\dot{\mathbf{W}} = \mathbf{F}(\mathbf{W})$ denote a semi-discrete version of the PDE (2.1.5). The first step in the standard energy method is to select an appropriate energy norm, i.e., a norm such that $\|\mathbf{W}(t)\|^2 = E_\Delta(t)$. Suppose that we can deal with an inner product norm $\|\mathbf{W}\|^2 = <\mathbf{W},\mathbf{W}>$. Then, if E_Δ is required to be constant in time, say, we have

$$\frac{d}{dt}\|\mathbf{W}(t)\|^2 = <\mathbf{F}(\mathbf{W}(t)), \mathbf{W}(t)> = 0.$$

A function $\mathbf{F}$ which satisfies this property for all vectors $\mathbf{W}$ is called conservative, on the analogy of the term used for the PDE.

Now consider the energy integral (2.1.7) and define the transformation of variables $v \to \frac{1}{2}\phi v$, $\phi \to \frac{1}{4}\phi^2$. Then E is in the form of a quadratic functional, viz.

$$E(t) = \int_0^D (v^2 + \phi^2)dy.$$

Next, introduce the inner product generated by the trapezoidal rule approximation E_Δ for E:

$$<\mathbf{W},\tilde{\mathbf{W}}> = \Delta y[\sum_{k=1}^{N_y-1} \mathbf{W}_k^T\tilde{\mathbf{W}}_k + \tfrac{1}{2}(\mathbf{W}_0^T\tilde{\mathbf{W}}_0 + \mathbf{W}_{N_y}^T\tilde{\mathbf{W}}_{N_y})],$$
$$\|\mathbf{W}\|^2 = E_\Delta. \tag{2.3.8}$$

With an elementary calculation it can now be proved that space-discretization of the transformed PDE

$$v_t = -\tfrac{1}{2}v\phi^{-1/2}v_y - \tfrac{1}{2}(v^2\phi^{-1/2})_y - \phi^{1/2}\phi_y,$$
$$\phi_t = -(\phi^{1/2}v)_y, \tag{2.3.9}$$

in the same way as described in Example 2.2.1, yields a semi-discrete approximation $\dot{\mathbf{W}} = \mathbf{F}(\mathbf{W})$ which is conservative with respect to the given energy norm. This particular ODE system reads

$$\begin{aligned}
\dot{\Phi}_0 &= -\frac{1}{\Delta y}\,\Phi_1^{1/2} V_1, \\
\dot{V}_k &= D_k - \frac{1}{2\Delta y}\,\Phi_k^{1/2}(\Phi_{k+1} - \Phi_{k-1}), \quad k = 1(1)N_y - 1, \\
\dot{\Phi}_k &= -\frac{1}{2\Delta y}\,(\Phi_{k+1}^{1/2} V_{k+1} - \Phi_{k-1}^{1/2} V_{k-1}), \quad k = 1(1)N_y - 1, \\
\dot{\Phi}_{N_y} &= \frac{1}{\Delta y}\,\Phi_{N_y-1}^{1/2} V_{N_y-1},
\end{aligned} \tag{2.3.10}$$

where D_k is given by

$$D_k = -\frac{1}{4\Delta y}\,[V_k \Phi_k^{-1/2}(V_{k+1} - V_{k-1}) + (V_{k+1}^2 \Phi_{k+1}^{-1/2} - V_{k-1}^2 \Phi_{k-1}^{-1/2})].$$

See [6] for more details. □

Open boundaries and stability.. For a dependent variable a boundary is called open if no boundary condition is present. Open boundaries normally lead to inaccuracies (see the discussion in Example 2.3.1) due to the use of one-sided difference approximations which tend to be less accurate than symmetric approximations, at least for problems with smooth solutions. An additional difficulty with open boundaries is that the use of higher order one-sided approximations may turn a stable approximation into an unstable one. We will illustrate the nuisance of unstable boundary conditions for the just described PDE (2.3.9). Recall that v is zero at the boundaries $y = 0$, D, while ϕ is not prescribed. Hence for ϕ the boundary is open.

EXAMPLE 2.3.4. As shown above the semi-discrete approximation (2.3.10) conserves the semi-discrete energy E_Δ. Let us apply the usual one-sided second-order difference approximation at the boundaries, instead of (2.2.4), for approximating $(\sqrt{\phi}\, v)_y$. The first and last equation of (2.3.10) are then replaced by

$$\begin{aligned}
\dot{\Phi}_0 &= \frac{-1}{2\Delta y}\,(4\Phi_1^{1/2} V_1 - \Phi_2^{1/2} V_2), \\
\dot{\Phi}_{N_y} &= \frac{1}{2\Delta y}\,(4\Phi_{N_y-1}^{1/2} V_{N_y-1} - \Phi_{N_y-2}^{1/2} V_{N_y-2}).
\end{aligned}$$

It is straightforward to prove that now

$$\dot{E}_\Delta = \frac{1}{4\Delta y}\,[(3\Phi_{N_y-1}^{1/2} V_{N_y-1} - \Phi_{N_y-2}^{1/2} V_{N_y-2})\Phi_{N_y} + (\Phi_2^{1/2} V_2 - 3\Phi_1^{1/2} V_1)\Phi_0].$$

Hence, $\dot{E}_\Delta \neq 0$ and the energy will increase as soon as the right hand side expression becomes positive. From then on we have to face severe instabilities the origin of which lies in the use of the second order one-sided differences at the boundary points. Note that E_Δ is again constant for the (physically unrealistic) boundary condition $\Phi = 0$. □

Curved boundaries. The domain Ω of our first model in subsection 2.1 is the rectangle (2.1.3). Finite differences are easily implemented on such simple domains. In applications, however, Ω may be rather irregular leading to curved boundaries. For example, part of $\partial\Omega$ may consist of a curved coastline. It shall be clear that such a domain is poorly approximated by an orthogonal grid. This poor representation of Ω will cause larger approximation errors α_Δ near the boundary $\partial\Omega$ than in the interior of the domain. These larger approximation errors, in turn, may increase the space approximation error η_Δ over a considerable part of Ω, if not the whole of Ω. To some extent it depends on the application whether this specific error increase is unacceptable. For in many practical computations the physical data, for example at a boundary, already contain inaccuracies which overshadow numerical errors due to a bad boundary representation or other numerical errors. In such applications one is satisfied with low accuracy finite difference models and orthogonal grids are still useful.

A cure for the above mentioned boundary inaccuracies is the use of curvilinear grids. Gridlines then can be chosen coincident with boundaries leading to a significantly more accurate discretization of the domain Ω. Clearly, the use of curvilinear grids does complicate the implementation of finite differences. Already the creation of Ω_Δ itself may become very cumbersome. For that reason one has developed so-called grid or mesh generators, computer programs which assist the engineer in setting up nonrectangular grids without irregularities such as too small corners between grid lines.

Loosely speaking, the derivation of approximations for partial derivatives on nonrectangular grid-elements are always based on a (local or global) coordinate transformation T which maps the nonrectangular grid-element onto a rectangular one where standard approximations are applicable. The effect of T is that one performs a standard space-discretization of a transformed PDE on a rectangular grid-element. The choice of T influences the accuracy of the discretization of course. DEKKER [4] has developed a method which minimizes the errors of derivative approximations on nonrectangular grid-elements. In case an explicit parametrization of the curvilinear grid-lines in the (x,y)-plane is available a suitable transformation is easily found [25]. The next example illustrates this.

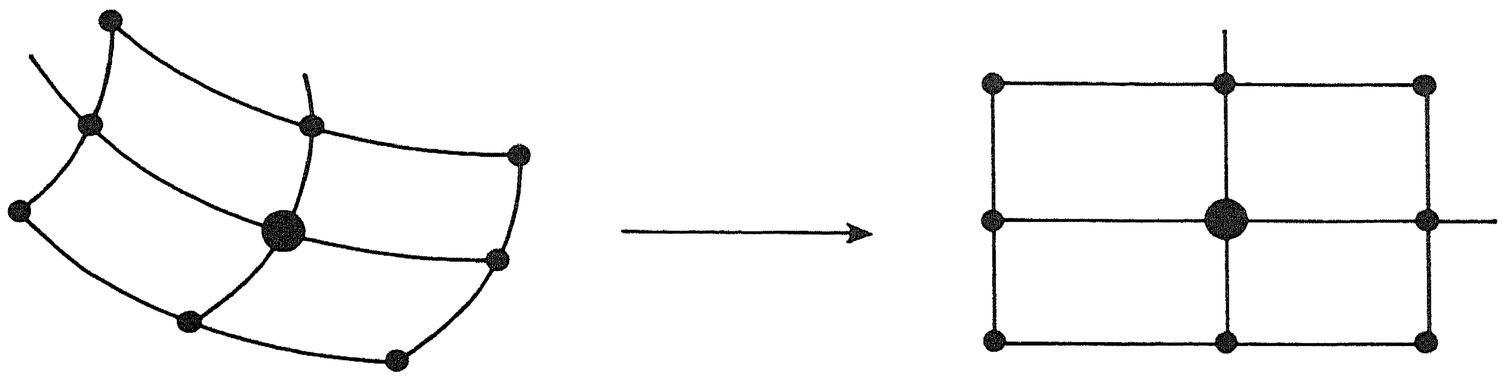


FIGURE 2.3.4. A curvilinear and a square grid-element

EXAMPLE 2.3.5. Consider Figure 2.3.4. Suppose that in the (x,y)-domain Ω of the PDE curvilinear gridlines $X=X(x,y)$, $Y=Y(x,y)$ have been defined. We seek a transformation T which maps the curvilinear grid onto a square grid in the (X, Y)-plane with grid distance Δ. Hence T is defined by

$$X(x,y) = j\Delta, \quad Y(x,y)=k\Delta,$$

j and k being gridindices in the square grid. Supposing that X and Y are differentiable, it holds that

$$\frac{\partial}{\partial x} = \frac{\partial X}{\partial x}\frac{\partial}{\partial X} + \frac{\partial Y}{\partial x}\frac{\partial}{\partial Y}, \quad \frac{\partial}{\partial y} = \frac{\partial X}{\partial y}\frac{\partial}{\partial X} + \frac{\partial Y}{\partial y}\frac{\partial}{\partial Y},$$

where X_x, X_y, Y_x and Y_y are explicitly known. Because of the transformation T standard differences can be used for approximating $\partial/\partial X$ and $\partial/\partial Y$ in the (X, Y)-plane.

In case no explicit parameterization $X(x,y)$, $Y(x,y)$ is available, one can approximate $X_x, \ldots, Y_y$ on the curvilinear grid in the (x,y)-plane. There are various possibilities to do so [4], [25]. Of course, in applications one mostly has to make this approximation. □

REMARK 2.3.6. By nature the finite element method leads to an easier way of handling curved boundaries. Differently from the field of hyperbolic PDEs, in recent years the finite element method has become quite popular for parabolic equations. PRAAGMAN [35] has implemented the finite element method for the shallow water equations and reports satisfactory results. More research is needed however for more definite conclusions how finite differences and finite elements compare to each other in the extensive and diverse field of hyperbolic equations, such as in shallow water applications. □

3. Time integrators

In this section we start with the time-continuous, semi-discrete system (2.2.6). In principle, any ODE solver can now be applied to this equation in order to obtain a numerical approximation to the solution of (2.2.6). Thus, using an initial value problem solver from a program library such as NAG or IMSL will provide us with a numerical solution of the SWEs. However, the costs both in terms of computation time and of storage will be enormous. The reason is that such library programs, being designed as general purpose methods, do not take into account the two characteristic properties of semi-discrete hyperbolic systems, in particular the semi-discrete SWEs:

A. *The large number of component equations in the system* (2.2.6) (3 *times the number of spatial grid points used in the semidiscretization*).

B. *The large, almost imaginary eigenvalue interval of the Jacobian matrix* $\partial F/\partial W$ *of the right-hand side in* (2.2.6)

Property A is obviously responsible for the excessive storage requirements when applying a general purpose method, at the same time implying that each integration step is relatively expensive. Property B causes the system (2.2.6) to be marginally stable; it is therefore expected that a numerical approximation to (2.2.6) will easily become unstable unless either small integration steps or special numerical approximations are used.

It is the purpose of this section to give a survey of possible integration techniques for solving (2.2.6) that take into account the properties A and B.

3.1. Runge-Kutta methods

Let $\mathbf{W}_n$, $n=0,1,2,\ldots$ denote numerical approximations to the exact solution $\mathbf{W}(t)$ of (2.2.6) at $t_n=t_0+n\Delta t$, Δt being the integration step. Then an important class of numerical approximations to (2.2.6) is given by

$$\begin{aligned}
\mathbf{W}_{n+1}^{(j)} &= \mathbf{W}_n+\Delta t\sum_{l=1}^{m} a_{j,l}\mathbf{F}_{n+1}^{(l)},\\
\mathbf{F}_{n+1}^{(l)} &:=\mathbf{F}(t_{n+1}^{(l)},\mathbf{W}_{n+1}^{(l)}),\quad j=1,2,\ldots,m,\\
\mathbf{W}_{n+1} &= \mathbf{W}_n+\Delta t\sum_{l=1}^{m} b_l\mathbf{F}_{n+1}^{(l)}.
\end{aligned}\tag{3.1.1}$$

This method is called an *m-stage Runge-Kutta method.* The Runge-Kutta parameters $a_{j,l}$ and b_l are determined by accuracy and stability conditions. The intermediate points $t_{n+1}^{(j)}$ are usually defined by

$$t_{n+1}^{(j)} = t_n+\Delta t\sum_{l=1}^{m} a_{j,l}.\tag{3.1.2}$$

In this case, the $\mathbf{W}_{n+1}^{(j)}$ are approximations to $\mathbf{W}(t_{n+1}^{(j)})$. We will assume that (3.1.2) is always satisfied.

Example 3.1.1. The most famous (and at the same time an appropriate time

integrator for the SWEs) is given by (KUTTA [26])

$$\mathbf{W}_{n+1}^{(1)} = \mathbf{W}_n, \quad \mathbf{W}_{n+1}^{(2)} = \mathbf{W}_n + \tfrac{1}{2}\Delta t \mathbf{F}_{n+1}^{(1)},$$
$$\mathbf{W}_{n+1}^{(3)} = \mathbf{W}_n + \tfrac{1}{2}\Delta t \mathbf{F}_{n+1}^{(2)}, \quad \mathbf{W}_{n+1}^{(4)} = \mathbf{W}_n + \Delta t \mathbf{F}_{n+1}^{(3)}, \tag{3.1.3}$$
$$\mathbf{W}_{n+1} = \mathbf{W}_n + \tfrac{1}{6}\Delta t[\mathbf{F}_{n+1}^{(1)} + 2\mathbf{F}_{n+1}^{(2)} + 2\mathbf{F}_{n+1}^{(3)} + \mathbf{F}_{n+1}^{(4)}].$$

If $\mathbf{F}(t, \mathbf{W})$ is sufficiently smooth it can be proved that (see e.g. [27])

$$\mathbf{W}_{n+1} - \mathbf{W}(t_{n+1}) = O((\Delta t)^4) \text{ as } \Delta t \to 0, \quad t_{n+1} \text{ constant.}$$

The method is said to be of order 4. Notice that the $\mathbf{W}_{n+1}^{(j)}$ are defined by (3.1.3) explicitly. In the particular case of the SWEs, this method when implemented on a computer requires 3 arrays for storing the $\mathbf{W}_{n+1}^{(j)}$ and $\mathbf{F}_{n+1}^{(j)}$ during the computation of an integration step. □

In Table 3.1.1 we present a few numerical results obtained by this method when applied to model 1 (equations (2.1.1)-(2.1.4)) in the semi-discrete form (2.2.6). These results refer to the relative height deviation defined by

$$\epsilon := \frac{h - h_{ref}}{\max |h_{ref} - \bar{h}_{ref}|},$$

with h_{ref} a sufficiently accurate reference solution and $\bar{h}_{ref}$ the mean value of h_{ref}. For a few grid points we have listed the accuracy expressed in terms of the number of correct significant digits, i.e.

$$sd := -{}^{10}\log(\max |\epsilon|).$$

TABLE 3.1.1. Model 1 with $\Delta x = \Delta y = 100 * 10^3$ m and $t_{end} = 48 * 3600$ sec.

Grid point	$\Delta t = 1200$	$\Delta t = 600$	$\Delta t = 300$
(48,00)	2.80	3.77	5.01
(48,12)	2.37	3.35	4.68
(48,24)	2.48	3.28	4.54
(48,36)	2.41	3.34	4.53

The reference solution for (2.2.6) was obtained by using a high order method with extremely small integration steps.

These results clearly reflect the fourth order behaviour of the Runge-Kutta integrator, i.e. on halving the integration step the *sd*-value should increase by $4 \log 2 \approx 1.2$. For $\Delta t \leqslant 600$ this (asymptotic) order property is shown.

In order to analyse the stability characteristics of a numerical method one often uses the *linear* equation

$$\dot{\mathbf{W}}(t) = \lambda \mathbf{W}(t), \quad \lambda \in \Lambda_n \tag{3.1.4}$$

as a stability test model. Here, Λ_n denotes the eigenvalue spectrum of $\partial \mathbf{F}/\partial \mathbf{W}$ at t_n. Applying (3.1.1) to (3.1.4) leads to the relation

$$\mathbf{W}_{n+1} = R(\lambda \Delta t)\mathbf{W}_n, \quad \lambda \in \Lambda_n, \tag{3.1.5}$$

where $R(z)$ is a rational function in z the coefficients of which are expressions in terms of the parameters $a_{j,l}$ and b_l. It can be shown [39] that

$$R(z) = \frac{\det[I - Az + \mathbf{e}\mathbf{b}^T z]}{\det[I - Az]}, \quad \mathbf{e} = [1, \ldots, 1]^T, \tag{3.1.6}$$

where A is the matrix $(a_{j,l})$; $j,l = 1, \ldots, m$, and $\mathbf{b}$ is the vector $(b_1, \ldots, b_m)^T$ (observe that the Runge-Kutta method is completely defined by the matrix A and the vector $\mathbf{b}$). The *stability region* $\mathbb{S}$ of a Runge-Kutta method is defined as the region in the complex z-plane where $|R(z)| \leqslant 1$. The method is said to be stable for a given problem at t_n if $\Delta t \lambda$ lies in the stability region. Notice that the stability region $\mathbb{S}$ is completely defined by the numerical method without reference to the particular problem to be solved. Evidently, if a method is stable at t_n, the numerical solutions of the test equations (3.1.4) satisfy the condition

$$\|\mathbf{W}_{n+1}\| \leqslant \|\mathbf{W}_n\|. \tag{3.1.7}$$

In many cases, the (linear) stability condition $\Delta t \Lambda_n \subset \mathbb{S}$ leads to satisfactory numerical solutions of nonlinear problems. But we should bear in mind that the above given analysis is based on the test equations (3.1.4) and should be applied with care to more general problems. For a discussion of nonlinear stability analysis we refer to DEKKER and VERWER [6].

Adopting $\Delta t \Lambda_n \subset \mathbb{S}$ as the stability condition it follows from property B that the SWEs require numerical methods the stability regions of which contain a relatively large imaginary interval $[-i\beta, i\beta]$ (notice that $\mathbb{S}$ is symmetric with respect to the real axis). For *implicit* methods this is easily achieved. However, from a practical point of view we are mainly interested in *explicit* Runge-Kutta methods ($a_{j,l} = 0$ for $j \geqslant l$) which turn out to have rather modest β-values.

EXAMPLE 3.1.2. Here we mention some well-known explicit Runge-Kutta methods and the corresponding stability function $R(z)$, which reduces to a polynomial for these explicit methods. Also the imaginary stability boundary β, the number of stages m and the order p are given. For the coefficients $a_{j,l}$ and b_l, defining the Runge-Kutta schemes, we refer to e.g. [27]

method of Euler; $R(z) = 1 + z$, $\beta = 0$, $m = 1$, $p = 1$,

method of Runge; $R(z) = 1 + z + \frac{1}{2}z^2$, $\beta = 0$, $m = 2$, $p = 2$,

method of Heun; $R(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3$, $\beta = \sqrt{3}$, $m = 3$, $p = 3$,

method of Kutta; $R(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4$, $\beta = 2\sqrt{2}$, $m = 4$, $p = 4$.

Thus, the methods using 3 or 4 stages in this example possess a nonvanishing stability boundary β on the imaginary axis. The corresponding stability condition $\Delta t \Lambda_n \subset \mathbb{S}$ can be written as

$$\Delta t \leqslant \frac{\beta}{S}, \quad S := \text{spectral radius of } \partial \mathbf{F}/\partial \mathbf{W}, \tag{3.1.8}$$

provided that Λ_n is purely imaginary. Since in the case of the SWEs, $S = O(\Delta^{-1})$, Δ being the mesh size on the spatial grid, condition (3.1.8) allows us to use grid parameters Δ and Δt of comparable magnitude as $\Delta, \Delta t \to 0$. However, in an actual computation the order constant in $S = O(\Delta^{-1})$ may be large (e.g. in computations with large values for the depth function); also, it is often allowed to use Δt-values which are large compared with Δ (see [38, p. 214]). In such cases, (3.1.8) may impose a severe limitation on the integration step Δt, just *for the sake of stability* and *not for the sake of accuracy*. It is therefore of interest to look for (explicit) Runge-Kutta methods with a large stability boundary β on the imaginary axis, that is to look for methods possessing a (so-called) stability polynomial $R(z) = P_m(z)$ of the form (3.1.6) which assumes values on the unit disc on the largest possible interval $[-i\beta, i\beta]$. Before giving results for this minimax problem we give a theorem which relates the order of the method to the specific form of $P_m(z)$.

THEOREM 3.1.1. *If the Runge-Kutta method* (3.1.1) *is of order p then*

$$\frac{d^j P_m}{dz^j}(0) = 1$$

for $j = 0, 1, \ldots, p$. *($P_m(z)$ is called consistent of order p.)* □

THEOREM 3.1.2. *If* $p \geqslant 1$ *and* $m \geqslant 2$ *then* $\beta \leqslant m - 1$ *(for m odd see* [17], *for m even see* [40]*).* □

THEOREM 3.1.3. *If* $p = 1, 2$ *and* $m = 2k + 1$, $k = 0, 1, 2, \ldots$, *then*

$$P_m(z) = T_k\left(1 + \frac{z^2}{2k^2}\right) + \frac{z}{k}\left(1 + \frac{z^2}{4k^2}\right) U_{k-1}\left(1 + \frac{z^2}{2k^2}\right) \tag{3.1.9}$$

solves the minimax problem and the corresponding β *value is the largest possible, i.e.* $\beta = m - 1$ [17]. □

This result has recently been extended [37]; now a polynomial is available for all values of m:

THEOREM 3.1.4. *For* $p = 1$ *the polynomial*

$$P_m(z) = i^{m-1} T_{m-1}\left[\frac{z}{i(m-1)}\right] + \frac{1}{2} i^m \left\{T_m\left[\frac{z}{i(m-1)}\right] - T_{m-2}\left[\frac{z}{i(m-1)}\right]\right\} \tag{3.1.10}$$

is the optimal polynomial and has $\beta = m-1$. □

THEOREM 3.1.5. *If* $p = 2,3,4$ *and* $m = 4$ *then*

$$P_4(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4 \tag{3.1.11}$$

solves the minimax problem and $\beta = 2\sqrt{2}$ [17]. □

In selecting a stability polynomial one should take into account that the larger m the more expensive an integration step. Hence, using large m-values in order to increase β, has to be paid for by m right-hand side evaluations. This suggests considering the effective (or scaled) stability boundary β/m. From Theorem 3.1.2 it follows that $\beta/m \leqslant 1 - 1/m$ so that it hardly pays to use a large value for m. In this connection, we observe that the *fourth order consistent* polynomial $P_4(z)$ given in Theorem 3.1.5 has an effective stability boundary $\beta/m = \frac{1}{2}\sqrt{2}$ which is already more than 70% of the asymptotic value of the *second order consistent* minimax polynomial of degree infinity. Therefore, the polynomial (3.1.11) is recommended as a stability polynomial if one decides to use an explicit Runge-Kutta method for the SWEs.

The next step is the choice of a Runge-Kutta method possessing (3.1.11) as its stability polynomial. An obvious choice is the fourth order method of Kutta (3.1.3) (see also Example 3.1.2), and in fact PRAAGMAN [35] used this method in solving the SWEs. An alternative might be the methods of MERLUZZI and BROSILOW [31] who (following ideas of STETTER [39]) developed methods which allow for global error estimation with low extra costs.

3.2. Linear multistep methods

A second important class of numerical approximations to the ODE (2.2.6) are the *linear k-step methods* defined by

$$\sum_{l=0}^{k} [a_l \mathbf{W}_{n+1-l} - \Delta t b_l \mathbf{F}_{n+1-l}] = 0, \tag{3.2.1}$$

$$\mathbf{F}_{n+1-l} := \mathbf{F}(t_{n+1-l}, \mathbf{W}_{n+1-l}),$$

where the coefficients a_l and b_l are determined by accuracy and stability conditions.

As in the case of Runge-Kutta methods we are particularly interested in explicit methods, i.e. $b_0 = 0$. However, *implicit* methods are also important for us as a starting point in constructing special predictor-corrector methods (see subsection 3.2.2).

3.2.1. Explicit linear multistep methods. Just as for Runge-Kutta methods, stability requirements for linear multistep methods are obtained by applying (3.2.1) to the test equations (3.1.4). This yields a relation of the form

$$[\rho(E)-\lambda\Delta t\sigma(E)]\mathbf{W}_{n+1-k} = 0, \quad \lambda\in\Lambda_n, \quad n+1\geqslant k, \tag{3.2.2}$$

where E is the shift operator defined by $E\mathbf{W}_n=\mathbf{W}_{n+1}$ and where $\{\rho,\sigma\}$ are the so-called *characteristic polynomials* defined by

$$\rho(\zeta) := \sum_{l=0}^{k} a_l\zeta^{k-l};\ \sigma(\zeta) := \sum_{l=0}^{k} b_l\zeta^{k-l}, \quad b_0=0. \tag{3.2.3}$$

The *stability region* $\mathbb{S}$ is now defined as the region in the complex z-plane where the *characteristic function* $\pi(\zeta,z):=\rho(\zeta)-z\sigma(\zeta)$ has all its roots ζ on the unit disc.

The stability condition, widely adopted in practical computations, reads $\Delta t\Lambda_n\subset\mathbb{S}$ so that we are again faced with the problem to construct a method the stability region $\mathbb{S}$ of which contains a large imaginary interval $[-i\beta,i\beta]$.

EXAMPLE 3.2.1. As an example of explicit linear multistep methods we mention the extensively used k-step *Adams-Bashforth* methods, which are characterized by their ρ-polynomial possessing the form $\rho(\zeta)=\zeta^k-\zeta^{k-1}$. The σ-polynomials may be found in [27]. For $k=2,3,4$ (yielding methods of order $p=2,3,4$, respectively) we have the respective imaginary stability boundaries $\beta=0$, $\beta=.72$ and $\beta=.43$. □

The stability boundaries given in this example are at the same time the effective boundaries because each integration step requires just one **F**-evaluation. A comparison with the results obtained for Runge-Kutta methods of the same order reveals that (cf. Example 3.1.2) the Adams-Bashforth method of order 3 has a larger effective stability interval along the imaginary axis, but the fourth order method does not. Both second order methods have $\beta=0$.

The maximization of the imaginary stability interval of explicit multistep method has been studied in JELTSCH and NEVANLINNA [24]:

THEOREM 3.2.1.

(a) *The imaginary stability boundary β of an explicit linear multistep method cannot exceed* 1.

(b) *Let $r\in[0,1)$ and $k\in\{2,3,4\}$ be given. Then there exists an explicit linear k-step method of order $p=k$ with $\beta=r$.* □

EXAMPLE 3.2.2.
Leap-frog method: $k=2$

$$\rho(\zeta) = \zeta^2-1, \quad \sigma(\zeta)=2\zeta, \quad \beta=1, \quad p=2$$

Jeltsch-Nevanlinna methods: $k=3,4$ $(\epsilon>0)$

$$\rho(\zeta) = (\zeta-1)(\zeta+1-\tfrac{2}{3}\epsilon)(\zeta-1+\epsilon),$$

$$\sigma(\zeta)=2\zeta(\zeta-1)+\tfrac{2}{3}\epsilon(2\zeta+1)+\frac{1}{18}\epsilon^2(\zeta^2-8\zeta+5), \quad \beta=1-O(\epsilon), \quad p=3.$$

$$\rho(\zeta) = (\zeta^2-1)(\zeta^2-2\frac{3-4\epsilon}{3-\epsilon}\zeta+1), \quad \sigma(\zeta)=\frac{6}{3-\epsilon}\zeta(\zeta^2+2(\epsilon-1)\zeta+1),$$

$$\beta=1-O(\epsilon), \quad p=4. \quad \square$$

In Table 3.2.1 results obtained by the leap-frog method are listed for problem (2.2.6) corresponding to model 1.

TABLE 3.2.1. Model 1 with $\Delta x=\Delta y=100*10^3$ m and $t_{end}=48*3600$ sec

Grid point	$\Delta t=400$	$\Delta t=225$	$\Delta t=75$
(48,00)	1.56	2.26	3.19
(48,12)	2.30	2.41	3.30
(48,24)	1.95	2.27	3.16
(48,36)	2.44	2.79	3.31

With the exception of the last grid point, the second order behavior is clearly shown for $\Delta t=75$. Note that the results of the fourth order *RK*-method (cf. Table 3.1.1) are much more accurate.

3.2.2. Predictor-corrector methods. The rather modest stability results obtained for explicit Runge-Kutta and linear multistep methods lead us to consider *implicit* methods. In particular we will study implicit linear multistep methods because of their simple structure. Thus, let $\mathbf{W}_{n+1}$ be defined by (3.2.1) with $b_0\neq 0$, or briefly

$$\mathbf{W}_{n+1}-b_0\Delta t\mathbf{F}(t_{n+1},\mathbf{W}_{n+1}) = \Sigma_n, \tag{3.2.4}$$

where Σ_n is a linear combination of $\mathbf{W}_j$ and $\mathbf{F}_j$ values with $j\leqslant n$. In order to solve this implicit relation we employ a *predictor-corrector method.*

Following [21] we define the m-point iteration scheme

$$\mathbf{W}_{n+1}^{(j)} = \sum_{l=1}^{j}[\mu_{jl}\mathbf{W}_{n+1}^{(l-1)}+\bar{\mu}_{jl}\Delta t\mathbf{F}(t_{n+1},\mathbf{W}_{n+1}^{(l-1)})]+\lambda_j\Sigma_n, \tag{3.2.5}$$

$$j=1, \ldots, m$$

where $\mathbf{W}_{n+1}^{(0)}$ is obtained by a suitable predictor method (e.g. an explicit linear multistep method) and $\mathbf{W}_{n+1}^{(m)}$ is accepted as an approximation to the exact solution of (3.2.4). By requiring that the j-th row sum of the matrices $M=(\mu_{jl})$ and $\bar{M}=(\bar{\mu}_{jl})$ are respectively given by $1-\lambda_j$ and $b_0\lambda_j$, we achieve that if

$\mathbf{W}_{n+1}^{(j)} \to \mathbf{W}$ as $j \to \infty$ then $\mathbf{W}$ equals the exact solution of (3.2.4). In doing so, (3.2.4) may be considered as the *corrector equation* and (3.2.5) as an correction iteration. The scheme (3.2.5) is said to be consistent with (3.2.4).

The iteration scheme (3.2.5) is conveniently characterized by the iteration polynomials

$$P_0(z) = 1, \quad P_j(z) = \sum_{l=1}^{j} [\mu_{jl} + \bar{\mu}_{jl} z] P_{l-1}(z), \tag{3.2.6}$$

$$j = 1,2, \ldots, m.$$

The consistency condition implies that $P_j(1/b_0) = 1$ for all j. The following theorem determines the accuracy of the predictor-corrector method [21]:

THEOREM 3.2.2. *Let* $\mathbf{W}_j = \mathbf{W}(t_j)$ *for* $j \leqslant n$, *then*

$$\mathbf{W}_{n+1}^{(m)} - \mathbf{W}(t_{n+1}) = [I - P_m(Z)]\,[\mathbf{W}_{n+1} - \mathbf{W}(t_{n+1})] + P_m(Z)[\mathbf{W}_{n+1}^{(0)} - \mathbf{W}(t_{n+1})] + O(\Delta t^q),$$

$$q \geqslant 3 + 2\min\{\tilde{p}, p\}, \quad Z := \Delta t \frac{\partial \mathbf{F}}{\partial \mathbf{W}}\,(t_{n+1}, \mathbf{W}_{n+1})$$

where p *and* $\tilde{p}$ *are the orders of accuracy of the corrector and the predictor respectively.* □

This theorem expresses the (local) error of the predictor-corrector method in terms of those of the predictor and the corrector plus higher order terms. It clearly shows that the solution of (3.2.4) is approximated better as $\|P_m(Z)\|$ is smaller. Furthermore, the theorem gives us the exact order of the method: let $P_m(z)$ have a zero at $z = 0$ of multiplicity r, then the predictor-corrector method is of order $p^* = \min\{p, \tilde{p} + r, 2 + 2\min\{\tilde{p}, p\}\}$.

Next, we consider the stability of (3.2.5). Assuming that we can find a predictor and an iteration polynomial $P_m(z)$ such that (3.2.4) is solved with sufficient accuracy, the stability properties are determined by those of (3.2.4). The following theorem is known:

THEOREM 3.2.3.

(a) *Only for* $p \leqslant 2$ *there exist linear multistep methods with an infinite imaginary interval of stability* [22].

(b) *For* $p > 2$ *the imaginary interval of stability cannot exceed* $[-i\sqrt{3}, i\sqrt{3}]$ [5],[23]. □

EXAMPLE 3.2.3. A few methods possessing an infinite imaginary interval of stability are [27]: *implicit Euler* ($k = 1$, $p = 1$), *the trapezoidal rule* ($k = 1$, $p = 2$) and the *backward differentiation method* ($k = 2$, $p = 2$). Within the class of methods with $p > 2$, the fourth order, 2-step *Milne-Simpson method* has the maximal attainable imaginary stability boundary $\beta = \sqrt{3}$. □

Theorem 3.2.3 indicates that we should be content with a first or second order corrector in our attempt to construct predictor-corrector methods with large imaginary stability boundaries. In particular, the implicit Euler and the second order backward differentiation method are recommendable because of their strong damping of higher frequencies (which are easily introduced by round-off errors).

Let us now consider the imaginary stability boundary β of the complete predictor-corrector method. We will do this by relating β to the *real* stability boundary β_{real} of the method. Since the derivation of β_{real} has been studied in some detail [21] we can avoid a lot of tedious computations. The following theorem is easily proved.

