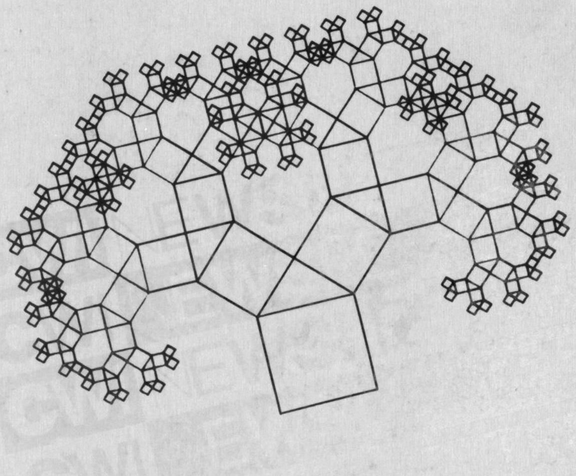
CWI NEWSLETTER

CWI
Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

Quarterly, Issue no. 6
March 1985



CWI NEWSLETTER

Number 6, March 1985

Editors

Arjeh M. Cohen Richard D. Gill Jo Ebergen

The CWI Newsletter is published quarterly by the Centre for Mathematics and Computer Science (Centrum voor Wiskunde en Informatica), Kruislaan 413, 1098 SJ Amsterdam, The Netherlands. The Newsletter will report on activities being conducted at the Centre and will also contain articles of general interest in the fields of Mathematics and Computer Science, including book reviews and mathematical entertainment. The editors encourage persons outside and in the Centre to contribute to the Newsletter. Normal referee procedures will apply to all articles submitted.

The Newsletter is available free of charge to all interested persons. The Newsletter is available to libraries on an exchange basis.

All correspondence should be addressed to: *The CWI Newsletter, Centre for Mathematics and Computer Science, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands.*

ISSN 0168-826x

Contents

Centre for Mathematics and Computer Science

Centrum voor Wiskunde en Informatica

The Pythagoras Tree as a Julia Set

H.A. Lauwerier
Centre for Mathematics and Computer Science
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

1. Introduction

Forty years ago in the dark days of the second world war the Dutch engineer A. Bosman constructed the so-called Pythagoras tree reproduced here in fig.1.1. It must have taken him many, many hours at the drawingboard. But now with a personal computer and a plotter a nice tree can be formed within an hour and generalizations can be made to order.

Our research on this tree actually started when we tried to determine the set of infinitesimally small squares formed in the limit if the construction is continued indefinitely. Let J denote the closure of this set then J, which we call the *blossom* of the Pythagoras tree, is a continuous curve which is invariant (i.e. mapped into itself) under two (!) similarity transformations A and B. Coordinates can be chosen in such a way that in complex notation

$$\begin{cases} A: \ z \mapsto 1 + (1+i)z/2, \\ B: \ z \mapsto 1 + (1-i)z/2. \end{cases} \tag{1.1}$$

We see that A has $1+i$ as its centre of rotation (or fixed point), the reduction factor $1/\sqrt{2}$ and the rotation angle $\pi/4$. For B the centre is at $1-i$ with reduction factor $1/\sqrt{2}$ and rotation angle $-\pi/4$. Both centres $1 \pm i$ are elements of J. More points of J (in fact a dense subset of it) can be obtained from them by subjecting them to a random sequence of operations of A and B. In this way fig.1.2 has been obtained as part of the blossom of the Pythagoras tree.

J is a continuous image of the unit interval $0 \leqslant r \leqslant 1$. Let r $(r<1)$ have the binary expansion

$$r = 0.r_1 r_2 r_3 \cdots \tag{1.2}$$

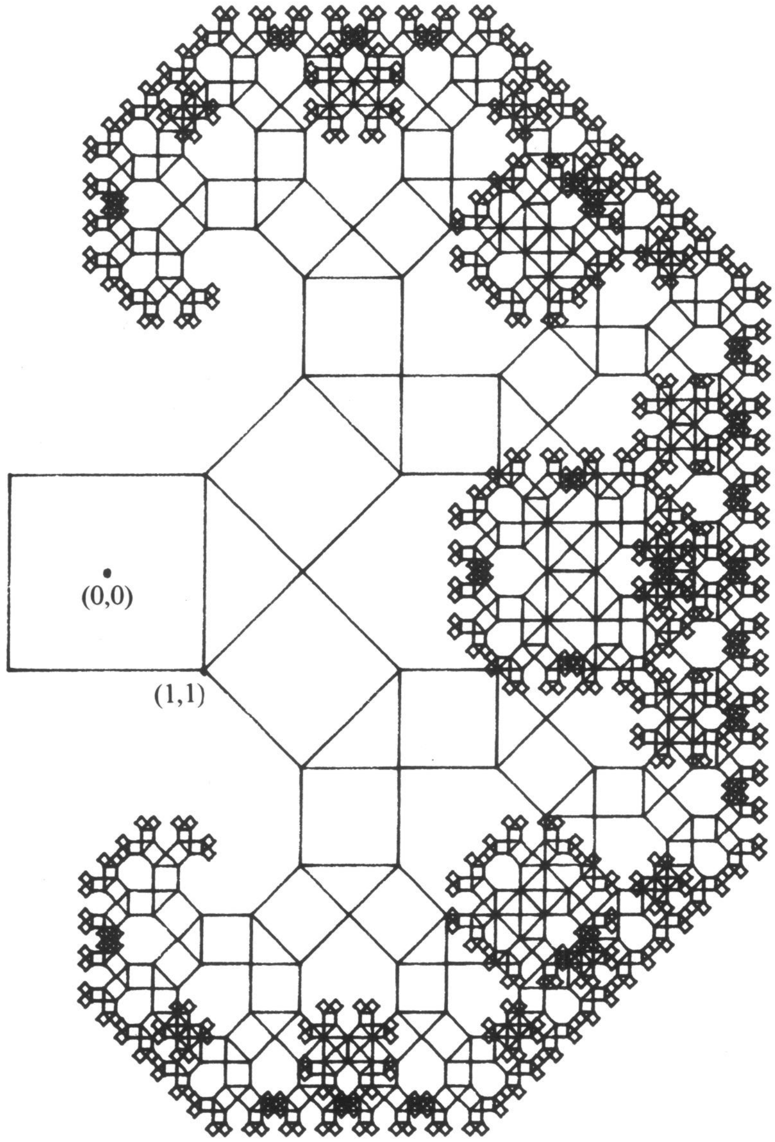


FIGURE 1.1. The Pythagoras tree

and define for $k \geqslant 1$

$$s_k = (1-r_k)a + r_k b, \tag{1.3}$$

with

$$a = (1+i)/2, \quad b = (1-i)/2. \tag{1.4}$$

This means that s_k is either a or b according to the value 0 or 1 of the kth binary digit of r. Then to r we may associate the following point of J

FIGURE 1.2. (Part of) the limit set of the Pythagoras tree

$$z = c_0 + c_1 + c_2 + c_3 + \cdots \tag{1.5}$$

where $c_0 = 1$ and for $k \geqslant 1$

$$c_k = s_k c_{k-1}. \tag{1.6}$$

Thus to $r = 0$ corresponds the point

$$z = 1 + a + a^2 + a^3 + ... = 1/(1-a) = 1 + i,$$

the fixed point of A, and to $r = 1/3 = .010101\cdots$ corresponds

$$z = 1 + a + ab + a^2b + a^2b^2 + a^3b^2 + \cdots = 3 + i.$$

On J the actions of A and B are then translated into

$$\begin{cases} A: & r \mapsto r/2, \\ B: & r \mapsto (1+r)/2. \end{cases} \tag{1.7}$$

Thus a random sequence of transformations A and B corresponds to a uniform distribution of numbers in (0,1) and accordingly to what one could call a uniform distribution of points on J.

The obvious generalization is to give a and b arbitrary complex values with $|a|<1$ and $|b|<1$. The problem of determining the conditions under which the resulting curve can be considered as the limit set of some Pythagoras tree will be taken up in the next section 2.

There is a little problem about rational numbers with a terminating binary expansion. Sequences like $.1000\cdots$ and $.0111\cdots$ represent the same rational number (1/2). However, a simple calculation shows that for $a+b = 1$ the corresponding sequences of complex numbers $1+b+ab+a^2b + a^3b + \cdots$ and $1+a+ab+ab^2+ab^3+ \cdots$ also represent the same point.

The overall situation is very reminiscent of the inverse logistic map in its complex form as studied by MANDELBROT [1]

$$z' = \pm \sqrt{z+\mu}. \tag{1.8}$$

We follow up this analogy in more detail in section 3. Its main features are as follows. For suitable values of μ this two-valued map (1.8) has a Julia set (cf.[1]) as the collection of limit points of random iterative sequences. The fixed points are $p/2$ and $1-p/2$ where $\mu = (p^2-2p)/4$ in the usual notation of the logistic map as $x \rightarrow px(1-x)$. In fact, both the more general version of (1.1)

$$z' = 1 + az \quad \text{or} \quad z' = 1 + bz, \tag{1.9}$$

and (1.8) can be considered as the members of a family of quadratic (2,2)-maps described by a relation of the form

$$F(z',z) = 0, \tag{1.10}$$

where F is a quadratic polynomial of its arguments. In particular the blossom of the Pythagoras tree and the San Marco attractors (cf.[1]) can be interpreted as Julia sets of the map (1.10). However, the theory of iterated analytic maps (cf.[2]) is only fully developed for the case that $z'(z)$ (or its inverse) is a single-valued meromorphic analytic function. The examples given here may give rise to an extension of the theory to algebraic functions of the kind (1.10).

2. The Pythagoras tree

In the introduction we have seen that the construction of BOSMAN's Pythagoras tree can be based upon sequences of complex numbers (1.5), (1.6) with a and b given by (1.4). In fig.2.1 the initial part of what we call the skeleton of a tree is given. The endpoints $P_k(z_k)$ of the successive branches can be labelled in such a way that

$$z_0 = 0, \quad z_1 = 1, \quad z_2 = 1+a, \quad z_3 = 1+b,$$
$$z_4 = 1+a+a^2, \quad z_5 = 1+a+ab, \quad \text{etc.}$$

E.g. for $k = 50$, which is 110010 in binary notation, we have

$$z_{50} = 1+b+ab+a^2b+a^2b^2+a^3b^2.$$

What we have done in fig.2.1 with the special values of a and b can be done for any values of a and b. In this way we obtain a similar tree. The question arises whether such a tree can be interpreted as the skeleton of a generalized Pythagoras tree. Can we put squares or quadrilaterals onto the branches?

Before that question can be answered we need a little more analysis of the tree of fig.2.1 which we now interpret as an illustration of the general case. The tree is transformed into itself by either similarity transformation

$$\begin{cases} A: \; z \mapsto 1 + az, \\ B: \; z \mapsto 1 + bz. \end{cases} \tag{2.1}$$

The fixed points of these transformations, $1/(1-a)$, $1/(1-b)$ are indicated in fig.2.1 by A and B. An endpoint with index k is transformed into an endpoint with a higher index. In particular

$$Az_{50} = z_{82}, \quad Bz_{50} = z_{114}.$$

The general rule is as follows. Let

$$2^m \leqslant k < 2^{m+1}$$

then symbolically

$$A(k) = k+2^m, \quad B(k) = k+2^{m+1}.$$

We now consider the central question under what conditions for a and b the tree of fig.2.1 can be blown up into a generalized Pythagoras tree. By this we understand a tree like fig.1.1 where the basic pattern is a triangle with similar quadrilaterals on its sides. In fig.1.1 the quadrilaterals are squares and the triangle is half a square. If the triangle is rectangular but not isosceles the tree is called an *oblique Pythagoras tree.* In all other cases the tree is called a *generalized Pythagoras tree.*

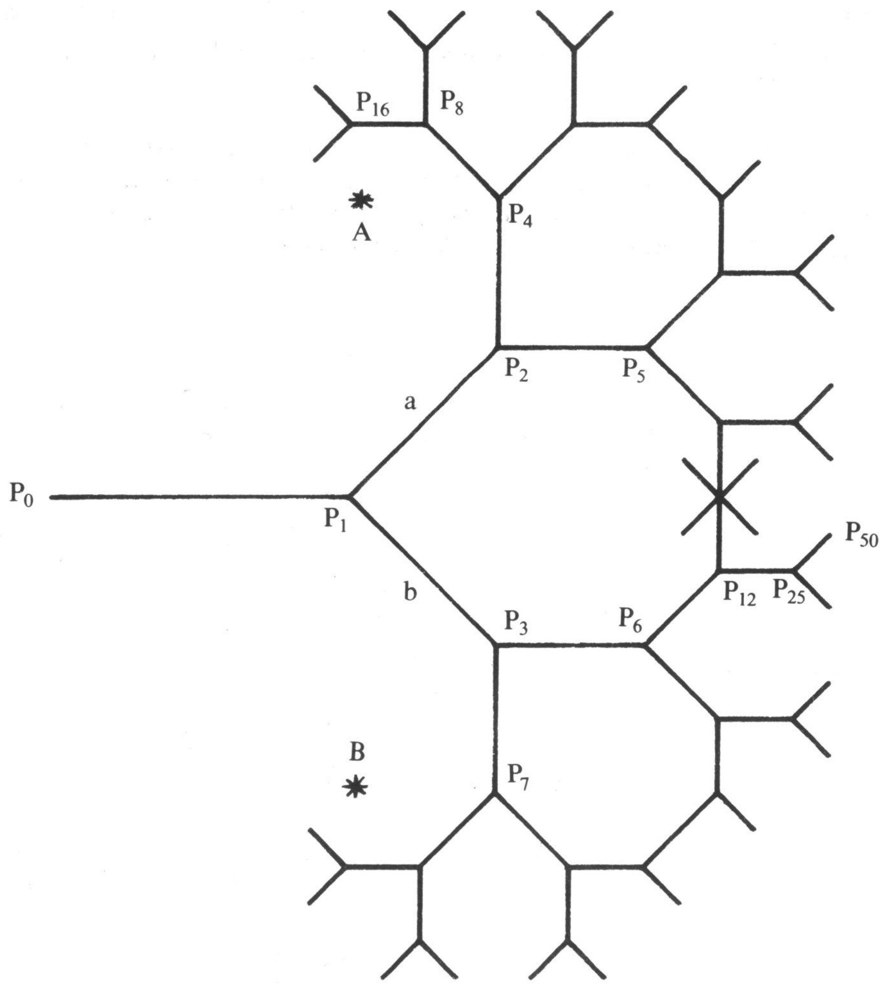


FIGURE 2.1. Initial part of the skeleton of the Pythagoras tree

In view of the similarity transformations (2.1) it is sufficient to consider the first three branches with the first three quadrilaterals as shown in fig.2.2. Let $UU'V'V$ be the first quadrilateral with $U' = A(U)$ and $V' = B(V)$ then there exists a point W which is both the A-image of V as well as the B-image of U. Labelling U and V by complex numbers u and v we obtain the condition

$$1+bu = 1+av$$

so that $bu = av$. This suggests the following construction. Let a and b be arbitrary complex numbers, of course with $|a| < 1$, $|b| < 1$ and a/b not real, then for any complex number λ a generalized Pythagoras tree can be constructed. The first quadrilateral is determined by the corners

$$\lambda a,\ \lambda b,\ 1+\lambda a^2,\ 1+\lambda b^2. \tag{2.2}$$

EXAMPLE. For $a = \frac{1}{2}(1+i)$, $b = \frac{1}{4}(3-2i)$ and $\lambda = 1$ we obtain a quadrilateral with the corner points $(1+i)/2$, $(3-2i)/4$, $(2+i)/2$, $(21-12i)/16$.