THEOREM 3.2.4. *Let $\beta_{real}(m)$ be the real stability boundary of the predictor-corrector pair using the iteration polynomial $Q_m(z)$. Then this predictor-corrector pair using the iteration polynomial $P_{2m}(z)=Q_m(b_0 z^2)$ has the imaginary stability boundary $\beta=\sqrt{\beta_{real}(m)/b_0}$.* □

EXAMPLE 3.2.4. In [21] a predictor-corrector pair consisting of the predictor

$$\mathbf{W}_{n+1} = 2\mathbf{W}_n - \mathbf{W}_{n-1}$$

and the second order backward differentiation corrector (see Example 3.2.3) is considered for which iteration polynomials are constructed such that $\beta_{real}(m)\uparrow 1.37m^2$ as $m\to\infty$. Hence, by Theorem 3.2.4 we can construct a $2m$-stage predictor-corrector method with $\beta\uparrow 1.43m$. Effectively, however, we obtain the value .72 for m sufficiently large. □

The methods suggested by Theorem 3.2.4 are not optimal. In order to get some insight into how good or poor these methods are we have done a numerical search for the optimal iteration polynomial $P_2(z)$ in the case of the predictor-corrector pair mentioned in Example 3.2.4. We found

$$P_2(z) = 1.0 - .408z + .272z^2, \quad \beta = 1.97.$$

3.3. Multigrid methods

A multigrid method can be used for solving the implicit relations obtained when an implicit Runge-Kutta or multistep method is applied to (2.2.6). Let us consider an implicit k-step method which requires the solution of equation (3.2.4) in each integration step. In the multigrid technique we do not only consider this equation but we define on a sequence of successively coarser grids a similar equation. Thus, we have a sequence of problems of the form

$$\mathbf{W}_{n+1} - b_0 \Delta t \mathbf{F}(t_{n+1}, \mathbf{W}_{n+1}) = \Sigma_n \tag{3.3.1}$$

where the number of components in $\mathbf{W}_{n+1}$, $\mathbf{F}$ and Σ_n correspond to the number of grid points in the grid considered. In fact, in the more advanced applications of the multigrid method the right-hand side vectors Σ_n are modified except for the Σ_n corresponding to the finest grid. We will not discuss

these modification but refer to the literature (e.g. [1], [15]). By first solving the coarsest grid problem one may construct a rather good initial approximation to the solution of the next finer grid problem, and so on. Usually, a linear interpolation procedure is applied. These approximations can be improved by adding certain correction terms for which we again refer to the literature. As a result we obtain initial approximations which differ from the solution of (3.3.1) only in the high frequency range. Thus, in order to solve the problems (3.3.1) one may use any iteration scheme (usually called *relaxation method*) that takes advantage of the fact that the initial approximation is incorrect only in the high frequency range. For functions $\mathbf{F}(t, \mathbf{W})$ the Jacobian matrix of which has a *real spectrum with an orthogonal eigensystem,* Gauss-Seidel relaxation or Incomplete LU relaxation [44] are widely used. Chebyshev relaxation (advocated in [20]) may be another possibility, particularly when vectorcomputers are to be used (cf. subsection 4.1). However, in the present case we do not have a Jacobian matrix with a real spectrum, but with an *imaginary spectrum* instead. For such problems there is hardly any experience.

Let us apply the iteration scheme (3.2.5) to (3.3.1), that is Σ_n is replaced by $\tilde{\Sigma}_n$. Omitting higher order terms in Δt we deduce from Theorem 3.2.2

$$\mathbf{W}_{n+1} - \mathbf{W}_{n+1}^{(m)} = P_m(Z)[\mathbf{W}_{n+1} - \mathbf{W}_{n+1}^{(0)}].$$

Since $\mathbf{W}_{n+1} - \mathbf{W}_{n+1}^{(0)}$ is supposed to contain only high frequency components, we should look for polynomials $P_m(z)$ such that the matrix $P_m(Z) = P_m(\Delta t \partial \mathbf{F}/\partial \mathbf{W})$ damps all high frequencies. In the case of the SWEs, the eigenvectors of $\partial \mathbf{F}/\partial \mathbf{W}$ corresponding to the eigenvalues with large imaginary parts present the high frequencies so that we should construct polynomials $P_m(z)$ satisfying the condition $P_m(1/b_0) = 1$, such that $|P_m(iy)|$ is as small as possible on an interval $a \leqslant |y| \leqslant b$. Here, $a = \alpha \Delta t S$ and $b = \Delta t S$, α being some parameter < 1 (say $\alpha = \frac{1}{2}$), and S is the spectral radius of $\partial \mathbf{F}/\partial \mathbf{W}$. As far as the authors know this minimax problem has not yet been solved for general m. For $m = 1$ we straightforwardly find that

$$P_1(z) = \frac{1 + z/b_0 b^2}{1 + 1/b_0^2 b^2} \quad \text{with} \quad \max_{[ia,ib]} |P_1(z)| = \frac{1}{\sqrt{1 + 1/b_0^2 b^2}} \tag{3.3.2}$$

is the minimax polynomial for all $0 \leqslant a \leqslant b$. For $m = 2$ and $m = 4$ a numerical computation for several intervals $[ia, ib]$ resulted in minimax polynomials with extremely small values for the odd degree coefficients. This suggests considering the polynomials $P_m(z) = Q_{m/2}(z^2)$ with m even. The minimax problem on $[ia, ib]$, i.e. $a^2 \leqslant -z^2 \leqslant b^2$, is solved by Chebyshev polynomials:

$$P_m(z) = \frac{T_{m/2}\left[\dfrac{b^2 + a^2 + 2z^2}{b^2 - a^2}\right]}{T_{m/2}\left[\dfrac{b^2 + a^2 + 2/b_0^2}{b^2 - a^2}\right]}, \quad m \text{ even}, \tag{3.3.3}$$

from which the damping factor immediately follows.

EXAMPLE 3.3.1. Let (3.3.1) correspond to the second order backward differentiation method ($b_0=2/3$) and choose $\alpha=½$, i.e. $a=b/2$. Then the polynomials (3.3.3.) damp the high frequencies by a factor

$$1/T_{m/2}\left[\frac{5}{3}+\frac{6}{b^2}\right].$$

In Table 3.3.1 a few values are listed.

TABLE 3.3.1. Damping factors obtained by (3.3.2) and (3.3.3) for $a=b/2$ and $b_0=2/3$

b	$m=1$	$m=2$	$m=4$	$m=6$	$m=8$	$m=10$	$m=12$
1	.83	.13	.009	$<10^{-3}$	~ 0	~ 0	~ 0
2	.95	.32	.05	.009	.001	~ 0	~ 0
4	.986	.49	.13	.04	.009	.003	~ 0
8	.997	.57	.19	.06	.02	.006	.002
16	.999	.59	.21	.07	.02	.008	.003

In order to illustrate that these polynomials are close to the optimal polynomials we explicitly give the fourth degree polynomial for $2a=b=4$:

$$[P_m(z)]_{optimal} \approx .58801+.028363z+.144657z^2+.002135z^3+.007261z^4$$

$$[P_m(z)]_{Chebyshev} \approx .62092+0.0z+.15144z^2+0.0z^3+.007572z^4$$

with respective damping factors .1344 and .1363. □

3.4. Splitting methods

Sofar we did not exploit the special structure of the right-hand side function $\mathbf{F}(\mathbf{W})$. We will now consider time integrators which take advantage of the specific form of $\mathbf{F}(\mathbf{W})$. These methods fall into the class of *one-stage splitting methods* defined by

$$\mathbf{W}_{n+1} = \mathbf{W}_n+\lambda\Delta t[\mathbf{G}(\mathbf{W}_{n+1},\mathbf{W}_n)+(\frac{1}{\lambda}-1)\mathbf{G}(\mathbf{W}_n,\mathbf{W}_n)] \tag{3.4.1}$$

or into the class of *two-stage splitting methods* defined by [19]

$$\begin{aligned}\mathbf{W}^{(1)}_{n+1} &= \mathbf{W}_n+\tfrac{1}{2}\Delta t[\mathbf{G}(\mathbf{W}^{(1)}_{n+1},\mathbf{W}_n)+(2\lambda-1)\mathbf{G}(\mathbf{W}_n,\mathbf{W}_n)]\\ \mathbf{W}_{n+1} &= \mathbf{W}_n+\tfrac{1}{2}\Delta t[\mathbf{G}(\mathbf{W}^{(1)}_{n+1},\mathbf{W}_n)+(2-\frac{1}{\lambda})\mathbf{G}(\mathbf{W}_n,\mathbf{W}_{n+1})+\\ &\quad (\frac{1}{\lambda}-1)\,\mathbf{G}(\mathbf{W}^{(1)}_{n+1},\mathbf{W}_{n+1})].\end{aligned} \tag{3.4.2}$$

Here, $\mathbf{G}(\mathbf{W},\tilde{\mathbf{W}})$ is a so-called splitting function satisfying the splitting condition

$\mathbf{G}(\mathbf{W},\mathbf{W})\equiv\mathbf{F}(\mathbf{W})$. The one-stage method is first order accurate for all λ. The two-stage method is second order for all λ. In the literature one meets also two-stage methods which use different splitting functions in the successive stages.

3.4.1. One-stage methods. The one-stage methods may be considered as a method in between the explicit and implicit Euler method. If $\lambda=0$ we have the *explicit* Euler method and if $\lambda=1$ with $\mathbf{G}(\mathbf{W},\tilde{\mathbf{W}})=\mathbf{F}(\mathbf{W})$ we obtain the *implicit* Euler method.

In the examples given below we have $\lambda=1$. Furthermore, the function $\mathbf{F}(\mathbf{W})$ is defined by the discretization of the right-hand side of (1.2) omitting the force term $\mathbf{r}$. Thus,

$$\mathbf{F}(\mathbf{W})=-\begin{bmatrix} UD_x+VD_y & -f & gD_x \\ f & UD_x+VD_y & gD_y \\ HD_x & HD_y & UD_x+VD_y \end{bmatrix}\mathbf{W},$$

where D_x and D_y are discretizations of $\partial/\partial x$ and $\partial/\partial y$, and U,V, and H are the diagonal matrices $diag(\mathbf{U})$, $diag(\mathbf{V})$ and $diag(\mathbf{H})$; we will also write H_x instead of D_xH, etc..

EXAMPLE 3.4.1. Fischer -Sielecki method [9]

$$\mathbf{G}(\mathbf{W},\tilde{\mathbf{W}})=-\begin{bmatrix} 0 & 0 & 0 \\ f & 0 & 0 \\ \tilde{H}_x+\tilde{H}D_x & \tilde{H}_y+\tilde{H}D_y & 0 \end{bmatrix}\mathbf{W}-\begin{bmatrix} \tilde{U}D_x+\tilde{V}D_y & -f & gD_x \\ 0 & \tilde{U}D_x+\tilde{V}D_y & gD_y \\ 0 & 0 & 0 \end{bmatrix}\tilde{\mathbf{W}}.$$

When implemented this splitting function generates a completely explicit scheme. The stability condition reads $\Delta t\leqslant 2\sqrt{\|g\mathbf{H}\|(S_x^2+S_y^2)}$ where S_x and S_y are the spectral radii of D_x and D_y. For a detailed discussion of the Fischer-Sielecki method and its modifications we refer to [18].

Navon's method [34]

$$\mathbf{G}(\mathbf{W},\tilde{\mathbf{W}})=-\begin{bmatrix} UD_x+\tilde{V}D_y & 0 & gD_x \\ f & UD_x+VD_y & gD_y \\ 0 & 0 & \tilde{U}D_x+\tilde{V}D_y+\tilde{U}_x+\tilde{V}_y \end{bmatrix}\mathbf{W}-\begin{bmatrix} 0 & -f & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\tilde{\mathbf{W}}.$$

By first solving the (linear) equation for $\mathbf{H}_{n+1}$ we obtain a (nonlinear) equation in $\mathbf{U}_{n+1}$ alone and a (nonlinear) equation for $\mathbf{V}_{n+1}$ alone.

3.4.2. Two-stage methods. In the case of (3.4.2), both stages should be defined in such a way that $\mathbf{W}_{n+1}^{(1)}$ and $\mathbf{W}_{n+1}$ can be 'conveniently' obtained, that is $\partial \mathbf{G}/\partial \mathbf{W}$ and $\partial \mathbf{G}/\partial \tilde{\mathbf{W}}$ are required to have a simple structure.

EXAMPLE 3.4.2. *Classical ADI splitting*

$$\mathbf{G}(\mathbf{W},\tilde{\mathbf{W}}) = -\begin{bmatrix} UD_x & 0 & gD_x \\ f & UD_x & 0 \\ HD_x & 0 & UD_x \end{bmatrix}\mathbf{W} - \begin{bmatrix} \tilde{V}D_y & -f & 0 \\ 0 & \tilde{V}D_y & gD_y \\ 0 & \tilde{H}D_y & \tilde{V}D_y \end{bmatrix}\tilde{\mathbf{W}}. \quad (3.4.3)$$

This splitting is the most natural one (cf.[19] for a survey), and leads to an ADI type splitting method. In fact, this type of method (with $\lambda = \frac{1}{2}$) was investigated by GUSTAFSSON [12] who gives a detailed discussion of the solution of the implicit relations (notice that the stages in (3.4.2) are one-dimensional implicit when using (3.4.3)). A linearized version of this ADI method has been considered by FAIRWEATHER and NAVON [8]. The (linear) stability analysis indicates unconditional stability for linear models with constant coefficients.

Leendertse's method

The scheme of Leendertse, in the way it was originally introduced [29], can also be formulated as a two-stage method of the form (3.4.2) with $\lambda = \frac{1}{2}$. To that end, however, we have to define two splitting functions, which are different for both stages. In the first stage, this method uses $\mathbf{G}(\mathbf{W},\tilde{\mathbf{W}})$ with components

$$\begin{aligned} \mathbf{G}_1 &= -[\tilde{U}_x\mathbf{U} + \tilde{U}_y\tilde{\mathbf{V}} - f\tilde{\mathbf{V}} + g\mathbf{H}_x] \\ \mathbf{G}_2 &= -[U\tilde{\mathbf{V}}_x + V\tilde{\mathbf{V}}_y + f\mathbf{U} + g\tilde{\mathbf{H}}_y] \\ \mathbf{G}_3 &= -[U\mathbf{H}_x + H\mathbf{U}_x + \tilde{V}\tilde{\mathbf{H}}_y + \tilde{H}\tilde{\mathbf{V}}_y]. \end{aligned} \quad (3.4.4)$$

By first (simultaneously) solving $\mathbf{U}$ and $\mathbf{H}$ and afterwards (explicitly) calculating $\mathbf{V}$, only tridiagonal systems have to be solved.

In the second stage a slightly different splitting function is used. To be more precise, the advective terms are replaced by

$$-[U_x\tilde{\mathbf{U}} + U_y\tilde{\mathbf{V}}], \quad -[U\mathbf{V}_x + \tilde{V}\mathbf{V}_y].$$

Solving this stage is similar to the first one, but now the rôles of $\mathbf{U}$ and $\mathbf{V}$ are interchanged.

In a later version of this scheme [30] there has been used a staggering in time for the velocity components; this results in a calculation of $\mathbf{U}$ and $\mathbf{H}$ at time levels $n+\frac{1}{2}$ and of $\mathbf{V}$ and $\mathbf{H}$ at levels n.

Stelling's method [38]
This method uses $\lambda = \frac{1}{2}$ and a splitting function $\mathbf{G}(\mathbf{W}, \tilde{\mathbf{W}})$ with components

$$\begin{aligned} \mathbf{G}_1 &= -[\tilde{U}_x \mathbf{U} + \tilde{U}_y \mathbf{V} - f\mathbf{V} + g\mathbf{H}_x] \\ \mathbf{G}_2 &= -[U\mathbf{V}_x + \tilde{V}\mathbf{V}_y + f\mathbf{U} + g\mathbf{H}_y] \\ \mathbf{G}_3 &= -[U\mathbf{H}_x + H\mathbf{U}_x + \tilde{V}\tilde{\mathbf{H}}_y + \tilde{H}\tilde{\mathbf{V}}_y] \end{aligned} \tag{3.4.5}$$

In the first stage (implicit in $\mathbf{W} = \mathbf{W}^{(1)}_{n+1}$) the vector $\mathbf{V} = \mathbf{V}^{(1)}_{n+1}$ can be solved 'conveniently' by using one-sided difference operators; then $\mathbf{U} = \mathbf{U}^{(1)}_{n+1}$ can explicitly be expressed in terms of $\mathbf{V}$ and $\mathbf{H} = \mathbf{H}^{(1)}_{n+1}$, and substitution into the equation for $\mathbf{H}$ yields an equation in $\mathbf{H}$ alone. The second stage is treated in a similar manner. Notice the strong-implicit treatment of the convection term. This method is claimed to be highly stable even in the presence of nonlinear terms.

4. Future developments

In this section we will briefly discuss a few aspects which may become important for the solution of the SWEs.

4.1. Vector processing

Since the numerical solution of a large-scale, realistic shallow water model is a tremendous task, a huge increase in computer speed as well as memory capacity is needed in order to obtain a sufficiently detailed simulation for engineering purposes. A useful alternative to the traditional scalar computer may be the so-called vector computer. The last type of machine is designed to enhance the concurrency of arithmetic operations, which results in a high system throughput. To be more precise, vectors (i.e., ordered sets of values) are operated with one single instruction.

Since 1984 the CWI has access to a CYBER 205, which is a vector processor, also called pipeline machine. Therefore, we will globally consider the consequences for solving the SWEs when using such a computer. (For a detailed discussion of parallel computing we refer to [16].)

In order to utilize the potential speed of a vector computer we have to satisfy certain constraints.

First of all there is the necessity to adapt the *computer program* to the architecture of the particular computer. More or less, this argument holds for any type of computer but on a vector processor the effects are more pronounced. More serious is the requirement to suit the *algorithm* to the specific architectorial nature of the computer. Traditionally, numerical algorithms were selected on their 'mathematical' qualities, for instance, the rate of convergence in iterative processes. When using a vector computer it is no longer true that algorithms which are 'mathematically' superior to others will result in a better (i.e., faster) performance, which is usually the case on a scalar computer. Therefore, to obtain optimal performance from a vector machine, it is necessary to construct an algorithm which is best suited to that particular machine

(running with a particular compiler). In doing so, we have to consider several aspects which will inhibit vectorisation. If vectorisation has to be performed by the compiler, only the source code will be examined. Evidently, DO-loops will be the most likely places where suitable sequences of operations can be found. Therefore, we should keep these loops going on, working on vectors the length of which is as large as possible. Hence, loops containing IF-statements, GOTO-statements or I/0-statements will inhibit vectorisation. Also certain index expression are a barrier to vectorisation, such as indirect addressing or nonlinear index expression. Moreover, calls to subroutines or functions within DO-loops make these loops nonvectorisable. The reason for this is that subprograms generally are compiled seperately.

However, the most restrictive aspect with respect to vectorisation is *recursion*, which means that in a sequence of evaluations the latest term depends on one or more of the previously computed terms, as, for example, in

```
      DO 10 I=2,N
   10 A(I) = A(I-1)+SCALAR*B(I)
```

Because the evaluation of a recurence relation *essentially* is a sequential process, recurrency conflicts with the nature of vectorisation. Recurrences are quite common in all fields of numerical analysis; examples are the calculation of the innerproduct of two vectors, solutions of linear equations by Gaussian elimination and in principle any iterative process in which a new approximation is calculated using previous approximations. Fortunately, for some of these problems manufacturers of vector computers provide a solution, e.g. the CYBER 205 has a special innerproduct instruction. The recursion which has our special attention occurs in the Gaussian elimination process used for solving tridiagonal systems. These systems frequently occur in the splitting methods as described in subsection 3.4. Therefore, in using these splitting methods we will have to use other techniques to solve the tridiagonal systems (such as recursive doubling or cyclic reduction). However, this usually requires additional arithmetic operations and storage.

A last aspect which we want to consider is the *portability*. Because FORTRAN originates from the fifties it lacks any feature for a standard treatment of vector processing. In consequence, each manufacturer developed his own dialect. Needless to say that this is disastrous for portability.

Notwithstanding these reservations, we think that it will be possible to take advantage of this vector processing machine by adapting both the program and the algorithm to this particular computer. Extensive tests have to show which algorithm is maximally benefitted from this type of computer.

Considering the various methods which are described in Section 3 we make a few remarks.

An aspect which seems to be in favour of the explicit methods (such as the Runge-Kutta schemes) is that these methods work with long vectors, whereas the splitting methods typically operate on vectors the length of which equals the number of points in one space-direction. This aspect is especially

important for the CYBER 205 because this machine has a relatively large start-up time, which is greatly amortized by executing long vectors. Furthermore, the explicit schemes require only **F**(**W**)-evaluations to solve (2.2.6). Because of their explicit nature, the vector **W** is known prior to the calculations within **F**, which makes these schemes — at least in principle — highly vectorisable.

The refinement of the model describing the SWEs will also have influence on the performance of a vector computer. Evidently, the more sophisticated the model is, the more complicated the program will be. For example, very irregular shapes of the boundary (or even time-dependent boundaries which require a flooding and drying procedure) or special treatment of the advection terms in the SWEs in the neighbourhood of the boundaries, etc. Such situations will cause the program to perform a lot of tests to detect these irregularities. These IF-statements as well as the enormous overhead will prevent the program from optimal performance.

A last facet is the ambiguity of the word *performance*. Is it merely the CPU time that counts or do we also take into account the time needed for transput of data? Another definition of performance could be in terms of costs on a particular computer installation or in terms of memory.

4.2 Vertical stratification

The general equations describing the motion of flow in shallow water are, in principle, three-dimensional. However, the enormous computational task such a 3-D system would require in numerical computations, is out of the scope of nowadays computers.

Therefore, in the momentum equation for the vertical velocity component, usually the following assumptions are made:

(i) the vertical acceleration is small with respect to the acceleration of gravity.

(ii) as the horizontal dimensions are large compared with the depth, the vertical velocity is small with respect to the horizontal velocity.

By these assumptions this momentum equation can be drastically simplified yielding a relation between pressure and gravity [7]. Then, this relation can be used to eliminate the pressure from the other momentum equations. A next step is introducing the depth-averaged horizontal velocity components

$$\bar{u} = \frac{1}{h}\int_0^h u\,dz \quad \text{and} \quad \bar{v} = \frac{1}{h}\int_0^h v\,dz.$$

Now, assuming that the free surface and the bottom are streamlines, which serve as vertical boundary conditions, the equations are integrated over the depth which eliminates the vertical velocity component from the system. It is this system which was considered in the previous sections, where the bars were dropped.

In many applications, this traditional approach has proven to be rather

satisfactorily. However if the fluid is not homogeneous with respect to temperature or salinity it may be necessary to consider a model with more than one layer. Now, in each of these layers the SWEs as defined in (1.2) are used, extended with a variable ρ denoting the density in that particular layer. Consequently, ρ will be a function of the temperature or the salinity. Evidently, these layers have to be connected by appropriate interaction conditions, i.e., vertical boundary conditions are imposed assuming that the borders between the layers are streamlines again. These models with vertical stratification have been studied in the literature (see e.g. [14], [41]) but are still in a rather premature stage of development. Further research in this field is necessary in order to obtain a more flexible treatment of realistic models. This technique of using layers is also valuable for modelling the three-dimensional circulation of a homogeneous sea (cf. [3]). Thanks to the ever-increasing computer power the refinement of the models is possible. This may eventually lead to models with many layers or even to fully three-dimensional calculations. Perhaps, in the end, it may give a better comprehension of the phenomenon of turbulence which is still so poorly understood.

4.3 Error estimation

A step forward in shallow-water calculations would be an estimation of the *global* error. The costs of such an estimation may be considerable. For this reason error estimation got little attention. However, a good estimation of the error would increase the reliability of the results, which will be a good starting point for probability calculations in civil engineering projects and thereby may lead to cheaper designs.

In future research on shallow-water equations at the CWI we will concentrate on the use of vector computers and on error estimation.

REFERENCES

1. A. BRANDT, N. DINAR (1979). Multi-grid solutions to elliptic flow problems. S.V. PARTER (ed.). *Numerical Methods for Partial Differential Equations,* Academic Press.
2. G. DAHLQUIST (1959). Stability and error bounds in the numerical integration of ordinary differential equations (thesis). *Transactions of the Royal Institute of Technology,* No. 130, Stockholm.
3. A.M. DAVIES (1980). On formulating a three-dimensional hydronamic sea model with an arbitrary variation of vertical eddy viscosity. *Computer Methods in Applied Mechanics and Engineering 22,* 187-211.
4. K. DEKKER (1980). Semi-discretization methods for partial differential equations on non-rectangular grids. *Int. J. Num. Meth. Engng. 15,* 405-419.
5. K. DEKKER (1981). Stability of linear multistep methods on the imaginary axis. *BIT 21,* 66-79.
6. K. DEKKER, J.G. VERWER (1984). *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations,* North-Holland, Amsterdam.

7. J.J. DRONKERS (1964). *Tidal Computations in Rivers and Coastal Waters,* John Wiley & Sons, New York.
8. G. FAIRWEATHER, I.N. NAVON (1980). A linear ADI method for the shallow-water equations. *J. of Comp. Phys. 37,* 1-18.
9. G. FISCHER (1959). Ein numerisches Verfahren zur Errechnung von Windstau und Gezeiten in Randmeeren. *Tellus 11,* 60-76.
10. D. GOTTLIEB, S.A. ORSZAG (1977). Numerical analysis of spectral methods: theory and applications. *CBMS-WSF. Regional Conference Series in Applied Mathematics, No 26,* SIAM Philadelphia.
11. A. GRAMMELTVEDT (1969). A survey of finite-difference schemes for the primitive equations for a barotropic fluid. *Monthly Weather Review 97, no 5,* 384-404.
12. B. GUSTAFSSON (1971). An alternating direction implicit method for solving the shallow water equations. *J. of Comp. Physics 7,* 239-254.
13. W. HANSEN (1956). Theorie zur Errechnung des Wasserstandes und der Strömungen in Randmeeren nebst Anwendungen. *Tellus 8,* 289-300.
14. N.S. HEAPS (1981). Three-dimensional models for tides and surges with vertical eddy viscosity prescribed in two layers. Part I, Mathematical formulation. *Geophys. J.R. astr. Soc. 64,* 291-302.
15. P.W. HEMKER (1981). Introduction to multi-grid methods. *Nieuw Arch. Wiskunde, Ser. (3) 29,* 71-101.
16. R.W. HOCKNEY, C.R. JESSHOPE (1981). *Parallel Computers,* Adam Hilger Ltd, Bristol.
17. P.J. VAN DER HOUWEN (1977). *Construction of Integration Formulas for Initial Value Problems,* North-Holland Publishing Company, Amsterdam.
18. P.J. VAN DER HOUWEN (1977). *Berekening van Waterstanden in Zeeën en Rivieren* (Dutch), MC Syllabus 33, Mathematical Centre, Amsterdam.
19. P.J. VAN DER HOUWEN, J.G. VERWER (1979). One-step splitting methods for semi-discrete parabolic equations. *Computing 22,* 291-309.
20. P.J. VAN DER HOUWEN, B.P. SOMMEIJER (1983). Analysis of Chebyshev relaxation in multigrid methods for non linear parabolic differential equations. *ZAMM 63,* 193-201.
21. P.J. VAN DER HOUWEN, B.P. SOMMEIJER (1983). Predictor-corrector methods with improved absolute stability regions. *JIMANA 3,* 417-437.
22. R. JELTSCH (1978). Stability on the imaginary axis and A-stability of linear multistep methods. *BIT 18,* 170-174.
23. R. JELTSCH, O. NEVANLINNA (1979). *Stability and Accuracy Discretizations for Initial Value Problems,* Oulu report, Helsinki.
24. R. JELTSCH, O. NEVANLINNA (1981). Stability of explicit time discretizations for solving initial value problems. *Num. Math. 37,* 61-91.
25. J. KOK, P.J. VAN DER HOUWEN, P.H.M. WOLKENFELT (1978). *A Semi-discretization Algorithm for Two-dimensional Partial Differential Equations,* Report NW 54/78, Mathematical Centre, Amsterdam.

26. W. KUTTA (1901). Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. *Z. Math. Phys. 46,* 435-453.
27. J.D. LAMBERT (1973). *Computational Methods in Ordinary Differential Equations,* John Wiley & Sons, London.
28. P. LANCASTER (1969). *Theory of Matrices,* Academic Press, New York and London.
29. J.J. LEENDERTSE (1967). *Aspects of a Computational Model for Long-period Water-wave Propagation,* Rand Corp., Mem. RM-5294, Santa Monica.
30. J.J. LEENDERTSE (1970). A water-quality simulation model for well-mixed estuaries and coastal seas. *Volume I, Principles of Computation,* Rand Corp. Mem. RM-6230- RC, Santa Monica.
31. P. MERLUZZI, C. BROSILOW (1978). Runge-Kutta integration algorithms with built-in estimates of the accumulated truncation error. *Computing 20,* 1-16.
32. A.R. MITCHELL, D.F. GRIFFITHS (1980). *The Finite Difference Method in Partial Differential Equations,* John Wiley & Sons, Chichester.
33. K.W. MORTON (1977). Initial-value problems by finite difference and other methods. D. JACOBS (ed.). *The State of the Art in Numerical Analysis,* Academic Press, London, New York, San Francisco.
34. I.M. NAVON (1978). Application of a new partly implicit time differencing scheme for solving the shallow-water equations. *Contrib. Atmospheric Phys. 51,* 281-305.
35. N. PRAAGMAN (1979). *Numerical Solution of the Shallow Water Equations by a Finite Element Method,* Thesis, Delft.
36. R.D. RICHTMYER, K.W. MORTON (1967). *Difference Methods for Initial Value Problems,* Interscience Publishers, New York.
37. P. SONNEVELD, B. VAN LEER (1985). A minimax problem along the imaginary axis. *Nieuw Arch. Wiskunde, Ser (4) 31,* 19-22.
38. G.S. STELLING (1983). *On the Construction of Computational Methods for Shallow Water Flow Problems,* Thesis, Delft.
39. H.J. STETTER (1973). *Analysis of Discretization Methods for Ordinary Differential Equations,* Springer-Verlag, Berlin.
40. R. VICHNEVETSKY (1983). New stability theorems concerning one-step numerical methods for ordinary differential equations. *Mathematics and Computers in Simulation 25,* 199-205.
41. C.B. VREUGDENHIL (1979). Two-layer shallow-water flow in two dimensions, a numerical study. *J. of Comp. Phys. 33,* 169-184.
42. T.J. WEARE (1976). Finite element or finite difference methods for the two-dimensional shallow water equations. *Computer Methods appl. Mech. Engin. 7,* 351-357.
43. T.J. WEARE (1976). Instability in tidal flow computational schemes. *Journal of the Hydraulics Division, ASCE, 102,* 569-580.
44. P. WESSELING (1982). Theoretical and practical aspects of a multigrid method. *SIAM J. Sci. Stat. Comp. 3,* 387-407.

Primality Testing[1]

H.W. Lenstra, Jr.
University of Amsterdam
P.O. Box 19268, 1000 GG Amsterdam, The Netherlands

Two fundamental problems from elementary number theory are the following:

(*a*) *(primality)* given an integer $n>1$, how can one tell whether n is prime or composite?

(*b*) *(factorization)* if n is composite, how does one find $a, b>1$ such that $n=ab$?

Many mathematicians have been fascinated by these problems throughout history. Among these are Eratosthenes ($\sim -284 - \sim -204$), Fibonacci ($\sim 1180 - \sim 1250$), Fermat (1601-1665), Euler (1707-1783), Legendre (1752-1833) and Gauss (1777-1855). Some of the fascination of the subject derives from the fact that, roughly speaking, problem (*a*) is 'easy' and (*b*) is 'difficult'. Suppose, for example, that two 80-digit numbers p and q have been proved prime; this is easily within reach of the modern techniques for dealing with (*a*). Suppose further, that the cleaning lady gives p and q by mistake to the garbage collector, but that the product pq is saved. How to recover p and q? It must be felt as a defeat for mathematics that, in these circumstances, the most promising approaches are searching the garbage dump and applying mnemo-hypnotic techniques. The 'numerologists' occupying themselves with primality and factorization do not accept this defeat. They imagine all composite numbers to be created by multiplication on the zeroth day of Creation, and they make it their task to unravel the mysteries involved in this process. In this connection, it is remarkable that no clairvoyants have ever been employed to identify Mersenne primes or to factor large numbers. Such an attempt might lead to new insights, if not in numerology then in parapsychology.

1. This paper is a revised version of one of the contributions to [19].

'Numerology' — this condescending term was, until recently, the fashionable one for the branch of science under discussion, in spite of the famous names listed above. Nowadays, a change in this attitude is noticeable. Partly, this change is due to an increased interest in general problems of feasibility of computations. The revival of the specific problems (a) and (b) has, in addition, been stimulated by their striking application in cryptography. For the details of this application we refer to [11]. Suffice it to say that, in this application, it is essential that (a) is 'easy' and that (b) is 'hard'. It is an ironic fact that the only existing evidence for the 'hardness' of (b) is the failure of generations of 'numerologists' to come up with an efficient factorization algorithm.

This lecture is devoted to a discussion of problem (a). For (b) we refer to [26] and [37], and the references given there.

In complexity theory, it is customary to call an algorithm *good* if its running time is bounded by a polynomial in the length of the input. For problems (a) and (b) the input is the number n, which can be specified by $[\log n/\log 2]+1$ binary digits. Thus the length of the input has the same order of magnitude as $\log n$.

A well known algorithm for solving (a) and (b) consists of trial divisions of n by the numbers less than or equal to $\sqrt{n}$. In the worst case, this takes $\sqrt{n}$ steps, which is exponential in the length of the input. We conclude that this algorithm is not 'good'.

Before one searches for a short proof that n is prime, or for a short proof that n is composite, it is a good question to ask whether such a proof exists. In this direction, we first have the following theorem; an *arithmetic operation* is the addition, subtraction or multiplication of two integers.

THEOREM 1. *If n is composite, this can be proved using only $O(1)$ arithmetic operations. Similarly if n is prime.*

PROOF. For composite n, the theorem is trivial; to prove that n is composite, it suffices to write down integers $a, b>1$ and to do the single multiplication necessary to verify that $ab=n$. Thus, in the composite case, the O-symbol is even superfluous. For prime n, the theorem is less obvious. It is an outgrowth of the negative solution of Hilbert's tenth problem [7], that there exists a polynomial in twenty-six variables

$$f\in\mathbb{Z}[\underline{A}, \underline{B}, \underline{C}, \ldots, \underline{X}, \underline{Y}, \underline{Z}]$$

with the property that the set of prime numbers coincides with the set of *positive* values assumed by f if non-negative integers are substituted for $\underline{A}, \underline{B}, \ldots, \underline{Z}$. Such a polynomial, of degree 25, is explicitly given in [12]. A similar polynomial in 10 variables of degree 11281 is constructed in [20; English translation]. To prove that a positive integer n is prime it now suffices to write down twenty-six non-negative integers $A, B, \ldots, Z$ and to do the bounded amount of arithmetic necessary to verify that $n=f(A, B, \ldots, Z)$. In fact, according to [12, Theorem 5] no more than 87 arithmetic operations are needed in this verification. This proves Theorem 1.

From a practical point of view Theorem 1 has two serious defects. The first is, that it tells us that certain proofs exist, but it does not tell us how to find them. Thus, F.N. Cole's proof that $2^{67}-1$ is composite consists of the single observation that

$$2^{67}-1 = 193707721 \cdot 761838257287.$$

But it had taken him 'three years of Sundays' to find his proof, and the methods that he employed are far more interesting than the final proof itself [6], [28].

With primes, the situation is slightly different. The proof that, for prime n, there exist non-negative integers A, B, ..., Z such that

$$n = f(A,B, \ldots, Z)$$

is completely constructive, see [12]. But for the polynomial from [12] it is not difficult to prove that the largest of $A,B, \ldots, Z$ necessarily exceeds

$$n^{n^{n^{n}}}$$

(For a much better polynomial in this respect, see [1, Theorem 3.5].) The second defect of Theorem 1 is, that it is clearly unrealistic to count an addition or multiplication of numbers of this size as a single operation. It is more realistic to count *bit* operations, which may be defined as arithmetic operations on numbers of one digit. Thus, we have:

THEOREM 2. *If n is composite, this can be proved using only $O((\log n)^2)$ bit operations.*

PROOF. It suffices to remark that the usual algorithm to multiply two numbers less than n requires no more than $O((\log n)^2)$ bit operations. This proves Theorem 2.

Using the fast multiplication routine of SCHÖNHAGE and STRASSEN [30], [35] we can replace $(\log n)^2$ in Theorem 2 by $(\log n)^{1+\epsilon}$, for any $\epsilon>0$, or more precisely by $O((\log n)\cdot(\log\log n)\cdot(\log\log\log n))$ (for $n>e^e$).

THEOREM 3 (PRATT [28]). *If n is prime, this can be proved using only $O((\log n)^4)$ bit operations.*

Again, using [30], we can replace $(\log n)^4$ by $(\log n)^{3+\epsilon}$, for any $\epsilon>0$.

PROOF. The proof relies on the structure of the group of units

$$(\mathbb{Z}/n\mathbb{Z})^* = \{(a \bmod n) : a\in\mathbb{Z},\ 0\leqslant a<n,\ \gcd(a, n)=1\}$$

of the ring $\mathbb{Z}/n\mathbb{Z}$ of integers modulo n. This is a finite abelian group of order $\phi(n)$, where ϕ is the Euler function. If n is a prime number, then $(\mathbb{Z}/n\mathbb{Z})^*$ is

cyclic of order $n-1$. Conversely, if $(\mathbb{Z}/n\mathbb{Z})^*$ has order $\geqslant n-1$, then n is a prime number. Thus we see that n is prime if and only if there exists $(a \bmod n) \in (\mathbb{Z}/n\mathbb{Z})^*$ of order $n-1$. If we assume n to be odd and write

$$n-1 = \prod_{i=0}^{k} q_i, \tag{1}$$

$$q_0 = 2$$

$$q_i \text{ prime } (1 \leqslant i \leqslant k) \tag{2}$$

then $(a \bmod n)$ has order $n-1$ in $(\mathbb{Z}/n\mathbb{Z})^*$ if and only if

$$a^{(n-1)/2} \equiv -1 \bmod n, \tag{3}$$

$$a^{(n-1)/q_i} \not\equiv 1 \bmod n, \quad \text{for} \quad 1 \leqslant i \leqslant k. \tag{4}$$

Therefore, to prove that n is prime, we can write down integers a, $q_0 = 2$, $q_1, \ldots, q_k$, verify that (1), (3) and (4) hold, and prove (2) recursively. This proof requires k multiplications in (1), and $k+1$ exponentiations $(\bmod n)$ in (3) and (4), plus what is needed for (2). So if $f(n)$ denotes the total number of multiplications and exponentiations in the proof, then

$$f(n) \leqslant k + k + 1 + \sum_{i=1}^{k} f(q_i)$$

where we define $f(2) = 1$. By induction we prove that $f(n) \leqslant 3 \cdot (\log n / \log 2) - 2$. This is true for $n = 2$, and if it holds for the q_i then

$$\begin{aligned} f(n) &\leqslant 2k + 1 + \sum_{i=1}^{k} (3(\log q_i / \log 2) - 2) \\ &= \left(\sum_{i=0}^{k} 3(\log q_i / \log 2)\right) - 2 \\ &= 3(\log(n-1)/\log 2) - 2 < 3(\log n / \log 2) - 2 \end{aligned}$$

as required.

We conclude that no more that $O(\log n)$ multiplications and exponentiations are needed. Each exponentiation in (3), (4) can be done by $O(\log n)$ squarings and multiplications $\bmod n$. Finally, each of these multiplications, squarings and multiplications $\bmod n$ (or mod a number smaller than n) can be done with $O((\log n)^2)$ bit operations. The total number of bit operations is therefore $O((\log n) \cdot (\log n) \cdot (\log n)^2) = O((\log n)^4)$. This proves Theorem 3.

Theorem 2 and 3 still have the first defect of Theorem 1: one is not told how to *find* the short proof whose existence is asserted. Nevertheless, the proof we have given of Theorem 3 is not exclusively of theoretical interest, and the same ideas are actually used in computer-assisted primality proofs. To illustrate

this, we begin with a particularly simple case, in which $n-1$ has no odd prime factors at all.

THEOREM 4 (PÉPIN, 1877). *Let* $n=2^m+1$, *with* $m>1$. *Then* n *is prime* $\Leftrightarrow$ $3^{(n-1)/2} \equiv -1 \bmod n$.

PROOF. The implication $\Leftarrow$ follows from the proof of Theorem 3, with $a=3$. Conversely, suppose that n is prime. Then n is not divisible by 3, since $n>3$, so m is even. Then $n \equiv 2 \bmod 3$ and $n \equiv 1 \bmod 4$, so quadratic reciprocity gives

$$\left(\frac{3}{n}\right) = \left(\frac{n}{3}\right) = \left(\frac{2}{3}\right) = -1.$$

By Euler's theorem, $\left(\frac{3}{n}\right) \equiv 3^{(n-1)/2} \bmod n$. This proves Theorem 4.

It is known that $n=2^m+1$ can only be prime if n is a power of 2; then n is one of the *Fermat numbers* $2^{2^k}+1$. For $k=0, 1, 2, 3, 4$ these numbers are actually prime, for $5 \leqslant k \leqslant 19$ and some other values (such as $k=2089$) they are known to be composite. It is reasonable to conjecture that they are, in fact, all composite for $k \geqslant 5$. The number F_{14} has been proved composite by Pépin's test, but no factor is known. To the uninitiated reader it may seem surprising that it is possible to prove that a number is composite, without the proof yielding a factorization. This is surprising indeed; the phenomenon will be further discussed at the end of this lecture. See [39, Sec. 5] and [3] for more information on the Fermat numbers.

For general n, the main difficulty of the above test is to find the complete factorization (1) of $n-1$. In the following variant only a partial factorization of $n-1$ is needed.

THEOREM 5. *Let* n *and* s *be integers satisfying*

$$n>1, \quad s>n^{1/2}.$$

Suppose that for every prime q *dividing* s *there exists an integer* a *(depending on* q*) satisfying*

$$a^{q^{m(q)}} \equiv 1 \bmod n, \quad \gcd(a^{q^{m(q)-1}}-1, n)=1 \tag{5}$$

where $m(q)$ *denotes the number of factors* q *in* s. *Then* n *is a prime number.*

PROOF. Let r be any prime dividing n and q any prime dividing s. From (5) we see that the order of $(a \bmod r)$ in the group $(\mathbb{Z}/r\mathbb{Z})^*$ equals $q^{m(q)}$, so by Lagrange's theorem $q^{m(q)}$ divides $r-1$. Since q is arbitrary, this implies that s divides $r-1$, so $r>s$. The inequality $s>n^{1/2}$ shows that n has at most one such prime factor. Hence n is prime, as required. This proves Theorem 5.

From the proof of Theorem 5 we see that the hypotheses imply that s divides $n-1$. To obtain a primality test from Theorem 5, one chooses s to be the largest divisor of $n-1$ that one is able to factor completely. For each q, the number a is constructed as follows. Search for an integer b such that

$$b^{n-1} \equiv 1 \bmod n, \quad b^{(n-1)/q} \not\equiv 1 \bmod n,$$

and put

$$a \equiv b^{(n-1)/q^{m(q)}} \bmod n.$$

If it is difficult to find such a number b, it is unlikely that n is prime, and one should attempt to show that n is composite, using Miller's method described below. The gcd in (5) is now equal to $\gcd(b^{(n-1)/q}-1, n)$, and it can be calculated efficiently using Euclid's algorithm. In fact, only one gcd-computation is necessary if one considers the product of the numbers $b^{(n-1)/q}-1 \bmod n$, with q ranging over the primes dividing s.

The critical condition of Theorem 5 is the inequality $s>n^{1/2}$ that must be satisfied by the completely factored part of $n-1$. There are several ways to replace this condition by a weaker one. Suppose, for example, that s only satisfies

$$s>n^{1/3}.$$

From the proof of Theorem 5 we see that every prime divisor of n is $1 \bmod s$, and the same is then true for *every* divisor. Hence, if n is composite, there exist integers x and y satisfying

$$n = (xs+1)(ys+1), \quad x>0, \quad y>0.$$

From $n<s^3$ it follows that $xy<s$, so $(x-1)(y-1)\geqslant 0$ implies that $0<x+y\leqslant s$. Since $x+y\equiv(n-1)/s \bmod s$ this means that we know the value of $x+y$. We also know that $n=(xs+1)(ys+1)$, so x and y can now be solved from a quadratic equation. Hence, if we add the hypothesis that the solution of this equation does not give rise to a non-trivial factorization of n, we still can conclude that n is a prime number.

A second method to relieve the condition $s>n^{1/2}$ makes use of lower bounds for the unknown prime factors of $n-1$. For a discussion of this technique, and references to the literature, see [39, Sections 10, 11].

Later in this lecture we shall consider a third type of generalization of Theorem 5, in which the role of $n-1$ is played by n^t-1, where t is some positive integer; see Theorem 11.

G.L. MILLER [21] introduced a different way to exploit the multiplicative structure of the integers $\bmod n$ in primality tests. It leads to the following theorem, in which 'GRH' denotes the generalized Riemann hypothesis, formulated in the course of the proof.

THEOREM 6 (MILLER). *Assume the validity of GRH. Then there exists an algorithm, described below, that in $O((\log n)^5)$ steps decides whether or not n is prime.*

This theorem has none of the defects of Theorem 1, 2 and 3, but it has a new one: the assumption of an unproved hypothesis.

Assume that n is odd, and write $n-1=u\cdot 2^k$, where u is odd and $k\geqslant 1$. Employing Rabin's terminology [29], we call an integer a a *witness* to the compositeness of n, or simply a witness for n, if the following three conditions hold:

$$n \text{ does not divide } a, \tag{6}$$

$$a^u \not\equiv 1 \bmod n, \tag{7}$$

$$a^{u\cdot 2^i} \not\equiv -1 \bmod n \text{ for } i=0, \ldots, k-1. \tag{8}$$

(Others say in this situation, that n is 'not a strong base a pseudoprime' ... .)

Whether or not a is a witness for n depends only on $a \bmod n$; so we may restrict to $0\leqslant a<n$. For a given such a, it takes only $O((\log n)^3)$ steps to check whether or not a is a witness for n, by the last paragraph of the proof of Theorem 3.

We note that witnesses are reliable: if a is a witness to the compositeness of n, then n is composite. To see this, suppose that (6), (7), (8) hold and that n is prime. By (6) and Fermat's theorem, $a^{u\cdot 2^k}=a^{n-1}\equiv 1 \bmod n$. Hence the last term in the sequence

$$a^u, a^{u\cdot 2}, \ldots, a^{u\cdot 2^k}$$

is $1 \bmod n$, but by (7) the first term is not $1 \bmod n$. Let $b=a^{u\cdot 2^i}$ be the last term in the sequence that is not $1 \bmod n$. Then $0\leqslant i\leqslant k-1$, and $b^2\equiv 1 \bmod n$ while $b \not\equiv 1 \bmod n$. Hence n divides $b^2-1=(b-1)(b+1)$ but it does not divide $b-1$. Therefore n divides $b+1$, which contradicts (8).

The algorithm referred to in Theorem 6 now runs as follows. We may assume that n is odd, and $n>1$. Check whether there is a witness a for n satisfying $0<a<70(\log n)^2$. If there is one, n is composite. If there is none, declare n to be prime. This algorithm clearly runs in time $O((\log n)^5)$.

To prove the correctness of the algorithm, we have to show that any composite odd n has a positive witness $a<70(\log n)^2$, if GRH is assumed. We sketch this proof only, referring to the literature for details.

First we describe the GRH as we need it. Let n be an arbitrary positive integer, and let $\chi:(\mathbb{Z}/n\mathbb{Z})^*\to\mathbb{C}^*$ (the group of non-zero complex numbers) be a group homomorphism. We view χ as a function on $\mathbb{Z}$ by $\chi(a)=\chi(a \bmod n)$ if $\gcd(a, n)=1$, and $\chi(a)=0$ otherwise. Such a function on $\mathbb{Z}$ is called a *character* modulo n. The L-series associated to χ is defined by

$$L(s, \chi) = \sum_{a=1}^{\infty} \frac{\chi(a)}{a^s}.$$

If χ is non-trivial, i.e. $\chi(a)\notin\{0, 1\}$ for some a, this series converges for all $s\in\mathbb{C}$ with $\mathrm{Re}(s)>0$. We say that $L(s, \chi)$ satisfies the generalized Riemann hypothesis if $L(s, \chi)\neq 0$ for all $s\in\mathbb{C}$ with $\mathrm{Re}(s)>\frac{1}{2}$. For trivial χ, this is only meaningful if $L(s, \chi)$ has been analytically continued; to avoid this, let us simply say that $L(s, \chi)$, for trivial χ, satisfies the generalized Riemann hypothesis

if and only if the classical Riemann hypothesis is true, which is equivalent to

$$\sum_{a=1}^{\infty} \frac{(-1)^a}{a^s} \neq 0 \quad \text{for all} \quad s \in \mathbb{C} \quad \text{with} \quad \tfrac{1}{2} < \mathrm{Re}(s) < 1.$$

The GRH in Theorem 6 is the conjunction of all generalized Riemann hypotheses described above.

LEMMA (ANKENY-MONTGOMERY). *There is an absolute constant c with the following property. Let χ be a non-trivial character modulo n, and suppose that $L(s, \chi)$ satisfies the generalized Riemann hypothesis. Then there exists $a \in \mathbb{Z}$, $0 < a < c \cdot (\log n)^2$, such that $\chi(a) \neq 0$ and $\chi(a) \neq 1$.*

PROOF. See [23, Theorem 13.1], or [13, Corollary 1.3] for a version in which also the classical Riemann hypothesis is needed.

COROLLARY. *Assume GRH, and let $G \neq (\mathbb{Z}/n\mathbb{Z})^*$ be a subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$. Then there exists $a \in \mathbb{Z}$ such that*

$$0 < a < c \cdot (\log n)^2, \quad \gcd(a, n) = 1, \quad (a \bmod n) \notin G,$$

with c as in the lemma.

PROOF. It suffices to apply the lemma to a non-trivial $\chi: (\mathbb{Z}/n\mathbb{Z})^* \to \mathbb{C}^*$ that is trivial on G.

Let now $n > 1$ be composite and odd. To finish the proof of Theorem 6, with an unspecified constant c instead of 70, it suffices, by the corollary, to exhibit a proper subgroup $G \subset (\mathbb{Z}/n\mathbb{Z})^*$ containing all non-witnesses a that are not divisible by n. For this we take (cf. [36])

$$G = \{(a \bmod n) \in (\mathbb{Z}/n\mathbb{Z})^* : a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \bmod n\}$$

where $\left(\frac{a}{n}\right)$ is the Jacobi symbol. It is a charming theorem of LEHMER [14, cf. 33] that $G \neq (\mathbb{Z}/n\mathbb{Z})^*$ for composite odd n. It is an equally charming result of SELFRIDGE [39, Theorem 17.2] that G contains all non-witnesses $(\bmod n)$ not divisible by n. This finishes the proof of Theorem 6.

Using additional arguments it can be proved that the generalized Riemann hypothesis is only needed for the L-series associated to characters χ of the form $\chi(a) = \left(\frac{a}{d}\right)$, where d runs over the positive integers that are $1 \bmod 4$ and either prime or the product of two distinct primes, see [16].

The value 70 for the constant is taken from [24, Théorème 4]; here again the classical Riemann hypothesis is needed, in addition to the generalized Riemann hypotheses just described. It is reported that WEINBERGER (unpublished) obtained sharper results.

The idea used in the proof of Theorem 6 has two other applications. The first is a fast primality test for small numbers:

THEOREM 7 (SELFRIDGE, WAGSTAFF). *Every odd composite n*

satisfying:	*has a witness among:*
$n<2047$	2
$n<1373653$	2,3
$n<2\cdot 10^9$, $n \neq$ 25326001, 161304001, 960946321, 1157839381	2,3,5
$n<25\cdot 10^9$, $n \neq 3215031751$	2,3,5,7

PROOF. By computer, see [27]. This proves Theorem 7.

The numbers in the left hand column are composite:

$$2047 = 23\cdot 89,$$
$$1373653 = 829\cdot 1657,$$
$$25326001 = 2251\cdot 11251,$$
$$161304001 = 7333\cdot 21997,$$
$$960946321 = 11717\cdot 82013,$$
$$1157839381 = 24061\cdot 48121,$$
$$3215031751 = 151\cdot 751\cdot 28351.$$

The test provided by Theorem 7 is easily implemented on a programmable pocket calculator. Thus, an HP-41C can decide the primality of an arbitrary $n<2\cdot 10^9$ within two minutes, using only 2, 3, 5 as possible witnesses.

The second application is based on the following theorem.

THEOREM 8 (RABIN). *Every odd composite n has at least* $\frac{3}{4}(n-1)$ *witnesses among* $\{1, 2, \ldots, n-1\}$.

The proof is an attractive exercise in elementary number theory, in which the Carmichael numbers play a role. See [29], [22]. This proves Theorem 8.

Rabin proposes the following primality test. Let m be a large integer, like 100, and choose randomly m integers $a_i \in \{1, 2, \ldots, n-1\}$, $1 \leqslant i \leqslant m$. If one of these a_i is a witness for n, then n is composite. If none of the a_i is a witness for n, then either n is prime or we have extremely bad luck. By Theorem 8, this bad luck occurs in at most one out of every 4^m cases. While this method is basically incapable of yielding rigorous primality proofs, it is in practical circumstances difficult to doubt that it yields correct answers. In any case, Rabin's method can be used to produce primes on a commercial basis: if found defective, they can easily be replaced.

If we try to remove the unproved assumption from Theorem 6 we are left with an algorithm that is no longer 'good':

THEOREM 9 (ADLEMAN, POMERANCE, RUMELY). *There is an algorithm that within* $(\log n)^{c' \log\log\log n}$ *steps decides whether or not* n *is prime, for* $n > e^e$. *Here* c' *denotes an effectively computable constant.*

A complete proof of this theorem can be found in [2] and [17]. A probabilistic version of the algorithm, which is somewhat easier to explain, will be described below. This version of the algorithm has been implemented by H. COHEN and A.K. LENSTRA on the CDC-Cyber 170-750 computer system of the SARA Computer Centre in Amsterdam, cf. [4], [5]. It is the only primality test in existence that can routinely handle numbers of up to 100 decimal digits, and it does so within approximately 45 seconds. Numbers of up to 200 decimal digits are dealt with within approximately 10 minutes.

The algorithm that we shall describe can be viewed as a special case of the following primality criterion.

THEOREM 10. *Let* $n > 1$ *be an integer. Then* n *is prime if and only if every divisor of* n *is a power of* n.

The proof is left to the reader.

To prove that n is prime using Theorem 10 we must check that any divisor of n is a power of n, and it clearly suffices to consider only *prime* divisors of n. Below we shall see how to do this without explicitly knowing the prime divisors of n. Actually, something weaker will be done: rather than showing that a prime r dividing n is a power of n, one attempts to show that this is true for the images of r and n in certain auxiliary groups, such as the group $(\mathbb{Z}/s\mathbb{Z})^*$ for an integer s that is coprime to n.

An example of this approach is provided by Theorem 5 and its proof: in that theorem we have $n \equiv 1 \bmod s$, and the proof proceeds by showing that any prime divisor r of n satisfies $r \equiv 1 \bmod s$, i.e. is congruent to a power of n modulo s. The following theorem provides a less trivial example.

THEOREM 11. *Let* n *and* s *be positive integers, and let* A *be a commutative ring with* 1 *containing* $\mathbb{Z}/n\mathbb{Z}$ *as a subring (with the same* 1*). Suppose that there exists* $\alpha \in A$ *satisfying the following conditions:*

(9) $\alpha^s = 1$,

(10) $\alpha^{s/q} - 1 \in A^*$ *(the group of units of* A*) for every prime* q *dividing* s,

(11) *the polynomial* $\prod_{i=0}^{t-1} (X - \alpha^{n^i})$ *has coefficients in* $\mathbb{Z}/n\mathbb{Z}$ *for some positive integer* t.

Then every divisor r *of* n *is congruent to a power of* n *modulo* s.

PROOF. We may assume that r is *prime*. Since r is a zero divisor (or zero) in A, there exists a maximal ideal M of A with $r \in M$. Let $\bar{A}$ be the field A/M, and $\bar{\alpha} = (\alpha \bmod M) \in \bar{A}$. By (9) and (10), the order of $\bar{\alpha}$ in $\bar{A}^*$ equals s. The polynomial $\prod_{i=0}^{t-1} (X - \bar{\alpha}^{n^i})$, which has $\bar{\alpha}$ as a zero, has coefficients in the

subfield $\mathbb{F}_r$ of $\bar{A}$ of cardinality r. Therefore $\bar{\alpha}^r$ is also a zero of this polynomial, so there exists $i \in \{0, 1, ..., t-1\}$ with $\bar{\alpha}^r = \bar{\alpha}^{n^i}$, i.e. $r \equiv n^i \bmod s$. This proves Theorem 11.

If we take $A = \mathbb{Z}/n\mathbb{Z}$ and $t = 1$, then condition (11) is trivially satisfied. It is easy to deduce Theorem 5 from Theorem 11, by choosing α equal to the product of the a's appearing in Theorem 5, taken modulo n.

The proof of Theorem 11 shows that the residue classes $1, n, n^2, ..., n^{t-1}$ modulo s are permuted upon multiplication by $(r \bmod s)$, for any prime r dividing n. Writing n as the product of its prime factors, we see that multiplication by $(n \bmod s)$ also permutes these residue classes, which just means that $n^t \equiv 1 \bmod s$. Hence s must be chosen to be a divisor of $n^t - 1$.

Let $t = 2$. In this case known prime factors of $n+1 = (n^2-1)/(n-1)$ can be used in addition to those of $n-1$ to build up the number s. Starting from Theorem 11 one can, for practically every primality test based on factors of $n-1$, devise a corresponding test based on factors of $n+1$. These tests are usually formulated in terms of Lucas functions [39, Sections 12, 13, 14]. In the simplest case, corresponding to Pépin's Theorem 4, the number $n+1$ is a power of 2:

THEOREM 12 (LUCAS-LEHMER). *Let $n = 2^m - 1$, with $m > 2$. Define $(e_k)_{k=1}^{\infty}$ by $e_1 = 4$, $e_{k+1} = e_k^2 - 2$. Then n is prime if and only if $e_{m-1} \equiv 0 \bmod n$.*

PROOF. First let m be *even*. Then n is divisible by 3, and not prime. Also $e_{m-1} \equiv -1 \bmod 3$ by induction, so $e_{m-1} \not\equiv 0 \bmod n$. This proves the theorem for even m. Assume now that m is *odd*, and define

$$A = (\mathbb{Z}/n\mathbb{Z})[T]/(T^2 - \sqrt{2}\,T - 1),$$

where $\sqrt{2}$ denotes any element of $\mathbb{Z}/n\mathbb{Z}$ with $(\sqrt{2})^2 = 2$; e.g., $\sqrt{2} = (2^{(m+1)/2} \bmod n)$. Denoting the image of T in A by α we have

$$A = \{a + b\alpha : a, b \in \mathbb{Z}/n\mathbb{Z}\}, \quad \alpha^2 = \sqrt{2}\,\alpha + 1.$$

Let $\beta = \sqrt{2} - \alpha = -\alpha^{-1}$ be 'the' other zero of $X^2 - \sqrt{2}\,X - 1$ in A. From $\alpha + \beta = \sqrt{2}$ and $\alpha\beta = -1$ it follows easily by induction on k that

$$\alpha^{2^k} + \beta^{2^k} = (e_k \bmod n) \in \mathbb{Z}/n\mathbb{Z}$$

for all $k \geqslant 1$. Now let first n be prime. The discriminant of $X^2 - \sqrt{2}\,X - 1$ equals 6, and from $n \equiv 1 \bmod 3$, $n \equiv -1 \bmod 8$ and quadratic reciprocity it follows that $\left(\frac{6}{n}\right) = -1$. Hence A is a quadratic field extension of $\mathbb{F}_n$, and α and β are conjugate over $\mathbb{F}_n$. By the theory of finite fields this implies that $\alpha^n = \beta$. Multiplying this by α we get $\alpha^{2^m} = -1$, so

$$(e_{m-1} \bmod n) = \alpha^{2^{m-1}} + \beta^{2^{m-1}} = \alpha^{2^{m-1}} + \alpha^{-2^{m-1}} = 0.$$

This proves the 'only if' part. Suppose, conversely, that $(e_{m-1} \bmod n) = 0$. Then

$$\alpha^{2^m} = -1, \quad \alpha^{2^{m+1}} = 1,$$

so (9) and (10) of Theorem 11 are satisfied with $s = 2^{m+1}$. Also, $\alpha^n = \alpha^{2^m - 1} = -\alpha^{-1} = \beta$, so the polynomial

$$(X-\alpha)(X-\alpha^n) = (X-\alpha)(X-\beta) = X^2 - \sqrt{2} \cdot X - 1$$

has coefficients in $\mathbb{Z}/n\mathbb{Z}$, which is condition (11) of Theorem 11 with $t=2$. From Theorem 11 and $n^2 \equiv 1 \bmod s$ it now follows that any divisor of n is congruent to 1 or n modulo s. But $s > n$, so this means that n is prime. This proves Theorem 12.

It is known that $n = 2^m - 1$ can only be prime if m is prime: then n is one of the *Mersenne numbers* $M_p = 2^p - 1$, p prime. These are known to be prime for 30 values of p:

2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279,
2203, 2281, 3217, 4253, 4423, 9689, 9941, 11213, 19937,
21701, 23209, 44497, 86243, 132049, 216091,

see [34]. It is reasonable to conjecture that $\#\{m < x: 2^m - 1 \text{ is prime}\}/\log x$ tends to a finite non-zero limit for $x \to \infty$. GILLIES [9] gives a probabilistic argument leading to the value $2/\log 2$ for the limit, but his reasoning is clearly in error since the same argument leads to a contradiction with the prime number theorem, cf. [10, § 22.20]. The number $e^\gamma/\log 2$, where γ is Euler's constant, has been proposed as a more likely value for the limit [25]; see also [38], [31].

If the complete factorization of $n-1$ is known then in practice it is easy to test n for primality, e.g. using Theorem 5. The same statement is true with $n-1$ replaced by $n+1$, using Theorem 11 with $t=2$. A combination of both tests is employed in the discovery of large *twin primes,* in the following way. Let m be a large number whose complete prime factorization is known; such a number can be found by multiplying together small numbers. Then $(m+1)-1$ and $(m-1)+1$ are completely factored, so we can apply an $(n-1)$-primality test to $m+1$ and an $(n+1)$-primality test to $m-1$. If both numbers turn out to be prime we have found a pair of twin primes. The largest known pair is

$$256200945 \cdot 2^{3426} \pm 1 = 2^{3426} \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 113 \cdot 151 \pm 1,$$

which have 1040 decimal digits. This pair was discovered by ATKIN and RICKERT [8].

We next discuss how Theorem 11 can for general t be used for primality testing. For A one takes a ring that if n were prime would be the field $\mathbb{F}_{n^t}$ of n^t elements. If n behaves as if it were a prime number then such a ring is in practice not difficult to construct: as in the proof of Theorem 12 one can take $A = (\mathbb{Z}/n\mathbb{Z})[T]/(f)$, where $f \in (\mathbb{Z}/n\mathbb{Z})[T]$ is a polynomial of degree t that passes a suitable irreducibility test (see [15, Sec. 5]). For s one takes the largest divisor

of n^t-1 that one is able to factor completely, and for α one takes an element of A^* of order s. If n is actually prime then α is usually easy to construct, by manipulating with elements of the form $\beta^{(n^t-1)/s}$, $\beta\in A$. In this case conditions (9) and (10) are clearly satisfied, and the polynomial in (11) is a power of the irreducible polynomial of α over $\mathbb{F}_n$ so it has certainly coefficients in $\mathbb{F}_n$. Suppose, conversely, that (9), (10) and (11) are found to be true. Then we cannot immediately conclude that n is prime, but we know, by Theorem 11, that any r dividing n is congruent to a power of n modulo s. If s is sufficiently large then this information can be used to finish the primality proof, in the following manner. Suppose that

$$s>n^{1/2}$$

(as in Theorem 5), and let r_i be determined by

$$n^i \equiv r_i \bmod s, \quad 0\leqslant r_i<s$$

for $0\leqslant i<t$. If n is composite then it has a non-trivial divisor r with $r\leqslant n^{1/2}<s$, and since r is congruent to a power of n modulo s it must be equal to one of the r_i. Hence, if we verify that none of the r_i is a non-trivial divisor of n, we have proved that n is prime. A similar but somewhat more involved procedure can be followed if s satisfies the weaker inequality $s>n^{1/3}$, see [18].

We refer to [17, Theorem 8.4] for a more flexible version of Theorem 11, in which it is possible to vary α with q, as in Theorem 5.

For very small values of t, such as $t=2$, 3, 4, 6, it is again possible to employ lower bounds for the unknown prime divisors of n^t-1, cf. [39, Sections 13-16] and the references given there. It is doubtful whether such lower bounds are equally useful for the larger values of t considered below.

To analyze the above algorithm we must know how to choose t such that $s>n^{1/2}$. We need the following theorem.

THEOREM 13 (ODLYZKO-POMERANCE). *There exists an effectively computable constant c'' with the following property. For every integer $n>e^e$ there exists a positive integer t satisfying*

$$t<(\log n)^{c''\log\log\log n}$$

$$t \text{ is squarefree}$$

such that the number

$$s=\prod_{q \text{ prime},\ q-1 \text{ divides } t} q$$

satisfies

$$s>n^{1/2}.$$

PROOF. See [2, Sec. 6]. This proves Theorem 13.

Let t be as in Theorem 13; the condition that t be squarefree is irrelevant for our present purpose. If q is a prime number for which $q-1$ divides t, then

$n^t \equiv 1 \bmod q$ by Fermat's theorem, unless q divides n. Hence, if s is as in the theorem, then s divides $n^t - 1$ provided that $\gcd(n, s) = 1$. Also, the complete factorization of s is known, and $s > n^{1/2}$. We conclude that these values for t and s can be used in the primality test described above. The resulting algorithm has, for prime n, an expected running time that is less than $(\log n)^{c' \log\log\log n}$ for some constant c'. This does not yet prove Theorem 9, since we have no such bound for the worst case running time. It appears that the size of t makes the test unsuitable for practical primality testing.

The test underlying Theorem 9 is closely related to the test just described. It depends on properties of *Gauss sums,* which we shall now consider. By ζ_m we denote a primitive m-th root of unity.

Let p and q be prime numbers not dividing n for which p divides $q-1$. We choose a character $\chi = \chi_{p,q}$ modulo q that has order p; i.e., $\chi: \mathbb{F}_q^* \to <\zeta_p>$ is a surjective group homomorphism, where $<\zeta_p>$ denotes the subgroup of $\mathbb{C}^*$ generated by ζ_p. Such a χ can be obtained by choosing a primitive root g modulo q and putting $\chi(g^i \bmod q) = \zeta_p^i$ for $i \in \mathbb{Z}$. We define the *Gauss sum* $\tau(\chi)$ by

$$\tau(\chi) = \sum_{x=1}^{q-1} \chi(x)\zeta_q^x.$$

This is an element of the cyclotomic ring $R = \mathbb{Z}[\zeta_p, \zeta_q]$. We have

$$\tau(\chi)^n \equiv \chi(n)^{-n} \cdot \tau(\chi^n) \bmod nR \quad \text{if } n \text{ is } \textit{prime}.$$

To prove this, notice that modulo nR we have

$$\begin{aligned} \tau(\chi)^n &\equiv \sum_{x=1}^{q-1} \chi(x)^n \cdot \zeta_q^{nx} \quad \text{(since } n \text{ is prime)} \\ &= \sum_{y=1}^{q-1} \chi(n)^{-n} \cdot \chi(y)^n \cdot \zeta_q^y \quad \text{(with } y \equiv nx \bmod q) \\ &= \chi(n)^{-n} \cdot \tau(\chi^n), \end{aligned}$$

as required. We investigate what can, conversely, be said about n if the following weaker condition is satisfied:

$$\tau(\chi)^n \equiv \eta(\chi)^{-n} \cdot \tau(\chi^n) \bmod nR \quad \text{for some } \eta(\chi) \in <\zeta_p>. \tag{12}$$

Let σ be the automorphism of R with $\sigma(\zeta_p) = \zeta_p^n$ and $\sigma(\zeta_q) = \zeta_q$. Then (12) can be written as

$$\tau(\chi)^{n-\sigma} \equiv \eta(\chi)^{-n} \bmod nR.$$

Raising both sides to the power $\sum_{i=0}^{p-2} n^{p-2-i}\sigma^i$ we obtain:

$$\tau(\chi)^{n^{p-1}-1} \equiv \eta(\chi) \bmod nR.$$

Now let r be any prime divisor of n. Then we know that (12), with n replaced by r and $\eta(\chi)$ by $\chi(r)$, is valid, so for the same reason we have

$$\tau(\chi)^{r^{p-1}-1} \equiv \chi(r) \bmod rR\,.$$

Combination of the last two congruences suggests that

$$\chi(r) = \eta(\chi)^{(r^{p-1}-1)/(n^{p-1}-1)} \tag{13}$$

for any prime r dividing n. To make this meaningful we have to explain how to interpret the fractional exponent. For this we need the following hypothesis on p:

$$v_p(r^{p-1}-1) \geqslant v_p(n^{p-1}-1) \quad \text{for every prime } r \text{ dividing } n, \tag{14}$$

where $v_p(m)$ denotes the number of factors p in m. If (14) is satisfied we can write $(r^{p-1}-1)/(n^{p-1}-1)=a/b$, with $a, b \in \mathbb{Z}$, $b \equiv 1 \bmod p$, and the residue class of $(r^{p-1}-1)/(n^{p-1}-1) \bmod p$ is then defined to be $(a \bmod p)$; this does not depend on the choice of a and b. Since $\eta(\chi)^p=1$ it is now meaningful to define the right hand side of (13) as $\eta(\chi)^a$.

With this interpretation it is straightforward to verify that (12) implies (13), provided that (14) is assumed. By induction on the number of prime factors one can now prove that (13) holds for *any* divisor r of n, prime or not. In particular, with $r=n$ we obtain $\chi(n)=\eta(\chi)$, so (13) now yields

$$\chi(r) = \chi(n)^{(r^{p-1}-1)/(n^{p-1}-1)} \tag{15}$$

for any r dividing n. Again we see that every divisor of n is a power of n, if images under χ are taken.

It is a vital question how to verify hypothesis (14). Trivially, we have

$$\text{if } n^{p-1} \not\equiv 1 \bmod p^2, \quad \text{then (14) holds.} \tag{16}$$

In [17, Sec. 2] it is proved that

$$\text{if (12) holds with } \eta(\chi) \neq 1, \text{ then (14) is true.} \tag{17}$$

The primality test based on the preceding theory runs as follows. Let t be a positive integer having all properties listed in Theorem 13, and let s have the same meaning as in that theorem. Choose, for every pair of prime numbers p, q with q dividing s and p dividing $q-1$ (so p dividing t) a character $\chi=\chi_{p,q}$ as above, and check that $\chi=\chi_{p,q}$ satisfies (12); we know that this is necessary for n to be prime. Next, attempt to prove that every prime p dividing t satisfies hypothesis (14). Usually, for each p there is a q dividing s with $\eta(\chi_{p,q}) \neq 1$, and then (17) applies. If there is no such q, and (16) does not apply either, one should test (12) for characters $\chi_{p,q}$ with q a prime *not* dividing s for which p divides $q-1$, until an example of $\eta(\chi_{p,q}) \neq 1$ is found.