The situation is sketched in fig.2.2. The quadrilateral is a trapezium here. A simple calculation shows that always $U'V' /\!/ UV$ when $a+b$ is real. When the vectors UU' and VV' are equal, the quadrilaterals are parallelograms. In that case we should have $\lambda(b^2-a^2) = \lambda(b-a)$ which gives the condition

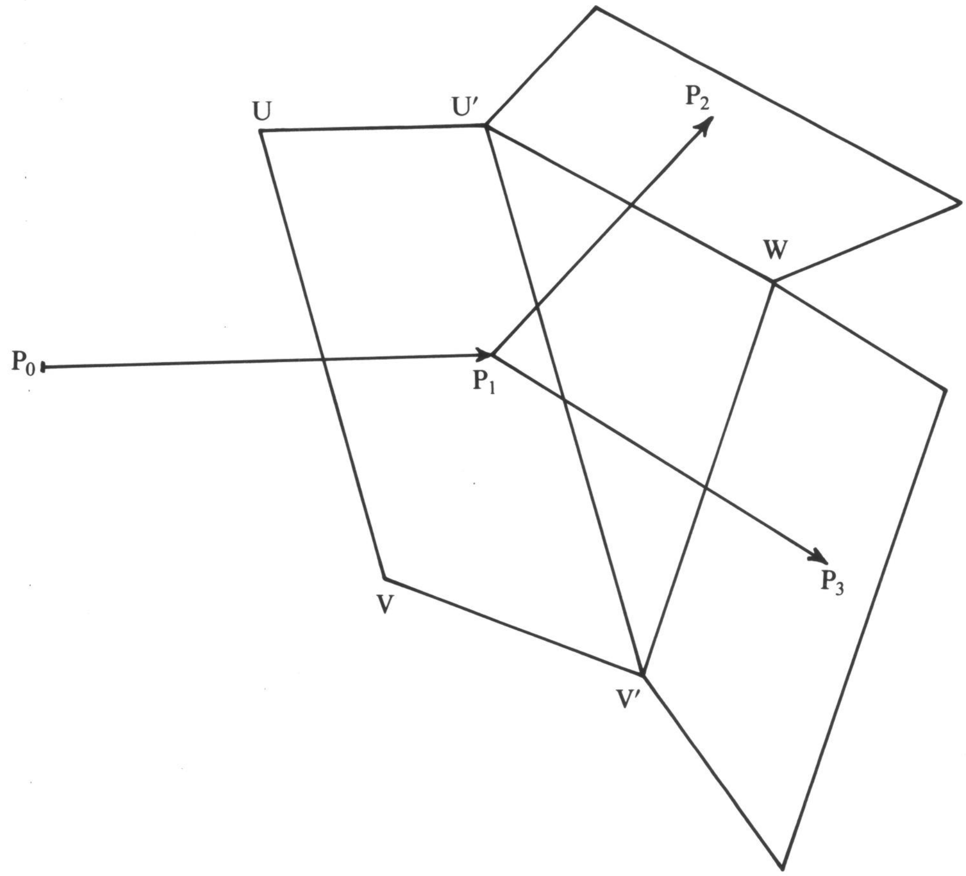


FIGURE 2.2. The beginning of a generalized Pythagoras tree

$$a+b = 1. \tag{2.3}$$

We may write

$$a = (1+ic)/2, \ b = (1-ic)/2 \tag{2.4}$$

where $c = i(b-a)$ is an arbitrary complex number for which $|a| < 1$ and $|b| < 1$. Thus c is restricted to a lens-shaped region bounded by the two circular arcs defined by $|c \pm i| < 2$.

The quadrilaterals are squares if

$$i(v-u) = u'-u$$

i.e. if

$$(1+\lambda a^2) - \lambda a = i\lambda(b-a).$$

Substitution of (2.4) gives the unique solution

$$\lambda = \frac{4}{c^2+4c+1}. \tag{2.5}$$

So given a and b satisfying (2.4) unless $c = -2 \pm \sqrt{3}$ a generalized Pythagoras tree with squares can be constructed. An oblique Pythagoras tree, i.e. a tree with squares and right-angled triangles, calls for a further specialization. A simple calculation shows that this requires that

$$c = -i\exp(2\alpha i), \ 0 < \alpha < \pi/4$$

and hence

$$\begin{cases} a = \cos^2\alpha + i\sin\alpha\cos\alpha, \\ b = \sin^2\alpha - i\sin\alpha\cos\alpha \end{cases} \tag{2.6}$$

Finally for $\alpha = \pi/4$ the original symmetric Pythagoras tree is obtained.

The corresponding geometric situation for an oblique tree is sketched in fig.2.3 (where $\alpha = 2\pi/9$).

A 'full' oblique Pythagoras tree with $\alpha = \pi/5$ is given in fig.2.4. The limit set of the infinitesimally small squares, its blossom, is given in fig.2.5. In the computer program it is obtained as the invariant set of the similarity transformations

$$z \mapsto 1+az, \quad z \mapsto 1+bz$$

with a and b given by (2.6). Each fixed point $1/(1-a)$ and $1/(1-b)$ is subjected to random sequences of similarity transformations.

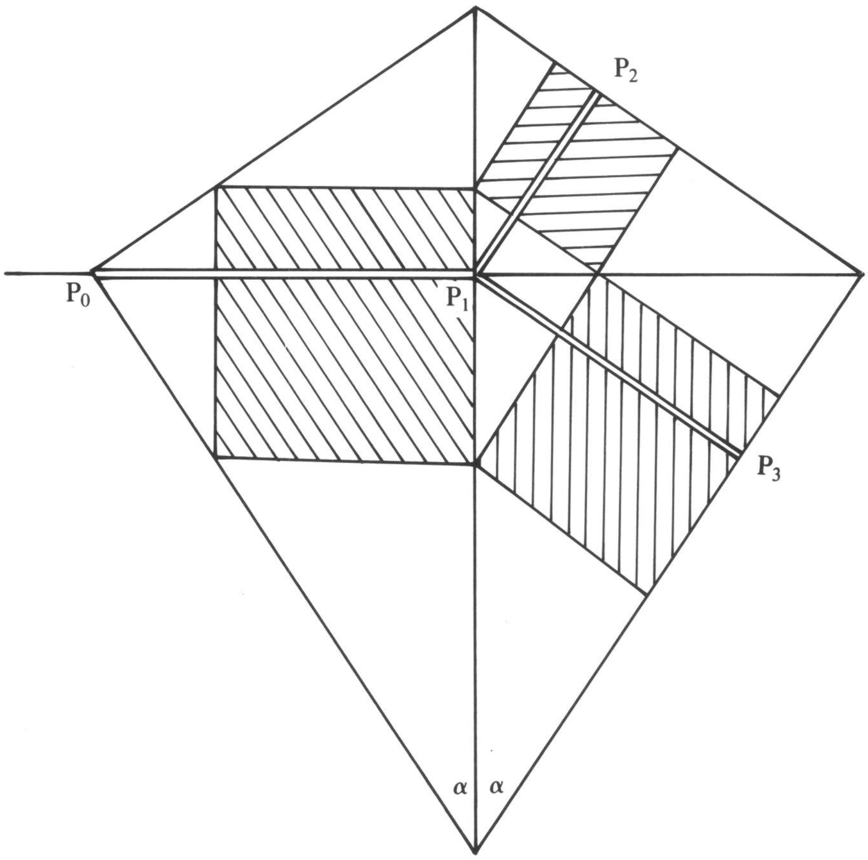


FIGURE 2.3. The basis of an oblique Pythagoras tree

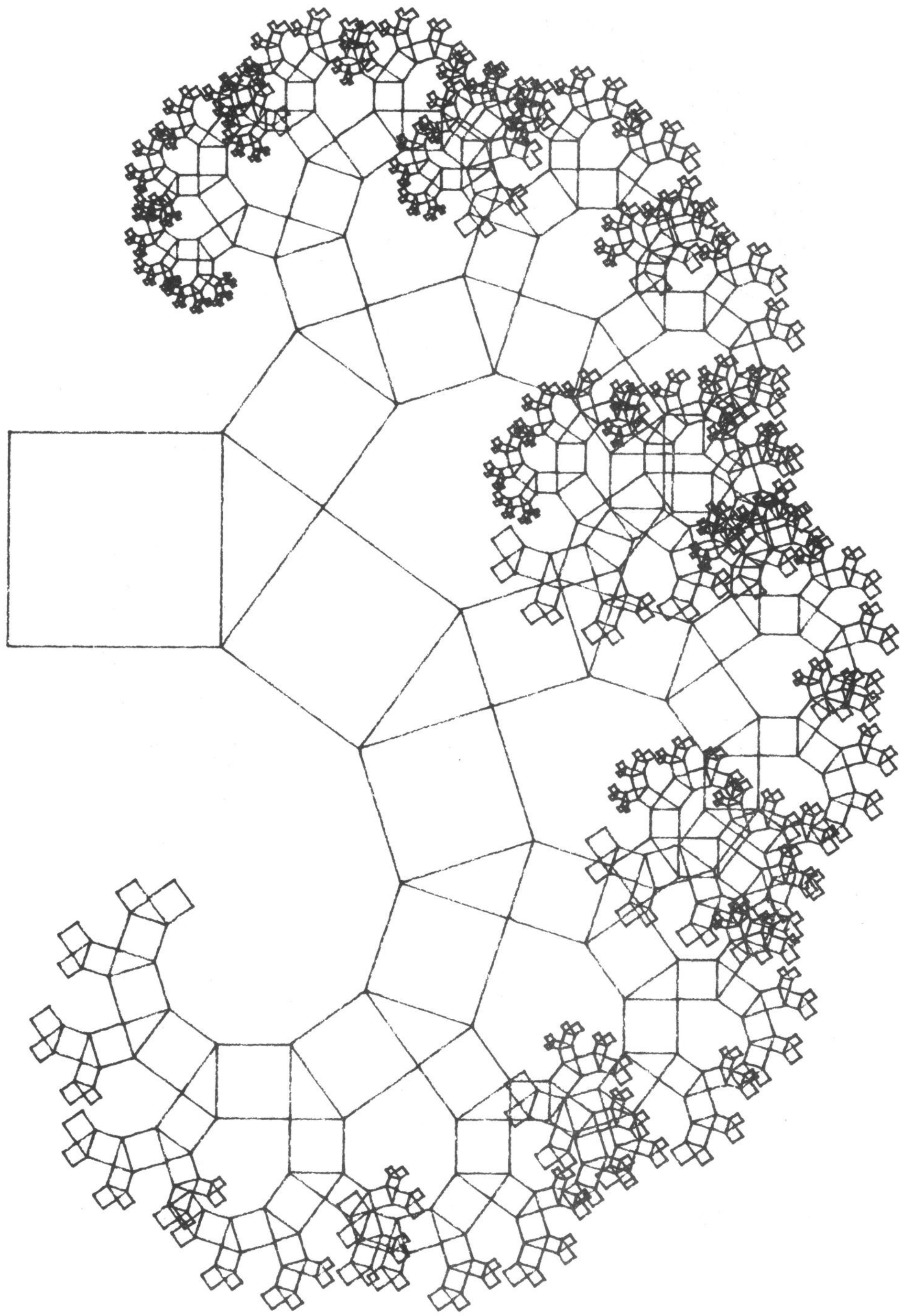


FIGURE 2.4. An oblique Pythagoras tree with $\alpha = \pi / 5$

FIGURE 2.5. The limit set of an oblique Pythagoras tree with $\alpha = \pi / 5$

3. Julia sets

A Julia set is a certain invariant set of an analytic map $z \mapsto f(z)$. It is obtained as the closure of the set of all unstable periodic points. Definitions, properties and many details can be found in the excellent survey paper by Blanchard [2]. In many cases the Julia set is a non-differentiable curve or a totally disconnected point set. The very special case

$$z \mapsto z^2$$

already shows many features of the general case. The Julia set is the unit circle here. It is densely covered by the pre-images of any of its points. It is a separatrix separating orbits converging to $z=0$ and orbits diverging to $z=\infty$. It is an attractor of the inverse map $z \mapsto \pm\sqrt{z}$.