At this stage of the algorithm one knows that every $\chi_{p,q}$, with p dividing $q-1$ and q dividing s, satisfies (15) for each r dividing n. We claim that this implies that each r is congruent to a power of n modulo s, so that the test can be completed in the same way as the test described before Theorem 13.

To prove the claim, let r divide n, and let $(i \bmod t)$ be determined by

$$i \equiv (r^{p-1}-1)/(n^{p-1}-1) \bmod p$$

(in the sense explained before) for each prime p dividing t; notice that here we use that t is squarefree. Then (15) implies that

$$\chi_{p,q}(r) = \chi_{p,q}(n^i)$$

for each pair p, q as above. For fixed q, the product of the primes p dividing $q-1$ equals $q-1$, so the characters $\chi_{p,q}$ generate the group of all characters modulo q; therefore $r \equiv n^i \bmod q$. Since this holds for all q dividing s, we conclude that $r \equiv n^i \bmod s$, as required.

The only non-deterministic part of the test is the verification of hypothesis (14). If n is composite it is conceivable that (14) is not satisfied, so that the algorithm will get stuck at this point. We refer to [2, Sec. 5] and [17, Sec. 5] for a variant that avoids hypothesis (14). It constructs an auxiliary number v such that from a set of conditions similar to (12) it can be deduced that any divisor r of n is congruent to a power of v, rather than a power of n, modulo s. This test is completely deterministic, and it has running time less than $(\log n)^{c' \log\log\log n}$ for $n > e^e$, where c' denotes an effectively computable constant. This concludes our sketch of the proof of Theorem 9.

There are several ways to improve the practical performance of the test [5], [17]. In the first place, the Gauss sums can be replaced by Jacobi sums, which belong to $\mathbb{Z}[\zeta_p]$ rather than $\mathbb{Z}[\zeta_p, \zeta_q]$. Secondly, characters of prime power order rather than of prime order can be employed, so that the condition that t be squarefree can be dropped. Finally, it is possible to combine the test with the tests described earlier depending on variants of Theorem 11. However, none of these improvements reduces the running time in a theoretically significant way.

As we noted in connection with the Fermat numbers, it is surprising that we can prove that a number is composite without actually finding a factor. To analyze this situation, let us assume that we proved n composite by exhibiting an integer a for which

$$a^{n-1} \not\equiv 1 \bmod n, \quad \gcd(a, n) = 1, \tag{18}$$

and applying Fermat's theorem that (18) is impossible for prime n. To see why this gives no factorization of n we must investigate how Fermat's theorem is proved. One proof is based on the remark that the map sending i to $a \cdot i \pmod n$ is a permutation of $\{1, 2, \ldots, n-1\}$, so

$$a^{n-1} \cdot (n-1)! = \prod_{i=1}^{n-1} (a \cdot i) \equiv \prod_{i=1}^{n-1} i = (n-1)! \bmod n.$$

Hence (18) implies that $(n-1)!$ has a non-trivial gcd with n, which tells us nothing more than that n is composite. Other proofs of Fermat's theorem have similar shortcomings. The situation would be different if factorials or binomial coefficients were easy to compute modulo n. This is clear from the proof of the following charming but useless theorem, in which we also consider 'division with remainder' as an arithmetic operation.

THEOREM 14 (SHAMIR). *There is an algorithm that for every composite n yields a non-trivial divisor of n, using no more than* $O(\log n)$ *arithmetic operations.*

PROOF. We notice that n is composite if and only if $1<\gcd(a_0!, n)<n$ for some positive integer a_0. Since $\gcd(a!, n)$ is a non-decreasing function of a, and is equal to 1, n for $a=1$, n respectively, we can determine such an a_0 by $O(\log n)$ bisections, provided that we know how to calculate $\gcd(a!, n)$.

Once we know $a!$, we can determine the gcd by Euclid's algorithm in $O(\log n)$ arithmetic steps. To calculate $a!$, we apply the formulae

$$(2b+1)! = (2b+1)\cdot(2b)!,$$

$$(2b)! = (b!)^2\cdot\binom{2b}{b}$$

$O(\log a)$ times. To calculate the binomial coefficient $\binom{2b}{b}$ needed here, we remark that $\binom{2b}{b}$ is the middle block of n binary digits in the binary expansion of $(2^n+1)^{2b}$, for $2b<n$; and the exponentiation can be done by $O(\log(2b))$ multiplications.

This algorithm, as we described it, takes $O((\log n)^3)$ arithmetic operations. For the modifications to bring it down to $O(\log n)$ we refer to Shamir's paper [32]. This concludes the proof of Theorem 14.

We notice that the best known deterministic factorization algorithm, which is due to Pollard, also depends on the calculation of factorials modulo n. This algorithm and several more practical ones are described in the papers of POMERANCE [26] and VOORHOEVE [37].

REFERENCES

1. L. ADLEMAN, K. MANDERS (1976). Diophantine complexity. *17th Annual IEEE Symp. on Foundations of Computer Science,* 81-88.
2. L.M. ADLEMAN, C. POMERANCE, R.S. RUMELY (1983). On distinguishing prime numbers from composite numbers. *Ann. of Math. 117,* 173-206.
3. R.P. BRENT, J.M. POLLARD (1981). Factorization of the eighth Fermat number. *Math. Comp. 36,* 627-630.
4. H. COHEN, A.K. LENSTRA (1985). *Implementation of a New Primality Test,* Report CS-R8505, CWI, Amsterdam; *Math. Comp.,* to appear.
5. H. COHEN, H.W. LENSTRA, JR. (1984). Primality testing and Jacobi sums. *Math. Comp. 42,* 297-330.
6. F.N. COLE (1903/4). On the factoring of large numbers. *Bull. Amer. Math. Soc. 10,* 134-137.
7. M. DAVIS (1973). Hilbert's tenth problem is unsolvable. *Amer. Math. Monthly 80,* 233-269.

8. M. GARDNER (1981). Mathematical games. *Scientific American 244* (2), 14-19.
9. D.B. GILLIES (1964). Three new Mersenne primes and a statistical theory. *Math. Comp. 18,* 93-97.
10. G.H. HARDY, E.M. WRIGHT (1979). *An Introduction to the Theory of Numbers,* 5th ed., Oxford University Press.
11. P.J. HOOGENDOORN. On a secure public-key cryptosystem, pp. 159-168 in [19].
12. J.P. JONES, D. SATO, H. WADA, D. WIENS (1976). Diophantine representation of the set of prime numbers. *Amer. Math. Monthly 83,* 449-464.
13. J.C. LAGARIAS, H.L. MONTGOMERY, A.M. ODLYZKO (1979). A bound for the least prime ideal in the Chebotarev density theorem. *Invent. Math. 54,* 271-296.
14. D.H. LEHMER (1976). Strong Carmichael numbers. *J. Austral. Math. Soc. Ser. A 21,* 508-510.
15. A.K. LENSTRA. Factorization of polynomials, pp. 169-198 in [19].
16. H.W. LENSTRA, JR. (1979). Miller's primality test. *Inform. Process. Lett. 8,* 86-88.
17. H.W. LENSTRA, JR. (1981). Primality testing algorithms (after ADLEMAN, RUMELY and WILLIAMS), Séminaire Bourbaki *33,* (1980/1981). no. 576, pp. 243-257 in: *Lecture Notes in Mathematics 901,* Springer, Berlin.
18. H.W. LENSTRA, JR. (1984). Divisors in residue classes. *Math. Comp. 42,* 331-340.
19. H.W. LENSTRA, JR., R. TIJDEMAN (eds.) (1982). *Computational Methods in Number Theory,* Mathematical Centre Tracts 154/155, Mathematisch Centrum, Amsterdam.
20. YU.V. MATIJASEVIČ (1981). Primes are nonnegative values of a polynomial in 10 variables. *Zap. Naučn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (LOMI) 68* (1977), 62-82 (Russian; English translation: *J. Soviet Math. 15,* 33-44).
21. G.L. MILLER (1976). Riemann's hypothesis and tests for primality. *J. Comput. System. Sci. 13,* 300-317.
22. L. MONIER (1980). Evaluation and comparison of two efficient probabilistic primality testing algorithms. *Theoret. Comput. Sci. 12,* 97-108.
23. H.L. MONTGOMERY (1971). Topics in multiplicative number theory. *Lecture Notes in Mathematics 227,* Springer, Berlin.
24. J. OESTERLÉ (1979). Versions effectives du théorème de Chebotarev sous l'hypothèse de Riemann généralisée. Journées Arithmétiques de Luminy, *Astérisque 61,* 165-167.
25. C. POMERANCE. (1981). Recent developments in primality testing. *Math. Intell. 3,* 97-105.
26. C. POMERANCE. Analysis and comparison of some integer factoring algorithms, pp. 89-139 in [19].

27. C. POMERANCE, J.L. SELFRIDGE, S.S. WAGSTAFF, JR. (1980). The pseudoprimes to $25 \cdot 10^9$. *Math. Comp. 35,* 1003-1026.
28. V.R. PRATT (1975). Every prime has a succinct certificate. *SIAM J. Comput. 4,* 214-220.
29. M.O. RABIN (1980). Probabilistic algorithm for testing primality. *J. Number Theory 12,* 128-138.
30. A. SCHÖNHAGE, V. STRASSEN (1971). Schnelle Multiplikation grosser Zahlen. *Computing 7,* 281-292.
31. M.R SCHROEDER (1983). Where is the next Mersenne prime hiding? *Math. Intell. 5,* 31-33.
32. A. SHAMIR (1979). Factoring numbers in $O(\log n)$ arithmetic steps. *Inform. Process. Lett. 8,* 28-31.
33. R. SOLOVAY, V. STRASSEN (1978). A fast Monte-Carlo test for primality. *SIAM J. Computing 6* (1977), 84-85; erratum: *7,* 118.
34. H.J.J. TE RIELE. Perfect numbers and aliquot sequences, pp. 141-157 in [19].
35. J.W.M. TURK. Fast arithmetic operations on numbers and polynomials, pp. 43-54 in [19].
36. J. VÉLU (1978). Tests for primality under the Riemann hypothesis. *SIGACT News 10,* 58-59.
37. M. VOORHOEVE. Factorization algorithms of exponential order, pp. 79-87 in [19].
38. S.S. WAGSTAFF, JR. (1983). Divisors of Mersenne numbers. *Math. Comp. 40,* 385-397.
39. H.C. WILLIAMS (1978). Primality testing on a computer. *Ars Combin. 5,* 127-185.

Algorithmics

Towards programming as a mathematical activity

Lambert Meertens

Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Of the various approaches to program correctness, that of "Transformational Programming" appears to be the most helpful in constructing correct programs. The essence of the method is to start with an obviously correct—but possibly hopelessly inefficient—algorithm, and to improve it by successively applying correctness-preserving transformations. The manipulations involved are akin to those used in mathematics. Two important impediments to this method are the verbosity of algorithmic notations, making the process cumbersome, and the semantic baroqueness of many primitives, making it hard to verify the validity of transformations. Computer Science can profit here from the lessons taught by the history of Mathematics. Another major step, comparable to one made long ago in Mathematics, is not to insist on the "executability" of algorithmic descriptions. This makes it possible to treat initial high-level specifications in the same framework as the final programs. Just as Mathematics evolved from "Transformational Arithmetic", Transformational Programming may come of age as "Algorithmics".

> Mathematical reasoning does play an essential role in all areas of computer science which have developed or are developing from an art to a science. Where such reasoning plays little or no role in an area of computer science, that portion of our discipline is still in its infancy and needs the support of mathematical thinking if it is to mature. RALSTON and SHAW [25]

0. INTRODUCTION

The historical roots of Mathematics and Computing are intertwined. If we ascertain the validity of a more efficient way of doing computations—more generally, of constructing a result—, we are performing mathematics.

Nowadays, we are happy to leave the actual computing to automata. Our task is to prescribe the process, by means of a program. But however great the speed of our automaton, our need for results is greater, and an important part of the Art of Programming is finding efficient computational methods. Whoever thinks now that programming as it is practised implies routinely giving mathematical justifications—albeit informal—of the "shortcuts" employed, is deceived. This would not be an issue if making an error in programming were exceptional. The current deplorable state of affairs can certainly be partially

ascribed to the ineptitude and ignorance of many programmers. But this is not the full explanation. It is true that Computer Science has yielded a number of results that make it possible to reason mathematically about programming, i.e., constructing a program that satisfies a given specification. But what is lacking is a manageable set of mathematical instruments to turn programming into an activity that is mathematical in its methods. To make it possible to discuss the—as yet hypothetical—discipline that would then be practised, I shall use the term "Algorithmics".

Mathematicians portrayed in cartoons are invariably staring at a blackboard covered with squiggles. To outsiders, mathematics = formulae. Insiders know that this is only the surface. But, undeniably, mathematics has only taken its high flight because of the development of algebraic notation, together with concepts allowing algebraic identities.[1]

The work reported on here has been motivated by the conviction that major parts of the activities of algorithm specification and construction should and can be performed in much the same way as that in which mathematicians ply their trade, and that we can profit in this respect from studying the development of Mathematics. Earlier work, based on the same conviction, can be found in GEURTS and MEERTENS[11] and MEERTENS[19]. In brief, the idea is that algorithms are developed by manipulating "algorithmic expressions". To be able to do this, we need a language that is capable of encompassing both specifications and programs. But, and this is important, this language should not be the union of two different languages, one a specification language, and the other a programming language. Rather, the language must be homogeneous: it must be possible to view all its expressions as specifications. Some of these expressions may, however, suggest a construction process more readily than others. Alternatively, all expressions can be viewed as abstract algorithms. Some of these algorithms may be so abstract, however, that they do not suggest an implementation.

The language should be comparable to the language used by mathematicians. Its *notations* give a convenient way to express *concepts* and thus facilitate reasoning, and also sustain more "mechanical" modes of transforming expressions (in the sense in which a mathematician transforms x^2-y^2 mechanically into $(x+y)(x-y)$).

In the long run, the development of algorithmics should give us "high-level" theorems, compared to which the few transformations we have now will look almost trivial. This is only possible through the growing development of higher-level concepts and corresponding notations. To get an idea of what I am dreaming of, compare the special product above with Cauchy's Integral Theorem, or with the Burnside Lemma.

1. The term "algebraic" is not used here in the technical modern sense (as in "algebraic data type"), but with the imprecise older meaning of "pertaining to Algebra" (as in "high-school Algebra"). The word "algebra" stems from the Arabic *al-jebr*, meaning "the [art of] recombining", originally used for bone setting. In the loose sense corresponding to that etymology, an identity like $\sin(x+y) = \sin x \cos y + \cos x \sin y$, in which the left-hand side is broken into constituents that are recombined to form the right-hand side, is algebraic.

The reader should carefully distinguish between

(i) the conviction—if not belief—that it is possible to create a discipline of "Algorithmics" that can be practised in the same style as Mathematics; in particular, by creating algorithmic derivations, using algorithmic expressions, with the same flavour as mathematical derivations and expressions;
(ii) the general framework around which the current investigations are built; namely a synthesis of an "algebraic" approach to data and to transformations (of data);
(iii) the concepts selected as worthy of a special notation in the language; and
(iv) the concrete notations and notational conventions chosen.

The program of research implied in (i) is closely related to the paradigm of "Transformational Programming"; see further Section 2. It is becoming increasingly clear (at least to me; I do not claim credit for the re-invention of the wheel) that a nice algebraic structure is a prerequisite for obtaining interesting results. Otherwise, no general laws can be stated, and so each step has to be proved afresh. (In fact, this is a truism, for what is an algebraic structure but a domain with operations, such that some general laws can be formulated.) This is also a major thought underlying the work on an "algebra of programs" of BACKUS[1]. A difference with the approach described here can be found in his motivation to overcome the "von Neumann bottleneck", resulting in a determined attempt to eschew variables for values (data, objects) even in their conventional mathematical roles, generally not considered harmful. More important is that Backus's "FP" framework is restricted to function *schemata*, and has (currently?) no place for an integrated algebraic view on data. (The approach described by GUTTAG, HORNING and WILLIAMS[12] allows algebraic specifications of data types but has more the nature of grafting them on FP than of integration.) It is clear, however, that the results obtained in his approach are valuable for the approach taken here, and that the correspondence merits further study. Integration of the data algebra with the algebra of operations on data can be found in the work by VON HENKE[13]. The emphasis there is on concepts; no attention is paid to notation.

The concepts and notations used here have grown out of my attempts to use the notations suggested by BIRD[4]. In trying to develop some small examples, I was struck by the similarity of many of the laws formulated in [4] (and some more I had to invent myself). Investigating this intriguing phenomenon, I discovered the higher-level algebraic framework underlying various similar laws. This incited me to introduce modifications to the notation, aimed at exhibiting similarities in the laws. These modifications have gone through various stages; for example, the symbols for sequence concatenation and set union were initially chosen to be similar; now they have been made identical.

The specific notational conventions, of all ideas presented here, should be given the least weight. This is not to say that I feel that good conventions are of secondary importance. It is obvious, however, that much work has still to be done to strike the right balance between readability, terseness, and

dependability (freedom of surprises). Only through the use in actual algorithmic developments, by a variety of people, can progress be made.

Two examples are included. They were chosen as being the first two not completely trivial problems that I tried to do in the present framework.

1. Mathematics for shortcuts in computation

In the Introduction, it was claimed that to ascertain the validity of a more efficient way of doing computations is to perform mathematics. This is still true if the reasoning is informal: the important thing is that it *could* be formalized. A beautiful example is the feat ascribed to Gauss as a young schoolboy. Asked to compute the sum of an arithmetic progression, he astounded his teacher by turning in the correct answer while the other pupils were still labouring on their first additions. We cannot, of course, know with certainty (if the story is true at all) what his reasoning was. But a plausible possibility is the following. Assume, for concreteness, that the task was to sum the first one-hundred terms of the arithmetic progression 534776, $534776+6207 = 540983$, $540983+6207 = 547190$, $\cdots$. Think of all those numbers, written in a column, and the same numbers in a second column, but this time in reverse order. So the first number in the second column is the number on the last line of the first column, which is $534776+99\times6207 = 1149269$. Next, add the numbers horizontally, giving a third column of one-hundred numbers.

$$
\begin{array}{rcrcr}
534776 & + & 1149269 & = & 1684045 \\
540983 & + & 1143062 & = & 1684045 \\
547190 & + & 1136855 & = & 1684045 \\
\vdots & & \vdots & & \vdots \\
1136855 & + & 547190 & = & 1684045 \\
1143062 & + & 540983 & = & 1684045 \\
1149269 & + & 534776 & = & 1684045 \\
\hline
S & + & S & = & 168404500
\end{array}
$$

FIGURE 1. Reconstruction of young Gauss's mathematical reasoning

Now we see a phenomenon that is not hard to explain. If we go down by one line, the number in the first column will increase by 6207. The number in the second column will *decrease* by the same amount. The sum of the two numbers on each line will, therefore, remain constant. So the third column will consist of 100 copies of the same number, namely $534776+1149269 = 1684045$. Now, call the sum of the numbers of the first column S. (This is the number to be determined.) The second column must have the same sum, for it contains the same numbers. The sum of the numbers in the third column is then $2S$. This sum is easy to compute: it equals $100\times1684045 = 168404500$. So $S = \frac{1}{2}\cdot168404500 = 84202250$. This "reconstruction" is rendered schematically in figure 1. It is noteworthy that the proof involves an intermediate construction that, if actually performed, would double the effort. The method is

easily generalized: if a is the first term of the progression, b is the increment and n is the number of terms to be added, we find $a+(n-1)b$ for the last term, and so $S = \frac{1}{2}n\{2a+(n-1)b\}$. The use of variables does not make the reasoning any less informal, of course.

Now, this was just an example, but substantial parts of mathematics consist of showing that two different construction methods will (or would) give the same result. Often one of the two is the original formulation of a problem to be solved, and the other one gives a construction that is much easier to perform.

It is also interesting to dwell for some time on the question of when we consider a mathematical problem solved. In mathematics we make no sharp distinction between the problem space and the solution space: both "problems" and "solutions" may have the form of construction methods. To call an answer a "solution" requires in the first place that it have the form either of a construction method, or of a *problem* for which we have, in our mathematical repertoire, a standard method for solving it. This requirement is not sufficient. For example, a mathematician will respond to the problem of determining the larger root of $x^2-2x-4 = 0$ by answering: $1+\sqrt{5}$, and consider the problem to be thereby solved. But what is the meaning of "$\sqrt{5}$" but: "the larger root of $x^2-5 = 0$"? So the problem is "solved" by reducing it to another problem. It is true that we have methods to approximate $\sqrt{5}$ numerically—for most purposes the best one is the Newton-Raphson method—but such methods will serve *equally well* to approximate the larger root of $x^2-2x-4 = 0$. Apparently, "to solve" does not simply mean: "to reduce to a case that we know how to handle". If that were the meaning, any quadratic equation would be its own solution. Out of the possibly many candidates for being solutions according to this requirement, mathematicians select one that allows a concise, elegant, formulation. We shall return to this issue in a discussion of mathematical notation, in Section 3.

2. Transformational Programming

The first published method for proving program correctness with mathematical rigour is that of Floyd [10]. Essentially the same method was suggested earlier by Naur [21]. Better known is the (semantically related) axiomatic approach of Hoare [14]. A technical objection to these methods is that they require the formulation of "intermediate assertions", i.e., predicates whose domain is the state space of an abstract machine; in more complicated cases, these predicates may grow into veritable algorithms themselves, and the conventional notations from predicate logic do not suffice to write them down. What makes program proving especially unsatisfactory is the following. The *activity* of programming, even in its present undisciplined form, already implicitly contains the essential ingredients for the construction of a correctness proof. These ingredients are present in the programmer's mind while developing the program. For example, a programmer may be heard muttering: "R must be at least 1 here, otherwise this code would not be reached. So I can omit this test and ...". None of this, however, is recorded.

Program proving requires now that a unique implicit correctness proof be made explicit *after the fact*. But such a reconstruction is in general much harder than to invent some proof in the first place. Also, it would be uneconomic to attempt to prove the correctness of a given program without verifying first that it handles several test cases successfully. But it is unrealistic to assume that programmers would go—unless forced—through the effort of proving apparently "working" programs correct.

This objection does not apply to the constructive approach advocated by DIJKSTRA [8], [9] and WIRTH [27], [28]. (The technical objection mentioned, however, does.) Here, the construction of the program is a *result* of the construction of the proof. Typical to the practical use of this approach, however, is that the program-under-construction is a hybrid, in which algorithmic notations are mixed with parts that are specified in natural language. For example, if we look over the shoulder of a programmer using this method of "stepwise refinement" or "top-down programming", we might see first:

"ensure enough room for T in *curbuf*"

in one stage of development, and in the next stage

while "not enough room for T in *curbuf*" **do**
  "ensure *nxtbuf* $\neq$ **nil**";
  curbuf, *nxtbuf* := *nxtbuf*, *nxtbuf*·*succ*
endwhile.

Although a big leap forward, the imprecision of the way the undeveloped parts are specified is unsatisfactory. In the example, it is probably the case that the task to "ensure enough room for T in *curbuf*" can be solved by emptying *curbuf*, and the task to "ensure *nxtbuf* $\neq$ **nil**" by the assignment *nxtbuf* := *curbuf*. But this would, in all likelihood, be incorrect, because of certain invariants to be maintained. It is, in principle, possible to attain the desired degree of precision, but the method itself does not incite the programmer to do so.

The same problem is not present in the method of "Transformational Programming"—at least, in its ideal form. In its essence, Transformational Programming is simple: start with an evidently correct—but possibly hopelessly inefficient—program, and bring this into an acceptable form by a sequence of "correctness-preserving" transformations. In contrast to mathematics, where the symmetrical relation "$=$", i.e., "is equal to", plays a central role, the central relation here is the asymmetric "may be replaced by",[1] denoted by "$\Rightarrow$". But at all stages, one has a correct program, with a precisely defined meaning. This way of manipulating a sequence of symbols

1. A simple example of this asymmetry is in the development of the task T = "Given a prime number p, find a natural number n such that n^2+n+p is composite". The development step that comes to mind (for a programmer) is to replace T by T' = "Find *the smallest* such natural number". A mathematician would probably replace the task by T'' = "Take $n = p$". Then $T \Rightarrow T'$ and $T \Rightarrow T''$. But T' and T'' are not interchangeable; for example, if $p = 2$, then T' finds $n = 1$, and in fact, they do not produce the same value of n for any value of p.

brings us closer to the ideal of "Algorithmics" aimed at. This is expressed in the following quote from a paper by BIRD[3], describing a new technique of program transformation: "The manipulations described in the present paper mirror very closely the style of derivation of mathematical formulas." There are several impediments to the application of this method. In the first place, the more usual algorithmic notations in programming languages suffer from verbosity. This makes manipulating an algorithmic description a cumbersome and tiring process. To quote [3] again: "As the length of the derivations testify, we still lack a convenient shorthand with which to describe programs." Furthermore, most programming languages have unnecessarily baroque semantics. In general, transformations are applicable only under certain conditions; checking these applicability conditions is all too often far from simple. The asymmetry of "$\Rightarrow$" makes these transformations also less general than is usual in mathematics. The requirement that the initial form be a program already (and "evidently correct", at that), is not always trivial to satisfy. In this respect, the method is a step backwards, compared to Dijkstra's and Wirth's approach. Finally, there is a very important issue: which are the correctness-preserving transformations? Can we give a "catalogue" of transformations? Before going deeper into that question, it is instructive to give an example.

Take the following problem. We want to find the oldest inhabitant of the Netherlands (disregarding the problem of there being two or more such creatures). The data needed to find this out are kept by the Dutch municipalities. Every inhabitant is registered at exactly one municipality. It is (theoretically) possible to lump all municipal registrations together into one gigantic data base, and then to scan this data base for the oldest person registered, as expressed in figure 2a in "pidgin ALGOL".

```
input dm, mr;
gdb := ∅;
for m ∈ dm do
   gdb := gdb ∪ mr[m]
endfor;
aoi := −∞;
for i ∈ gdb do
   if i·age > aoi then
      oi, aoi := i, i·age
   endif
endfor;
output oi.
```

FIGURE 2a. Program A for determining the oldest inhabitant

A different possibility is to determine the oldest inhabitant for each municipality first. The oldest person in the set of local Methuselahs thus obtained is the person sought. This is expressed in figure 2b.

Replacing (possibly within another program) program A by program B is then a transformation. Were there no inhabitants of the Netherlands, both

```
input dm, mr;
slm := ∅;
for m ∈ dm do
    alm := −∞;
    for i ∈ mr[m] do
        if i·age > alm then
            lm, alm := i, i·age
        endif
    endfor;
    slm := slm ∪ {lm}
endfor;
aoi := −∞;
for i ∈ slm do
    if i·age > aoi then
        oi, aoi := i, i·age
    endif
endfor;
output oi.
```

FIGURE 2b. Program B for determining the oldest inhabitant

programs would have an undefined result. This is generally not seen as affecting the applicability of the transformation $A \Rightarrow B$. But if—assuming at least one inhabitant in the country—some municipality had no registered inhabitants, then program A would have a defined result, whereas the outcome of B might be undefined. (The problem is that in the line "$slm := slm \cup \{lm\}$" the variable lm has no defined value if the empty municipality is the first one to be selected by "**for** $m \in dm$ **do**".) So the transformation $A \Rightarrow B$ has the following applicability condition:

$$(\forall m \in dm\colon mr[m] = \emptyset) \vee (\forall m \in dm\colon mr[m] \neq \emptyset).$$

We happen to know that for the given application this condition is satisfied, but it is easy to think of applications of this transformation where it is less obvious and has to be checked. Overlooking such conditions that are only exceptionally not satisfied is a typical source of programming errors. Note that a human interpreter of the original descriptions in natural language would almost certainly handle exceptional cases reasonably.

How large must a catalogue of transformations be before it is reasonable to expect it to contain this transformation? Obviously, unmanageably large. It is possible to have a manageable catalogue, and to require proofs of other transformations that are not in the catalogue. But how do you prove such a transformation? Hopefully, again with transformations, otherwise the practitioner of Transformational Programming needs two proof techniques instead of one. But what transformations will gradually transform A into B?

As another example, consider young Gauss's "transformation". This may be expressed as

```
input a, b, n;                                  input a, b, n;
sum, t := 0, a;                                 output (n / 2) × (2×a+(n−1)×b)
for i from 1 to n do                    ⇒
   sum, t := sum+t, t+b
endfor;
output sum
```

Again, this is an unlikely transformation to be catalogued. Now compare this to the mathematical derivation:

$$\sum_{i=1}^{n}\{a+(i-1)b\} = \tfrac{1}{2}\left[\sum_{i=1}^{n}\{a+(i-1)b\} + \sum_{i=1}^{n}\{a+(i-1)b\}\right] =$$

$$\tfrac{1}{2}\left[\sum_{i=1}^{n}\{a+(i-1)b\} + \sum_{i=1}^{n}\{a+(n-i)b\}\right] = \tfrac{1}{2}\sum_{i=1}^{n}\{2a+(n-1)b\} =$$

$$\tfrac{1}{2}n\{2a+(n-1)b\}.$$

It is usual in presenting such derivations to omit obvious intermediate steps, and this one is no exception. For example, the first step has the pattern $S = \frac{1}{2}(S+S)$; a complete derivation would have $S = 1S = (\frac{1}{2}\cdot 2)S = \frac{1}{2}(2S) = \frac{1}{2}(S+S)$. Nevertheless, the only step that possibly requires looking twice to check it is the substitution of $n+1-i$ for one of the two summation variables i.

In what follows, an attempt is made to sketch an "algorithmic language" to overcome the drawbacks mentioned. To give a taste of what will be presented there, here, in that language, is the "transformation" $A \Rightarrow B$ of the oldest-inhabitant problem:

$$\uparrow_{age}/+/mr * dm = \uparrow_{age}/(\uparrow_{age}/mr) * dm .$$

Comparing this with figure 2a and 2b should explain my complaint about the verbosity of algorithmic languages. And yet that pidgin is a terse language when compared to those mountains of human achievement, from FORTRAN to Ada.® Note also the reinstatement of the symmetric "=", which will be explained in Section 6.

The emphasis on the similarity with Mathematics creates a clear difference with much of the work in the area of Transformational Programming, such as that of the Munich CIP group (BAUER *et al.* [2]). In that work, the emphasis is on creating a tool for mechanical aid in, and the verification of, program development. The prerequisite of mechanical verifiability puts its stamp on a language. Note that the language of Mathematics has not been developed with any regard to mechanical verifiability; the only important factor has been the sustenance offered in reasoning and in manipulation of formulae. In this respect, the approach of, e.g., BIRD [3] is much more closely related, even if its framework is different. To quote that paper once more: "[...] we did not start

out, as no mathematician ever does, with the preconception that such derivations should be described with a view to immediate mechanization; such a view would severely limit the many ways in which an algorithm can be simplified and polished." The main point is, perhaps, that in my view the language should be "open", whereas mechanical verifiability requires a closed and frozen language. To prevent misunderstanding of my position, I want to stress that I sympathize with the thesis that systems for the complete verification of a development are extremely valuable, and that research and development in that area should be vigorously pursued. I hope—and, in more optimistic moments, expect—that the different line of approach followed here will, in the long run, contribute to better methods for program design and development, and to better systems for mechanical assistance in these tasks.

3. The role of notation in Mathematics

When Cardan breached his pledge of secrecy to Tartaglia and published the first general method for solving cubic equations in his *Ars Magna* (1545), he described the solution of the case $x^3+px = q$ as follows [my translation]:

> RULE
> Raise the third part of the coefficient of the unknown to the cube, to which you add the square of half the coefficient of the equation, & take the root of the sum, namely the square one, and this you will copy, and to one [copy] you add the half of the coefficient that you have just multiplied by itself, from another [copy] you subtract the same half, and you will have the Binomium with its Apotome, next, when the cube root of the Apotome is subtracted from the cube root of its Binomium, the remainder that is left from this, is the determined value of the unknown.

This description strikes us as clumsy, but at the time, no better method was available. This "clumsiness" stood directly in the way of mathematical progress. Take, in contrast, a description of the same solution in present-day notation:

Solution of the equation $x^3+px = q$.

Let $c = \sqrt{d}$, *where* $d = \left(\frac{p}{3}\right)^3+\left(\frac{q}{2}\right)^2$, *and let* $b = c+\frac{q}{2}$ *and* $a = c-\frac{q}{2}$.

Then $x = \sqrt[3]{b}-\sqrt[3]{a}$ *is a root of the equation.*

What are the advantages of this notation? Obviously, it allows for a more concise description. Also, in Cardan's description, there might be some doubt whether "the half of the coefficient" itself, or its square, has to be added to and subtracted from the copies. In present-day notation, there is (in this case) no room for this doubt, and in general, parentheses will disambiguate (if necessary) anything. Both of these advantages, however, are insignificant compared to what I see as the major advantage of the "algebraic" notation used now, namely that it is possible to manipulate the formula for x algebraically.

So we see readily that

$$\begin{aligned} x^3 &= b - 3\sqrt[3]{b^2 a} + 3\sqrt[3]{ba^2} - a \\ &= (b-a) - 3\sqrt[3]{ba}\,(\sqrt[3]{b} - \sqrt[3]{a}) \\ &= q - (3\sqrt[3]{ba})\,x, \end{aligned}$$

and since

$$ba = c^2 - \left(\frac{q}{2}\right)^2 = \left(\frac{p}{3}\right)^3,$$

we see that indeed $x^3 + px = q$. No more than high-school mathematics was needed to verify the solution. A similar verification is impossible for the formulation in natural language. If, at the time, our notations had been available, then the solution of the cubic equation would not have had such a romantic history. A disadvantage of modern notation is its *suggestion* of abstruseness, of being an esoteric code. Undeniably, people can only profit substantially from the major advantage mentioned above if they not only know the meaning of the diverse squiggles, but are intimately familiar with them, which takes time and practice. I want to emphasize, however, that a description in natural language, as the one given by Cardan, is utter gibberish too to the mathematically uneducated reader. This point would have been obvious, had I chosen to use the "most literal" translation of the words in the Latin original, instead of present-day terminology. The rule would then have started: "Bring the third part of the number of things to the cube, ...".

In Section 1 I stated that a requirement for "solutions" is that their formulation be "elegant". This issue is connected to that of notation. It is matter of context, taste, conventions and tacit agreement between mathematicians, what constitutes "elegance". It is hard for us to understand why the ancient Egyptians were so keen on expressing fractions in terms of quantities $\frac{1}{n}$, as in

$$\frac{41}{45} = \frac{1}{2} + \frac{1}{5} + \frac{1}{9} + \frac{1}{10} = \frac{1}{2} + \frac{1}{3} + \frac{1}{20} + \frac{1}{36}.$$

For some reason, forms like $\frac{41}{45}$ did not belong to their solution space, but quantities like $\frac{1}{9}$ did. If we were to agree that, say, $Q(p, q, r, s)$, denoting the largest root of the equation $x^5 + px^3 + qx^2 + rx + s = 0$, belongs to our solution space, then suddenly the general quintic equation becomes solvable "algebraically". There is a reason for mathematicians not to take this way out. The squiggle approach is helpful only if mathematical practitioners can acquire sufficient familiarity with the squiggles, which imposes a limit on their number. Given this limitation, some criterion must determine which concepts are the winners in the contention for a notational embodiment. Two aspects determine the viability of a proposed notation. One is the *importance* of the concept: is it just applicable in some particular context, or does it come up again and again? The other is the amenability to *algebraic manipulation*: are there simple powerful algebraic identities expressible in terms of the notation considered? The Q-notation suggested above will be found lacking in both respects.