Much attention has been paid to the properties of the quadratic map

$$z \mapsto z^2 - \mu \tag{3.1}$$

in the literature. Only for $\mu = 0$ and $\mu = 2$ do we have a Julia set in the form of a simple curve or arc. For $\mu = 3/4$ the Julia set has a nice shape called the 'San Marco attractor' by Mandelbrot. It is given in fig.3.1. The computer program is very similar to that for the blossom of the Pythagoras tree. Points of the Julia set are obtained from the iteration process

$$z_{k+1} = \sigma_k \sqrt{\mu + z_k}, \tag{3.2}$$

where σ_k, $k \in \mathbb{N}$ is a random sequence of $\pm$ 1's and where $z_0 = -1/2$, a critically stable fixed point which is an element of the Julia set.

The Julia set of (3.1) is invariant under the two transformations

$$\begin{cases} A: \; z \mapsto \sqrt{\mu+z}\,, \\ B: \; z \mapsto -\sqrt{\mu+z}. \end{cases} \tag{3.3}$$

the two inverses of (3.1). If this is compared with the corresponding transformations (2.1) of the generalized Pythagoras tree, we observe a striking similarity. The limit set of a Pythagoras tree and the Julia set of the quadratic transformation (3.3) appear to have common features. We have seen that the limit points of the Pythagoras tree formed from

$$z \mapsto 1 + az, \quad z \mapsto 1 + bz \tag{3.4}$$

with $|a| < 1$, $|b| < 1$ are explicitly given by

$$z = 1 + \sum_{k=0}^{\infty} a_0 a_1 a_2 \cdots a_k, \tag{3.5}$$

with

$$\begin{cases} a_k = a & \text{if } r_k = 0, \\ a_k = b & \text{if } r_k = 1, \end{cases} \tag{3.6}$$

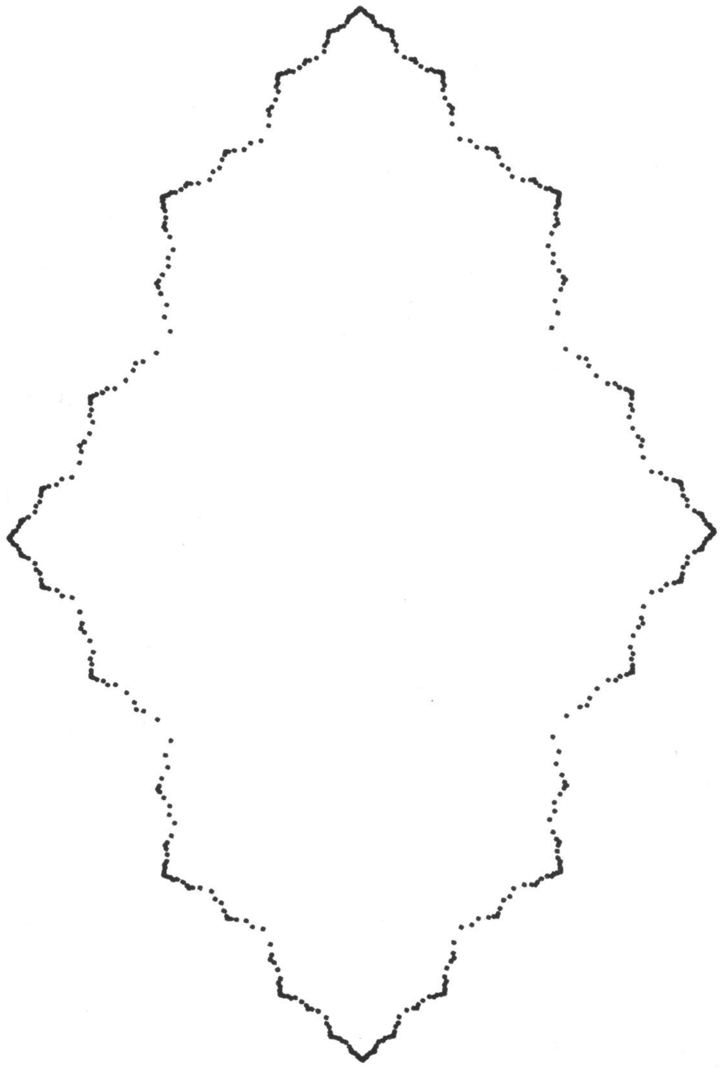


FIGURE 3.1. The San Marco attractor, the Julia set of $z \mapsto z^2 - 3/4$

where r_k is the kth binary digit of the binary expansion of a fraction r. Thus to each point of the limit set J corresponds a real number of [0,1]. (There is a little ambiguity for binary expansions terminating in an endless string of zeros or ones, but this concerns only a countable subset of J.)

The dynamics on the limit set J can be described by (see fig.3.2)

$$\begin{cases} Az: & r \mapsto r/2, \\ Bz: & r \mapsto (1+r)/2. \end{cases} \tag{3.7}$$

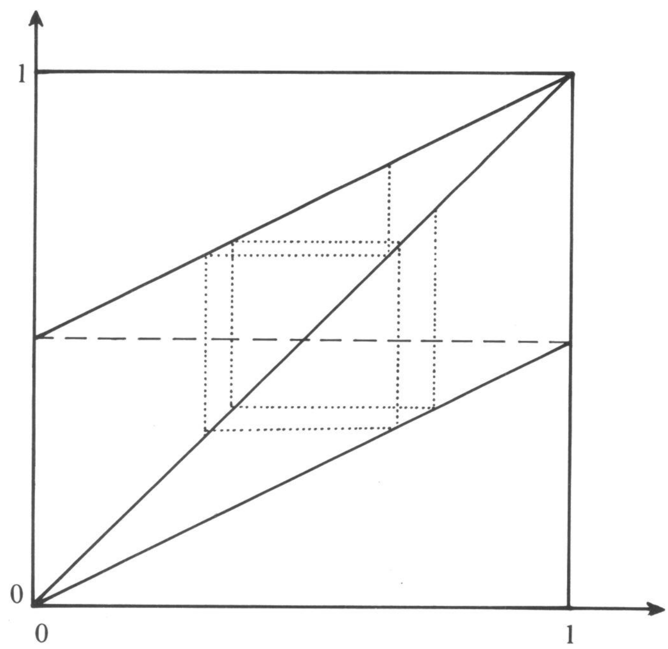


FIGURE 3.2.

This double-valued transformation has a unique inverse which is perhaps the simplest transformation showing chaotic behaviour.

Let us next consider the quadratic map (3.3) for the case $\mu = 2$. Then we may use the parametrization

$$z = 2\cos \pi r. \tag{3.8}$$

Substitution gives at once

$$\begin{cases} Az: & r \mapsto r/2, \\ Bz: & r \mapsto 1 - r/2, \end{cases} \tag{3.9}$$

the well-known tent map closely related to the map (3.7).

Thus there is every reason to extend the notion of the Julia set to non-unique analytic mappings. In both cases we have considered here the limit set J has the same chaotic behaviour. If z is an arbitrary point of J, then the sequences formed by subjecting z either to A or to B in some pseudo-random manner, e.g. prescribed by the binary digits of the binary expansion of a fraction, almost never converge. A generalized Julia set may then be defined as the limit set of all sequences found in this way from two or more analytic transformations, A, B etc. provided it exists. If the transformations are the branches of the inverse of a single-valued analytic function this coincides with the traditional definition. It would be tempting to sketch a general theory but, in our opinion, it is better to start with a number of interesting special cases.

We end this paper by considering the following (2-2)-complex map in which both the Pythagoras tree and the quadratic map are combined. We consider

$$F(w,z) = 0 \tag{3.10}$$

where F is a quadratic polynomial. It is assumed that $w=z=\pm 1$ are fixed points with multipliers dw/dz equal to a and b. Then F is determined by a further single complex parameter c and can be written as

$$(w-az-1+a)(w-bz+1-b)+c(w-z)^2 = 0. \tag{3.11}$$

For $c = 0$ this reduces to the Pythagoras map

$$\begin{cases} w = 1 + a(z-1), \\ w = -1 + b(z+1). \end{cases} \tag{3.12}$$

For $c = -ab = -\frac{1}{2}(a+b)$ the quadratic map is obtained in the form

$$\tfrac{1}{2}\sqrt{(1+c)/c}\,(w^2-1) + w - z = 0. \tag{3.13}$$

The maps (3.13) and (3.1) are equivalent with the following relation between the parameters

$$4\mu c = 1. \tag{3.14}$$

In particular the San Marco attractor is obtained for $a = -1$, $b = 1/3$, $c = 1/3$ as

$$w^2 + w - 1 = z. \tag{3.15}$$

As an illustration we consider the special case

$$a = (1+i)/2, \; b = (1-i)/2, \; c = -\tfrac{1}{4}.$$

The multipliers are those of the Pythagoras tree (1.1). The value $c = -\frac{1}{4}$ is chosen halfway the value $c = 0$ of the Pythagoras tree and $c = -\frac{1}{2}$ of the quadratic map. The result shown in fig.3.3 looks like the blossom of a generalized Pythagoras tree but has the cauliflower structure of known Julia sets of the quadratic map (see [2]). Shown are 1000 pre-images of the point $z = 1$ which is a fixed point of (3.13). It is a safe conjecture that this Julia set is entirely disconnected.

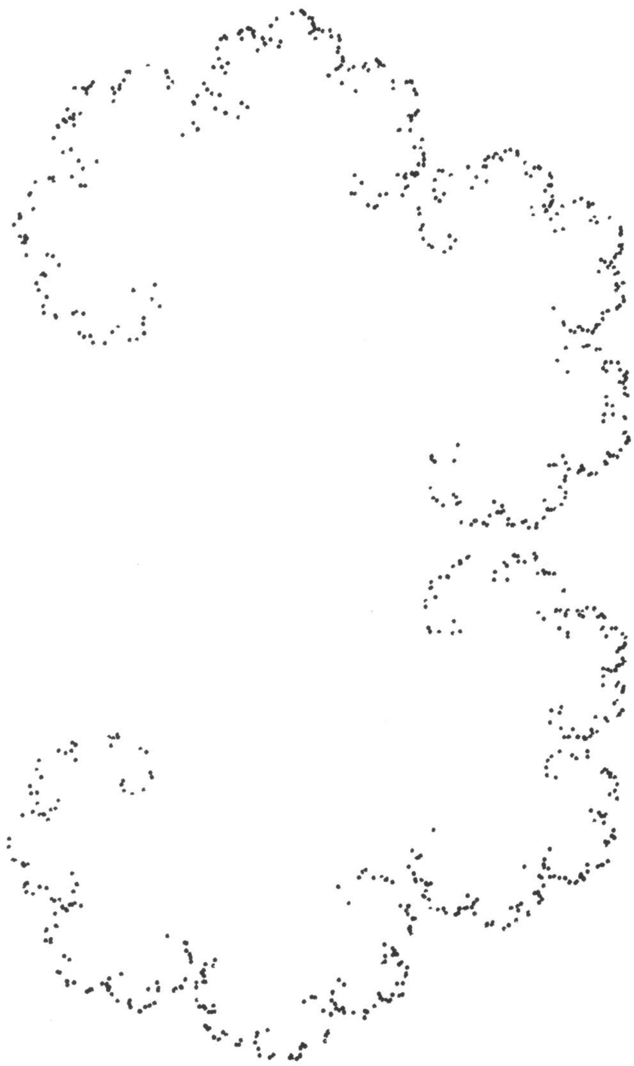


FIGURE 3.3. A generalized Julia set

The last illustration may give an idea of what to expect in a more general situation. We took

$$\begin{cases} A: \ z \mapsto 1 + iz/2, \\ B: \ z \mapsto a(1+z^2)/z, \end{cases} \qquad (3.16)$$

with $a = 4/5$. The fixed point of A is $0.8 + i0.4$. The fixed points of B are ± 2. In fig.3.4 we have shown a representative part of the corresponding generalized Julia set with $z = 2$ as the starting-point of a random sequence.

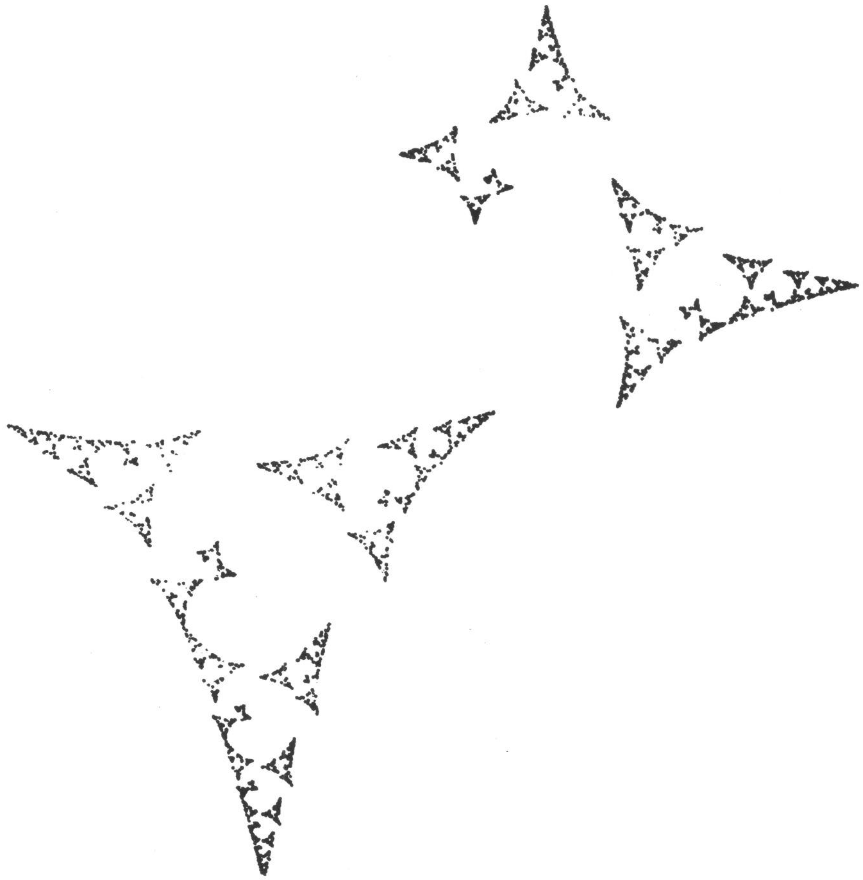


FIGURE 3.4. A generalized Julia set

REFERENCES

1. B. MANDELBROT (1980). Fractal aspects of the iteration of $z \to \lambda z(1-z)$. *Ann. New York Acad. Sci.* *357,* 249-259.
2. P. BLANCHARD (1984). Complex analytic dynamics. *Bull. Am. Math. Soc.* *11,* 85-141.
3. M.F. BARNSLEY, A.N. HARRINGTON (1985). A Mandelbrot set for pairs of linear maps. *Physica 15D,* 421-432.

Mathematical Libraries in Ada [1]

Jan Kok
Centre for Mathematics and Computer Science
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

In this paper a description is given of past and present activities at the CWI related to the programming language *Ada*. After a survey of Ada's history and its design aims, more details are given of language features that are of special interest for the implementation of numerical methods and the construction of large scientific libraries. Characteristics of these libraries, enhanced by the use of Ada, are portability, reliability, and reusability. The paper concludes with an outline of future activities which will take place in international co-operation.
KEY WORDS: Ada, high level language, scientific libraries, portability, reliability, reusability.

1. Introduction

At the end of the sixties it was recognized that the production of computer software was not keeping pace with the increase in availability and possibilities of modern computers. Causes of the so-called 'software crisis' were found in the old-fashionedness of programming methods and tools. Independent efforts to provide a new programming language for common use (one of the tools) merely contributed to the crisis, because of the confusion of languages and the incompatibility of systems.

In a programme launched by the US Department of Defense for modernising both tools and methods, one of the results was the definition of a new programming language for common use. This language, called Ada [1], was primarily designed for the production of large portions of maintainable software for real-time applications. Furthermore Ada's use should enhance software characteristics such as readability, modularity, portability, reusability, and reliability.

With respect to *scientific* computation, Ada was intended in the first place for use in those calculations invoked in software for real-time applications. But the readability, modularity, portability, reusability, and reliability, which make programming in large teams possible as well, make Ada also useful for the production of software for large-scale scientific computation on mainframes.

1. Ada is a registered trademark of the US Government (AJPO).

The need for mathematical software in Ada with the same qualities as software for real-time applications is not yet widely recognized by users. It is generally assumed that computer manufacturers have completely satisfied the demand for software for all kinds of computations. Here, many enjoyable case stories of application failures are tacitly ignored [2]. Perhaps it is the rise of a modern language like Ada which now clearly exposes the shortcomings of previous hardware and software provisions. Possibilities not offered by most other languages are user demands for various real precisions, constructs for writing portable and reusable software, and safe interaction between separate library modules.

One may expect that software produced before, in older languages and probably in many versions for all kinds of machines and systems, should be reconsidered for use in Ada, and that re-design of mathematical software in agreement with the Ada design goals might turn out to be very profitable.

An early initiative was taken by the National Physical Laboratory (NPL) in Teddington and the Centrum voor Wiskunde en Informatica (CWI), to investigate both the new possibilities given by the language Ada for the design of mathematical software and for the construction of scientific libraries for large-scale computation, and to offer solutions for the problems encountered. This led to a project by NPL and CWI that was sponsored by the European Community. In this project, which lasted from 1982 to 1983, guidelines were produced for the design of large, modular, scientific libraries in Ada [4].

In the following sections first a summary of Ada's history and design are given, with a discussion of Ada's design goals and some information about Ada's availability. Next, language features are described which open new ways for the implementation of efficient and portable mathematical software for large scientific libraries. Finally, completed and ongoing activities are described.

2. Ada's history

The story of Ada starts in 1974 when Ada's foster parent, the US Department of Defense (DoD), initiated a programme for obtaining more profits from their software budget, in particular concerning software for embedded systems. After an analysis of software costs, with the conclusion that too much was spent on maintenance and conversion, characteristics were gathered for a programming language in common for all types of machines in use, and for all kinds of applications.

When it became clear that no existing language satisfied the requirements, proposals were requested for the development of a new language. From the selection of four proposals for further evaluation, in 1979 one language design was chosen to become the common high-level language. Its name would be *Ada*, in honour of Augusta Ada Lovelace, daughter of Lord Byron, and as the assistant of Charles Babbage one of the first programmers. In 1980 this language was standardized by the DoD, and a revised version was accepted as ANSI standard in 1983 (ANSI: American National Standards Institute). The process for its becoming an ISO standard continues.

One of the characteristics of Ada is that its text should be clear, readable, and easy-to-understand: the possibility of programming errors and typing errors not being detected should be very low. This characteristic is achieved by:

- offering language expressions for:
  - the declaration of (abstract) data types;
  - making packages of sets of data types, operations, and routines;
  - defining library modules, where components of library modules can be packages as well as individual routines, i.e. procedures, functions, or operators;
  - recovery from aborts by the so-called *handling of exceptions*;
  - programming for distributed systems by the so-called *tasking* facility;
- detection of errors at an early stage by:
  - Ada's strong typing rules and scope rules;
  - the small number of error-prone language concepts;
  - several constructs that encourage structured programming;
  - the inhibition of access to abstract data types from outside;
- possibilities for the design of portable programs by means of language concepts for:
  - obtaining hardware or system information through environmental parameters;
  - connecting appropriate hardware provisions to user-defined data types (and also operators, exceptions, and tasks);
  - reusing software through the *generic* concept (see section 3).

The availability of Ada is still very small. The amount of language constructs and their semantic intricacies make the writing of a compiler and run-time system a major software project. Moreover, only systems which correctly process an official suite of about 2000 test programs are allowed to be called *Ada* (compiler validation). And although the language requirements were such that compilers ought to be able to generate efficient object code, not much can be said about this yet, since the first efforts are spent on obtaining *correct* object code, not *fast*. For Ada to become a success, many more compilers for different machines must become available within the next few years.

Meanwhile, the creation of a new language became just one component of the much larger DoD activity for improving software technology. A related project is the construction of *Ada Programming Support Environments* (APSE) as the indispensable working environments for Ada programmers and for the execution of Ada programmes. An APSE should contain besides the Ada compiler and run-time system, the command language, an Ada-directed editor, libraries, verifiers, a debugger, monitors, data bases, etc.

3. USE OF ADA FOR NUMERICAL SOFTWARE

During 1982 and 1983 the National Physical Laboratory, Teddington, and the Centrum voor Wiskunde en Informatica were engaged in an investigation of the possibilities of designing large modular scientific libraries in Ada. The project was sponsored by the Commission of the European Community and culminated in the production of a set of guidelines [4] which include recommendations on the ways in which Ada can and might be used in this context.

The language features which might make Ada particularly useful for the redesigning of numerical software and for the construction of coherent scientific libraries, are:

- Several *floating-point types* can be defined. E.g.

  type REAL **is digits** 8;

  (Meaning: a new floating-point type named REAL is made by specifying a minimal number of 8 significant decimal digits.)
  If hardware floating-point types with different mantissa lengths are available, then the hardware type best suitable to the user-defined type will be associated with it. If the user's requirements cannot be met, the program will not run.
- Type definitions and associated operator definitions can be made for all kinds of numerical data structures, e.g.
  VECTOR, MATRIX, COMPLEX, RATIONAL, etc.
- A *package* in Ada is a capsule containing related definitions of types, operators, and procedures. Moreover, the physical structure of data types can be hidden from the user. For example,

```
package RATIONAL_FIELD is
   type RATIONAL is record
                  NUMERATOR : INTEGER;
                  DENOMINATOR : POSITIVE;
                end record;

   function "+" ( A, B : RATIONAL ) return RATIONAL;

   -- analogous declarations for "-", "*", "/", ...

 end RATIONAL_FIELD;
```

  The declarations of this package can be obtained together by mentioning: **use** RATIONAL_FIELD.
  If the inner declaration is replaced by:

```
type RATIONAL is private;
```

this would make the structure of the type RATIONAL hidden.

- Language concepts are available for the construction of large pre-compiled libraries containing as modules packages, but also individual routines. The language rules enable that consistency checks for parameters (in compile time) will always be possible.
- Modules can be parameterized with *types*, *operators* and other subprograms, as well as with individual values and variables. Such modules are called *generic* subprograms or packages, and it allows library packages to be constructed once for a whole class of user-defined types and the operations thereon.
  For example, for sorting values in a one-dimensional array only one generic sort procedure need be written:

```
generic
  -- ( with three generic parameters:
    type EL_TYPE is private;
    type AR_TYPE is array ( INTEGER range <> ) of EL_TYPE;
    with function "<" ( A, B : EL_TYPE ) return BOOLEAN
        is <>;
  -- )
procedure GEN_SORT ( X : in out AR_TYPE );
```

This generic procedure is a template for sorting procedures. A concrete sorting procedure can be obtained by substituting a linear array type, its component type, and the operator "<" acting on values of the component type. The specification of such a procedure would be

```
procedure SORT_INTEGERS ( X : in out INTEGER_ARRAY );
```

- *Exceptions* and the recovery from raised exceptions can be used for the description of all abnormal events. E.g., when the declaration of an exception SINGULAR_MATRIX is available, a routine can execute the statement

```
raise SINGULAR_MATRIX;
```

to cause abnormal termination of the called routine. However, the user can catch raised exceptions in so-called *exception handlers,* in order to execute some recovery action.

- The concept of *tasks* is a high-level concept for describing in a readable and reliable way concurrent activities and the safe communication between tasks, and between a user and his tasks. Tasks can be used to describe in a clear way the provision of numerical services to processes in embedded systems, but also the processing of distributed computations for special architectures, especially for general-purpose multi-processors.

A conclusion of the NPL/CWI investigation was that Ada is a useful language and several Ada features are of much interest for the redesigning and construction of large scientific libraries. Difficulties caused by idiosyncracies of the language syntax and semantics can be overcome satisfactorily in most of the cases, and solutions were offered. Moreover, since elementary mathematical provisions in Ada style were expected to be indispensable for Ada programmers, a standard definition and the early implementation of basic functions packages were emphatically recommended in [3].

In [4] also examples of library components were given, and language items were listed which ought to be cleared (by Ada implementors) or preferably changed in a future language revision.

For more information and details one is referred to the complete report [4].

4. Future activities

Members of the CWI Numerical Mathematics Department are currently co-operating with colleagues from six European Community countries (and Sweden) in the EC-supported *Ada-Europe Numerics Working Group.* This group has already produced several documents on all aspects of the implementation of numerical libraries in Ada and the design of new methods. Co-operation is sought with a US counterpart.

A problem the group is facing is that the need for newly-designed mathematical software with properties that Ada can provide (reliable, well-designed, readable, and portable modules in coherent packages) is not generally recognized. Ada offers the possibility of imposing a hierarchical structure onto libraries, through encapsulation of related declarations. Furthermore, consistency checks of passed parameters are maintained throughout the library's life cycle. Finally, relying on the mathematical provisions made available by a computer manufacturer has sometimes been a bad experience. It is regrettable that the very experts on the analysis of the results of long computations have been ignored so often.

Meanwhile, co-operation between the CWI and the Numerical Algorithms Group (NAG, Oxford) and NPL (among others) has been established, with the final aim of making all common numerical provisions available to Ada programmers. As a start, pilot implementations will be made of basic modules of numerical libraries in Ada, for which subsidies have been granted.

REFERENCES

1. ANSI/MIL-STD 1815 A. (1983). *Reference Manual for the Ada Programming Language.*
2. R.L. BABER (1984). Software development: science or patchwork? *CWI Newsletter 2,* 18-34.
3. J. KOK, G.T. SYMM (1984). A proposal for standard basic functions in Ada. *Ada Letters Vol. IV.3*, 44-52.
4. G.T. SYMM, B.A. WICHMANN, J. KOK, D.T. WINTER (1984). *Guidelines for the Design of Large Modular Scientific Libraries in Ada,* NPL Report DITC 37/84, CWI Report NM-N8401.

Calculating by Steam

Book Review

Harry Cohen

ANTHONY HYMAN (1982). *Charles Babbage - Pioneer of the Computer,* Oxford, University Press.
ANDREW HODGES (1983). *Alan Turing - The Enigma,* Burnett Books.

Those who would like to see the world's oldest computer, should go to London. In the Science Museum, between a 250 million times enlarged model of the DNA molecule and a collection of antique navigation instruments, there is a construction that reminds one of some old-fashioned gas-meters. It is the Difference Engine built by Babbage in 1832 and still in perfect working order. Of course, it is not the oldest calculating aid. The abacus has been in use for thousands of years and the first multiplicator, the slide rule, was invented in the seventeenth century. Also from those days are the first calculators that deserve the name of machine. The ingenious clockworks of Pascal and Leibniz could add or even multiply numbers, but they were not automatic machines. When a calculation involved more than one step, the result had to be read each time and the apparatus readjusted. The difference engine, however, only needs a few starting instructions and then goes through the whole cycle without any additional assistance. It is as automatic as a modern washing-machine.

The England of 160 years ago, in which this engine originated, was a quiet little world in which production was still largely manual. Electricity was known, of course, as a natural phenomenon, but industrial applications were not yet thought of. Even the use of steam was still in its infancy. Darwin and Marx had not yet disturbed mankind. There was no place in London that was more than a quarter of an hour's walk from the edge of town. Few realized that the Industrial Revolution had already begun.