4. Notational conventions for functions and operations

A program operates on input and produces output. Whether that input be a "value", a data base, or a stream of requests, say, is immaterial to this abstract viewpoint. Similarly, it is immaterial if the output consists of values, modifications to a data base, or a stream of responses. In the usual approaches to programming languages, the distinction is, unfortunately, paramount in the concrete embodiment of the program. This obscures the deeper similarities in possible program development steps. So the first thing required is a uniform notation, reflecting a unified conceptual framework. The notation used here is that of a "function" operating on an "object". The result is a style that may be called "functional". However, I feel that the cherished distinction between a functional (or "applicative") style of programming, and a procedural (or "imperative") one, is not as deep as supporters/opponents of one or the other style would make it appear. A much deeper difference is the distinction between viewing an algorithmic expression, be it denoted as a function definition or as a while program, as an *operational* prescription for an automaton, or as an *abstract* specification determining a relationship between input and output. The price paid for taking the latter viewpoint is that this abstraction may make it hard to express some transformations that derive their relevance from performance characteristics of certain types of architecture. Such a transformation makes sense only if we commit ourselves to a decision on how the abstract specification is mapped to a process on a machine—although in due time several natural "canonical" mappings for various architectures may emerge. Moreover, if the inverse mapping is not defined, a low-level transformation may lack a high-level counterpart. (This problem occurs in high-level programming languages as well: try to express in Pascal, say, the low-level optimization that the storage for a global array variable that will no longer be referenced can be used for other purposes.) Since computing resources will always remain scarce—relative to our unsatiable need for processing—this is not a minor inconvenience. Some consolation can be found in the thought that many of these transformations are well understood and can be automated relatively well (e.g., recursion elimination; tabulation techniques; low-level data structure choice), possibly sustained by "implementation hints" added to the program text.

The main ingredients of our language will be "objects", (monadic, or unary) "functions", and (dyadic, or binary) "operations". Functions always take an object as argument, and return an object. Operations are written in infix notation, and may take an object, a function or an operation as left operand and an object as right operand. They return an object. Function application is (notationally) not treated as an operation (although, from a mathematical point of view, it is one, of course). It is simply denoted by juxtaposition, usually leaving some white space for legibility or to delineate the boundary between the lexical units involved. So, if f is a function and x is an object, $f\,x$ stands for the application of f to x. If g is then applied to $f\,x$, this may be denoted by $g\,f\,x$. Function composition, usually written in mathematics in the form $g \circ f$, is *also* denoted by juxtaposition, without intervening operation.

This makes expressions such as $h\,g\,f$ and $g\,f\,x$ ambiguous. But semantically, there is no ambiguity: the expressions specify the same, since $(h\,g)\,f$ denotes the same function as $h\,(g\,f)$, and $(g\,f)\,x$ the same object as $g\,(f\,x)$. (The reader should note that these identities are algebraic, and about the simplest ones possible.) In fact, the wish to omit as many parentheses as possible without depending on priority rules motivated this unconventional convention. In particular, it removes the somewhat annoying disparity between an identity expressed on the object level, as in

$$f\,(g(x)) = g'(f\,(x)),$$

and its expression as functional identity, as in

$$f \circ g = g' \circ f.$$

A drawback is that this convention does not indicate how to denote the application of a functional (higher-order function) to a function argument; in the general case, a function may be so generic that it might both be composed with and be applied to another function. An example is the identity function; in that particular case, the distinction is semantically unimportant, but for other functions it is not. So some operation will be needed to denote function application in the general case. (Actually, it turns out possible to denote function application with the operations provided in the sequel, but only in a clumsy way.)

If $\times$ is an operation, then $x \times y$ denotes the application of $\times$ to x and y. In general, parentheses are needed to distinguish, e.g., $f(x \times y)$ from $(f\,x) \times y$. The interpretation of $f\,x \times y$ in the absence of parentheses is $f(x \times y)$. In a formula $x \times y \times z$, the absence of parentheses implies, likewise, the interpretation $x \times (y \times z)$. This convention is similar to the right-to-left parsing convention of APL.

Note. In derivations, chains may occur like $e_1 = e_2 = \cdots$. The connective signs ("$=$" etc.) in these chains are meta-signs, and are not to be confused with operations (in particular, the *operation* $=$, which takes two operands and delivers a truth value). They will always give precedence to the operations in the expressions e_i.

A further reduction of the number of parentheses is made possible by the following convention. An expression of the form "$\alpha\,;\,\beta$" stands for "$(\alpha)\,\beta$". The—purely syntactic—operator ";" takes lower precedence than the semantic operations. If several ";"s occur, they group from left to right: "$\alpha\,;\,\beta\,;\,\gamma$" stands for "$((\alpha)\,\beta)\,\gamma$".

An important convention is the following: If $\times$ is some operation, and x is an acceptable left operand for $\times$, then the notation "$x\times$" stands for the *function* $\lambda y\colon x \times y$. Note that $x \times y$ is now syntactically, but not semantically, ambiguous, since $(x\times)\,y$ denotes the same object as $x \times y$. In the notation $f\,x\times$ the meaning is always $f(x\times)$, so it denotes a functional composition. If the meaning $(f\,x)\times$ is intended, parentheses are required (or, equivalently, the notation $f\,x\,;\times$ can be used). This convention makes it also possible to define the meaning of an operation $\times$ in the following form:

Let x be ... . Then $x\times$ denotes the function F_x .

The meaning of $x\times y$ is then that of $F_x\ y$.

Now, for example, $1+\sqrt{}$ is defined: its meaning is $1+;\sqrt{} = \lambda y\colon 1+y;\circ\sqrt{} = \lambda x\colon 1+\sqrt{x}$.

Finally, if $\times$ is an operation that takes two objects as operands, and f and g are functions, then $f\times g$ stands for the function $\lambda x\colon (fx;\times g\ x)$.

The aim of these conventions is only to increase the usability of the formal language. The proof is therefore in the practical use. It will take time, and the experience of a variety of practitioners of Algorithmics, to find the most helpful notational conventions. Note that the current mathematical practice of using the sign "+" for addition and juxtaposition for multiplication, and to give multiplication precedence, has taken its time to become universally accepted—after the general idea of using an algebraic notation was already commonly accepted. Also, if the language is as open as the language of Mathematics, it is possible to adopt other conventions locally when this is more helpful in dealing with the problem at hand.

To define functions and operations concisely, we use, in addition to lambda forms, the convention of BURSTALL and DARLINGTON [6]. For example, the following lines define the Fibonacci function:

$$Fib\ 0 \Leftarrow 0;$$

$$Fib\ 1 \Leftarrow 1;$$

$$Fib\ n+2 \Leftarrow Fib\ n;+Fib\ n+1\,.$$

The variables on the left-hand side of " $\Leftarrow$ " are dummy variables for which values are to be substituted such that the left-hand side matches the actual function application; then the right-hand side, after applying the same substitutions, is equal to the function application and may replace it in a formula. This step is known as "Unfold"; the reverse operation as "Fold". A canonical evaluation can be defined by systematically unfolding, thus providing an operational semantics. BURSTALL and DARLINGTON show that an amazingly large number of transformations can be expressed as a sequence of Unfold/Fold steps. As long as $\Leftarrow$ is interpreted as equality, this is generally safe. If $\Leftarrow$ is interpreted in terms of the canonical evaluation, then a Fold step may introduce non-termination where it was not present.

5. STRUCTURES

In giving an algorithmic description, we are generally not only concerned with elementary values, like numbers and characters. These are combined into larger objects with a certain structure. For example, in some application we may want to compute on polynomials, represented as a sequence of coefficients, or with a file of debtors. The usual algorithmic approach to such aggregate structures has grown from the aim of obtaining an efficient mapping to the architecture of concrete computational automata. For the purposes of Algorithmics, we need a more algebraic approach. The domain of data on which a program operates usually has some algebraic structure. This fact

underlies the work in the field of algebraic data types. However, since the *motivation* there is not to obtain a simple algebra, but to achieve representation abstraction, the types as specified by way of example in the papers in this field are not usually algebraically (in the *al-jebr* sense) manageable. If they are, as for example the type of natural numbers, or the type of McCarthy's *S*-expressions, the structure of algorithms operating on objects of these types tend to reflect the structure of the objects. In algebraic terms, the function relating the input to the output is a homomorphism. This observation underlies the work by VON HENKE[13]. (The work by JACKSON[15]—best known outside of *Academia*—can be viewed as based on the same idea, although the term "homomorphism" is not used there.)

Let us start with algebraic structures that are about as simple as possible. Using the notation of MCCARTHY[17], we have

$$S_D = D \oplus S_D \times S_D .$$

This defines a domain of "*D*-structures", each of which is either an element of the (given) domain *D* (e.g., numbers, or sequences of characters), or is composed of two other *D*-structures. To practitioners of computer science, it is virtually impossible to think of these structures, McCarthy's "*S*-expressions", without a mental picture of an implementation with car and cdr fields from which arrows emerge. To mathematicians, however, this domain is simply a free groupoid, about the poorest (i.e., in algebraic laws) possible algebra, and computer-scientists will have a hard time explaining to them how arrows enter (or emerge from) their mental picture.

We need some notation for constructing such structures. We construct a *D*-structure by using the function "^" and the operation "+". If x is an element of D, then $\hat{}x$ will stand for the corresponding element of S_D. The monadic function $\hat{}$ is, of course, an injection. It is a semantically rather uninteresting function, and it could be left unwritten in many cases without ambiguity. As a compromise, the application of $\hat{}$ to x is written as $\hat{x}$ if this is typographically reasonable. If s and t are *D*-structures, then $s+t$ denotes the *D*-structure composed of s and t. The set S_D consists then of all structures that can be built from D by a finite number of applications of $\hat{}$ and $+$. (It is also useful to allow an infinite number of applications; this possibility will be ignored here to keep the treatment simple.)

The diligent reader will have noticed an important difference between the structures defined now, and the *S*-expressions as used for LISP. The value **nil** is missing. We can introduce it by writing (using "0" instead of "**nil**"):

$$S_D = D \oplus \{0\} \oplus S_D \times S_D .$$

Algebraically, however, this makes little difference; the domain obtained is isomorphic with $S_{D \oplus \{0\}}$, i.e., the one obtained by the previous construction if D is first augmented with an element 0. It becomes more interesting if we impose an algebraic law: $s+0 = 0+s = s$. This gives about the poorest-but-one possible algebra. Now we have a more dramatic deviation from the *S*-expressions, for it is certainly not the case that, e.g., cons(*s*, **nil**) = *s*.

The previous law is known as the *identity* law, and an element 0 satisfying this law is called an "identity (element)". Note that an identity can always be *added*, but that there is at most one identity in a groupoid.

We can go further and consider structures on which other algebraic laws are imposed. Of particular interest are the laws of *associativity*: $s+(t+u) = (s+t)+u$; of *commutativity*: $s+t = t+s$; and finally of *idempotency*: $s+s = s$. The interesting thing now is that the structures obtained correspond to familiar data structures: we get, successively, *sequences*, *bags*,[1] and *sets*. For sets, $\hat{}$ is the function $\lambda x: \{x\}$ and $+$ is the set union $\cup$. The identity law gives us the empty sequence, bag or set. This relationship between familiar algebraic laws and familiar data structures has been pointed out by BOOM[5]. Sequences correspond to what are known in algebra as monoids (or semigroups if there is no identity).

The usual way of characterizing sequences algebraically uses an operation "append (or prepend) an element". The choice between using "append" and "prepend" as the primitive operation introduces an asymmetry. The introduction of sequences by imposing associativity is quite symmetric. This way of introduction gives a uniform approach, exhibiting the essential and deep similarity between binary labelled trees (the *S*-expressions), sequences, bags and sets. This can be used to express laws that apply to all these kinds of structures. To stress the similarity, $+$ will be used in all cases; a disadvantage is that the type has then (at least in some cases) to be clear from the context. The notation S_D will likewise be used for all domains of such structures, and not be reserved for the free *S*-expressions.

To prove laws, we can use the following lemma:

INDUCTION LEMMA. *Let f and g be two functions defined on S_D, satisfying, for all $x \in D$ and s and $t \in S_D$:*
(i) $f0 = g0$,
(ii) $f\hat{x} = g\hat{x}$, *and*
(iii) $fs+t = gs+t$, *using the induction hypothesis that* $fs = gs$ *and* $ft = gt$.
Then $f = g$.

PROOF. By induction on the complexity of the function argument.

If S_D has no identity, then part (i) can of course be omitted. It is sometimes easier, in particular for sequences, to replace (ii) and (iii) together by $fs+\hat{x} = gs+\hat{x}$, which gives the traditional induction on the length. The advantage of the lemma as stated here is that it allows many laws to be proved independently of the algebraic richness of S_D.

To express interesting laws we first need some general operations, that also play an important role in Backus's FP. The notation used here for "applied-to-all" has been taken from [4]; the APL notation is used for "inserted-in".

1. Bags (or *multi-sets*), underrepresented in mathematics, are ubiquitous in computer science. They differ from sequences in that the elements have no order, and from sets in that an element can occur more than once.

Applied-to-all. Let f be a function in $D_1 \to D_2$. Then $f*$ stands for the function in $S_{D_1} \to S_{D_2}$ satisfying
(i) $f*0 = 0$,
(ii) $f*\hat{x} = \hat{}fx$, and
(iii) $f*s+t = f*s; +f*t$.

So f is applied to each "member" (elementary component) of its argument, and the result is a structure of the function values obtained. For example, if s is the set of numbers 0 through 9, then $1+*s$ is the set 1 through 10. For $f*$ to be well defined, it is required that $+$ on S_{D_2} have at least the same algebraic richness as its counterpart on S_{D_1}: if $+$ on S_{D_1} is associative, then so is $+$ on S_{D_2}, and so on. If S_{D_1} has no identity, we can simply omit part (i) from the definition. A similar remark can be made in most cases in the sequel: the laws are presented for structures with identity, but can easily be amended to cover identity-less structures.

Inserted-in. Let $\times$ be an operation in $D \times D \to D$. Then $\times/$ stands for the function in $S_D \to D$ satisfying
(i) if $\times$ has an identity e (so that $e \times x = x \times e = x$), then $\times/0 = e$,
(ii) $\times/\hat{x} = x$, and
(iii) $\times/s+t = \times/s; \times\times/t$.

So if $\times$ stands for the conventional multiplication operation, $\prod_{x \in S} x$ is a more familiar notation for $\times/s$. However, inserting an operator $\times$ in a structure s is only meaningful if $\times$ has at least the same algebraic richness as the operation $+$ used to construct the structure. This means that if $\times$ is multiplication, then the notation $\times/s$ is not allowed if s is a set, for (in general) $x \times x \neq x$. Otherwise, we would obtain contradictions like $2 = \times/\hat{2} = \times/\hat{2}+\hat{2} = \times/\hat{2}; \times\times/\hat{2} = 2 \times 2 = 4$. (Alternatively, we could define the insertion as an indeterminate expression, depending on the choice of representatives from the congruence classes induced by the laws of $+$.)

The classes of functions $f*$ and $\times/$ are special cases of the homomorphisms definable on S_D. By combining them in the form $\times/f*$, all such homomorphisms can be expressed. This can be stated in the form of another lemma:

HOMOMORPHISM LEMMA. *Let the function $g \in S_D \to D'$ be a homomorphism, i.e., let there exist a function $f \in D \to D'$ and an operation $\times \in D' \times D' \to D'$ with identity $\times/0$, satisfying, for all $x \in D$ and s and $t \in S_D$:*
(i) $g0 = \times/0$,
(ii) $g\hat{x} = fx$,
(iii) $gs+t = gs; \times gt$.
Then $g = \times/f*$.

PROOF. By the induction lemma. For part (i), we have $g0 = \times/0 = \times/f*0$. For part (ii), $g\hat{x} = fx = \times/\hat{}fx = \times/f*\hat{x}$. For part (iii), by the induction hypothesis $gs = \times/f*s$ and $gt = \times/f*t$. Then $gs+t = gs; \times gt = \times/f*s; \times\times/f*t = \times/f*s+t$.

Note that this gives an algebraic formulation of the "Divide and Rule" paradigm. For part (iii) tells us that to rule a structure s that is not atomic (i.e., to compute $g\,s$), we can divide s in two parts, rule these, and combine the results appropriately.

The operations $*$ and $/$ give rise to three important new laws.

LAW 1. *Let* $f \in D_2 \to D_3$ *and* $g \in D_1 \to D_2$. *Then* $(f\,g)* = f* g*$.

LAW 2. *Let* $f \in D \to D'$, $\times \in D \times D \to D$ *and* $\times' \in D' \times D' \to D'$ *satisfy*
$f\,x \times y = f\,x\,;\, \times' f\,y$ *and* $f \times /\,0 = \times' /\,0$.
Then $f \times / = \times' / f*$.

LAW 3. *Let* $\times \in D \times D \to D$ *and let* $+$ *operate on* S_D.
Then $\times / + / = \times / \times /*$ *(where these functions operate on* S_{S_D}*).*

PROOF. The proof (by induction) of law 1 is straightforward. Law 2 is an application of the homomorphism lemma, by taking $f \times /$ for g and $\bar{\times}'$ for $\times$. Law 3 is an application of the same lemma, with $\times /$ for f and $\times / + /$ for g.

Each of these laws corresponds to a whole set of program transformations. Since the law $g*x + y = g*x\,;\, + g*y$ holds, and $g* + /0 = + /0$ (since 0 is the identity of $+$, we have $+/0 = 0$), we can apply law 2, with $g*$ for f and $+$ for both $\times$ and $\times'$, to obtain

COROLLARY. *Let* $g* \in S_D \to S_{D'}$. *Then* $g* + / = + / g**$.

The importance of the corollary is that it has no condition to be verified, in contrast to the complex applicability condition of the law from which it was derived.

This game can be continued on more complicated algebras. The simple cases dealt with above, however, already give rise to a surprisingly fruitful range of identities. For example, the identity mentioned in Section 2, which in functional form reads $\uparrow_{age} / + / mr* = \uparrow_{age} / (\uparrow_{age} / mr)*$, in which mr is used as a function, is derived as follows

$$\uparrow_{age} / + / mr* = \uparrow_{age} / \uparrow_{age} / * mr* \quad \text{(by law 3, using } \uparrow_{age} \text{ for } \times).sp - .1v$$
$$= \uparrow_{age} / (\uparrow_{age} / mr)* \quad \text{(by law 1).}$$

This identity applies then to trees, sequences, bags and sets. Indeed, the transformation $A \Rightarrow B$ is valid, irrespective of whether the inhabitants are registered in orderly ledgers, or in bags. It is possible that $\uparrow_{age} /$ is not meaningful on the structures considered, but then both sides of the identity are meaningless.

A particular type of structure is obtained by taking the point domain $\{\iota\}$, containing one single element ι. Assume $+$ is at least commutative, and define $1 = \hat{\iota}$. Then each member of $S_{\{\iota\}}$, except 0, can be written in the form $1 + \ldots + 1$. In this particular case, associativity implies commutativity, since the 1s are indistinguishable. (This is not true if we allow infinite structures.) If identity, associativity and commutativity are the only laws for $+$, so that, e.g., $1+1 \neq 1$, then $S_{\{\iota\}} = \mathbb{N}$, the natural numbers, and $+$ has the conven-

tional meaning of addition. If idempotency holds too, we obtain a set with two elements, 0 and 1, which will be identified with "false" and "true", respectively. The meaning of $+$ on this domain is that of $\vee$, the "logical or" operation.

6. Fictitious values

Since antiquity mathematicians have been confronted with equations that, although not inconsistent, were nevertheless "impossible". A simple example is the equation $s+8 = 5$. If a shepherd adds eight sheep to his flock, it is impossible that the result is that the flock contains five sheep. And yet, discovered the mathematicians, it is possible to practise an internally consistent mathematics with fictitious quantities such as "3 short". In this way the notion of "number" has been extended from natural to, successively, integral, rational, algebraic, real and complex numbers. Today we are so familiar with all this that it is hard to realize what triumph of intellect the invention must have been to denote "nothing", something "non-existent", with a symbol like "0". Why has mathematics gone the way of accepting "fictitious values" on an equal footing? The answer must be that for mathematical practice the simplicity of the algebraic laws prevailed over semantic doubts about the necessary extensions of the notion of "value". Nowadays, we feel no qualms in stating that the set of primes that are also squares is empty, rather than that such a set is "impossible". Only one century ago, this was not so easy. The well-known mathematician C.L. Dodgson—well-known for other than his mathematical writings—advocated that universal quantification over such an "impossible" set would stand for a contradiction. Nobody could have worded the arguments better than he, but nothing has stopped mathematics from going the way of algebraic simplicity, in spite of all "common sense", leading to the currently universally accepted interpretation, which is just the reverse. So now we have

$$(\forall x \in S: p(x)) \supset (\forall x \in S': p(x)) \text{ for all } p \text{ iff } S' \subset S .$$

The Carrollean definition would have required, instead of "iff $S' \subset S$", the much more complicated "iff $S = \emptyset \vee S' \neq \emptyset \wedge S' \subset S$". Yet it is important to realize that all this is a matter of convenience, and not of mathematical necessity. If, for example, we define $<$ between sets over an ordered domain by

$$S < T \text{ iff } \forall s \in S: \forall t \in T: s < t,$$

then under the present interpretation $<$ is not transitive, whereas it would have been so, had nineteenth-century "common sense" prevailed. So the advantages of the current convention are not unequivocal.

The problem that arises in the oldest-inhabitant problem treated in Section 2 if some municipality is without inhabitants, can be solved by introducing the fictitious value "Nobody". In more mathematical terms, the domain of inhabitants forms a semi-lattice (disregarding inhabitants of equal age), and, as is well known, it is always possible to add some bottom element to it. If we

denote the operation of the semi-lattice by "$\uparrow_{age}$", then the oldest inhabitant of a set s of inhabitants is given by $\uparrow_{age}/s$, and so this "Nobody" is $\uparrow_{age}/0$. If Nobody is next compared to somebody, somebody will be chosen, since $s\uparrow_{age}\uparrow_{age}/0 = s$. This explains why "$\Rightarrow$" could be replaced by "$=$". In general, if some operation $\times$ has no identity in its domain, we can extend the domain by adding $\times/0$ as its identity. The properties of $\times/0$ are completely determined by the relevant algebraic laws. In particular, we see that it is an identity of $\times$ from $x \times \times/0 = \times/\hat{x}; \times\times/0 = \times/\hat{x}+0 = \times/\hat{x} = x$. Such a fictitious value can drastically simplify an algorithmic description; for that reason, it is not uncommon to find the notation ∞ in algorithms described in "pidgin ALGOL". The important insight is that such a domain extension is, in general, consistent. Inconsistencies can arise through additional laws, or through interference between laws involving several operations in a domain. To give an example of the possible pitfalls, let the operation $\ll$ be defined by

$$x \ll y \Leftarrow x.$$

This operation is associative, since $(x \ll y) \ll z = x \ll (y \ll z)$. The function $\ll/$ selects the first element of a sequence (or the leftmost element of a tree). Now consider $\ll/0$, where 0 is the empty sequence. Then $\ll/0; \ll x = x$, since $\ll/0$ is the identity of $\ll$. But from the definition of $\ll$, we have $\ll/0; \ll x = \ll/0$. So $x = \ll/0$ for arbitrary x. The problem arises since the law $x \ll y = x$ has already assigned a value to a formula containing the newly introduced identity. In fact, each element is a so-called right-identity of $\ll$; if a semi-group contains both a left- and a right-identity, then it is well known that they must coincide. If, for algorithmic purposes, a fictitious element $\ll/0$ is desirable, we must choose between two possibilities to retain consistency: either restrict the law $x \ll y = x$ to $x \neq \ll/0$, or use $\ll/0$ as a right-identity only (in which case the law $\ll/s+t = \ll/s; \ll \ll/t$ requires, of course, the restriction $s \neq 0$). Which solution is best depends on the context.

For the applicability of the methods of "transformational programming" and especially of "programming by stepwise refinement", it is important that algorithmic descriptions allow a certain amount of "indeterminacy". We may then find descriptions like "Let x be an element of s". The correctness of the algorithm does not depend on the element chosen, and so permits arbitrary choice. This type of "arbitrariness" should not be confused with the intended chaotic arbitrariness of pseudo-random generators. It only indicates a freedom that is left in realizing the algorithm, and which can be used, e.g., to achieve a simplification through a judicious choice of x. Now what if $s = 0$, the empty structure? The usual approach is then that the meaning of "Let x be an element of s" is "undefined", an entity that is loved by semanticists but best avoided by programmers. Let us use the symbol $\square$ to denote an unspecified choice: the operation of making an arbitrary choice between two values. So $x \square y$ is a specification that is satisfied by any solution for x, but also by any solution for y. The expression $1 \square 2$ may yield 1, but may as well yield 2 (but not 3). The operation $\square$ is associative: $(x \square y) \square z$ is equivalent to $x \square (y \square z)$. It

is also commutative and idempotent. So $[]/s$ stands for an "arbitrary" choice from the structure s. Choosing from an empty structure can now be described with the formula $[]/0$. But no choice is possible, so what is the meaning of this formula? The answer is: "Nothing". A more learned answer is that $[]/0$ represents the unsatisfiable specification. In essence, the question is as unanswerable as the question what it means to take the square root of -1. The meaning of $[]/0$ is given by the algebraic laws it satisfies; beyond that, it has no inherent meaning, any more than ∞, $\sqrt{-1}$, $\sqrt{2}$, $\frac{1}{2}$ or, for that matter, -3 have one. So, in particular, its meaning is that it satisfies $x[][]/0 = x$. In words, if we may choose "freely" between x and Nothing, then we must choose x.

An important identity for $[]$ is

$$f x [] y = f x ; [] f y .$$

This corresponds to what is known in Formal Semantics as the "monotonicity" of f. We know then, from law 2 of Section 5, that $f[]/ = []/f*$. A prerequisite for general applicability of this law here, is, however, that the function be "strict", i.e., that the identity $f[]/0 = []/0$ be satisfied as well. (In Formal Semantics, a function f is called "(error-)strict" or "bottom preserving" if $f(x)$ is "undefined" (or "the error value") whenever x is. The pseudo-value $[]/0$ can serve here, more or less, as a denotation of an "error value".) Many other identities require that the functions involved be strict. That a function is indeed strict will sometimes follow from its definition. In other cases, such as for the constant function $0\ll$, it does not; if strictness is not necessary, we have to specify what we want. It is, of course, possible to take strictness of functions as an immutable characteristic of the framework. But this is undesirable. In particular, if $[]/0$ is an identity of the operation $[]$, this gives simpler algebraic laws. Since then $x[][]/0 = x$, the function $x[]$ cannot be strict for satisfiable x, and so the identity $x[][]/s = []/x[]*s$ requires the restriction $s \neq 0$. A reasonable convention appears to be that a function f is only strict if the algebraic identities assign no other meaning to $f[]/0$, or, of course, if strictness is explicitly specified. Then $\hat{}$, $+$, and all functions of the forms $f*$ and $\times/$, are strict. Moreover, $=$ must be strict, to prevent pathological paradoxes as would be created by $fx \Leftarrow$ if $fx = []/0$ then x else $[]/0$.

We can now define the asymmetric relation $\Rightarrow$ in terms of $=$ and $[]$, for $p \Rightarrow q$ has the same meaning as $p = p[]q$. A consequence is that $p \Rightarrow []/0$ for each p; for that reason programmers are well advised not to interpret "$\Rightarrow$" too literally as "may be replaced by": otherwise, "Nothing" would remain of programming.

7. Abstract algorithmic expressions

The expressions we have encountered until now are algorithms, in the sense that we could construct an automaton that accepts such expressions and—provided that the value of all variables is known—produces a result in a finite amount of time. The first mathematical formulae were, likewise, computational prescriptions. When we now manipulate formulae, it is the exception

rather than the rule that we are concerned with the efficiency of evaluating the formula; whether we replace $x^2 - y^2$ by $(x + y)(x - y)$, or prefer the replacement in the opposite direction, depends on the context. Likewise, we must abandon our fixation on efficiency if algorithmics is to enjoy a fruitful development. In general, developing an efficient algorithm will require that we first understand the problem, and for this we need simple algorithmic expressions; but to simplify an expression we have to shed our old habits. In mathematics, a formula like $\limsup_{n \to \infty} a_n^{1/n}$ shows that the thought of a constructive prescription has been abandoned. For algorithmics, it is similarly useful not to cling to the idea that every algorithmic expression must be interpretable by an automaton. An interesting step, that has not yet been explored, is to extend the notion of "structure" to structures whose finite constructibility is not guaranteed, or is even provably impossible. So, for example, the function *infrep* defined by

$$infrep\ x \Leftarrow \hat{x} + infrep\ x$$

would define an infinite structure of x's.

For the time being, the primary purpose is to allow algorithmic expressions that serve purely as specifications. An example of a possible specification is, in natural language, "a counterexample to Fermat's Last Theorem". Even though we do not know, at the time of writing, how to construct one, we can (in theory) recognize one if it exists. But even the uncertainty about the existence of a counterexample does not make the specification vague; it has a precise and well-understood meaning. Allowing such "unexecutable" specifications to be expressed in the language of algorithmics makes it possible to keep the complete trajectory, from the initial (formal) specification to the final algorithm, in one unified framework. Many transformational derivations start with an expression that is theoretically executable, but not in practice; in particular, they tend to take the form of "British Museum" algorithms, in which a finite but exceedingly large search space is examined. An advantage is that one may hope to run this initial "specification" for a very small example. A disadvantage is that it is not always trivial to give an expression for the proper search space; the requirement that it be finite may increase the distance from the true specification. Also, it is not unthinkable that this step might introduce an error (some relevant case not included in the search space); particularly so since it precedes the formal development. It turns out that we can use one particular "unexecutable" expression to denote a "sufficiently large" search space. It will be denoted by "$\mathbb{U}$", and its meaning is, informally, the "universe" of all possible objects that are meaningful, i.e., of the right type, in the given context. The trick is that the notation $P : s$, where P is a predicate, stands for the collection of elements of s that satisfy P. A more traditional notation is $\{x \in s \mid P(x)\}$; however, ":" works also on structures other than sets. The meaning of $\{x \in \mathbb{U} \mid P(x)\}$ is then understood to be the same as that of the common notation $\{x \mid P(x)\}$. So, if C is a predicate testing for the property of being a counterexample to Fermat's famous claim, then $C : \mathbb{U}$ specifies *all* counterexamples, and $\Box / C : \mathbb{U}$ specifies *a* counterexample.

8. Semantics for algorithmic expressions

How important it is to have a formal semantics for algorithmic expressions depends on the degree to which we want to place confidence in the meaningfulness of purely formal manipulations. My feeling is that in the current stage, a requirement that each proposed construction be accompanied by a formal definition of its meaning, so that each transformation could be formally justified, would be stifling. After all, great progress had been made in, e.g., Analysis, before Cauchy developed a firm foundation, and the paradoxes involved in summing divergent series have not led to disaster. Well-known examples where theory followed the application are Heaviside's "Operational Calculus" and Dirac's δ-notation. In due time, if the approach to Algorithmics investigated here proves its worth, possible paradoxes can be resolved by introducing higher-level concepts similar to, e.g., uniform convergence, to tighten the conditions of some theorems.

Still, some form of semantics would help to reason about aspects of proposed constructions. It is well known that we need extremely sophisticated mathematical constructions to define denotational semantics for expressions involving unbounded indeterminacy, and the desire also to allow infinite objects in the domain of discourse will hardly simplify matters. This seems to defeat the original motivation for defining semantics in a denotational way, namely to define meanings in clearer terms (i.e., better amenable to formal reasoning) than possible under the usual operational approach. In our case, the situation is even worse. For the intention is that the algorithmic expressions serve equally well as *specifications*. But specifications requiring an inordinate mathematical ability to understand them in the first place, are pretty useless. An operational semantic definition is, of course, out of the question (but see the next Section). A possible approach is the following.

Let $\mathfrak{E}$ stand for the set of algorithmic expressions. It is assumed that, next to the usual well-formedness criteria, other aspects, such as typability, are prerequisites for acceptability as an expression of $\mathfrak{E}$. To simplify the treatment, we assume that $\mathfrak{E}$ is recursive, and that $\mathfrak{E}$ contains a recursive subset $\mathfrak{V}$ of expressions that are identified with "values" (e.g., "2", or "$\lambda x : x+1$"). Intuitively, we can interpret an expression e of $\mathfrak{E}$ as "specifying" one, or more, or possibly no, elements of $\mathfrak{V}$. Define $\mathfrak{B}(e)$ to be the set $\{v \in \mathfrak{V} \mid e$ "specifies" $v\}$. Alternatively, we can interpret e as a "task" to find or construct some element of $\mathfrak{V}$. That task might have several solutions, or be impossible. Define $e \Rightarrow e'$ to mean: the task e can be solved by solving the task e'. The relation $\Rightarrow$ is a subset of $\mathfrak{E} \times \mathfrak{E}$. We can think of $\Rightarrow$ as "may be transformed to". The relation $\Rightarrow$ is reflexive and transitive (which may be ensured by taking the reflexive and transitive closure of some initial relation). Under the interpretation of an expression e as specifying elements of $\mathfrak{V}$, we would certainly expect e to specify a given $v \in \mathfrak{V}$ whenever $e \Rightarrow v$. On the other hand, if $v \in \mathfrak{B}(e)$ has been established, then v is a solution of the task e, so we have $e \Rightarrow v$. It follows that $\mathfrak{B}(e) = \{v \in \mathfrak{V} \mid e \Rightarrow v\}$. This gives a characterization of $\mathfrak{B}$ in terms of $\Rightarrow$. If we define the relation $\equiv \subset \mathfrak{E} \times \mathfrak{E}$ by $e \equiv e'$ iff $e \Rightarrow e'$ and $e' \Rightarrow e$, then $\equiv$ is an equivalence relation. We can, in the usual

way, step from $\mathfrak{E}$ (and $\mathfrak{V}$) to the equivalence classes induced by $\equiv$ in these sets. For convenience, the classes may still be denoted by some representative; but where formerly we had to write $e \equiv e'$, now we have $e = e'$.

When may a task e be replaced by a task e'? A requirement is certainly that any solution to e' be a solution to the original task e. So $e \Rightarrow e'$ requires $\mathfrak{B}(e') \subset \mathfrak{B}(e)$. We take this as the characterization of $\Rightarrow$ in terms of $\mathfrak{B}$, replacing "requires" by "iff". This has some consequences. Call an expression f "flat" if $\mathfrak{B}(f)$ is the empty set. An example of a flat expression is $\square/0$ (assuming that we do not admit this pseudo-value in the distinguished company of the proper values). Then we find, for any e, $e \Rightarrow \square/0$. But $\square/0$ can hardly be considered a reasonable replacement for e, unless e happens to be flat too. So, possibly, a more reasonable characterization of $\Rightarrow$ in terms of $\mathfrak{B}$ might additionally require the "preservation of definedness", meaning that a non-flat expression may not be replaced by a flat one. This gives rise to rules that are more complicated, which is a reason for rejecting this approach. Instead, it is better to accept the validity of $e \Rightarrow \square/0$, with the consequence that the meaning of $\Rightarrow$ does not correspond exactly to the intuitive notion of "may (as a task) be replaced by". The preservation of definedness has then to be proved separately for derivations involving $\Rightarrow$. It is generally easier to do this once than to check it for each individual derivation step.

There is another important difference between the usual formal treatment of the refinement relation between algorithms (see, e.g., MEERTENS[19]), and the relation $\Rightarrow$. For, in the usual treatment, one has $\square/0 \Rightarrow e$ for any e. This is unacceptable here, since we would then find that each $e = \square/0$. See, however, the notion of "total variant" of a function defined below.