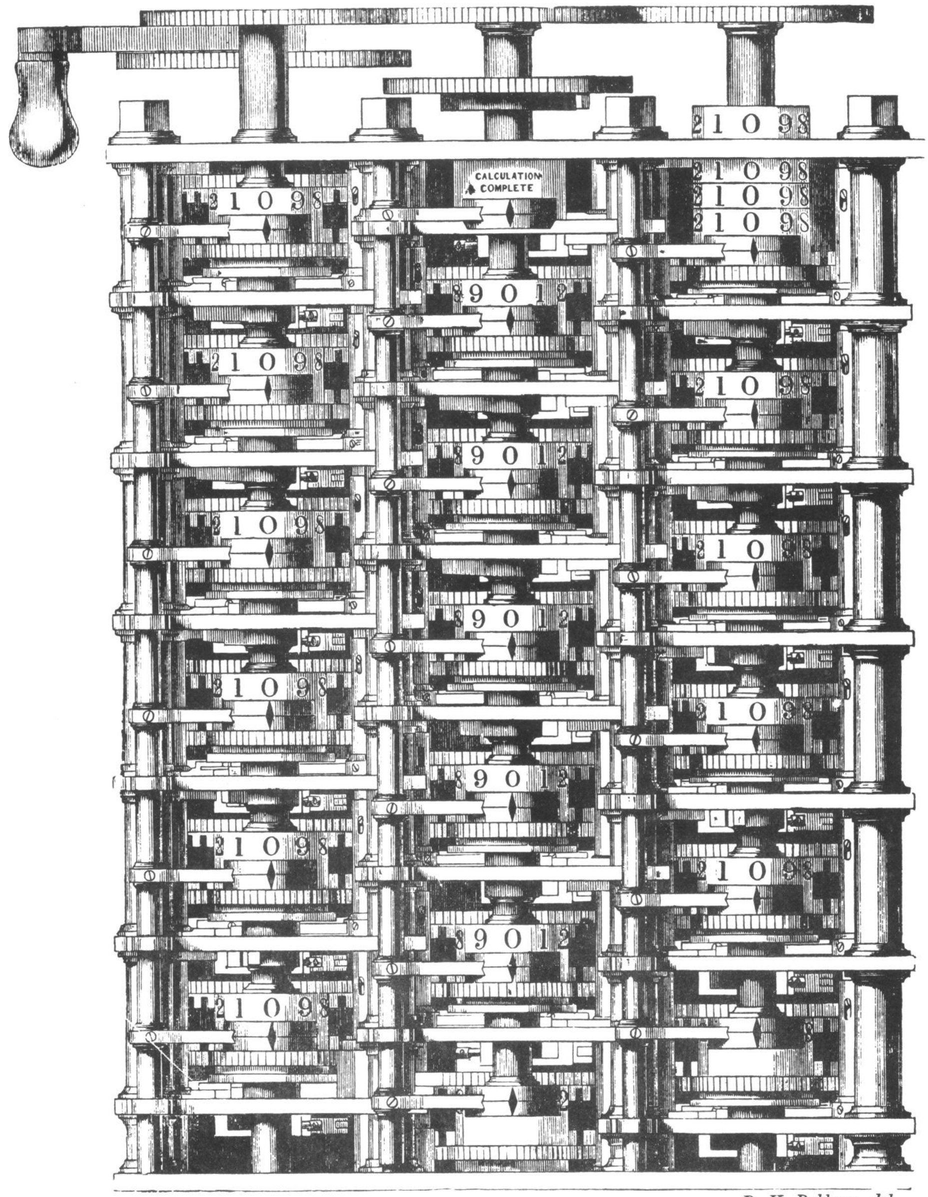
Charles Babbage (1791-1871), however, had seen it all coming. He belonged to the London upper middle class and had sufficient means to occupy himself all his life with unpaid scientific research. Hyman's biography gives a good description of this social background. Babbage's family life, his travels, his circle of friends: all is vividly portrayed. The discussion of Babbage's books gives us an idea of how universal the work of one man could still be in those days. In a theological treatise, for example, he tried to show that the plan of creation corresponds to the pattern of a quartic equation. But first and foremost he was interested in mathematics and physics, engine-building, and economics, three branches of science that were closely connected in his world-picture. Unlike the classical economists Adam Smith and Ricardo, he did not consider agriculture but industry as the pivot of the economic system. And the progress of industry depended on science and technology. This point of view was behind his decision on how to spend his life.

He first had an inspiration when round about 1820 he met a French engineer who applied the principle of the division of labour, as described by Adam Smith, to the making of logarithmic tables. The success was amazing. The whole job, that would otherwise have taken a lifetime, was now done in a few years (division of labour: 6 scientists, 8 trained assistants, and ca. 60 executors who could only do additions and subtractions). Babbage's ideas went much farther. An operation that consists of the continuous repetition of a simple action, such as addition, could in principle be mechanized. That means better quality (in this case fewer mistakes), faster results, and cheaper production. Driven by steam it might become even faster and cheaper. Babbage began to dream of logarithmic tables 'as cheap as potatoes'.

Shortly after he began to design a machine that would make mathematical tables. The working of such a machine is not really difficult to understand. Suppose we want to have a table of the squares of all the numbers from 1 to 1000, i.e., 1, 4, 9, 16, 25, etc. If we subtract from each square the preceding square, we find the so-called difference series: 3, 5, 7, 9, etc. These numbers can also be found in another way, viz., by starting with 3 and then adding 2 each time. Having found the difference series in this way, the computation of the squares in only a simple trick:

$$
\begin{aligned}
1+3 &= 4\\
4+5 &= 9\\
9+7 &= 16\\
16+9 &= 25
\end{aligned}
$$

This table then can be obtained by means of addition. For a great many other mathematical tables there are similar tricks for computing each term by two or more additions of preceding terms. They are always based on a difference series, hence the name Difference Engine. In this way there is no need for multiplication. This is a great advantage, for addition is easier for a machine (just as it is for a human being) than multiplication. So Babbage could keep this part of his machine quite simple. The rest of the mechanism, however, occasioned so much brain-racking that its design and construction took over ten years. The outcome was the Difference Engine which, after the necessary



B. H. Babbage, del.

Impression from a woodcut of a small portion of Mr. Babbage's Difference Engine No. 1, the property of Government, at present deposited in the Museum at South Kensington. (Facsimile of frontispiece from 'Passages from the Life of a Philosopher' published in 1864.)

adjustments, could work quite independently. The results were even recorded in matrices of printing type, so that both reading and printing errors were avoided. The whole contraption, however, had to be kept in motion by turning a handle without interruption; the dream of calculating by steam was never realized.

Babbage still had all sorts of improvements in mind, but for financial reasons he had to leave it at this first model. Originally, the project was subsidized by the government, but this was stopped by the politicians' short-sightedness. They did not even want to take over the machine. Eventually, it was put in Babbage's drawing-room, and moved to a museum later on.

Babbage realized that he would never be able to build another complete machine, but he had sufficient means at least to continue to design and experiment. His new plans concerned a far more sophisticated calculating engine, the Analytical Engine, which could also solve mathematical equations. With each improvement the design began to resemble a modern computer more and more closely. For example, there was a clear division between the memory and the processing unit (Babbage called it 'store' and 'mill', according to the layout of cotton mills in those days). Both parts were run by a 'control'. To this end the control was given the necessary commands on punched cards made of tin in which the program was recorded. For the output of the results Babbage had first thought of the matrix press of the Difference Engine or of a line printer. Later it proved to be simpler to have the results recorded by the machine on punched cards that could then serve as a program for an automatic printing press. In this way, moreover, human errors were avoided completely.

Quite a modern feature were the special commands to react to interim results. In the design of the Analytical Engine punched cards were connected in such a way as to form a chain. Normally, they would pass the 'control' in fixed order. After one of the special commands, however, the mechanism would depart from this pattern whenever an interim result satisfied certain conditions. The whole chain would then move a few places, for example. In other words: an automatic modification of the program. This approaches what is nowadays called conditional branching. (Without this facility our computers would not be so uncannily clever; Hodges calls it the mechanization of the word IF.) This is as far as the resemblance to a modern computer goes. In Babbage's machine the binary system was (deliberately) not used and the program could not be stored in the memory. Of course, the Analytical Engine was not electronic, nor even electrical. This made the parts so large and their movements so slow that the working of the machine, if it would ever have been built, could have been followed with the naked eye. The 4.50 meter high colossus would have needed a few seconds for an addition and a few minutes for a multiplication. The whole machine has been described by Ada, Countess of Lovelace, a daughter of Byron. According to her, its features were so universal that in principle it could compose music. She also wrote some programs for the engine.

The question has been raised whether at this stage Babbage's plans were still realistic. Hyman is inclined to think that, with sufficient funding, construction would have been technically feasible. But Babbage was far ahead of his time. After 1840 the gap between pure science and the art of engineering grew wider and wider, and the Engine 'fell through this gap into a century's oblivion.'

Indeed, for the next hundred years little was heard about calculating engines. Babbage's brain-children were sleeping in the Science Museum. The mathematics student Turing, who had been in Cambridge since 1931, was not aware of their existence. He, too, had invented a calculating machine, but one that will never be on view in a museum. Of the design only one part is known, viz., a tape that is divided into little squares. It can move one square at a time, either forwards or backwards. Then there is a simple operation: if the new square is empty, it may be marked with a cross; if it already contains a cross, it may be erased. Then there is another cycle. Turing has shown that such a tape-machine, if supplied with the necessary clockwork, can do additions. After some changes in the mechanism it can also do multiplications or other arithmetical operations. He even described a universal machine that could do 'anything'. We shall never know, however, how it was supposed to function, as Turing never worked out the technical aspects of his ideas. There are no models or blueprints. Indeed, it never was his intention to construct anything. The so-called Turing machines were no more than abstract constructs intended to give a precise meaning to the notion of 'effective procedure', which needed clarification in the context of mathematical logic at that time.

Alan Turing (1912-1954) was regarded as an eccentric in Cambridge, and also in later years he always remained the incredibly intelligent outsider in whatever circles he moved. In his biography, Hodges has treated this aspect at great length. Alan's parents, who lived in India, sent him to England when he was two years old, to be brought up by strangers. He became a withdrawn little boy, but won all the prizes at school. Later he applied himself to such an individual sport as long-distance running, and as an adult he asked his mother a teddybear for Christmas. His fellow students were shocked by the frankness with which he admitted his homosexual inclinations. As a topic for conversation it was not taboo in Cambridge, but in those days the actual practice of it was limited to certain exclusive circles (hence, 'higher sodomy') to which Turing was not admitted. After all, this was the time in which King George V is supposed to have said: 'I thought men like that shot themselves.'

For his intellectual achievements, however, and this is Cambridge, too, he was openly honoured and rewarded. Yet, he left for America in 1936 to graduate. While there, he built an apparatus in his spare time derived from the Turing machines, with which text could be encoded. In the mean time, Europe was heading for war. Information was as important as guns were, and it was necessary to intercept as much as possible of the wireless communications between the enemy forces. But naturally the Germans sent their messages in code, so that each message had to be deciphered. To that end Churchill created a new service, the famous Bletchley Park. A puzzle club of well-meaning amateurs rapidly developed into a tight organization of 10,000 people (mostly women, by the way). Turing, who was back in England by now, was put in control of a department that was concerned with the movements of the enemy submarine fleet. German submarines were particularly active in the Atlantic. The outcome of the war depended on American material (and later on American men), and supplies were almost exclusively by sea. There was no lack of messages, but decoding often took so long that in the mean time the information had become useless. That was why Turing concentrated all his efforts on cracking the code itself. The code was defined by the wiring of the Enigma, the electro-mechanical coding and decoding machine, which belonged to the equipment of every German unit.

Turing replaced primitive techniques of trial and error by refined statistical analyses and introduced punched cards. During 1941 they succeeded in reducing losses of ships by 50%. This success did not last long, for the Germans regularly changed the Enigma key setting. On the other hand, they were so careless as to transmit their weatherforecasts, the contents of which were easily guessed, in the same code. Hodges thinks that they never realized that their code was being tracked all the time. Failure of a submarine mission was invariably attributed to espionage or treason, never to the silent crew of Bletchley Park, 'the geese who laid the golden eggs and never cackled,' as they were called by Churchill.

In 1942 Turing was sent to the United States to learn about the use of electronics in data handling. The subject was in the air, mathematicians and physicists all over the country were doing research, but all threads came together at one man, Professor Von Neumann of Princeton University. Turing had met him several times. Although scientifically they were on the same track, their approach was as different as can be. While the lonely hobbyist Turing soldered his own models, Von Neumann had organized his project as a large-scale enterprise. He visited universities, was on all committees, had his own professional journal, and made an immortal name for himself in the history of computer science.

Having returned to England in 1943, Turing changed his course on the basis of the newly acquired knowledge. Bletchley Park could now do without him, and he applied himself to the construction of a machine for speech-encipherment that would secure military telephone-traffic across the ocean. But his imagination was already much farther ahead. With the use of electronics the Turing machine, that was originally meant for thought experiments only, could now actually be built. It would even be possible to imitate the human brain by electronic means (in those days radio valves!). Of course this imitation was not in a physiological or psychological sense, but as a logical system. As soon as the war was over, Turing took up a post in a government laboratory to realize this vision. The design of the ACE (Automatic Computing Engine; the use of the word 'engine' is in honour of Babbage) was largely due to him. It promised to be a very fast computing machine with an enormous memory, but for the rest the hardware was a reflection of his modest lifestyle: a minimal machine, no built-in gadgets, the type of installation that is appreciated by very clever programmers only. It could have become the first real computer, but Turing had little or no consideration for 'user-friendliness'. Because of his inflexible behaviour, construction was out of the question for the time being. After all kinds of difficulties he left the laboratory.

After this incident, his fame declined. In 1948 he eventually accepted an appointment at Manchester University. There they had just finished building a computer, so that there was little original work left for him to do. It did not seem to bother him; his attention was now directed rather towards such questions as are frequently heard again these days: can a computer think? Well, said Turing, let's first see if a computer can do arithmetic. Each calculating machine is designed in such a way that if the input is 47+21, the output is 68. In Babbage's engine one could actually see the three numbers (the positions of the gear-wheels). In an electronic machine, however, they are not really present; those investigating the inside will only find some pulses darting to and fro. A computer does not do arithmetic in the way a human being does. It does come up with the desired result, but it is found in a completely different way. If you insist on still calling this 'doing arithmetic', was Turing's conclusion, then with as good reason I can state that in principle a computer can think. The same goes for playing chess, learning, making decisions, etc. He expected, by the way, that by the end of this century the definitions of all of these words would have been sufficiently expanded to make such discussions superfluous. (Compare the word 'computer' that 15 years earlier had been used for human calculators only.)

In Manchester, too, he was the eccentric genius again. People laughed about that lanky person who put an old tie round his waist to keep his trousers up, who changed his bicycle with his own hands into a moped, and in spring wore a gasmask for hayfever. Shrink-proof clothing he washed himself, the rest went to his mother.

His mastery as a programmer was openly acknowledged. He could make a computer do conditional branching, even if this was not provided for in the hardware. Primitive peripherals were forced to question and answer techniques in a way that is very much like a modern teenager playing games with his personal computer nowadays. Only once was he outrivalled by a colleague who wrote a program that made the computer (then still provided with a hooter) play the national anthem when it had performed its task.

In the mean time the engineers around him were a step ahead again; they were experimenting with transistors. Alan Turing was not involved. As Hodges points out, he had become the Trotski of the computer revolution.

In 1952 he was arrested for homosexuality. He was placed on probation for one year, with the condition that he submitted to a hormone treatment that rendered him temporarily impotent. A year after the end of the probation term, a few days before his forty-second birthday, he killed himself.

Translated from Dutch from: NRC Handelsblad, Supplement 'Mens en Bedrijf', 21 March 1984

Abstracts

of Recent CWI Publications

When ordering any of the publications listed below please use the order form at the back of this issue.

CWI Tract 2. J.J. Dijkstra. *Fake Topological Hilbert Spaces and Characterizations of Dimension in Terms of Negligibility.*
AMS 57N20, 54F45, 54G99, 57N15; 109 pp.

Abstract: The main result of this monograph is the construction of a sequence X_{-1}, X_0, X_1,... of topologically complete AR spaces that have, among other things, the following properties: (1) X_n has the weak discrete approximation property; (2) homeomorphisms between compact subsets of X_n can be extended; (3) $X_n \times X_n$ is homeomorphic to ℓ^2, the separable Hilbert space; (4) if $A \subset X_n$ is a σ-compactum, then A is strongly negligible iff dim $A \leqslant n$; (5) if $A \subset X_n$ is a compactum of fundamental dimension at most n, then A is negligible (in particular, if $B \subset X_n$ is an m-cell, then B is negligible and B is strongly negligible iff $m \leqslant n$).
Our results were motivated by Toruńczyk's theorem that every complete AR with the strong approximation property is homeomorphic to ℓ^2. Since in ℓ^2 every σ-compact set is strongly negligible (Anderson), (4) shows that $X_n \not\approx \ell^2$ and that $X_n \not\approx X_m$ if $n \neq m$. The spaces X_n show that the properties (1,2,3,5) and the "if" part of (4) do not characterize ℓ^2 and therefore block a possible generalization of Toruńczyk's theorem. As a by-product we get a characterization of topological dimension in terms of negligibility. Our construction is inspired by ideas in Anderson, Curtis & van Mill, Trans. Amer. Math. Soc. **272** (1982) 311-321, but to get (4) and (5) a more delicate process was necessary and quite a lot of additional machinery had to be developed.

CWI Tract 12. W.H. Hundsdorfer. *The Numerical Solution of Nonlinear Stiff Initial Value Problems*
AMS 65L05; 138 pp.

Abstract: In this monograph an analysis is presented of a general class of one-step methods for the numerical solution of stiff initial value problems. This class of methods includes implicit Runge-Kutta methods as well as semi-implicit methods such as Rosenbrock methods, ROW-methods and adaptive Runge-Kutta methods. The main subjects we consider are the feasibility of the methods

(meaning that the implicit - algebraic - equations are uniquely solvable), and their nonlinear stability properties.

CWI Tract 13. D. Grune. *On the Design of ALEPH.*
CR D.3.1, D.3.4, F.4.2; 194 pp.

Abstract: The tract 'On the Design of ALEPH' emphasizes the similarity that exists between grammars and programs. The relation between the input and the output of a program can be described by a (two-level or affix) grammar, which can then be viewed as parsing the input while at the same time producing the output. The context information gathered during parsing and needed during production is kept in parameters (metanotions or affixes). The tract explains the above principles in detail and then proceeds to base a practical system on them, ALEPH. A global design of a compiler for ALEPH is given, augmented with a detailed description of some of its parts. The ALEPH Manual is included in an appendix.

CWI Tract 14. J.G.F. Thiemann. *Analytic Spaces and Dynamic Programming.*
AMS 28A05, 28A33, 49C20, 60B99, 90C39; 96 pp.

Abstract: Parts of the theory of analytic topological spaces are developed within a purely measure-theoretic framework, and applied to dynamic programming. This results in a formalism for dynamic programming involving no topology. Moreover, this formalism allows more general strategies than existing formalisms. The main measure-theoretic tools used, viz. universal measurability, Souslin sets, and a measurable-space structure on sets of probabilities, are treated in an introductory chapter, which makes the exposition self-contained.

CWI Tract 15. F.J. van der Linden. *Euclidean Rings with Two Infinite Primes.*
AMS 12A45, 12A25, 12A30, 12A90, 12J10, 10C25, 10E15, 10E20, 14G15; 196 pp.

Abstract: An infinite prime of a subring of a global field K is an equivalence class of non-trivial valuations of K that does not correspond to a prime ideal of the ring. In this book all earlier results about the classification of (norm-) Euclidean subrings with two infinite primes of global fields are combined. Examples of rings with two infinite primes are the rings of integers of real quadratic, complex cubic, and totally complex quartic fields. Apart from earlier results also new results are given. In particular a total classification is given for subrings with two infinite primes in complex quadratic fields. For rings of integers of certain quartic fields new results are derived. For instance all Euclidean rings of integers of cyclic quartic totally complex fields are determined. As a slight extension to the subject of this book not only Euclidean rings are investigated but also rings with a Euclidean ideal class.

CWI Syllabus 3. W.C.M. Kallenberg et al. *Testing statistical hypotheses: worked solutions.*
AMS 62F03, 62H15, 62G10; 310 pp.; **key words:** hypothesis tests, invariance, unbiasedness.

Abstract: E.L. Lehmann's book 'Testing Statistical Hypotheses', Wiley & Sons, New York, was first published in 1959 and has since become a classic in the statistical literature. It remains the standard introduction to hypothesis testing, giving special attention to the role of the principles of unbiasedness and invariance. Its 228 exercises continue to stimulate and challenge. This CWI syllabus contains complete worked solutions to all the exercises in 'Testing Statistical Hypotheses'. It should be especially useful to those using the book in statistics courses and for private study.

CWI Syllabus 4. J.G. Verwer (ed.). *Colloquium 'Topics in Applied Numerical Analysis'.* Vol. 1.
AMS 65XX06; 253 pp.

CWI Syllabus 5. J.G. Verwer (ed.). *Colloquium 'Topics in Applied Numerical Analysis'.* Vol. 2.
AMS 65XX06; 229 pp.

Abstract: The colloquium 'Topics in Applied Numerical Analysis' was held at the Department of Numerical Mathematics of the CWI during the academic year 1983/1984. The aim of this colloquium was to draw attention to the widespread use of numerical mathematics in real life scientific problems, as well as to foster co-operation between mathematicians working in an academic environment and representatives from industries and institutes where the numerical solution of real life problems is studied. The proceedings, consisting of two volumes, contain in complete form all 24 papers presented by the speakers in the colloquium. The greater part of the papers deal with practical problems, mainly arising in the engineering sciences.

CS-R8413. P.J.W. ten Hagen & J. Derksen. *Parallel input and feedback in dialogue cells.*
CR I.3.6; 15 pp.; **key words:** interaction, man-machine communication, Computer Graphics.

Abstract: In this paper a specification method for interaction is outlined based on a new programming concept called 'dialogue cells'. The method supports parallel input and separation of a dialogue part from the algorithmic part of an application. Graphics interaction can be fully integrated. Some problems associated with parsing parallel inputs are analyzed.

CS-R8416. P.M.B. Vitányi. *Distributed elections in a ring of processors using Archimedean time.*
AMS 68A05, 68B20, 94C99; CR C.2, D.4, F.2.2; 13 pp.; **key words:** distributed control, local area networks - rings, operating systems, communication management - message sending, algorithms using time, Archimedean time distributed systems, time-independent correctness and termination, robustness, accelerated efficiency by improved synchronicity, extrema-finding in a ring network.

Abstract: Unlimited synchronism is intolerable in real physically distributed computer systems. Such systems, synchronous or not, use clocks and timeouts. Pure synchronism can not exist either. Therefore the appropriate model is in between synchronous and asynchronous: the magnitudes of elapsed absolute time in the system need to have *finite* ratios, that is, they satisfy the axiom of Archimedes. Under this restriction of asynchronicity logically time-independent algorithms can be derived which are better (in number of message passes) than is possible otherwise. An example is the problem of decentralized extrema-finding in a circular configuration of processors, that is, reaching distributed agreement on the choice of a single processor. Each processor has a unique name (integer) and does not know the size of the ring. The election can be instigated by any processor at any time: also when an election is in progress but the processor is as yet unaware of it. Asynchronous rings have been shown to need at least $N \log N$ message passes on the average and a unidirectional order $N \log N$ worst-case message pass solution is known. We give a logically time-independent (unidirectional deterministic) solution using order sN message passes in the worst case, with s a measure of the asynchronicity of the system. For synchronous systems $s=1$ and even for asynchronous systems we can eliminate s by choice of the parameters in the Protocol. The result depends on the processors using subjective clocks but its correctness and termination is independent of the time assumptions. Consequently, some basic subtilities associated with distributed computations are highlighted. For instance, even the known nonlinear *average-case* lower bound on the number of message passes is cracked by the *worst-case* performance of the new solution. For the synchronous case, in which the necessary assumptions hold *a fortiori,* the method is -asymptotically- the most efficient one yet, and of optimal order of magnitude. The deterministic algorithm is of -asymptotically- optimal bit complexity, and, in the synchronous case, also yields an optimal bound to determine the ring size. All of these results

improve the known ones. A result on distributed sorting in unidirectional rings is discussed.

CS-R8417. S.J. Mullender. *A secure high-speed transaction protocol.*
AMS 68A05, 68B20; CR C.2.2, C.2.4, D.4.4; 9 pp.; **key words:** transaction protocols, connectionless protocols, capabilities, local-area networks.

Abstract: Most computer networks use a byte stream protocol for communication between processes, which suffer from two important drawbacks: the addressing mechanisms provided are often process-dependent or location-dependent, and communication is slow. While carrying out research into distributed operating systems at the Free University of Amsterdam and the Centre for Mathematics & Computer Science, we have developed a transaction-oriented transport protocol for the *Amoeba* distributed operating system, aimed for high speed, with an addressing mechanism that is not only more general, but provides a protection mechanism as well. The basic mechanism for communication between processes is the transaction: a client process sends a request to a server process, which carries out the request and returns a reply. Protection is provided by using ports, chosen from a sparse address space, for addressing services. These ports serve as a "capability" for communication with the service. Through its simplicity, the transaction protocol achieves much higher transmission rates than other protocols executing on similar hardware (about 300 Kbytes/sec process-to-process). The protection mechanism and the mechanisms for realising high transmission speeds, are described.

CS-R8419. S.J. Mullender. *Distributed systems management in wide area networks.*
AMS 68A05, 68B20; CR 2.2, C.2.4, D.4.4, D.4.7; 10 pp.; **key words:** service model. distributed systems, wide-area networks, COST-11.

Abstract: While quite a few distributed operating systems for local-area networks exist, hardly any work has been done to date on distributed operating systems for wide-area networks. In Europe, a number of public networks are now operational, with gateways between some of them. However, the use of these networks is still mostly restricted to *"remote login"* and, in some cases, simple *file transfer* operations. To study these problems and to find structural solutions for efficient and simple use of national and international networks the working group *"Distributed Systems Management"* was founded within COST 11. Recently, this working group has submitted a research proposal to COST 11 to realise an infrastructure for the implementation of distributed services in a wide-area network in a European collaborative effort. The model underlying the research is the *service model* used in many local-area network distributed operating systems. The research project is described, and the proposed infrastructure is discussed in some detail.

CS-R8420. J.A. Bergstra & J.W. Klop. *A complete inference system for regular processes with silent moves.*
AMS 68B10, 68C01, 68D25, 68F20; CR F.1.1, F.1.2, F.3.2, F.4.3; 59 pp.; **key words:** concurrency, process algebra, regular processes, complete inference system, parallel composition, invisible steps.

Abstracts: We study the notion of bisimulation between process graphs with silent or invisible steps (τ-steps). This leads to a normalisation or minimalisation result for regular processes, and furthermore to a complete proof system for regular processes with τ-steps and subject to operations + (alternative composition), . (sequential composition) and || (parallel composition or free merge), thereby answering a question of Milner and proving the consistency of a version of Koomen's fair abstraction rule.

CS-R8421. J.A. Bergstra & J.W. Klop. *Algebra of communicating processes.*
AMS 68B110, 68C01, 68D25, 68F20; CR F.1.1, F.1.2, F.4.3; 42 pp.; **key words:** concurrency, communicating processes, process algebra, merge, data flow networks, regular processes, invisible steps.