If we start with some definition of $\mathfrak{B}$, next derive $\Rightarrow$ from that definition, and use $\Rightarrow$ then to find $\mathfrak{B}$, this will be the original function we started with. If, however, we start with some definition of $\Rightarrow$, use that to define $\mathfrak{B}$ and use this function to determine $\Rightarrow$, the latter relation may be larger than the original one. Next to transitivity and reflexivity, a "complete" relation $\Rightarrow$ satisfies a stronger closure property:

$$\text{If } \{v \in \mathfrak{V} \mid e' \Rightarrow v\} \subset \{v \in \mathfrak{V} \mid e \Rightarrow v\}, \text{ then } e \Rightarrow e'.$$

In this way, a relation $\Rightarrow$ can be specified by giving an initial subset, in the form of rules like

$$e_1 \square e_2 \Rightarrow e_i, \; i = 1, 2.$$

But this still does not give the full story. A pleasant property of expression-forming constructions is *monotonicity*: if $C[e]$ stands for an expression containing e as a *constituent* sub-expression, and $e \Rightarrow e'$, then we want to be able to conclude that $C[e] \Rightarrow C[e']$. This property is postulated for all constructions admitted to our language (and so $\mathfrak{B}$ is excluded).

It is necessary to give a meta-rule for $\Rightarrow$ on functions, since equality of functions is not in general decidable. (The notion of "function" includes here our binary operations.) A reasonable rule appears to be:

META-RULE FOR $\Rightarrow$ ON FUNCTIONS.
Let f *and* $f' \in D \to \mathbb{V}$ *(where* $D \subset \mathbb{V}$*), and let* $f\,v \Rightarrow f'\,v$ *for all* $v \in D \cup \{\square/0\}$.
Then $f \Rightarrow f'$.

This rule makes a choice between several possibilities for defining $\Rightarrow$ on functions. The possibility chosen seems to be the more manageable rule. If functionals (higher-order functions) can operate on functions involving indeterminacy, the meta-rule must be used with caution. For assuming the reasonable identity $f \square g; x = fx; \square\, g\,x$, we are led to conclude that $f \square g = \lambda x: (fx; \square\, g\,x)$. Now take $f = \mathrm{id}$ $(= \lambda x: x)$, $g = 3\ll$ $(= \lambda x: 3)$, and let $h = \lambda x: x \square 3$. Then $h = f \square g$. But if $F = \lambda\phi: (\phi 1; +\phi 2)$, then we find $Ff \square g = Ff; \square\, Fg = 1{+}2; \square\, 3{+}3 = 3 \square 6$, whereas $Fh = h\,1; +h\,2 = 1 \square 3; +2 \square 3 = 3 \square 4 \square 5 \square 6$.

The converse rule "If $f \Rightarrow f'$, then $f\,v \Rightarrow f'\,v$" results if the monotonicity postulate is applied to function application. A consequence is that if f is a partial function, but f' is total (i.e., never yields $\square/0$), then $f \Rightarrow f'$ cannot hold. However, it is often desirable to turn partial functions into total ones. For example, a problem specification may prescribe that error messages be given if certain conditions are not met. It may then be preferable to treat these error messages initially as "instances" of $\square/0$. Call f' a "variant" of f if $f\,v \Rightarrow f'\,v \neq \square/0$ whenever $f\,v$ is not flat. A useful curiosity is that if f is "determinate" (see below), then $f' \Rightarrow f$. This is also a sufficient condition to show that a determinate function f' is a variant of f. A "total variant", finally, is a variant that is a total function.

We also need rules for function applications. Unfortunately, the simple rule

$$(\lambda x: C[x])\,e = C[e]$$

is not enough. One counter-example is found by considering $f\,1 \square 2$, where $f = \lambda x: x - x$. Mechanical textual substitution gives $1 \square 2; -1 \square 2 = -1; \square\, 0 \square 1$, which, together with the above meta-rule, would lead to the conclusion that function application is not monotonic (or, worse, that $0 \Rightarrow 1$). Another problem is given by taking $h\,\square/0$, where $h = \lambda x: x \square 3$ is—for the moment—taken to be a strict function. Textual substitution results in $\square/0; \square\, 3 = 3$, which is inconsistent with the identity characterizing strictness, namely $h\,\square/0 = \square/0$. Therefore, the rule for function application needs the condition that the expression for the argument is "determinate" (see below) and non-flat if the function is specified to be strict. This corresponds, roughly, to what is known as "call-by-value" semantics. Note, however, that it is not required to *evaluate* the argument; all that is needed is that we exhibit certain properties, for which some sufficiency conditions can even be given in terms of syntactic criteria. If the function definition does not involve more than a single occurrence of the argument, then indeterminacy of the argument is no problem. The reason that functions are non-strict by default should now be apparent: this choice simplifies the applicability condition of the rule. Note that for strict functions it is always safe to use the rule in the "Fold" direction, namely $C[e] \Rightarrow (\lambda x: C[x])\,e$.

An expression e is determinate if, for any two values v_1 and v_2 such that $e \Rightarrow v_1$ and $e \Rightarrow v_2$, we have $v_1 = v_2$. It seems reasonable to require all values to be determinate, which implies that $\Rightarrow$ and $=$ coincide on $\mathcal{V}$. All values are, by definition, non-flat. The function-application rule could then be stated by restricting the argument to values (as was already done for the meta-rule), with the advantage that the notions of "determinacy" and "flatness" need not be used. A problem arises, however, if we want to define $\mathcal{B}(h)$, where h is as above (but not strict). Since h is obviously indeterminate (we have both $h \Rightarrow \mathrm{id}$ and $h \Rightarrow 3\ll$), we do not want to allow $\lambda x : x[]3$ as element of $\mathcal{V}$. No enumerable collection of determinate lambda forms, however, can capture the meaning of h. This is related to the problem mentioned above for equality of functions.

A function definition may contain several occurrences of the argument, as in

$$\mathrm{abs}\ x \Leftarrow \mathrm{if}\ x < 0\ \mathrm{then}\ -x\ \mathrm{else}\ x\,.$$

Suppose we want to show the equality

$$\mathrm{abs}\ 2\times e = 2\times \mathrm{abs}\ e\,.$$

This is easily proved by the Unfold/Fold method:

$$\begin{aligned}&\mathrm{abs}\ 2\times e = \mathrm{if}\ (2\times e) < 0\ \mathrm{then}\ -(2\times e)\ \mathrm{else}\ (2\times e) = \\ &\mathrm{if}\ e < 0\ \mathrm{then}\ 2\times -e\ \mathrm{else}\ 2\times e = 2\times \mathrm{if}\ e < 0\ \mathrm{then}\ -e\ \mathrm{else}\ e = \\ &2\times \mathrm{abs}\ e\,.\end{aligned}$$

Unfortunately, the condition for the function-application rule is not satisfied if e is indeterminate. And yet, it is easy to see that in this particular case no harm is done. This insight can be generalized to the following meta-rule:

META-RULE FOR INDETERMINATE UNFOLD/FOLD.
Let $C[e]$ and $C'[e]$ be expressions containing e as a constituent expression, and let e occur at most once in $C'[e]$.
If there is a derivation $C[e] \Rightarrow C'[e]$ for determinate *e, and e is* uninterpreted *in that derivation, then $C[e] \Rightarrow C'[e]$ is also valid for indeterminate expressions e.*

This allows one to use, e.g., $e - e \Rightarrow 0$ or $1 \cdot e = e$, the latter by applying the meta-rule in both directions. This meta-rule is a corollary of the rules given above, as the following derivation shows:

$$C[e] \Rightarrow (\lambda x : C[x])\,e \Rightarrow (\lambda x : C'[x])\,e \Rightarrow C'[e]\,.$$

The middle step is an application of the meta-rule for $\Rightarrow$ on functions, together with the monotonicity property.

9. EXECUTABLE EXPRESSIONS

In going from specification to implementation, we can stop the development when we have an expression that has an obvious translation in terms of a program (i.e., it belongs to the "solution space"). If that translation is so obvious, then we can wonder if it could not be delegated to a machine. If that is possible at all (and it is certainly possible for some subset of the language $\mathcal{E}$ of

algorithmic expressions), then we effectively have a machine for executing some expressions. This would eliminate an uninteresting step that might easily introduce clerical errors. It also opens the possibility of having the machine apply certain optimizations that are hard to express without spoiling the clarity of the expressions, but that are nevertheless obvious (e.g., replacing recursion by iteration, or eliminating redundant computations).

In the current stage of this work, a serious effort to define an "executable subset" of the algorithmic expressions is still out of the question. We may wonder, however, what properties we would require of a hypothetical machine for executing expressions. Let $\mathfrak{E}$, $\mathfrak{V}$ and $\Rightarrow$ be as in the previous section. A possible approach is that the machine tries to mimic $\Rightarrow$, going through a sequence $e_1 \Rightarrow e_2 \Rightarrow \cdots$, hopefully ending up in a member of $\mathfrak{V}$. To the machine, the forms it operates on are states, rather than expressions. It is realistic to assume that the machine may have to attach some bookkeeping information to the expressions. To simplify the discussion, this possibility will be ignored. Obviously, we may not assume that the machine is capable of accepting all expressions of $\mathfrak{E}$ as states.

Let $\mathfrak{P}$ be a subset of $\mathfrak{E}$, standing for the "executable" expressions, i.e., the expressions that the machine is designed to cope with. (The letter $\mathfrak{P}$ has been chosen here because to us these expressions are programs for the machine.) We assume that $\mathfrak{P}$ and $\mathfrak{P} \cap \mathfrak{V}$ are recursive sets. Now we define $p \rightarrow p'$ to mean: if the machine is in the state p, it can, possibly, switch next to the state p'. So $\rightarrow$ is a subset of $\mathfrak{P} \times \mathfrak{P}$. There is no reason to require that the machine be deterministic, but it makes sense to assume that $\rightarrow$ is at least recursively enumerable. There must be some halting condition for the machine. A simple criterion is to have the machine halt if its state is a value, i.e., a member of $\mathfrak{V}$. This is then the output. For the sake of simplicity, we require all values to be "dead-end states", where p is a dead-end state if no state is reachable via $\rightarrow$ from p. Now we have two requirements:

> *Soundness.* Let $\rightarrow^*$ stand for the transitive and reflexive closure of $\rightarrow$. Then, for all $p \in \mathfrak{P}$ and $v \in \mathfrak{V}$, if $p \rightarrow^* v$, then $p \Rightarrow v$.
>
> *Preservation of Definedness.* Let p be an arbitrary non-flat member of $\mathfrak{P}$ (where the non-flatness is with respect to $\mathfrak{E}$). Then (a) if $p \rightarrow^* p'$, and p' is a dead-end state, then it is a value; and (b) there does not exist an infinite sequence of states $p_0, p_1, \cdots$ such that $p = p_0 \rightarrow p_1 \rightarrow \cdots$.

The first requirement is simply that the machine produce no wrong answers. The second one requires that *if* the program p, viewed as an expression, specifies a result (some value), then the machine will output a value when started in state p. Part (a) prohibits the machine from reaching a dead end without producing output (which, if it can be detected, can be interpreted as abortion of the program), whereas part (b) forbids infinite loops. It is, of course, in general undecidable whether the machine will halt if started in a given state p, so the proof would depend heavily on properties of $\Rightarrow$, such as monotonicity, and possibly of $\mathfrak{P}$.

A relation $\rightarrow$ satisfying the requirements for soundness and for preservation of definedness, may be called an "operational semantics" for $\mathcal{P}$. Note that different machines may correspond to different executable subsets of $\mathcal{E}$, and even that two machines operating on the same set $\mathcal{P}$ may differ in their operational semantics. So there is no such thing as *the* subset of executable expressions. In fact, let $\mathcal{P}$ be *any* executable subset, with operational semantics $\rightarrow$. Then it is always possible—provided that $\mathcal{E}$ is sufficiently expressive—to find some pair $\langle e, v\rangle \in \mathcal{E} \times \mathcal{V}$ such that $e \notin \mathcal{P}$ and $e \Rightarrow v$. Then $\mathcal{P} \cup \{e, v\}$ is also an executable subset, with operational semantics $\rightarrow \cup \{\langle e, v\rangle\}$. So there do not even exist maximal executable subsets of $\mathcal{E}$.

The "canonical evaluation" of programs in the style of BURSTALL and DARLINGTON [6] is one prime candidate for being an operational semantics. Some expressions have obvious translations into an imperative style, like $\uparrow_{age}/+/mr*dm$ into the program of figure 2a of Section 2. $\mathcal{P}$ could be restricted to such programs, which could then be "compiled" into "pidgin ALGOL". Yet another possibility is translation into FP.

A problematic aspect is the evaluation of expressions such as $x \,[]\, y$. It is easy to imagine a machine that would always go to a state $x' \,[]\, y$ if $x \rightarrow x'$ for some x'. Note, however, that the machine is forced, by virtue of the requirement of preservation of definedness, to try the other choice if the preferred choice leads to a dead end without output. This corresponds, in a limited sense, to what is sometimes called "angelic nondeterminism". Operationally, however, no "nondeterminism" need be involved in this. But the same is also required if the first choice may lead to an infinite loop. Fortunately, the machine need not decide beforehand if this undecidable contingency will arise; it is sufficient if the evaluations of the alternatives are "dovetailed" (interleaved) in a fair way, i.e., not excluding some alternative indefinitely. In the context of a recursive function definition, this provides "automatic backtracking", where $[]/0$ takes the role of "Fail". To give a stronger example, consider

$$f x \Leftarrow \text{if } x = 0 \text{ then } f0 \,[]\, 1 \text{ else } 1.$$

It is then guaranteed that $f0 = 1$, since $f0 \Rightarrow f0 \,[]\, 1 \Rightarrow f1 \Rightarrow 1$, and no other value than 1 could be a possible outcome. Although this may not be the most pleasant thing to implement, neither is it prohibitively difficult or expensive, and certainly not if occurrences of $[]$ in "executable code" are the exception rather than the rule. It will often be possible to exhibit the non-flatness of expressions by a static analysis. If x is known to be non-flat, then the step $x \,[]\, y \rightarrow x$ is allowed.

10. SOME MORE BASIC OPERATIONS

If x and y denote two objects, $\langle x, y\rangle$ denotes an object that is a pair consisting of those two objects. The functions π_1 and π_2 allow the retrieval of the components from the pair, so, e.g., $\pi_2\langle x, y\rangle = y$. If $x \in D_1$ and $y \in D_2$, the pair $\langle x, y\rangle \in D_1 \times D_2$. If orderings are defined on the component domains, then the product domain is assumed to be ordered lexicographically, unless a different order is specified.

We have already encountered the operation $\ll$, which selects its left operand: $x \ll y = x$. An important application is that $x \ll$ denotes the constant function $\lambda y\colon x$. The operation $\gg$ selects its right operand (and so $x \gg$ is, for each x, the identity function id).

If x is a determinate object (meaning that no choice of the type $[]$ is involved), then $P?x$, where P is a predicate (i.e., a function returning a truth value), stands for $x \ll_* Px$. This formulation has probably no immediately obvious meaning to the reader. Remember that "false" and "true" are identified with 0 and $1 = \hat{\iota}$, respectively. So, if Px is false, $P?x = x \ll_* 0 = 0$. If Px is true, $P?x = x \ll_* 1 = x \ll_* \hat{\iota} = \hat{}x \ll \iota = \hat{x}$. We see now that $P?x$ means "if Px then $\hat{x}$ else 0". The operation ? is mainly (but not only) useful as auxiliary operation to define other operations. An important application is in the definition of a "filter": a function to "extract" all members of a structure satisfying a given property. The function $+/P?*$ returns the structure of all P-satisfying members of its argument. For example, if Px holds, but Py does not, we obtain $+/P?*\hat{x}+\hat{y} = +/(\hat{}P?x; +\hat{}P?y) = +/\hat{\hat{x}}+\hat{0} = +/\hat{\hat{x}}; ++/\hat{0} = \hat{x}+0 = \hat{x}$. It is important enough to merit a shorter notation; for this, we use $P:$, which we have already encountered. For example, the filter $x=:$ extracts all elements equal to x. We can then define

$$x\in \Leftarrow 0\neq x=:$$

to test for membership of x.

Some laws that use : are:

$$P:+/ = +/P:*;$$

$$x=:\mathbb{U} = \hat{x};$$

$P:f_* = f_*(Pf):$, provided that f is determinate;

$P:Q: = P\wedge Q;:$ (remember that $P\wedge Q; x = Px; \wedge Qx$).

The proof of the first, least obvious, law, is $P:+/ = +/P?*+/ = +/+/P?** = +/P:*$, in which the middle step is an application of the corollary of Section 5. The second law cannot be proved from previous laws, since no previous law involves $\mathbb{U}$; instead, it can be viewed as a (partial?) characterization of $\mathbb{U}$. The derivation of the third law is left as an exercise to the interested reader. (Hint: use the meta-rule for $\Rightarrow$ on functions from Section 8 to show first that $fx;\ll = fx\ll$, and next that $P?f = f_*(Pf)?.$) The last law is most easily proved by proving it first for determinate predicates P and Q (by considering all possibilities of assigning truth values to Px and Qx), and then using the last meta-rule of Section 8.

An example of the use of these laws is given by

$$x\in P:\mathbb{U} = 0\neq x=:P:\mathbb{U} = 0\neq P:x=:\mathbb{U} = 0\neq P:\hat{x} = 0\neq P?x = Px.$$

Another important property connected with : needs some terminology. Call an operation $\times\in D\times D\to D$ "selective" if $[]\Rightarrow\times$, i.e., for all x and $y\in D$,

$x \,[]\, y \Rightarrow x \times y$. Examples of selective operations are $[]$ itself, $\ll$, $\gg$, and $\downarrow_f$ and $\uparrow_f$, to be defined below. The property is then:

> If $\times$ is selective and $\times/P{:}s \Rightarrow x \neq []/0$ for some structure s, then $P\,x \Rightarrow 1$.

The crucial step in the proof is $[]/P{:}s \Rightarrow \times/P{:}s$.

Another useful application of ? is in the definition of $\rightarrow$, where the predicate $p\rightarrow$ is defined by $[]/p \ll ?$, in which p is a proposition, i.e., an expression whose value belongs to the domain of truth values. (Since the operation ? requires a predicate as first operand, the operation $\ll$ is used to turn the proposition p into a predicate.) Then $p\rightarrow x; []\, q\rightarrow y$ specifies, indeterminately, x or y, but x is only specified if p can be satisfied, and y if q can be. For example, assume that p holds and q does not. Then we find $p\rightarrow x; []\, q\rightarrow y = []/p \ll ?x; [][]/q \ll ?y = []/\hat{x}; [][]/0 = x\,[][]/0 = x$. So the combination of $\rightarrow$ with $[]$ gives "guarded expressions", whose meaning is not primitive but is obtained by composing the meanings of the individual operations. Note that $0[]1; \rightarrow x = x$, since $0[]1; \rightarrow x = 0\rightarrow x; []1\rightarrow x$.

An important law for $\rightarrow$ is:

$$f\,p\rightarrow\ = p\rightarrow f, \text{ provided that } f \text{ is strict.}$$

Since $p\rightarrow$ is obviously strict, we have $p\rightarrow q\rightarrow\ = q\rightarrow p\rightarrow\ (= p \wedge q; \rightarrow)$.

If x and y are elements of a semi-lattice with greatest lower bounds, then $x\downarrow y$ stands for the greatest lower bound of x and y. The expression $\downarrow/0$ stands then for the top of the semi-lattice. If it has no top already, it can be extended with one in a consistency-preserving way. It is often profitable to identify $\downarrow/0$ with $[]/0$. The operation $\uparrow$ is defined similarly. Although it is likewise often useful to define $\uparrow/0 = []/0$ if the (semi-)lattice has no bottom, it is generally unsafe to use this device for both $\downarrow$ and $\uparrow$ if they can appear mixed in a formula.

On structures, we can define a default partial ordering

$$s \leqslant t \text{ iff } 0[]1; \ll{:}t \Rightarrow s.$$

So $s \leqslant t$ if s can be obtained by omitting some (possibly none) of the members of t. For sequences, $\leqslant$ corresponds then to "is a (possibly non-contiguous) subsequence of". For sets, natural numbers, and truth values, we find as meanings, respectively, "$\subset$", the traditional "$\leqslant$", and implication. Structures for which the construction operation $+$ is associative and commutative form now a lattice, and $\downarrow$ gives, e.g., "$\cap$" for sets and "$\wedge$" for truth values. The operation $\uparrow$ is then defined as well. Note that $\uparrow/0 = 0$, since 0 is an identity of the operation $\uparrow$.

The operation $<_f$, where f is a determinate function, is defined by

$$x <_f y \Leftarrow fx; <fy,$$

and $=_f$, $>_f$, etc., are defined similarly.

The operation $\downarrow_f$, for a determinate function f whose range is a domain with a total ordering, is defined by

$$x \downarrow_f y \Leftarrow (x \leqslant_f y ; \rightarrow x) \,[]\, (y \leqslant_f x ; \rightarrow y).$$

An identity relating $\downarrow_f$ to $\downarrow$ is $f \downarrow_f / = \downarrow / f*$. The operation $\uparrow_f$ is defined similarly. It is again often helpful to define $\downarrow_f / 0 = []/0$ or $\uparrow_f / 0 = []/0$, with the same *caveat* for mixed use.

Finally, we need a function # to count the number of elements of a structure. This can be done by mapping each element to ι, so $\#\hat{x} + \hat{y} = \hat{\iota} + \hat{\iota} = 1+1 = 2$. So we can define # as $\iota \ll *$. There is a surprise, though: on sets (and more generally, on all structures with idempotency) this # refuses to count properly. The problem is that #, as defined, is a homomorphism. But the number-of-elements function on sets is not. That "number of elements" cannot be defined as a homomorphism on sets follows from the breakdown of the law $\# +/ = +/ \#*$ (an application of the corollary of Section 5) for sets; in particular, $\#s ; +\#s$ for a non-empty set s differs from $\#s + s = \#s$. The function $\iota \ll *$ is only defined on sets as a mapping to the *set* $S_{\{\iota\}}$, which is the domain of truth values, and it tests then for non-emptiness.

11. First example: A Text-Formatter

The following problem specification, copied from Bauer *et al.* [2], is a reformulation (under the heading "Text editor") of the original specification (under the heading "Line editing problem") given in Naur [22].

> "A text, i.e. a non-empty sequence of words separated by blanks (BL) or new line characters (NL), is to be re-structured according to the following rules:
> (1) every two words are separated by exactly one BL or NL;
> (2) the first word is preceded by NL; the last character is neither BL nor NL;
> (3) each line is at most MAX characters long (not counting NL); within this range, it contains as many words as possible.
> The input line is required to start with NL; further, no word must contain more than MAX characters."

As a first step, we aim at more abstraction. This can be done by assuming that a type "word" is already given, and that the function #, applied to a word, will give its length (some natural number). Then the input can be viewed as a single "line", i.e., a sequence of words, whereas the output is a sequence of lines. This abstract view makes requirements (1) and (2), the clarification "(not counting NL)" of (3) and the first part of the last sentence irrelevant, since they deal with the concrete representation of sequences of lines in terms of some character code. More important is that it guarantees that the algorithmic development will work for different representations. (If more concreteness is nevertheless required, it is still advantageous to split the problem into a more algorithmic part, and the treatment of the concrete representation. For the latter, mappings from the types "sequence of words" and "sequence of lines" to the type "sequence of (character or 'BL' or 'NL')" have to be defined, and the abstract algorithm obtained has to be transformed

to work on this new concrete representation. Techniques for effecting a change of representation are given in BURSTALL and DARLINGTON [6] and MEERTENS [18]. Hopefully, it will be possible in some future to leave such low-level transformations to an automated system.)

Next we have to make the natural-language specification more precise. The meaning of "A text ... is to be re-structured" is best expressed as a requirement on the relationship between the input and the output:

(0) the output, "unstructured", is the original input.

Furthermore, requirement (3) is best split into two parts:

(3a) each line of the output is at most of length MAX;
(3b) each line of the output contains as many words as is possible within the constraints imposed by (0) and (3a).

An observation can now be made: the specification is symmetric with respect to the directions left-to-right and right-to-left. More precisely, let *rev* be a function that takes a sequence as argument and returns the reverse sequence as result. Then we have:

> If a function *f* "solves" (0), (3a) and (3b) (i.e., for each acceptable input line *i*, *f i* is acceptable output), then so does *rev* ∗ *rev f rev* (= *rev rev* ∗ *f rev*).

From (3b) we can derive the following requirement:

> No line of the output starts with a word that would have fit at the end of the previous line.

For, otherwise, that line contains fewer words than possible. Expressed very informally, this means: lines are "eager" to accommodate words as long as there is enough room. Because of the symmetry, a solution must then also satisfy the mirror-image "reluctant" requirement:

> No line of the output ends with a word that would have fit at the start of the following line.

But it is not hard to give input for which the "eager" and the "reluctant" requirements are, together, impossible to satisfy. An example, if MAX = 13, is the input "`Impossible.to.satisfy.in.both.ways!`". The unique "eager" solution is then

```
Impossible.to
satisfy.in...
both.ways!...
```

The "reluctant" solution is different:

```
Impossible...
to.satisfy...
in.both.ways!
```

Something is wrong. The "reluctant" approach tends to leave as much white space on the first line as possible. This is, by application of real-world knowledge, typographically undesirable. The "eager" approach, in contrast, leaves the last line unfilled. This is, if not typographically desirable, then at least neutral. This suggests to us replacing (3b) by:

(3b′) each line *but the last, if any,* of the output contains as many words as is possible within the constraints imposed by (0) and (3a).

However, this still does not solve the "eager" *vs.* "reluctant" problem: just add a 13-character "word" (e.g., "`Exasperating!`") to the end of the example input given above. The problem with the specification seems to reflect our conditioning to think in terms of left-to-right. Whereas (0) and (3a) are "boundary conditions", (3b) is an "objective", namely, "Do not waste more space than necessary"; more precisely:

(3b″) minimize the total white space on the output, not counting the last line.

This approach was suggested to me by Robert Dewar. There is still a tiny problem left: if the last line is completely filled, then another empty line may be added without penalty in terms of the white-space objective. So a second objective, subordinate to the previous one, is to minimize the number of lines of the output.

Now we are ready to start giving a formal treatment of the problem. This will be done in an unusually detailed way, comparable to the minuteness of the steps in $S = 1S = (\frac{1}{2}\cdot 2)S = \frac{1}{2}(2S) = \frac{1}{2}(S+S)$. We use the letter r for the input ("raw"), and c for the output ("cooked"). The proposition that the input/output constraints are satisfied, is denoted by $r{\sim}c$. If, furthermore, *obj* denotes the objective function, then the problem is to determine, for given input r,

$$f\,r \Leftarrow \downarrow_{obj}/\,r{\sim} : \mathbb{U}.$$

In words: take any *obj*-minimizing object c such that $r{\sim}c$. We put $\downarrow_{obj}/0 = \mathbb{U}/0$. We must define $\sim$ and *obj*. If *len* is a function giving the length of a single line, then $\sim$, expressing that the two constraints (0) and (3a) are satisfied, can be defined as:

$$r{\sim}c \Leftarrow +/c = r;\ \wedge\uparrow/len*c \leqslant \text{MAX}.$$

The *len* of a line is the sum of the lengths of its words, plus 1 for each space between a pair of words. A simple way to obtain this result, is to add 1 to the length of each word before summing, and to subtract 1 from the sum. For an empty line, we have to define its length separately:

$$len\ 0 \Leftarrow 0;$$

$$len\ l{+}\hat{w} \Leftarrow -1;\ +\,+/(1+\#)*l{+}\hat{w}.$$

For a line consisting of a single word, we have, of course, $len\,\hat{w} = -1; + +/(1+\#)*\hat{w} = -1; +(1+\#)w = \#w$. The objective function is defined by

$$obj\,c \Leftarrow \langle ws\,c,\ \#c\rangle,$$

where the "white-space" function ws gives the white space on its argument (not counting the last line). The white space left on a single line is given by the function $ws_1 = \text{MAX} - len$. This quantity has to be summed over all lines but the last. This gives us the definition:

$$ws\,c'+\hat{l} \Leftarrow +/ws_1*c'.$$

To make the function total, we also define

$$ws\,0 \Leftarrow 0.$$

We turn now first to the question whether it is possible to satisfy the constraints, not bothering about the objective. One extreme approach to satisfy (0) is to have a one-line page, or $c = \hat{r}$. This is likely to violate constraint (3a). Since the white space does not matter, we can try the other extreme: use a separate line for each word. This would give us $c = \hat{}*r$. Then (0) is, of course, satisfied, but what about (3a)? Since $len\,\hat{} = \#$, we find

$$\uparrow/\,len*c = \uparrow/\,len*\hat{}*r = \uparrow/\,(len\,\hat{})*r = \uparrow/\,\#*r.$$

So, if $\uparrow/\,\#*r \leqslant \text{MAX}$, i.e., each word on the input is at most MAX long, we have $r \sim \hat{}*r$, so the problem posed is solvable. Next, we show that this condition is not only sufficient, but also necessary. If $l \neq 0$,

$$len\,l = -1; + +/(1+\#)*l \geqslant -1; +\uparrow/(1+\#)*l = -1; +1+\uparrow/\#*l = \uparrow/\#*l.$$

In the given context, $\uparrow/0 = 0$, since line lengths are natural numbers. Then, if $l = 0$, $len\,l = 0 = \uparrow/\,\#*l$, so no condition $l \neq 0$ is necessary for the inequality $len\,l \geqslant \uparrow/\,\#*l$. Now we have

$$\uparrow/\,len*c \geqslant \uparrow/\uparrow/\,\#**c = \uparrow/\,\#*+/c.$$

If $r \sim c$ is satisfied, $+/c = r$ and $\uparrow/len*c \leqslant \text{MAX}$, so

$$\uparrow/\,\#*r = \uparrow/\,\#*+/c \leqslant \uparrow/len*c \leqslant \text{MAX}.$$

In conclusion,

$$fr \neq []/0 \text{ if and only if } \uparrow/\,\#*r \leqslant \text{MAX}.$$

To "synthesize" f, we must derive some properties of $\sim$ and obj. In the first place, empty lines can be deleted from the output without violating the constraints. For

$$+/c_1+\hat{0}+c_2 = (+/c_1)+(+/\hat{0})+(+/c_2) = (+/c_1)+0+(+/c_2) = (+/c_1)+(+/c_2) = +/c_1+c_2.$$

Also, $\uparrow/len*\hat{0} = \uparrow/\hat{}len\,0 = \uparrow/\hat{0} = 0$, so

$$\uparrow/len \circ c_1 + \hat{0} + c_2 = (\uparrow/len \circ c_1)\uparrow(\uparrow/len \circ \hat{0})\uparrow(\uparrow/len \circ c_2) = (\uparrow/len \circ c_1)\uparrow 0\uparrow(\uparrow/len \circ c_2) = (\uparrow/len \circ c_1)\uparrow(\uparrow/len \circ c_2) = \uparrow/len \circ c_1 + c_2 .$$

Combining these two gives

$$r \sim c_1 + c_2 \text{ if and only if } r \sim c_1 + \hat{0} + c_2 .$$

Next, we show that empty lines are always disadvantageous in terms of the objective. To show this, we have to distinguish several cases, because of the form of the definition of *ws*. First, we treat the case where the empty line considered is not the last line. Since

$$+/ws_1 \circ \hat{0} = +/\hat{}\, ws_1\ 0 = ws_1\ 0 = \text{MAX} - len\ 0 = \text{MAX},$$

we have

$$ws\ c_1 + \hat{0} + c_2 + \hat{l} = +/ws_1 \circ c_1 ; + \text{MAX} + +/ws_1 \circ c_2 \geqslant +/ws_1 \circ c_1 ; + +/ws_1 \circ c_2 = +/ws_1 \circ c_1 + c_2 = ws\ c_1 + c_2 + \hat{l}.$$

If the empty line *is* the last, but not the only one, we find

$$ws\ c_1 + \hat{l} + \hat{0} = +/ws_1 \circ c_1 + \hat{l} = +/ws_1 \circ c_1 ; + +/ws_1 \circ \hat{l} \geqslant +/ws_1 \circ c_1 = ws\ c_1 + \hat{l}.$$

Finally, if the whole document consists of just one empty line,

$$ws\ \hat{0} = ws\ 0 + \hat{0} = +/ws_1 \circ 0 = +/0 = 0 = ws\ 0.$$

So in all cases

$$ws\ c_1 + \hat{0} + c_2 \geqslant ws\ c_1 + c_2 .$$

Since

$$\#\, c_1 + \hat{0} + c_2 = (\#c_1) + (\#\hat{0}) + (\#c_2) = (\#c_1) + 1 + (\#c_2) > (\#c_1) + (\#c_2) = \#\, c_1 + c_2 ,$$

we have

$$obj\ c_1 + \hat{0} + c_2 > obj\ c_1 + c_2 .$$

We may conclude that it is never helpful to consider output containing empty lines. This can be expressed formally by inserting a filter that sifts out pages with empty lines, e.g., by replacing $\mathbb{U}$ in the definition of f by $0 \notin : \mathbb{U}$. On the set of pages without empty lines, *obj* has the same ordering as *ws*, so we can replace $\downarrow_{obj}$ in the definition of f by $\downarrow_{ws}$. We can now also use for the *len* function the uniform definition

$$len\ l \Leftarrow -1 ; + +/(1 + \#) \circ l ,$$

since we know that the function is not applied to an argument 0. This allows us to do some elementary mathematics. If $c \neq 0$, we can put $c = c' + \hat{l}$, so

$$ws\,c = ws\,c' +\!\!\!\!\!\!\hat{}\; \hat{l} = +/ws_1 * c' = +/(\text{MAX} - len) * c' =$$
$$+/(\text{MAX} - (-1) + +/(1 + \#) *) * c' =$$
$$+/(\text{MAX} + 1; - +/(1 + \#) *) * c' =$$
$$\text{MAX} + 1; \times \# c'; - +/ +/(1 + \#) ** c' =$$
$$\text{MAX} + 1; \times \# c'; - +/(1 + \#) * +/c'.$$

If, furthermore, $r \sim c$, then $r = +/c$, so

$$len\,r = len\,+/c = len\,+/c' +\!\!\!\!\!\!\hat{}\; \hat{l} = -1; + +/(1 + \#) * +/c' +\!\!\!\!\!\!\hat{}\; \hat{l} =$$
$$+/(1 + \#) * +/c'; + (-1) + +/(1 + \#) * l =$$
$$+/(1 + \#) * +/c'; + len\,l,$$

so that we have

$$+/(1 + \#) * +/c' = len\,r; -len\,l.$$

Combining these two gives us: if $r \sim c$ and $c = c' +\!\!\!\!\!\!\hat{}\; \hat{l}$,

$$ws\,c = \text{MAX} + 1; \times \# c'; - (len\,r; - len\,l).$$

In using this formula to compare the outcome of *ws* on two different non-empty pages that both meet the constraints, we can replace the part "$-(len\,r; -len\,l)$" by "$+ len\,l$", since r, and therefore $len\,r$, is fixed. Since then, moreover, $len\,l < \text{MAX} + 1$, the quantity $\#c'$ prevails over $len\,l$ in the comparison. This leads us to consider the simpler function

$$lpos\,c' +\!\!\!\!\!\!\hat{}\; \hat{l} \Leftarrow \langle \#c', len\,l \rangle.$$

On non-empty pages, the ordering of *ws* is that of *lpos*. If we also define

$$lpos\,0 \Leftarrow \langle 0, 0 \rangle,$$

we may even drop the restriction to non-empty pages.