Abstract: 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.

CS-R8422. J.W. de Bakker & J.N. Kok. *Towards a uniform topological treatment of streams and functions on streams.*
AMS 68B10; CR D.3.1, F.3.2, F.3.3; 18 pp.; **key words:** streams, functional programming, denotational semantics, metric topology, Hausdorff metric, trace theory, typed lambda calculus, concurrency.

Abstract: We study the semantics of functional languages on streams such as Turner's SASL or KRC. The basis of these languages is recursive equations for (functions on) finite or infinite sequences. The paper presents a start towards a mathematical (denotational) description of such languages using tools from metric topology. The description is based on the Banach fixed point theorem and a restricted version of a typed lambda calculus. To a system of recursive stream (function) declarations a system of functions is associated in an appropriate topological domain. These functions have to be contracting in certain arguments and non distance increasing in others; a syntax is designed which ensures the right interplay between these conditions. Nondeterminism is handled by considering compact sets of streams, and preservation of compactness is another important technical issue. Not all concepts in a language such as KRC are covered, and some indications on possible extensions of the framework are provided.

CS-R8423. J. Seiferas, P.M.B. Vitányi. *Counting is easy.*
AMS 68C40, 68C25, 68C05, 94B60, 10-00; CR F.1.1, F.1.3, F.2.3, B.3.2, E.1, E.4; 8 pp.; **key words:** counting, number representation, coding, counter machine, multicounter machine, augmented counter machine, real-time simulation by oblivious one-tape Turing machine.

Abstract: An exposition is presented of the recent result that many independent counts with simultaneous zero-check can be maintained in real-time on an oblivious single-head tape unit using the information-theoretical storage minimum. Some extensions of that result are also given. The treatment is informal and aims at making the central ideas more transparent.

CS-R8424. S.J. Mullender, P.M.B. Vitányi. *Distributed match-making for processes in computer networks.*
AMS 68C05, 68C25; C.2.1. F.2.2, G.2.2; 16 pp.; **key words:** locating objects, locating services, computer networks, network topology.

Abstract: Locating services in a computer network is usually done by broadcasting "where are you" messages. In many networks this is an efficient method, because the network medium is itself a broadcast medium. In other networks, such as large store-and-forward networks, broadcasting is considerably more costly than sending a message directly to its destination. Here we examine methods for locating services that are less expensive than broadcasting in terms of message passes or "hops." For these methods we investigate the complexity in terms of needed storage, in terms of message passes and in terms of processing needed. The general problem consists of distributed match-making between processes, such as server processes and client processes, in computer networks. The processes are assumed to be mobile and not have fixed addresses. When the servers assist the clients in getting themselves found, it appears that, in many mesh networks, match-making can be done in $\Theta(\sqrt{N})$ message passes, where N is the number of nodes in the network. Conventional broadcast methods for locating services need a minimum of $\Theta(N)$ message passes to do the broadcast. The theoretical limitations of distributed match-making are established, and the techniques are applied to several network topologies.

CS-R8501. J. Heering. *Partial evaluation and ω-completeness of algebraic specifications.*
AMS 03B40, 03C05, 68B10, 68C20; CR D.1.2, D.3.4, F.3.2, F.4.1, I.1.1, I.2.2; 14 pp.; **key words:** algebraic specification, ω-completeness, initial algebra semantics, equational logic, structural induction, ω-rule, program transformation, program optimization, partial evaluation, mixed computation, symbolic computation, propagation of incomplete information, constant propagation, combinatory logic, ω-extensionality.
Abstract: Suppose $P(x,y)$ is a program with two arguments, whose first argument has a known value c, but whose second argument is not yet known. *Partial evaluation* of $P(c,y)$ results (or rather: should result) in a specialized residual program $P_c(y)$ in which "as much as possible" has been computed on the basis of c. In the literature on partial evaluation this is often more or less loosely expressed by saying that partial evaluation amounts to "making maximal use of incomplete information." In this paper a precise meaning is given to this notion in the context of initial algebra specifications and term rewriting systems. It turns out that, if maximal propagation of incomplete information is to be achieved, as a first step it is necessary to add equations to the algebraic specification in question until it is *ω-complete* (if ever). The basic properties of ω-complete specifications are discussed, and some examples of ω-complete specifications as well as of specifications that do not have a finite ω-complete enrichment are given.

CS-N8501. N.W.P. van Diepen. *Integer-square-root. An example of and introduction to program transformations.* (In Dutch.)
AMS 68B10; CR I.2.2; 12 pp.; **key words:** program transformation, program verification.
Abstract: This paper is an intuitive introduction to the field of transformational programming. A short introductory description of partial correctness proofs in Hoare Logic is given. Next two exhaustive examples of correctness preserving program transformations are given. Both examples start with an evidently correct solution to the problem of computing the Integer-Square-Root ($\lfloor\sqrt{n}\rfloor$) of a given positive integer n. These programs are then optimized using program transformations.

OS-R8411. R.M. Karp, J.K. Lenstra, C.J.H. McDiarmid & A.H.G. Rinnooy Kan. *Probabilistic analysis of combinatorial algorithms: An annotated bibliography.*
AMS 68E10, 68C25, 90C05, 90C10, 90C35; 26 pp.; **key words:** probabilistic analysis, combinatorial algorithm, random graph, matching, stable set, colouring, Hamiltonian cycle, assignment, traveling salesman, location, linear programming, simplex method, bin packing, scheduling, knapsack, branch-and-bound, local search.
Abstract: This annotated bibliography reviews the literature on the probabilistic analysis of combinatorial algorithms for problems defined on (unweighted and weighted) graphs, Euclidean problems, linear programming, packing and covering problems, and of branch-and-bound and local search methods.

OS-N8402. M.W.P. Savelsbergh. *Vehicle routing and computer graphics.*
AMS 68V20, 69G30, 69G12; 6 pp.; **key words:** vehicle routing problem, interactive optimization, computer graphics, colour graphics, Graphical Kernel System.
Abstract: After introducing the Vehicle Routing Problem, we discuss the advantages of an interactive optimization approach and make some remarks on the underlying optimization method

to be used. In addition, we emphasize the importance of a colour graphics user interface and relate our experiences with the Graphical Kernel System graphics package.

OS-N8403. H.M.C.A. Hop. *Time dependent behaviour of queuing systems and the practical application of relaxation times.* (In Dutch.)
AMS 60K25, 69D58, 90B22; 37 pp.; **key words:** queuing system, relaxation time, time dependent behaviour, simulation.
Abstract: We study the amount of time required for the average number of customers in a network of $M/M/1$ stations to approach its stationary value up to 1%, given initial conditions that differ from the equilibrium situation. In particular the ratio between the above defined time and the relaxation time of the average number of customers in the network is investigated. For certain models this ratio is determined by the solution of the system of differential equations that describes the model. The numerical results are confirmed by simulations.

OS-N8404. M.W.P. Savelsbergh. *Vehicle routing and computer graphics.* (Dutch version of OS-N8402.)

NM-R8414. H.J.J. te Riele & R.W. Wagenaar. *Numerical solution of a first kind Fredholm integral equation arising in electron-atom scattering.*
AMS 65R20, 81GXX; 7 pp.; **key words:** Fredholm integral equation of the first kind, regularization, electron-atom scattering.
Abstract: The regularization method of Phillips and Tihonov is applied to a first kind Fredholm integral equation arising in the field of electron-atom scattering.

NM-R8415. H.J.J. te Riele. *Some historical and other notes about the Mertens conjecture and its recent disproof.*
AMS 01A55, 10A20; 6 pp.; **key words:** Mertens conjecture.
Abstract: This paper answers some questions about the recent disproof of the Mertens conjecture by Odlyzko and te Riele. In particular, the roles of Stieltjes and Mertens are sketched, and some comments are given on the electronic communication between Amsterdam and Murray Hill and on the publicity around the disproof.

NM-R8416. H.J.J. te Riele. *A program for solving first kind Fredholm integral equations by means of regularization.*
AMS 65R20, 65V05, 81GXX; 17 pp.; **key words:** Fredholm integral equation of the first kind, regularization, elastic electron-atom scattering, indirect measuring, dispersion relation.
Abstract: A program is described for solving a Fredholm integral equation of the first kind with help of the regularization method of Phillips and Tihonov. This type of problem frequently arises in the mathematical analysis of physical problems, such as elastic electron-atom scattering.

NM-R8501. B.P. Sommeijer, P.J. van der Houwen & B. Neta. *Symmetric linear multistep methods for second-order differential equations with periodic solutions.*
AMS 65L05; 9 pp.; **key words:** numerical analysis, second-order differential equations, periodic solutions, linear multistep methods.
Abstract: Special symmetric linear multistep methods for second-order differential equations without first derivatives are proposed. The methods can be tuned to a possibly a priori knowledge of the user on the location of the frequencies that are dominant in the exact solution. On the basis of such extra information the truncation error can be considerably reduced in magnitude.

Numerical results are compared with results produced by the symmetric methods of Lambert and Watson and the method of Gautschi.

MS-R8413. P. Groeneboom. *Brownian motion with a parabolic drift and Airy functions.*
AMS 60J65, 60J75, 62E20, 62G05; 45 pp.; **key words:** Brownian motion, parabolic drift, Airy functions, Cameron-Martin-Girsanov formula, Feynman-Kac formula, Bessel process, density estimation.

Abstract: Let $\{W(t): t \geqslant s\}$ be Brownian motion, starting at x at time s. The densities of first passage times of the process $\{W(t) - ct^2 : t \geqslant s\}$ are determined analytically in terms of Airy functions; the joint distribution of the maximum and the location of the maximum of this process is also expressed in terms of Airy functions. Corresponding results are given for two-sided Brownian motion. The structure of a jump process of locations of maxima of Brownian motion with respect to a family of parabolas is derived. This process plays a fundamental role in describing the limiting global behavior of certain estimators of densities and distribution functions. As a probabilistic side result the distribution of excursion integrals is obtained.

MS-R8414. C. van Eeden. *Mean integrated squared error of kernel estimators when the density and its derivative are not necessarily continuous.*
AMS 62G05; 11 pp.; **key words:** density estimation, mean integrated squared error, optimal kernel.

Abstract: Asymptotic properties of the mean integrated squared error *(MISE)* of kernel estimators of a density function, based on a sample $X_1, ..., X_n$, were obtained by Rosenblatt and Epanechnikov for the case when the density f and its derivate f' are continuous. They found, under certain additional regularity conditions, that the optimal choice h_{n0} for the scale factor $h_n = Kn^{-\alpha}$ is given by $h_{n0} = K_0 n^{-1/5}$ with K_0 depending on f and the kernel; they also showed that $MISE\ (h_{n0}) = \mathcal{O}(n^{-4/5})$ and Epanechnikov found the optimal kernel. In this paper we investigate the robustness of these results to departures from the assumptions concerning the smoothness of the density function. In particular it is shown, under certain regularity conditions, that when f is continuous but is derivative f' is not, the optimal value of α in the scale factor becomes 1/4 and $MISE\ (h_{n0}) = \mathcal{O}(n^{-3/4})$; for the case when f is not continuous the optimal value of α becomes 1/2 and $MISE\ (h_{n0} = \mathcal{O}(n^{-1/2})$. For this last case the optimal kernel is shown to be the double exponential density.

MS-N8401. P. Groeneboom. *An investigation into head-way time on a 2x2 lane motorway.* (In Dutch.)
AMS 62G05, 62P99; 40 pp.; **key words:** semi-Poisson model, Brantson's generalized queuing model.

Abstract: Measurements of head-way times on a 2x2 lane motorway were analysed with the help of various statistical models. The distribution of head-way times was separated into two components. One component corresponds to drivers whose speed is determined by the car in front of them ('followers'), the other corresponds to drivers for which this is not the case ('leaders'). Various models from the literature were further developed and large-sample variances of the estimates were determined. With analysis very clear differences were found in the structure of traffic at different locations, and the results also led to hypotheses on the difference in structure during wet and dry weather.

AM-R8415. H.E. de Swart & J.T.F. Zimmerman. *Tidal rectification in lateral viscous boundary layers of a semi-enclosed basin.*
AMS 35A35, 76D30; 15 pp.; **key words:** residual current, Strouhal number primitive perturbation method, global renormalization.

Abstract: The rectified flow, induced by divergence of the vorticity flux in lateral oscillatory viscous boundary layers along the side-walls of a semi-enclosed basin, is studied as a function of the Strouhal number, κ, equivalent to the Reynolds number. It is shown that for small Strouhal numbers the ratio of the rectified flow and the tidal current amplitude is proportional to κ, but for larger κ values the behaviour is exponential. The latter conclusion is reached at by using a global renormalization of the vorticity equation.

AM-R8416. H.A. Lauwerier & M.B. van der Mark. *Chaos and order in an optical ring cavity.*

AMS 58F14, 58F13, 78A10; 16 pp.; **key words:** iterative maps, chaotic behaviour, strange attractor.

Abstract: This study orginates from an attempt to understand the bifurcation behaviour in a certain two-dimensional iterative map considered by Ikeda et al. in a theoretical study of optical bistability. The map can be written in complex coordinates z and $\bar{z}$ in the following form: $z' = A + Bz \exp i(z\bar{z} - \beta)$. The complex coordinate z has the meaning of an electric field vector, A is the amplitude of the incoming wave, B measures the dissipation of the electric energy and β is called a mistuning parameter. The mapping is considered from a mathematical point of view by using the usual techniques of two-dimensional maps. The parameter B determines whether the map is Hamiltonian ($B = 1$), i.e. measure preserving, or dissipative ($B < 1$). Various plots are given showing the chaotic behaviour of the map, periodic points, continuous invariant curves, isolated structures, strange attractors, etc.