If we combine the above findings, we obtain the following definition for f:

$$f\,r \Leftarrow \downarrow_{lpos} / r \sim : 0 \notin : \mathbb{U}.$$

This formulation makes it possible to find solutions of $f\,r +\!\!\!\!\!\!\hat{}\; \hat{w}$ in terms of solutions of $f\,r$. The effect, as we will see, is that of following the "eager" strategy. We may thereby lose some other, equally optimal, solutions. Expressed in words, the crucial idea is the following. Suppose c is the result of formatting a given input text r. We can "truncate" c by "erasing" the last word on its last line, and the last line itself if it then becomes empty. Then the two data c-truncated and w, together with the knowledge that c-truncated was obtained by erasing w from an optimal solution c, suffice to reconstruct c uniquely. (It is assumed that the value of MAX is known.) Moreover, c-truncated is then an acceptable way of formatting r-truncated, and although it need not be an optimal solution, there is no harm done by replacing it by an optimal one. It follows then that an optimal solution for r (since we know it to exist) can be formed from an optimal solution for r-truncated. This will now be shown more formally. We define

$$Trnc\ c'+\hat{}l+\hat{w} \Leftarrow (l \neq 0; \rightarrow c'+\hat{l})\ \square\ (l = 0; \rightarrow c');$$
$$Trnc\ r'+\hat{w} \Leftarrow r'.$$

(Note that the function *Trnc* is "overloaded" here: the two definitions operate on arguments from different domains.) So suppose $r \sim c$, and among all possible solutions the *lpos* of c is minimal. Suppose, moreover, $c \neq 0$, so $r = +/c \neq 0$ (remember that empty lines are excluded), and we can put

$$c = c'+\hat{}l+\hat{w}_1;$$
$$r = r'+\hat{w}_2.$$

From $r \sim c$ we have $r'+\hat{w}_2 = +/c'+\hat{}l+\hat{w}_1 = +/c'; +l+\hat{w}_1$, so $r' = +/c'; +l$ and $w_1 = w_2$. (Note that we used the knowledge that $+\hat{}$ is injective here. The conclusion would be unwarranted if $+$ were commutative or idempotent.) We can now drop the subscripts on w. Let $c_T = Trnc\ c$. Then

$$c_T = Trnc\ c'+\hat{}l+\hat{w} = (l \neq 0; \rightarrow c'+\hat{l})\ \square\ (l = 0; \rightarrow c'),$$

in which c' and l are still to be determined. We see that c' and l satisfy

$$(l \neq 0; \rightarrow c_T = c'+\hat{l})\ \square\ (l = 0; \rightarrow c_T = c').$$

If $c_T = 0$, the first alternative cannot apply (since $c'+\hat{l} \neq 0$), so then $l = 0$. Otherwise, we can put $c_T = c'_T+\hat{l}_T$, and so

$$(c_T \neq 0; \wedge l \neq 0; \rightarrow \langle c', l\rangle = \langle c'_T, l_T\rangle)\ \square$$
$$(l = 0; \rightarrow \langle c', l\rangle = \langle c_T, 0\rangle),$$

or

$$\langle c', l\rangle = (c_T \neq 0; \wedge l \neq 0; \rightarrow \langle c'_T, l_T\rangle)\ \square\ (l = 0; \rightarrow \langle c_T, 0\rangle).$$

The conditions on l have now lost their significance, since they are satisfied by both possible choices. If we put

$$c_1 = c'_T+\hat{}l_T+\hat{w},\quad c_2 = c_T+\hat{\hat{w}},$$

we find that $c = c'+\hat{}l+\hat{w}$ has to satisfy

$$c = (c_T \neq 0; \rightarrow c_1)\ \square\ c_2.$$

Since c has to satisfy $\uparrow/len*c \leqslant$ MAX, the first choice is open only if, moreover, $len\ l_T+\hat{w} \leqslant$ MAX, and the second one if $len\ \hat{w} = \#w \leqslant$ MAX. The remaining indeterminacy has to be resolved using the minimality of *lpos c*. If both choices are still open, c_1 has to be chosen, since

$$lpos\ c_1 = \langle \#c'_T, len\ l_T+\hat{w}\rangle < \langle 1+\#c'_T, len\ \hat{w}\rangle =$$
$$\langle \#c_T, len\ \hat{w}\rangle = lpos\ c_2.$$

The choice is now determinate, and $c = c_T \# w$, where $\#$ is defined by

$$0 \# w \Leftarrow \#w; \leqslant \text{MAX}; \rightarrow \hat{\hat{w}};$$
$$c'_T+\hat{l}_T; \# w \Leftarrow (len\ l_T+\hat{w}; \leqslant \text{MAX}; \rightarrow c_1)\ \square$$
$$(len\ l_T+\hat{w}; > \text{MAX}; \wedge (\#w; \leqslant \text{MAX}); \rightarrow c_2).$$

It has to be verified next that $Trnc\ r \sim Trnc\ c$. In the first place,

$$+/Trnc\ c = +/Trnc\ c' +\hat{} l +\hat{} w =$$
$$+/(l \neq 0 ; \rightarrow c' +\hat{} l) \,[]\, (l = 0 ; \rightarrow c') =$$
$$(l \neq 0 ; \rightarrow +/c' +\hat{} l) \,[]\, (l = 0 ; \rightarrow +/c') = +/c' ; +l = r' =$$
$$Trnc\ r' +\hat{} w = Trnc\ r .$$

It is intuitively obvious that erasing words cannot increase line lengths, so that $\uparrow/len * c \leqslant \text{MAX}$ implies $\uparrow/len * Trnc\ c \leqslant \text{MAX}$. However, we will derive this also formally, just to show how this is done. We reinstate—temporarily—$len\ 0 = 0$. Then

$$\uparrow/len * c' +\hat{} 0 = \uparrow/(len * c') +\hat{} len\ 0 = \uparrow/(len * c') +\hat{} 0 =$$
$$\uparrow/len * c' ; \uparrow\uparrow/0 = \uparrow/len * c' ; \uparrow 0 = \uparrow/len * c' ; \uparrow\uparrow/0 = \uparrow/len * c' .$$

So

$$\uparrow/len * c = \uparrow/len * c' +\hat{} l +\hat{} w = \uparrow/(len * c') +\hat{} len\ l +\hat{} w =$$
$$\uparrow/len * c' ; \uparrow len\ l +\hat{} w \geqslant \uparrow/len * c' ; \uparrow len\ l =$$
$$\uparrow/(len * c' ; +\hat{} len\ l) = \uparrow/len * c' +\hat{} l =$$
$$(l \neq 0 ; \rightarrow \uparrow/len * c' +\hat{} l) \,[]\, (l = 0 ; \rightarrow \uparrow/len * c' +\hat{} 0) =$$
$$(l \neq 0 ; \rightarrow \uparrow/len * c' +\hat{} l) \,[]\, (l = 0 ; \rightarrow \uparrow/len * c') =$$
$$\uparrow/len * (l \neq 0 ; \rightarrow c' +\hat{} l) \,[]\, (l = 0 ; \rightarrow c') = \uparrow/len * Trnc\ c .$$

We have now $Trnc\ r \sim Trnc\ c$.

Finally, it must be shown that replacing $Trnc\ c$ in $c = Trnc\ c ; +\!\!+ w$ by an arbitrary realization of $f\ Trnc\ r$ does no harm to the minimality of $lpos\ c$. (The verification that the result still satisfies $r \sim c$ is straightforward and is omitted here.) If $Trnc\ r = 0$, there is no choice but taking $c = 0 +\!\!+ w$. Otherwise, putting $c = c_T +\!\!+ w = c'_T +\hat{} l_T ; +\!\!+ w$, we have

$$lpos\ c = lpos\ c'_T +\hat{} l_T ; +\!\!+ w =$$
$$lpos\ (len\ l_T +\hat{} w ; \leqslant \text{MAX} ; \rightarrow c_1) \,[]\, (len\ l_T +\hat{} w ; > \text{MAX} ; \rightarrow c_2) =$$
$$(len\ l_T +\hat{} w ; \leqslant \text{MAX} ; \rightarrow lpos\ c_1) \,[]\, (len\ l_T +\hat{} w ; > \text{MAX} ; \rightarrow lpos\ c_2) .$$

If we define

$$\langle m, n \rangle = lpos\ c_T ,$$

we find $\#c'_T = m$ and $len\ l_T = n$. Then

$$len\ l_T +\hat{} w = len\ l_T ; +1 +\#w = n +1 +\#w,$$

and so

$$lpos\ c_1 = \langle \#c'_T, len\ l_T +\hat{} w \rangle = \langle m, n +1 +\#w \rangle ;$$
$$lpos\ c_2 = \langle \#c_T, len\ \hat{w} \rangle = \langle \#c'_T +\hat{} l_T, len\ \hat{w} \rangle =$$
$$\langle \#c'_T ; +1, len\ \hat{w} \rangle = \langle m +1, \#w \rangle .$$

We can now simplify the expression for $lpos\ c$ to

$$(n +1 +\#w ; \leqslant \text{MAX} ; \rightarrow \langle m, n +1 +\#w \rangle) \,[]$$
$$(n +1 +\#w ; > \text{MAX} ; \rightarrow \langle m +1, \#w \rangle).$$

This expression is non-strictly monotonic in $\langle m, n\rangle = lpos\, c_T$, so taking c_T to be a realization of $f\,Trnc\,r$, which minimizes *lpos*, guarantees that $lpos\, c$ is minimized too. Summing up, we have

$$f0 = 0;$$

$$fr +\!\!+ \hat{w} = Trnc\, fr +\!\!+ \hat{w};\ +\!\!+ w \Rightarrow f\,Trnc\, r +\!\!+ \hat{w};\ +\!\!+ w = fr;\ +\!\!+ w.$$

After these lengthy preparations (but remember that most of the derivations were aimed at exhibiting obvious facts), we can now formulate an "implementation" of f:

$$ff\,0 \Leftarrow 0;$$

$$ff\, r +\!\!+ \hat{w} \Leftarrow ff\, r;\ +\!\!+ w.$$

This function satisfies $f \Rightarrow ff$ and it preserves the definedness of f; i.e., if $fr \neq \square/0$, then $ff\, r \neq \square/0$. The standard technique of recursion elimination gives the obvious iterative "eager" algorithm. Note also that $fr = \square/0$ implies $ff\, r = \square/0$. This is a consequence of $f \Rightarrow ff$, since then $\square/0 \Rightarrow fr \Rightarrow ff\, r \Rightarrow \square/0$. It is easy to define a total variant of ff by making $+\!\!+$ total, e.g. by removing the conditions "$\#w;\ \leqslant$ MAX" from its definition.

Some final remarks to this example: The length of the derivation is mainly due to the small steps taken, but also to some degree to the presentation, which emphasized the algorithmic analysis and synthesis. If one were to "guess" the definition of ff, then the verification is somewhat shorter. Note, in particular, that the need to handle $\mathbb{U}$ did not arise.

The final development phase was an example of "Formal Differentiation" (or "Finite Differencing") (PAIGE[23], PAIGE and KOENIG[24]). This term stands for a widely applicable technique for improving algorithms. It is of special interest here because it is often especially fit to the improvement of high-level algorithms that have been (semi-)automatically synthesized. The essential idea is that of "incremental" computation. Let x' be the result of applying a "small" variation to x. For many functions f, it is more efficient to compute the value of fx' from the result of fx and the variation, than to compute it afresh. It can be seen that this is a special case of the "Divide and Rule" paradigm. If x is the result of sequentially making small variations, then fx can also be computed sequentially. A challenging problem, not addressed here, is to develop general algebraic techniques for *deriving* expressions for "formal derivatives". For a not very general but interesting algebraic technique, see SHARIR[26].

The eager strategy (also known as "greedy" strategy) is a special case of formal differentiation in the context of optimization problems. A higher-level derivation would have run, schematically: (i) show that f satisfies the conditions of some "eagerness" theorem; (ii) apply the theorem to give ff as implementation. There appears to be a relationship with matroid theory here (KORTE and LOVÁSZ[16]). It remains to be investigated if this can be expressed conveniently in the framework pursued here. If so, it would be a good example of the "higher-level" theorems aimed at. A different choice for

the objective function (e.g., minimize the sum of the squares of the white space on each line) would have invalidated its applicability. Still, an important gain in efficiency is possible for many other objective functions (e.g., for the least-squares objective), namely by applying the technique of dynamic programming. An algebraic approach to this technique can be found in CUNINGHAME-GREEN [7], and a specific application of this approach in an algorithmic development in MEERTENS and VAN VLIET [20].

12. SECOND EXAMPLE: THE AMOEBA FIGHT SHOW

The following problem is of interest because it is the first problem that I tried to tackle algebraically without already knowing a reasonable algorithm for it—or seeing one immediately. It was passed on to me by Richard Bird. Its origin is, as far as I know, a qualifying exam question from CMU. Since I do not know the original formulation of the problem, it is given here in a setting of my own devising.

What with the rising prices of poultry, a certain showman has modernized his *Amazing Life-and-Death Rooster Fight Show*, and replaced his run of prize-fighting cocks by a barrel of cannibalistic amoebae. As is well known, amoebae have an engrossing way of tackling an opponent: it is simply swallowed, hide and hair.[1] It follows from the Law of Conservation of Mass that the weight of the winner then increases by that of the loser. Each show stages a tournament between n amoebae (where n is some positive natural number), consisting of a sequence of $n-1$ duels (two amoebae staged against each other). At the end of the tournament, all that remains is the final victor (although it encompasses, in some sense, all losers). The showman wishes to maximize the throughput of his enterprise by minimizing the time taken by one show. The time needed for a single duel, he has found experimentally, is proportional to the weight of the lighter contestant (about one minute for each picogram). At the start of a show, the amoebae are lined up in a microscopic furrow. Each two adjacent fighters are kept apart by a removable partition. (This set-up has been chosen thus because of limitations in the state of the art of micro-manipulation. For similar reasons, the initial arrangement cannot be controlled.) Each time a partition is removed, the two amoebae now confronting each other engage in a life-and-death duel.

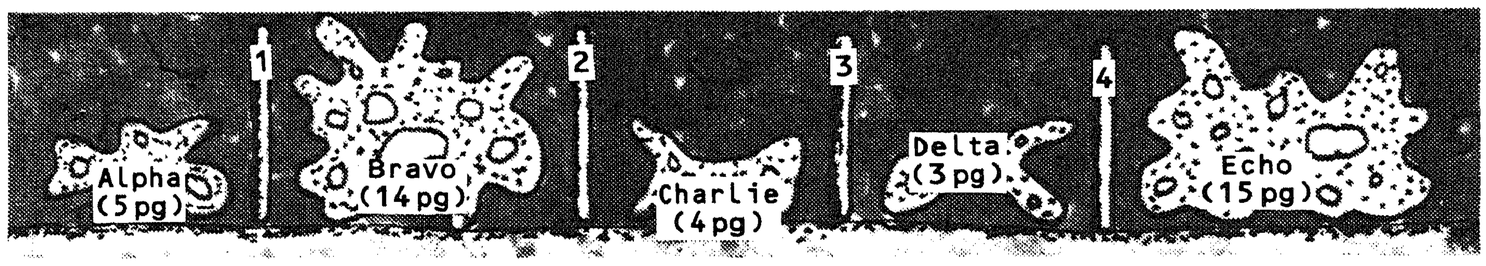


FIGURE 3. Five amoebae lined up before the tournament (magnification: 500 ×)

1. For amoebae, this terminology is not entirely appropriate. The hapless victim is, in fact, engulfed by the attacker's bulging around and completely enveloping it, *membrane* and *pseudopodia*.

The showman thinks the best strategy is to have, each time, the lightest amoeba fight against its heaviest neighbour. His assistant suspects that it is better to choose the pair whose weight difference is largest. In the situation sketched in figure 3, these two strategies give rise to the same sequence of duels. First, the showman removes partition 4, and Delta and Echo fight. After 3 minutes, Echo has consumed Delta. Next, partition 3 is lifted, and Charlie enters the arena against Echo. The unequal battle takes 4 more minutes. Echo weighs now, after having feasted on Delta and Charlie, $15+3+4 = 22$ picograms. The next step is the removal of partition 1. It takes Bravo 5 minutes to gobble up Alpha. When the last partition is taken away, the battle of the champions starts. In spite of Bravo's putting up a heroic resistance, pseudopod after pseudopod wraps around its body, and after 19 exciting minutes the last visible part disappears into Echo's innards. The whole tournament has taken $3+4+5+19 = 31$ minutes. Unaware of the fact that a different sequence of duels would have required less than half an hour, the showman and his assistant start clearing the house for the next show.

Let us see if we can do better. The process of amoeba fusion in a tournament creates a tree structure on top of the original sequence of amoebae. For the example, that tree is $\hat{A}+\hat{B}; +\hat{C}+\hat{D}+\hat{E}$, where A stands for Alpha, etc. Each node corresponds to a sub-tournament. Since the structure of the tree gives sufficient information to determine the tournament, even if the elements are not amoebae, it is simplest to work directly with the sequence of the *weights* of the amoebae. Let $w\,t$, for a given tournament tree t, stand for the final weight of the champion of t, $d\,t$ for its duration, and $wd\,t$ for the pair $\langle w\,t,\ d\,t\rangle$. For the trivial case of a one-amoeba "tournament" we have

$$wd\,\hat{w} = wd_0\,w \Leftarrow \langle w,\ 0\rangle .$$

Then we find

$$wd\,t_L+t_R = wd\,t_L;\times wd\,t_R ,$$

where the operation $\times$ is given by

$$\langle w_L ,\ d_L\rangle \times \langle w_R ,\ d_R\rangle \Leftarrow \langle w_L+w_R ,\ d_L+d_R+w_L\downarrow w_R\rangle .$$

(The operation $\times$ is commutative, but, of course, not associative.) So, by the homomorphism lemma, we can express wd by

$$wd = \times/\,wd_0\ast .$$

The function d can be re-defined as $\pi_2\ wd$. If $T\,s$ is the set of all possible tournament trees that can be put on top of an initial configuration s, the problem can be specified as: Determine $\downarrow_d/\ T\,s$. The property characterizing a member t of $T\,s$ is $s = +/\hat{}\ast t$, in which the inserted operation $+$ introduces associativity. Then

$$T\,s \Leftarrow (s = +/\hat{}\ast)\colon \mathbb{U} .$$

It would be possible, of course, to develop an algorithm for determining T, after which we would have an algorithm for the whole problem. But

obvious that any tournament with this property is optimal. The step from here to a linear-time algorithm is simple, if not trivial. One possible algorithmic formulation is

$$\downarrow_d / \; T\, s \Rightarrow t\, s\,,$$

where t is defined recursively by

$$t\, \hat{w} \Leftarrow \hat{w}\,;$$

$$t\, \hat{w}_1 + s' + \hat{w}_n \Leftarrow (w_1 \leqslant m_R;\ \rightarrow \hat{w}_1 + t\, R) \,[]\, (w_n \leqslant m_L;\ \rightarrow t\, L;\ + \hat{w}_n),$$
$$\text{where } L = \hat{w}_1 + s',\ \ m_L = \uparrow / L,\ \ R = s' + \hat{w}_n,\ \ m_R = \uparrow / R.$$

The correctness follows directly from the preceding proof, since it has been shown that $d\, t\, s = \downarrow / d * T\, s$.

Our showman is probably more interested in a simple method that tells him when to lift which partition, than in determining a tree. It should be obvious that we can advise him to remove, each time, any partition keeping the heaviest amoeba apart from a neighbour. It is not hard to derive this formally from the given expression for t.

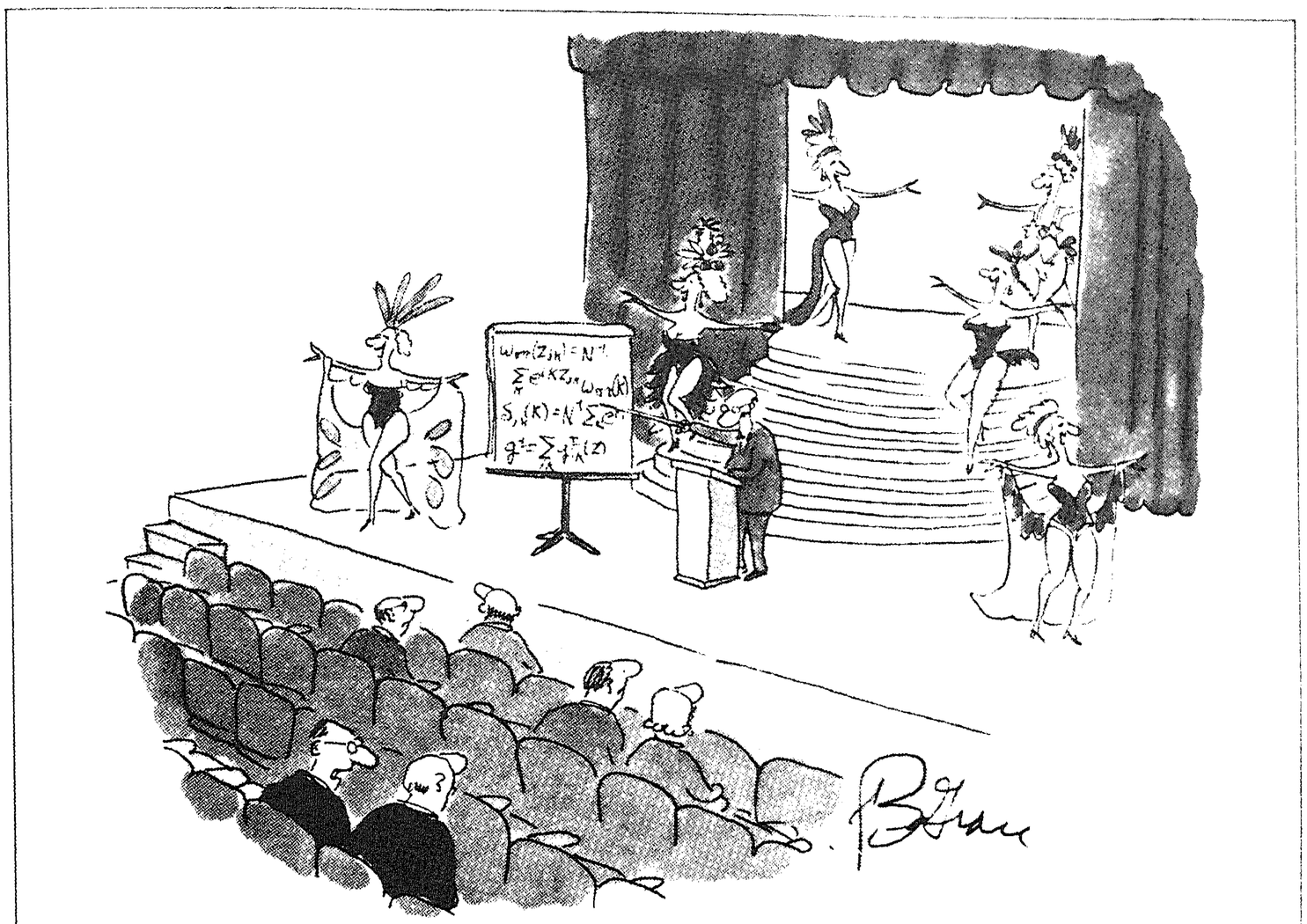
13. Conclusion

An attempt has been made here to convince the reader that the ideal of a discipline of "Algorithmics" can be realized. If the account was possibly unconvincing, then, I suspect, a major culprit is perhaps the shock of being exposed to a set of unfamiliar squiggles. In my first endeavours, exploring the suggestions of Bird [4], I found that the only way to proceed was to translate the formulae continually into familiar "operational" concepts. Now, after having played with these notations for some time, I find myself applying transformations without being conscious of an operational meaning. The reader is invited to try and undergo the same experience. A good starting point is to derive

$$\#P : +/ \;=\; +/ +/ (\iota \ll * P\,?)\ **.$$

This is a meaningful and useful transformation; the two formulae are readily translated into "pidgin Algol", and the resulting programs are each about 10 lines long.

Much work has to be done to develop the current set of concepts and notations beyond the initial attempts presented here. Important points are the discovery and formulation of "algebraic" versions of higher-level programming paradigms and strategies, and the development of techniques to assess something like the concrete "complexity" of an expression in the absence of an operational model in which time and space are meaningful notions. Other issues to be investigated are the introduction of infinite objects, of ways to express some form of concurrency, and of suitable notations for handling algebraically more complex structures than the ones dealt with here.



"Personally, I think he's trying to cover up a shaky theory."

ACKNOWLEDGEMENTS

The cartoon by Bud Grace, Copyright © 1984, B. Grace, is reprinted here by the kind permission of the artist. I am indebted to Steven Pemberton of CWI and to Norman Shulman of NYU for scrutinizing earlier versions and suggesting many improvements.

REFERENCES

1. J. BACKUS (1978). Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *Comm. ACM 21,* 613–641.
2. F. L. BAUER *et al.* (1981). Programming in a wide-spectrum language: a collection of examples. *Science of Computer Programming 1,* 73–114.
3. R. S. BIRD (1977). Improving programs by the introduction of recursion. *Comm. ACM 20,* 151–155.
4. R. S. BIRD (1981). *Some Notational Suggestions for Transformational Programming.* WG 2.1 working paper NIJ-3 (unpublished).
5. H. J. BOOM (1981). *Further Thoughts on Abstracto.* WG 2.1 working paper ELC-9 (unpublished).
6. R. M. BURSTALL, J. DARLINGTON (1977). A transformation system for developing recursive programs. *J. ACM 24,* 44–67.
7. R. CUNINGHAME-GREEN (1979). *Minimax Algebra. Lecture Notes in Economics & Mathematical Systems 166,* Springer, Berlin, 1979.
8. E. W. DIJKSTRA (1968). A constructive approach to the problem of program correctness. *BIT 8,* 174–186.

9. E. W. DIJKSTRA (1971). Notes on structured programming. O.-J. DAHL, E. W. DIJKSTRA, C. A. R. HOARE. *Structured Programming,* Academic Press.
10. R. W. FLOYD (1967). Assigning meanings to programs. J. T. SCHWARTZ (ed.). *Proc. Symp. Appl. Math., Vol. 19, Mathematical Aspects of Comp. Science* 19–32, AMS, Providence, RI.
11. L. GEURTS, L. MEERTENS (1978). Remarks on Abstracto. *ALGOL Bull. 42,* 56–63.
12. J. GUTTAG, J. HORNING, J. WILLIAMS (1981). FP with data abstraction and strong typing. *Proc. 1981 Conf. on Functional Programming Languages and Computer Architecture* 11–24, ACM.
13. F. W. VON HENKE (1976). An algebraic approach to data types, program verification, and program synthesis. *Proc. Math. Foundations of Comp. Science '76, Lecture Notes in Comp. Science 45,* 330–336, Springer, Berlin.
14. C. A. R. HOARE (1969). An axiomatic basis for programming language constructs. *Comm. ACM 12,* 576–580.
15. M. A. JACKSON (1975). *Principles of Program Design.* A.P.I.C. Studies in Data Processing 12, Academic Press.
16. B. KORTE, L. LOVÁSZ (1981). Mathematical structures underlying greedy algorithms. F. GÉCSEG (ed.). *Fundamentals of Computation Theory, Lecture Notes in Comp. Science 117,* 205–209, Springer, Berlin.
17. J. MCCARTHY (1963). A basis for a mathematical theory of computation. P. BRAFFORT, D. HIRSCHBERG (eds.). *Computer Programming and Formal Systems* 33–70, North-Holland.
18. L. MEERTENS (1977). From abstract variable to concrete representation. S. A. SCHUMAN (ed.). *New Directions in Algorithmic Languages 1976* 107–133, IRIA, Rocquencourt.
19. L. MEERTENS (1979). Abstracto 84: the next generation. *Proc. of the 1979 Annual Conf.* 33–39, ACM.
20. L. MEERTENS, J.C. VAN VLIET (1976). Repairing the parenthesis skeleton of ALGOL 68 programs: proof of correctness. G. E. HEDRICK (ed.). *Proc. of the 1975 Int. Conf. on ALGOL 68* 99–117, Oklahoma State University, Stillwater.
21. P. NAUR (1966). Proof of algorithms by general snapshots. *BIT 6,* 310–316.
22. P. NAUR (1969). Programming by action clusters. *BIT 9,* 250–258.
23. R. PAIGE (1981). *Formal Differentiation.* UMI Research Press.
24. R. PAIGE, S. KOENIG (1982). Finite differencing of computable expressions. *ACM Trans. on Programming Languages and Systems 4,* 402–454.
25. A. RALSTON, M. SHAW (1980). Curriculum '78—Is Computer Science really that unmathematical? *Comm. ACM 23,* 67–70.
26. M. SHARIR (1982). Some observations concerning formal differentiation of set theoretic expressions. *ACM Trans. on Programming Languages and Systems 4,* 196–225.
27. N. WIRTH (1971). Program development by stepwise refinement. *Comm. ACM 14,* 221–227.
28. N. WIRTH (1973). *Systematic Programming.* Prentice-Hall.

Uniform Asymptotic Expansions of Integrals

N.M. Temme
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

The purpose of the paper is to give an account of several aspects of uniform asymptotic expansions of integrals. We give examples of standard forms, the role of critical points and methods to construct the expansions.

1. Introduction

Asymptotic expansions of integrals is an important topic of classical analysis. Many results are available for the well-known higher transcendental functions of mathematical physics and probability theory, and for integrals occurring as solutions of physical problems.

Here we are concerned with uniform expansions of integrals of the type

$$I(z) = \int_C f(t)e^{-z\phi(t,\alpha)}dt \tag{1.1}$$

where C is a contour in the complex t-plane and z is a large parameter. Note that the value of $I(z)$ depends on the parameter α. We suppose that for certain values of α the asymptotic behaviour of $I(z)$ will change.

For obtaining uniform expansions the following major steps can be distinguished:

(i) trace the points on C or near C that significantly contribute to $I(z)$;
(ii) transform the integral into a standard form;
(iii) construct a formal uniform expansion;
(iv) investigate the asymptotic properties of the expansion;
(v) construct error bounds;
(vi) extend the results to wider domains of the parameters.

The first three are most frequently the only possibilities to investigate in practical problems. In applications this formal approach is usually accepted. Often the contributions in the expansion have a physical meaning and then just the form of the expansion is the ultimate requirement. In a systematic study of uniform asymptotic expansions the remaining steps should be incorporated. Also, in numerical applications efficient error bounds are particularly

important and in this area point (v) cannot be forgotten.

The above points are not the only problems to be investigated. Several problems arising in physics (for instance in optics and in scattering theory) yield integrals which are generalizations of Airy-type integrals. Then the approximants are higher transcendental functions which fall outside the classical ones. The computational problems for these generalizations are not easy to solve.

In this paper we discuss several aspects of the steps enumerated above. We give definitions of asymptotic expansions, we consider critical points and various methods and techniques to construct the coefficients and, for some cases, error bounds. Several unsolved problems are mentioned.

A standard reference work for asymptotic expansions is OLVER [11], also for special functions; see also OLVER [12] for uniform expansions for special functions. WONG [27] gives a survey with recent results on error bounds for asymptotic expansions of integrals.

This paper is concerned with the classical aspects of asymptotic analysis. Recently new investigations of integrals have been initiated by MASLOV and HÖRMANDER, see DUISTERMAAT [6]. Uniformity problems are cast into the theory of unfoldings of singularities. This approach falls outside the scope of the present publication. An introduction to Maslov's work can be found in POSTON, STEWART [13].

2. DEFINITIONS OF ASYMPTOTIC EXPANSIONS

We use the terminology of generalized asymptotic expansions. First we introduce the concept of asymptotic scale:

a sequence of functions $\{\phi_n(x)\}$ is called an *asymptotic sequence* or *scale* when $\phi_{n+1}(x)=o[\phi_n(x)]$ as $x\to\infty$.

Then we have the definition:

the formal series $\Sigma_{n=0}^{\infty} f_n(x)$ is said to be an *asymptotic expansion* of $f(x)$ *with respect to the scale* $\{\phi_n\}$ if

$$f(x)-\sum_{n=0}^{N} f_n(x) = o[\phi_N(x)] \quad as \quad x\to\infty, \quad N=0,1,\ldots; \tag{2.1}$$

in this case we write

$$f(x)\sim\sum_{n=0}^{\infty} f_n(x); \quad \{\phi_n(x)\} \quad as \quad x\to\infty.$$

In uniform expansions it is required that the 'o' sign holds uniformly (with respect to $\alpha\in A$, say). This general set up is extensively described in ERDÉLYI, WYMAN [7].

When $f_n=\phi_n$ we have a Poincaré type asymptotic expansion; when $f_n=\phi_n=x^{-n}$ we obtain the definition of Poincaré and Stieltjes, who both introduced the definition of this kind in 1886.

Observe that in (2.1) no requirements are put on $\{f_n\}$: it need not be an asymptotic scale. Rather useless expansions may arise (from an asymptotical

point of view) when it is not. Also, we can take the scale too rough to measure the error in (2.1).

EXAMPLE 2.1. Take $f_n(x)=(x+n)^{-2}$, and $\phi_n(x)=\log^{-n}x$, $x>1$, $n=0,1,2,\ldots$. Then we have

$$\sum_{n=N+1}^{\infty}(x+n)^{-2} = \mathcal{O}(x^{-1})=o[\phi_m(x)] \quad as \quad x\to\infty$$

for all N,m. So we can write

$$f(x)\sim\sum_{n=0}^{\infty}(x+n)^{-2};\ \{\log^{-n}x\} \quad as \quad x\to\infty$$

where for f we can take the convergent sum, which represents $d^2 ln\,\Gamma(x)/dx^2$ (Γ is the Euler gamma function).

Some expansions are provided with a 'thin' scale in which successive terms become more and more indistinguishable. The following example is in WIMP [24], a survey on uniform scale functions and asymptotic expansion of integrals.

EXAMPLE 2.2. The coefficients a_n of the expansion $\Gamma(1-t)=\Sigma_{n=0}^{\infty}a_n t^n$, $|t|<1$, satisfy the expansion

$$a_n\sim\sum_{k=0}^{\infty}\frac{(-1)^k}{k!(k+1)^{n+1}}\ ;\ \{\phi_k(n)\} \quad as \quad n\to\infty,$$

where $\phi_k(n)=(k+1)^{-n}$. The series converges rather fast. However, the scale satisfies $\phi_{k-1}(n)/\phi_k(n)=(1+1/k)^{-n}$, which indeed is $o(1)$ as $n\to\infty$, $k\geqslant 1$, k fixed. But as k increases this ratio tends to unity (n fixed).

For some functions we need a *compound asymptotic expansion.* That is we have a decomposition

$$\begin{aligned} f(x) &= A_1(x)f_1(x)+\ldots+A_k(x)f_k(x) \\ f_k(x)&\sim\sum_{j=0}^{\infty}f_{jk}(x);\quad \{\phi_{jk}\} \quad as \quad x\to\infty, \end{aligned} \tag{2.2}$$

where, for each k, $\{\phi_{jk}(x)\}$ is an asymptotic scale. In complicated problems the f_k are not known a priori.

It may be rather difficult to investigate whether an expansion is uniform with respect to a parameter α. A non-uniformity may be recognized when in (2.2) $A_k(x)$, $f_{jk}(x)$ or $\phi_{jk}(x)$ are singular at certain values of the uniformity parameter α, whereas $f(x)$ remains regular for these values.

EXAMPLE 2.3. Consider the exponential polynomial in the form

$$e_n(x) = e^{-x}\sum_{s=0}^{n}\frac{x^s}{s!}, \quad x>0, \quad n=0,1,2,\ldots.$$

We have $\lim_{x\to\infty}e_n(x)=0$, $\lim_{n\to\infty}e_n(x)=1$; so the first limit cannot be

uniformly valid when n grows with x. Asymptotic expansions for large n, which are uniform with respect to unrestricted real values of x can be given in terms of error functions. Any approximation in terms of elementary functions breaks down when x passes the value n, which is not a singularity for $e_n(x)$. The function $e_n(x)$ is related to the incomplete gamma functions and to the Poisson distribution. For more information we refer to WONG [25], TRICOMI [21], and [19].

3. CRITICAL POINTS

There is a systematic approach to obtain the asymptotic expansion of (1.1). We have to look for certain distinguished points whose immediate neighbourhoods determine completely the asymptotic behaviour of the integral. Such points are called *critical points* by VAN DER CORPUT [5]. Possible candidates are:

- the end points of the contour;
- singular points of the integrand;
- stationary or saddle points of ϕ (i.e., where $\partial\phi/\partial z$ vanishes).

The contribution of a single critical point to the asymptotic value of $I(z)$ is known for a great variety of critical points. We mention some key words in this respect: Watson's lemma, the method of Laplace, the method of steepest descent, the method of saddle points, the principle of stationary phase and the method of Darboux. We give a formulation of one of the most important tools.

LEMMA (WATSON). *Consider the Laplace integral*

$$I(z) = \int_0^\infty e^{-zt} f(t)dt. \tag{3.1}$$

Assume that

(i) *f is locally integrable on $[0, \infty)$;*

(ii) *$f(t) \sim \Sigma_{s=0}^{\infty} a_s t^{(s+\lambda-\mu)/\mu}$ as $t \to 0^+$, μ, λ fixed, $\mu > 0$, $\mathrm{Re}\lambda > 0$;*

(iii) *the abscissa of convergence of (3.1) is not $+\infty$.*

Then,

$$I(z) \sim \sum_{s=0}^{\infty} \Gamma\left(\frac{s+\lambda}{\mu}\right) a_s z^{-(s+\lambda)/\mu} \tag{3.2}$$

as $z \to \infty$ in the sector $|\arg z| \leqslant \frac{1}{2}\pi - \delta (< \frac{1}{2}\pi)$ where $z^{(s+\lambda)/\mu}$ has its principal value.

PROOF. See OLVER [11, p. 113]. □

Oberve that (3.2) is obtained by substituting (ii) into (3.1) and by interchanging the order of summation and integration. In (ii) λ and μ are fixed. When

$\lambda = \mathcal{O}(z)$ (or larger) the expansion (3.2) has no meaning. A modification of the lemma is needed then to give a uniform expansion, see [20].

When in (1.1) α ranges over a domain A the critical points may be variable. For certain values in A two or more critical points may coalesce. Usually, the form of the expansion changes and it is unlikely that the sum of the contributions of each critical point will be uniformly valid. For instance, coefficients of the several expansions may become singular when α takes these distinguished values.

The systematic approach of van der Corput to add several contributions from the critical points was an important step to take away part of the mystery of asymptotics. In uniform problems it is also important to systematize. We can single out the following possibilities for (1.1):

- singularity coincides with stationary point;
- end-point of contour coincides with stationary point;
- two stationary points coincide.

In VAN DER WAERDEN [23], CHESTER, FRIEDMAN, URSELL [4] and BLEISTEIN [1] important contributions are given for these cases.

By introducing several auxiliary parameters much more situations can occur. Some of them correspond with important physical applications or with problems for the well-known special functions of mathematical physics. A survey is given by OLVER [12].

The approximants in uniform expansions are usually more complicated than the elementary functions used in earlier days. Now we use error functions, Airy functions, Bessel functions, parabolic cylinder functions, etc. The computational problem has been solved for most of these functions, and now they are accepted as approximants.

In classifying relevant cases of coalescing critical points it is instructive to look at approaches via the WKB or Liouville-Green methods for differential equations. Most functions from mathematical physics can be investigated in both directions: they have an integral representation and they satisfy a differential equation. See again [12] for more details on this point.

4. EXAMPLES OF STANDARD FORMS

In the table we give standard forms of integrals for which well-known special functions are used as approximants. We give the critical points, the coalescence of which causes uniformity problems, and references to the literature.

	Standard form	Approximant	Critical points	References
(4.1)	$\int_{-\infty}^{\infty} e^{-zt^2} \frac{f(t)}{t-i\alpha} dt$	Error function	$t=0,\ t=i\alpha$	[23]
(4.2)	$\int_{-\infty}^{\infty} e^{-zt^2} \frac{f(t)}{(t-i\alpha)^{\mu}} dt$	Parabolic cylinder function	$t=0,\ t=i\alpha$	[1]
(4.3)	$\int_{-\infty}^{\alpha} e^{-zt^2} f(t) dt$	Error function	$t=0,\ t=\alpha$	[19]
(4.4)	$\int_{0}^{\infty} t^{\beta-1} e^{-z(\frac{1}{2}t^2-\alpha t)} f(t) dt$	Parabolic cylinder function	$t=0,\ t=\alpha$	[1], [8] [14], [25]
(4.5)	$\int_{c} e^{z(\frac{1}{3}t^3-\alpha t)} f(t) dt$	Airy function	$t=\pm\sqrt{\alpha}$	[4], [10]
(4.6)	$\int_{0}^{\infty} t^{\alpha-1} e^{-zt} f(t) dt$	Gamma function	$t=0,\ t=\alpha/z$	[20]
(4.7)	$\int_{\alpha}^{\infty} t^{\beta-1} e^{-zt} f(t) dt$	Incomplete gamma function	$t=0,\ t=\alpha$	[9], [15] [17], [28]
(4.8)	$\int_{0}^{\infty} t^{\beta-1} e^{-z(t+\alpha/t)} f(t) dt$	Bessel function	$t=0,\ t=\pm\sqrt{\alpha}$	[18]
(4.9)	$\int_{\alpha}^{\infty} f(\sqrt{t^2-\alpha^2}) \sin zt\, dt$	Bessel function	$t=\pm\alpha$	[22], [26]
(4.10)	$\int_{0}^{\infty} \frac{\sin z(t-\alpha)}{t-\alpha} f(t) dt$	Sine integral	$t=0,\ t=\alpha$	[29]

REMARKS

1. Functions f are supposed to be regular in neighbourhoods of the critical points.
2. The integrals reduce to their approximants when $f=1$, except in (4.9) where it occurs for $f(t)=t^{\beta}$.
3. Quite different integrals may have the same approximants.
4. Different intervals of integration are investigated too.
5. In (4.5), (4.8) two saddle points coalesce with each other when $\alpha=0$; both cases are different, however. In (4.8) we have an additional critical point at $t=0$ (end point and singularity).

6. In all cases elementary approximants can be used for fixed values of the uniformity parameter α.
7. Several of the examples need further investigations with respect to the construction of error bounds and the determination of maximal regions of validity.

5. Transformation to standard forms

Once the critical points are located and the asymptotic phenomena are recognized, a next step may be a transformation to one of the standard forms. To obtain an optimal representation, such a transformation may be rather complicated. As a consequence, it may cause serious problems for the construction of error bounds and for the computation of the coefficients. In this section we consider two examples. The first one (on incomplete gamma functions) is relatively simple; the second one is more difficult to investigate due to the role of the uniformity parameter.

5.1 Incomplete gamma functions

These are defined by

$$P(a,x) = \frac{1}{\Gamma(a)}\int_0^x t^{a-1}e^{-t}dt, \quad Q(a,x) = \frac{1}{\Gamma(a)}\int_x^\infty t^{a-1}e^{-t}dt. \tag{5.1}$$

We consider positive values of x and a. The function $e_n(x)$ of Example 2.3 is a special case: $e_n(x) = Q(n+1,x)$. We are interested in the asymptotic expansion which is valid for $a\to\infty$ and $x\in[0,\infty)$ (uniformly). The function $t^a e^{-t}$ attains its maximal value at $t=a$. When x and a are nearly equal this point is close to the end points of the intervals of integration in (5.1). Hence, we recognize (4.3). We rewrite $P(a,x)$ in the form

$$P(a,x) = \frac{e^{-a}a^a}{\Gamma(a)}\int_0^{x/a} e^{-a(t-1-\ln t)}dt.$$

The transformation into the standard form is defined by the mapping $\zeta:\mathbb{R}^+\to\mathbb{R}$, given by

$$\tfrac{1}{2}\zeta^2(t) = t-1-\ln t, \quad \operatorname{sign}\zeta(t) = \operatorname{sign}(t-1). \tag{5.2}$$

The result is

$$P(a,x) = \frac{e^{-a}a^a}{\Gamma(a)}\int_{-\infty}^{\eta} e^{-\frac{1}{2}a\zeta^2} f(\zeta)d\zeta, \tag{5.3}$$

$$Q(a,x) = \frac{e^{-a}a^a}{\Gamma(a)}\int_{\eta}^{\infty} e^{-\frac{1}{2}a\zeta^2} f(\zeta)d\zeta,$$

where $\eta = \zeta(\lambda)$, $\lambda = x/a$, $f(\zeta) = t^{-1}dt/d\zeta = \zeta/(t-1)$, $f(0) = 1$.

By tracing the (complex) singularities of the mapping in (5.2) we can infer that f is analytic in the strip $|\mathrm{Im}\zeta|<\sqrt{2\pi}$. To give the main steps in this analysis we observe that ζ is analytic at $t=1$, but not at $t_n=\exp(2\pi i n)$, $n=\pm1,\ \pm2,\ldots$. (To obtain a sufficiently large ζ-domain we have to consider more than the principal sheet of the Riemann surface of the logarithm in (5.2)). Corresponding ζ-points follow from (5.2): $\frac{1}{2}\zeta_n^2=-2\pi i n$; the points with $n=\pm1$ are nearest to the real line, $|\mathrm{Im}\zeta_{\pm1}|=\sqrt{2\pi}$.

This information is useful for estimating coefficients and remainders in the asymptotic expansions of (5.3). A first approximation to the functions in (5.3) is obtained by replacing $f(t)$ by $f(0)=1$. Then the integrals can be written in terms of the normal probability functions or error functions. In [19] the complete expansion is given, which is uniformly valid with respect to $\eta\in\mathbb{R}$, (or $x\in[0,\infty)$).

5.2 Anger function of large order

A second example is from OLVER [11, p. 352]. The integral is

$$A_{-\nu}(a\nu)=\int_0^\infty e^{-\nu(a\sinh t-t)}dt,\quad a>0,\quad \nu>0. \tag{5.4}$$

$A_\nu(z)$ is a so-called Anger function, which is related to Bessel functions; ν is the large parameter, a is restricted to $(0,1]$, where $a=1$ is a critical value. Write $a=1/\cosh\alpha$. Saddle points in (5.4) are zeros of $d[a\sinh t-t]=\cosh t/\cosh\alpha-1$. When $a\in(0,1)$ two real saddle points are $\pm\alpha$, which coalesce with each other when $a\to1$.

A transformation to the standard form (4.5) is obtained by using the mapping $\zeta:\mathbb{R}\to\mathbb{R}$ that is defined by

$$\sinh t/\cosh\alpha-t=\tfrac{1}{3}\zeta^3-\eta\zeta. \tag{5.5}$$

To make $\zeta(t)$ regular at $t=\pm\alpha$ the only possible choice for η is

$$\tfrac{2}{3}\eta^{\frac{3}{2}}=\alpha-\tanh\alpha. \tag{5.6}$$

So we obtain

$$A_{-\nu}(a\nu)=\int_0^\infty e^{-\nu(\frac{1}{3}\zeta^3-\eta\zeta)}f(\zeta)d\zeta, \tag{5.7}$$

where $f(\zeta)=dt/d\zeta$. The singularities of f arise from complex singular points of $\zeta(t)$. These arise from complex solutions of the equation $\cosh t=\cosh\alpha$, i.e., $t_k^{\pm}=\pm\alpha+2k\pi i$, $k=\pm1,\pm2,\ldots$. Corresponding $\zeta_k^{\pm}$ values follow from (5.5). For small values of a (i.e., large α and η) we have $\zeta_k^{\pm}\sim\sqrt{\eta}+\eta^{-1/4}\sqrt{2\pi i k}$. So, the singularities of $f(\zeta)$ are rather close to the saddle point $\zeta=\sqrt{\eta}$, when η is large.

As OLVER [11] shows this distance is not too small for obtaining an

expansion for (5.7) that is uniformly valid with respect to $a \in [0,1]$, or $\eta \in [0,\infty)$. In Section 6 some more details about the expansion will be given.

REMARK. In both examples (5.3) and (5.7) the asymptotic nature of the expansion follows from the singularities of $f(\zeta)$, where f is considered as an analytic function of the complex variable ζ. This approach is natural for the special functions considered here. In a more general approach, where it may be assumed that f belongs to a function class C^k, the method of proof is quite different, of course.

6. THE CONSTRUCTION OF THE FORMAL EXPANSION

Several methods are available to obtain various kinds of asymptotic expansions. Roughly speaking we have the following three possibilities:

1. Expansions at the critical points;
2. Integration by parts;
3. Residue methods.

The third method is well known in the theory of Laplace and Mellin transformations for obtaining back transforms; see BLEISTEIN, HANDELSMAN [3] for a lot of information on the use of Mellin transforms in asymptotics. It will not be considered here. A new method based on a combination of 1. and 2. is discussed in Section 7.

The expansion at $t = 0^+$ in Watson's lemma (Section 3) is an example of 1. To obtain uniform expansions integration by parts should not be done in a straightforward way. We now demonstrate a method of BLEISTEIN [1] that is very useful in various types of integrals.

6.1.

We consider (4.8) and we write $\mu^2 = \alpha$. Saddle points occur at $t = \pm\mu$, where μ is supposed to be positive. The first step is the representation

$$f(t) = a_0 + b_0 t + (t - \mu^2/t)g(t) \tag{6.1}$$

where a_0, b_0 follow from substitution of $t = \pm\mu$. We have

$$a_0 = \tfrac{1}{2}[f(\mu) + f(-\mu)], \quad b_0 = \frac{1}{2\mu}[f(\mu) - f(-\mu)].$$

Denoting (4.8) by $I(z)$ we obtain upon inserting (6.1) into (4.8)

$$I(z) = a_0\Phi_0 + b_0\Phi_1 + I_1(z) \tag{6.2}$$

where Φ_0, Φ_1 are modified Bessel functions

$$\Phi_j = 2(\mu/\sqrt{z})^{\beta+j} K_{\beta+j}(2\mu\sqrt{z}), \quad j = 0,1.$$

An integration by parts gives

$$I_1(z) = \int_0^\infty t^{\beta-1} e^{-z(t+\mu^2/t)}(t - \mu^2/t)g(t)dt$$

$$= -\frac{1}{z}\int_0^\infty t^\beta g(t)de^{-z(t+\mu^2/t)} = \frac{1}{z}\int_0^\infty t^{\beta-1}e^{-z(t+\mu^2/t)}f_1(t)dt,$$

with $f_1(t)=t^{1-\beta}\frac{d}{dt}[t^\beta g(t)] = \beta g(t)+tg'(t)$. We see that $zI_1(z)$ is of the same form as $I(z)$. The above procedure can now be applied to $zI_1(z)$ and we obtain for

$$I(z) = \int_0^\infty t^{\beta-1}e^{-z(t+\mu^2/t)}f(t)dt, \tag{6.3}$$

the formal expansion

$$I(z) \sim \Phi_0 \sum_{s=0}^{\infty}\frac{a_s}{z^s} + \Phi_1 \sum_{s=0}^{\infty}\frac{b_s}{z^s}, \quad \text{as} \quad z\to\infty, \tag{6.4}$$

where we define inductively $f_0(t)=f(t)$, $g_0(t)=g(t)$ and for $s=1,2,...,$

$$f_s(t) = t^{1-\beta}\frac{d}{dt}[t^\beta g_{s-1}(t)] = a_s + b_s t + (t-\mu^2/t)g_s(t),$$

$$a_s = \tfrac{1}{2}[f_s(\mu)+f_s(-\mu)], \quad b_s = \frac{1}{2\mu}[f_s(\mu)-f_s(-\mu)].$$

6.2.

Next we show that it is rather easy to obtain an expansion in which β acts as a second uniformity parameter. Then we exploit fully the fact that the Bessel functions in Φ_j are functions of two variables. The form of the new expansion is exactly as in (6.4), with the same Φ_j, but with different coefficients.

We write $\beta=2\nu z$, $\nu\in\mathbb{R}$. The saddle points $t_\pm$ are now zeros of $d[t+\mu^2/t-2\nu\ln t]/dt$, which gives $t_\pm=\nu\pm(\nu^2+\mu^2)^{1/2}$. The modification of (6.1) is

$$f(t) = c_0 + d_0 t + (t-2\nu-\mu^2/t)h_0(t)$$

and we obtain for (6.3) the formal expansion

$$J(z) \sim \Phi_0 \sum_{s=0}^{\infty}\frac{c_s}{z^s} + \Phi_1 \sum_{s=0}^{\infty}\frac{d_s}{z^s}, \quad \text{as} \quad z\to\infty. \tag{6.5}$$

Now the coefficients follow from

$$\tilde{f}_0(t) = f(t), \quad \tilde{f}_s(t) = t\frac{d}{dt}h_{s-1}(t) = c_s + d_s t + (t-2\nu-\mu^2/t)h_s(t),$$

$$c_s = \frac{t_+\tilde{f}_s(t_-)-t_-\tilde{f}_s(t_+)}{t_+-t_-}, \quad d_s = \frac{f_s(t_+)-f_s(t_-)}{t_+-t_-}.$$

6.3.

When f of (6.3) is analytic, say in the strip $|\mathrm{Im}\, t|<a$, then each f_s, $\tilde{f}_s$ and hence all coefficients are well defined for all $\nu\in\mathbb{R}$, $\mu\geqslant 0$. Extension to complex values of ν, μ and z is possible when more is known about f.

The expansions (6.4), (6.5) might be applied to the confluent hypergeometric function

$$U(a,b,z) = \frac{1}{\Gamma(a)}\int_0^\infty t^{a-1}(1+t)^{b-a-1}e^{-zt}dt, \tag{6.6}$$

where $\mathrm{Re}\, a>0$, $b\in\mathbb{C}$, $\mathrm{Re}\, z>0$.

In [18] (6.6) was transformed into (6.3) and an expansion was obtained by expanding f at the critical point $t=0$. The range of the parameters was rather limited but we obtained a manageable error bound, which was very useful in a numerical algorithm for (6.6).

Using the procedure for (6.5) we expect to be able to construct for (6.6) an expansion for $a\to\infty$, that is uniformly valid with respect to $z\in[0,\infty)$, $b\in(-\infty,\nu a)$, $\nu<1$, ν fixed. Further research is needed, however, to transform (6.6) into (6.3), to investigate the asymptotic nature of (6.4), and to construct error bounds.

6.4.

In OLVER [11] a uniform expansion of (5.7) is obtained by expanding $f(\zeta)$ at the critical point $\zeta=\sqrt{\eta}$. By writing $f(\zeta)=\Sigma q_s(\zeta-\sqrt{\eta})^s$, the expansion

$$A_{-\nu}(\nu a) \sim \sum_{s=0}^{\infty} q_s \frac{\pi Qi_s(\nu^{\frac{2}{3}}\eta)}{\nu^{(s+1)/3}}, \qquad \nu\to\infty \tag{6.7}$$

follows. It is shown to be uniform with respect to $a\in[0,1]$, or $\eta\in[0,\infty)$ (see (5.6)). Here $Qi_s(y)$ is related to Airy functions,

$$Qi_s(y) = \frac{1}{\pi}\int_0^\infty e^{-\frac{1}{3}t^3+yt}(t-\sqrt{y})^s dt, \quad s=0,1,\ldots.$$

An integration by parts procedure for (5.7) is used by WONG [27]; the result is supplied with an error bound.

6.5.

The standard form (4.6) is investigated in [20] in both directions: integration by parts and expansion of f at the critical point $t=\alpha/z$. The asymptotic nature of the expansions is discussed and error bounds are given. The integration by parts procedure gives for

$$F_\lambda(z) = \frac{1}{\Gamma(\lambda)}\int_0^\infty t^{\lambda-1}e^{-zt}f(t)dt \tag{6.8}$$

the expansion

$$F_\lambda(z) \sim z^{-\lambda} \sum_{s=0}^{\infty} f_s(\mu) z^{-s}, \quad z \to \infty \tag{6.9}$$

where

$$f_0(t) = f(t), \quad f_{s+1}(t) = t \frac{d}{dt} \frac{f_s(t) - f_s(\mu)}{t - \mu}, \quad s = 0, 1, 2, \ldots \tag{6.10}$$

with $\mu = \lambda / z$. In [20] it is shown that $\{f_s(\mu) z^{-s}\}$ is an asymptotic scale and that (6.9) is a Poincaré type expansion that is uniform with respect to $\mu \in [0, \infty)$. The main condition on f is that its singularities are not too close to $t = \mu$: let R_μ denote the radius of convergence of the Taylor expansion of f at $t = \mu$, then we require $R_\mu^{-1} = \mathcal{O}[(1+\mu)^{-\kappa}]$, $\mu \geqslant 0$ ($\kappa \geqslant \frac{1}{2}$, κ fixed).

7. A NEW CLASS OF POLYNOMIALS

In the previous sections three different types of expansions of the integrand function f are used for obtaining an asymptotic expansion:

(i) in (3.1) an expansion at the fixed critical point $t = 0$;
(ii) for (6.7) an expansion at the movable saddle point $\zeta = \sqrt{\eta}$;
(iii) (6.4) and (6.5) are expansions that in fact are based on a two-point interpolation process for f.

The computation of the coefficients in the asymptotic expansion and the construction of error bounds becomes progressively more difficult in the above cases. Especially this is true when f is defined in terms of implicitly defined relations due to transformations to standard forms. When f is analytic in a neighbourhood of the critical points, in the first two cases representations of the coefficients are available in terms of Cauchy integrals. In general, such a representation is missing in the third case.

Therefore, a new approach to construct the coefficients of a uniform expansion is worth to mention. In this section we describe a recent method of SONI and SLEEMAN [16], where a set of polynomials is introduced to expand the function f. An interesting by-product of the method is a Cauchy-type integral for the coefficients that generalizes the representation for the Taylor expansion. We return to (6.8) to demonstrate the method (in [16] it is given for (4.4), but it has much wider applications).

Consider the formal expansion

$$g(t) = \sum_{s=0}^{\infty} \alpha_s P_s(t) \tag{7.1}$$

where g is defined by $f(t) = f(\mu) + (t - \mu) g(t)$ and where it is assumed that $\{P_s\}$ satisfies the following conditions:

(i) $P_0(t) = 1, \quad P_1(t) = t.$
(ii) $P_s(0) = 0, \quad s = 1, 2, \ldots.$
(iii) $tP'_s(t) = (t - \mu) P_{s-1}(t), \quad s = 2, 3, \ldots.$

Then there is a unique polynomial solution $\{P_s\}$ satisfying the above three

conditions; the coefficients $\{\alpha_s\}$ in (7.1) can be computed recursively and they appear as coefficients in the expansion (6.9).

SKETCH OF THE PROOF. From the recursion (iii) it follows that the first polynomials are

$$P_0(1) = 1, \quad P_1(t)=t, \quad P_2(t)=\tfrac{1}{2}t^2-\mu t, \quad P_3(t)=\tfrac{1}{6}t^3-\tfrac{3}{4}\mu t^2+\mu^2 t.$$

The remaining P_s follow from (ii) and (iii) by writing

$$P_s(t) = \int_0^t (\tau-\mu)\tau^{-1}P_{s-1}(\tau)d\tau.$$

From (ii) we infer $\alpha_0 = g(0)$. Formal differentiation of (7.1) gives (with (iii))

$$tg'(t) = \alpha_1 t+\alpha_2(t-\mu)P_1(t)+\alpha_3(t-\mu)P_2(t)+\ldots,$$

from which we obtain $\alpha_1 = g'(\mu)$. Next we write

$$g_1(t) := t\frac{g'(t)-g'(\mu)}{t-\mu} = \alpha_2 t+\alpha_3 P_2(t)+\alpha_4 P_3(t)+\ldots.$$

Applying again the operator $t\dfrac{d}{dt}$ we get $\alpha_2 = g'_1(\mu)$. In this way all coefficients α_s can be computed. To show that they turn up in (6.9) we insert (7.1) into (6.8) and we obtain (the term $s=0$ gives no contribution)

$$F_\lambda(z) \sim z^{-\lambda}f(\mu)+\sum_{s=1}^{\infty}\alpha_s\psi_s \tag{7.2}$$

$$\psi_s = \frac{1}{\Gamma(\lambda)}\int_0^\infty t^{\lambda-1}e^{-zt}(t-\mu)P_s(t)dt = \frac{1}{z}\psi_{s-1} \quad (s=2,3,\ldots),$$

where we used (iii). Hence it follows that $\psi_s = \lambda/z^{\lambda+s+1}$, and that (7.2) can be written as

$$F_\lambda(z) \sim z^{-\lambda}[f(\mu)+\alpha_1\mu/z+\alpha_2\mu/z^2+\ldots].$$

This gives the relation (using the unicity property of asymptotic expansions) $\alpha_s = f_s(\mu)/\mu$, $s=1,2,\ldots$. □

The above method generalizes Watson's lemma: the expansion at $t=0$ in (3.1) is now replaced by the expansion (7.1). The polynomials P_s reduce to $t^s/s!$ when $\mu\to 0$. Hence in that event (7.1) is the Maclaurin expansion of $g(t)$, when g is analytic. It also may give a new approach for obtaining error bounds; some ideas are worked out in [16]. Furthermore, it gives an explicit representation of $f_s(\mu)$ or α_s. This result is not in [16].

First we compute the coefficients $Q_s(z)$ in the expansion

$$\frac{1}{z-t} = \sum_{s=0}^{\infty}Q_s(z)P_s(t). \tag{7.3}$$

Using (i), (ii), (iii) above we obtain

$$Q_0(z) = 1/z, \quad Q_{s+1}(z) = -\frac{d}{dz}[zQ_s(z)/(z-\mu)].$$

For analytic functions g we have

$$g(t) = \frac{1}{2\pi i}\int \frac{g(z)}{z-t}dz = \frac{1}{2\pi i}\int g(z)\sum_{s=0}^{\infty} Q_s(z)P_s(t)dz.$$

Hence, formally, we obtain for α_s of (7.1)

$$\alpha_s = \frac{1}{2\pi i}\int g(z)Q_s(z)dz = \frac{1}{2\pi i}\int \frac{f(z)-f(\mu)}{z-\mu}Q_s(z)dz.$$

Since $f(\mu)$ gives no contribution we arrive at

$$f_s(\mu) = \frac{\mu}{2\pi i}\int \frac{f(z)Q_s(z)}{z-\mu}dz, \quad s=1,2,.... \tag{7.4}$$

The contour encircles $z=\mu$ in positive direction and no singularities of f.

EXAMPLE. Take $f(z)=1/(z+1)$. The residue at $z=-1$ gives at once

$$f_s(\mu) = \frac{\mu}{\mu+1}Q_s(-1), \quad s=1,2,....$$

8. SOME REMARKS ON ERROR BOUNDS

Special functions of mathematical physics are frequently treated as examples to demonstrate the methods of asymptotical analysis. Functions of hypergeometric type satisfy a differential equation and they have integral representations. Error bounds for the remainders in the expansions of special functions are derived most frequently from a differential equation. In Olver's work, see [11], general methods are derived to obtain strict and realistic error bounds. For a survey on error bounds for expansions of integrals we refer to WONG [27], where also a chapter on uniform expansions is included. Wong's conclusion is that the error theory for uniform expansion is still in its infancy. We agree with him that it is important to develop the theory. In many applications there is no choice between integrals and differential equations.

In a recent paper URSELL [22] demonstrates how the maximum-modulus theorem for analytic functions can be used to bound the error term. In [20] error bounds are given for (6.9). We now review the method of that paper.

The remainder in (6.9) is defined by

$$F_\lambda(z) = z^{-\lambda}[\sum_{s=0}^{n-1} f_s(\mu)z^{-s} + z^{-n}E_n(z,\lambda)], \tag{8.1}$$

$$E_n(z,\lambda) = \frac{z^\lambda}{\Gamma(\lambda)}\int_0^\infty t^{\lambda-1}e^{-zt}f_n(t)dt, \tag{8.2}$$

where f_n is the iterated function given in (6.10). From the conditions on f it

follows that

$$|f_n(t)| \leqslant M(\mu)(1+t)^{p-n}, \quad t \geqslant 0, \tag{8.3}$$

where $M(\mu)$ is bounded for fixed finite values of $\mu=\lambda/z, \mu \geqslant 0$; p is a fixed real number. So we obtain for E_n the bound (we consider real positive values of λ, z)

$$|E_n(z,\lambda)| \leqslant \frac{M(\mu)z^{\lambda}}{\Gamma(\lambda)} \int_0^{\infty} t^{\lambda-1} e^{-zt}(1+t)^{p-n} dt. \tag{8.4}$$

When $p-n \leqslant 0$ the integral is easily estimated and we obtain $|E_n(z,\lambda)| \leqslant M(\mu)$; when $p-n>0$ we have to accept the integral in the bound (although it is a well-known special function, see (6.6)). Another point is that (8.3) may be sharp for t-values far from the interesting point $t=\mu$. In that case, the right-hand side of (8.4) may grossly overestimate $|E_n|$. To obtain a more manageable and more realistic bound we define real numbers σ_n such that

$$|f_n(t)| \leqslant M|f_n(\mu)|[(t/\mu)^{-\mu} e^{t-\mu}]^{\sigma_n}, \quad t>0, \tag{8.5}$$

where M is a fixed constant exceeding unity; $f_n(\mu)$ is supposed to be non-vanishing. Now the estimate is sharp at $t=\mu$ and the bound is expressed in terms of $f_n(\mu)$, which is part of the asymptotic scale. For E_n we obtain

$$|E_n(z,\lambda)| \leqslant M|f_n(\mu)|R_n, \tag{8.6}$$

$$R_n = (1-\sigma_n/z)^{-\lambda}[(\lambda-\mu\sigma_n)/e]^{\mu\sigma_n}\Gamma(\lambda-\mu\sigma_n)/\Gamma(\lambda).$$

When $z-\sigma_n$ and $\lambda=\mu z$ are large, R_n is close to unity, which follows from the Stirling approximation of the gamma functions. Observe that σ_n does not depend on z, when we consider μ as an independent uniformity parameter. When σ_n is a bounded function of μ on $[0, \infty)$, (8.6) gives a sufficient condition to prove that (6.9) is a uniform expansion with respect to the scale $\{f_s(\mu)z^{-s}\}$ (when $f_s(\mu)$ happens to be zero for some μ, s the scale has to be modified). Of course the bound is useful when σ_n is not too large.

The best value of σ_n in (8.5) is given by

$$\sigma_n = \sup_{t>0} \frac{ln|f_n(t)/[Mf_n(\mu)]|}{t-\mu-\mu\ln(t/\mu)}.$$

It should be remarked that, in general, it is rather difficult to compute σ_n, especially when f is obtained from a transformation to standard form.

The above method modifies a method of Olver [11] for Laplace integrals of the form (3.1) (non-uniform case).

References

1. N. Bleistein (1970). Uniform asymptotic expansion of integrals with stationary point near algebraic singularity. *Comm. Pure Appl. Math. 19,* 353-370.

2. N. BLEISTEIN (1967). Uniform asympotic expansions of integrals with many nearby stationary points and algebraic singularities. *J. Math. Mech. 17,* 535-560.
3. N. BLEISTEIN, R.A. HANDELSMAN (1975). *Asymptotic Expansions of Integrals,* Holt, Rinehart, and Winston.
4. C. CHESTER, B. FRIEDMAN, F. URSELL (1957). An extension of the method of steepest descents. *Proc. Cambridge Philos. Soc. 53,* 599-611.
5. J.G. VAN DER CORPUT (1934). Zur Methode der stationären Phase I, *Compositio Math. 1,* 15-38.
6. J. J. DUISTERMAAT (1974). Oscillatory integrals, Lagrange immersions and unfolding of singularities. *Comm. Pure Appl. Math. 27,* 207-281.
7. A. ERDÉLYI, M. WYMAN (1963). The asymptotic evaluation of certain integrals. *Arch. Rational Mech. Anal. 14,* 217-260.
8. A. ERDÉLYI (1970). Uniform asymptotic expansions of integrals. R.P. GILBERT, R.G. NEWTON (eds.). *Analytical Methods in Mathematical Physics,* 149-168, Gordon & Breach.
9. A. ERDÉLYI (1974). Asymptotic evaluation of integrals involving a fractional derivative. *SIAM J. Math. Anal. 5,* 159-171.
10. B. FRIEDMAN (1959). Stationary phase with neighboring critical points. *J. Soc. Indust. Appl. Math. 7,* 280-289.
11. F.W.J. OLVER (1974). *Asymptotics and Special Functions,* Academic Press.
12. F.W.J. OLVER (1975). Unsolved problems in the asymptotic estimation of special Functions. R. ASKEY (ed.). *Theory and Application of Special Functions,* 99-142, Academic Press.
13. T. POSTON, I.N. STEWART (1978). *Catastrophe Theory and its Applications,* Pitman.
14. L.A. SKINNER (1980). Uniformly valid expansions for Laplace integrals. *SIAM J. Math. Anal. 11,* 1058-1067.
15. K. SONI (1983). A note on uniform asymptotic expansions of incomplete Laplace integrals. *SIAM J. Math. Anal. 14,* 1015-1018.
16. K. SONI, B.D. SLEEMAN (1982). *On Uniform Asymptotic Expansions and Associated Polynomials,* Report DE 82:4, UDDM, Dundee.
17 N.M. TEMME (1976). Remarks on a paper of Erdélyi. *SIAM J. Math. Anal. 7,* 767-770.
18. N.M. TEMME (1981). On the expansion of confluent hypergeometric functions in terms of Bessel functions. *J. Comput. Appl. Math. 7,* 27-32.
19. N.M. TEMME (1982). The uniform asymptotic expansion of a class of integrals related to cumulative distribution functions. *SIAM J. Math. Anal. 13,* 239-253.
20. N.M. TEMME (1985). Laplace type integrals: transformation to standard form and uniform asymptotic expansion. *Quart. Appl. Math. XLIII,* 103-123.
21. F.G. TRICOMI (1950). Asymptotische Eigenschaften der unvollständigen Gammafunktion. *Math. Z. 53,* 136-148.

22. F. URSELL (1984). Integrals with a large parameter: Legendre functions of large degree and fixed order. *Math. Proc. Camb. Phil Soc. 95,* 367-380.
23. B.L. VAN DER WAERDEN (1951). On the method of saddle points. *Appl. Sci. Res. Ser. B2,* 33-45.
24. J. WIMP (1980). Uniform scale functions and the asymptotic expansion of integrals. *Ordinary and Partial Differential Equations* (Proc. Fifth Conf., Univ. Dundee, 1978) 251-271, *Lecture Notes in Math.* 827, Springer.
25. R. WONG (1973). On uniform asymptotic expansion of definite integrals. *J. Approximation Theory 7,* 76-86.
26. R. WONG (1980). On a uniform asymptotic expansion of a Fourier-type integral. *Quart. Appl. Math. XXXVIII,* 225-234.
27. R. WONG (1980). Error bounds for asymptotic expansions of integrals. *SIAM Review 22,* 401-435.
28. A.S. ZIL'BERGLEIT (1977). Uniform asymptotic expansions of some definite integrals. *USSR Comput. Maths. Math. Phys. 16,* 36-44.
29. A.S. ZIL'BERGLEIT (1978). A uniform asymptotic expansion of Dirichlet's integral. *USSR Comput. Maths. Math. Phys. 17,* 237-242.