AM-R8417. H.J.A.M. Heijmans. *Structured populations, linear semigroups, and positivity.*

AMS 92A15, 47D05; 17 pp.; **key words:** structured populations, positive irreducible semigroup, Perron-Frobenius theory, essential spectrum, peripheral spectrum, renewal theorem, generation expansion.

Abstract: A general model represented by a first-order partial differential equation with boundary and initial conditions, describing the growth of a biological population is formulated and it is indicated how several physiological structures fit into this model. It is shown that the semigroup associated with the problem is positive and irreducible. Therefore the Perron-Frobenius theory of positive semigroups can be applied in order to obtain relevant information on the spectrum of the semigroup, and this can be used to characterize the large time behaviour of solutions.

AM-R8418. J. Aten & J. Grasman. *Quantitative models describing the kinetics of tumour cell proliferation: a comparison with experimental data.*

AMS 62P10; 6 pp.; **key words:** cell cycle, transition probability model.

Abstract: Pedigrees of cells from a mouse osteosarcoma line are analyzed. The generation time data of cells of 12 pedigrees is used to estimate the parameters in two transition probability models of the cell cycle.

AM-R8419. J.A.J. Metz & F.H.D. van Batenburg. *Holling's "hungry mantid" model for the invertebrate functional response considered as a Markov process. Part I: The full model and some of its limits.*

AMS 92A15, 60J25, 60K30; 46 pp.; **key words:** satiation, functional response, predation behaviour, structured population models, first order partial differential equations with transformed arguments, Markov processes, time scale arguments, approximating stochastic processes.

Abstract: In this paper we give an analytical reformulation of Holling's (1966) simulation model for invertebrate predatory behaviour. To this end we represent a population of predators as a frequency distribution over a space of (physiological) states. The functional response of a predator is calculated from the (stable) equilibrium of its state as a function of prey density. Starting from the general model various other models, some of them new and some of them old, are obtained as limit processes, the more interesting of which will be studied in further papers in this series.

AM-R8420. J.A.J. Metz & F.H.D. Batenburg. *Holling's "hungry mantid" model for the invertebrate functional response considered as a Markov process. Part II: Negligible handling time.*

AMS 92A15, 60J25, 60K30; 33 pp.; **key words:** satiation, functional response, predation behaviour, first order partial differential equations with transformed arguments, Markov processes, approximation stochastic processes, van Kampen expansion.

Abstract: In this paper we analyse a stochastic model for invertebrate predation taking account of the predator's satiation. This model approximates Holling's "hungry mantid" model when handling time is negligible (see part I). For this model we derive equations from which we can calculate the functional response and the variance of the total catch. Moreover we study a number of approximations which can be used to calculate these quantities in practical cases in a relatively simple manner.

AM-R8502. S-N. Chow, O. Diekmann & J. Mallet-Paret. *Stability, multiplicity, and global continuation of symmetric periodic solutions of a nonlinear Volterra integral equation.*

AMS 45D05, 45G10, 45M05, 45M10; 32 pp.; **key words:** singular perturbation, global Hopf bifurcation, Volterra convolution integral equation, invariant cone, periodic solutions, stability, slowly oscillating solutions, Floquet multipliers.

Abstract: Results on existence, multiplicity, stability, global continuation, and limiting behaviour when $\epsilon\downarrow 0$ of periodic solutions of

$$\text{(E)} \qquad x(t)=\frac{1}{2\epsilon}\int_{1-\epsilon}^{1+\epsilon} f(x(t-\tau))\,d\tau$$

are derived for the case of a nonlinear function f having certain monotonicity and symmetry properties. The proofs are based on the following two observations: (i) the right-hand side of (E) defines an operator which maps a cone of two-periodic functions with symmetry and positivity properties into itself; and (ii) slowly oscillating solutions of the linear variational equation correspond to dominant Floquet multipliers.

AM-R8503. H.A. Lauwerier. *The Pythagoras tree as a Julia set.*

AMS 30D05; 12 pp.; **key words:** Pythagoras tree, Julia set, iterated mapping, self-similarity.

Abstract: Various properties of the so-called Pythagoras tree are considered, especially with respect to iterated mappings, self-similarity, and Julia sets. The approach is based on binary representations of real numbers and on the use of complex variables.

AM-R8504. O. Diekmann, H.J.A.M. Heijmans & H.R. Thieme. *On the stability of the cell size distribution II. Time-periodic developmental rates.*

AMS 92A15; 23 pp.; **key words:** size-dependent population growth, reproduction by fission, first order partial differential equation with transformed argument, stable size distribution, spectral theory of strongly positive quasi-compact linear operators on Banach lattices, essential spectrum.

Abstract: A deterministic model for the growth of a size-structured proliferating cell population is

analyzed. The developmental rates are allowed to vary with time. For periodically varying rates stability of the cell size distribution is shown under similar conditions for the growth rate of individual cells as found before in the time-homogeneous case. Strongly positive quasi-compact linear operators on Banach lattices serve as powerful abstract tools. Finally the autonomous case is revisited and the conditions for stability are relaxed.

AM-R8505. O. Diekmann, R.M. Nisbet, W.S.C. Gurney & F. van den Bosch. *Simple mathematical models for cannibalism: A critique and a new approach.*
AMS 92A15; 17 pp.; **key words:** age structured population dynamics, cannibalism, egg eating predators, steady states, stability, Hopf bifurcation.
Abstract: In this paper we show how to incorporate a functional response in recent models of Gurtin, Levine, and others for egg cannibalism. Starting from a relatively complicated model with vulnerability spread over an age interval of finite duration ϵ, we arrive at a much simpler model by passing to the limit $\epsilon \downarrow 0$. It turns out that survivorship through the vulnerable stage is implicitly determined by the solution of a scalar equation. Subsequently we study the existence and stability of steady states and we find (analytically in a simple case, numerically in more general situations) curves in a two-dimensional parameter space where a nontrivial steady state loses its stability and a periodic solution arises through a Hopf bifurcation.

PM-R8414. E.P. van den Ban. *A convexity theorem for semisimple symmetric spaces.*
AMS 22E30, 22E46, 43A85; 42 pp.; **key words:** semisimple symmetric space, Iwasawa decomposition, convexity.
Abstract: We study an Iwasawa type projection related to a semisimple symmetric space and prove a generalization of Kostant's convexity theorem for it.

PM-R8415. A.M. Cohen. *Point-line characterizations of buildings.*
AMS 51B25; 18 pp.; **key words:** buildings, incidence systems, Lie geometry.
Abstract: A survey is given of recent results in synthetic Lie geometry: axiomatic characterizations of buildings in terms of points and lines. These notes are an extended version of a lecture given at the CIME (International Mathematical Summer Center) Meeting on 'Buildings and the Geometry of Diagrams', Como (Italy), August 30, 1984.

PM-R8501. T.H. Koornwinder. *Special orthogonal polynomial systems mapped onto each other by the Fourier-Jacobi transform.*
AMS 33A65, 33A30, 33A75; 10 pp.; **key words:** Fourier-Jacobi transform, Hankel transform, Whittaker transform, Jacobi polynomials, Laguerre polynomials, Wilson polynomials, continuous dual Hahn polynomials, tridiagonalization of differential operators.
Abstract: The Fourier-Jacobi transform, which generalizes the Mehler-Fock transform, is shown to map an orthogonal basis involving Jacobi polynomials onto an orthogonal basis involving Wilson polynomials. One limit case is the Hankel transform mapping Laguerre functions onto itself. Another limit case is the Whittaker transform mapping Laguerre functions to continuous dual Hahn polynomials. The orthogonal basis involving Jacobi polynomials tridiagonalizes the Jacobi function differential operator. Group theoretic interpretations are briefly discussed. This is a preliminary report not containing full proofs.

PM-R8502. E.P. van den Ban. *On the holomorphic continuation of the Iwasawa and a related decomposition.*
AMS 22E30, 43E85; 31 pp.; **key words:** semisimple Lie group, Iwasawa decomposition, holomorphic continuation, symmetric space.

Abstract: Let G be a real semisimple adjoint Lie group, G_c its complexification. In this paper we study the holomorphic continuation to G_c of a decomposition of G which essentially generalizes the Iwasawa decomposition. The results are of interest for the analysis on a semisimple symmetric space.

CWI Activities

Spring 1985

With each activity we mention its frequency and (between parentheses) a contact person at CWI. Sometimes some additional information is supplied, such as the location if the activity will not take place at CWI.

Study group on Analysis on Lie groups. Joint with University of Leiden. Biweekly. (E.P. van den Ban)

Seminar on Algebra and Geometry. Coxeter Groups and Combinatorics. Biweekly. (A.E. Brouwer)

Study group on Cryptography. Biweekly. (J.H. Evertse)

Colloquium 'STZ' on System Theory, Applied and Pure Mathematics. Twice a month. (J. de Vries)

Study group 'Biomathematics'. Lectures by visitors or members of the group. Joint with University of Leiden. (J. Grasman)

Study group on Nonlinear Analysis. Lectures by visitors or members of the group. Joint with University of Leiden. (O. Diekmann)

Progress meetings of the Applied Mathematics Department. New results and open problems in biomathematics, mathematical physics and analysis. Weekly. (N.M. Temme)

National Study Group on Statistical Mechanics. Joint with Technological University of Delft, Universities of Leiden and Groningen. Monthly. University of Amsterdam. (H. Berbee)

Lecture course on Basic Principles of Statistics of Dependent Observations (martingale approach) by A.N. Shiryayev (Steklov Math. Institute, Moscow). 6,7 and 8th May. (R.D. Gill)

Seminar on Probability Inequalities and Related Topics, given by M.L. Eaton. Joint with University of Amsterdam (and held there). Weekly. (R.D. Gill)

Progress meetings on Combinatorial Optimization. Biweekly. (J.K. Lenstra)

System Theory Days. Irregular. (J.H. van Schuppen)
Study group on System Theory. Biweekly. (J.H. van Schuppen)
National colloquium on Optimization. Irregular. (J.K. Lenstra)
Study group on Differential and Integral Equations. Lectures by visitors or group members. Biweekly. (H.J.J. te Riele)
Study group on Numerical Flow Dynamics. Lectures by group members. Every wednesday. (J.G. Verwer)
Study group on Hyperbolic Systems. Every wednesday. (P.W. Hemker)
Progress meetings on Numerical Mathematics. Weekly. (H.J.J. te Riele)
Seminar National Concurrency Project. Joint with Universities of Leiden, Utrecht, Nijmegen and Amsterdam. 22 February, 22 March and 24 May. (J.W. de Bakker)
National Study Group on Concurrency. Joint with Universities of Leiden, Utrecht, Nijmegen and Amsterdam. 18 January, 8 February, 8 March and 5 April. University of Utrecht. (J.W. de Bakker)
ESPRIT/LPC Advanced School on Current Trends in Concurrency. 10-21 June at 'De Leeuwenhorst', Noordwijkerhout. Invited speakers:
E.A. Ashcroft (SRI International, Menlo Park, USA), H.P. Barendregt (University of Utrecht, The Netherlands), M. Diaz (L.A.A.S., Toulouse, France), G. Levi (University of Pisa, Italy), E.-R. Olderog (University of Kiel, West Germany), A. Pnueli (Weizmann Institute, Rehovot, Israel), F.B. Schneider (Cornell University, Ithaca, USA), P.S. Thiagarajan (University of Aarhus, Denmark). (J.W. de Bakker)
Post-academic course on Modern Techniques in Software Engineering. 9,10,23 and 24 May. (J.C. van Vliet)
Post-academic course on *B*. 7 January - 9 May. Twice a week. (L. Geurts)
Study group on Graphics Standards. Monthly. (M. Bakker)
Study group on Dialogue Programming. (P.J.W. ten Hagen)

Visitors to CWI from Abroad

K. Burrage (University of Auckland, New Zealand) January - February. J.C. Butcher (University of Auckland, New Zealand) 4 January. W. Dahmen (University of Bielefeld, West Germany) 20 March. H. Dym (The Weizmann Institute of Science, Rehovot, Israel) 25 February. B. Fiedler (SFB 123 Heidelberg, West Germany) 11-22 March. M. Ghil (Courant Institute, New York, USA) 14 January. J.W.P. Hirschfeld (University of Sussex, Brighton, UK) 25 March. K.H. Hofmann (TH, Darmstadt, West Germany) 26-30 March. C.J. Holland (Department of the Navy, London, UK) 12-14 February. A. Korányi (Washington University, St. Louis, USA) 21-25 January. E.-R. Olderog (University of Kiel, West Germany) 21 March - 4 April. J.B. Orlin (M.I.T., Cambridge, USA) 1 September 1984 - 1 May 1985. H. Schlichtkrull (University of Copenhagen, Denmark) 24 February - 3 March. M. Smorodinski (University of Tel Aviv, Israel) 4 February. G.J. Székely (Eötvös L. University, Budapest, Hungary) 6-8 March. K. Voss (GMD, Bonn, West Germany) 22 February. L.D. Wittie (SUNY, Stonybrook, USA) 21-22 January.

Order Form for CWI Publications

Centre for Mathematics and Computer Science
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

☐ Please send the publications marked below on an exchange basis
☐ Please send the publications marked below with an invoice

	Publication code	Price per copy	Number of copies wanted
☐	CWI Tract 2 *)	Dfl. 16.70	....
☐	CWI Tract 12 *)	20.30	....
☐	CWI Tract 13 *)	28.60	....
☐	CWI Tract 14 *)	14.30	....
☐	CWI Tract 15 *)	27.40	....
☐	CWI Syllabus 3 *)	45.10	....
☐	CWI Syllabus 4 *)	36.90	....
☐	CWI Syllabus 5 *)	33.30	....
☐	CS-R8413	3.70	....
☐	CS-R8416	3.70	....
☐	CS-R8417	3.70	....
☐	CS-R8419	3.70	....
☐	CS-R8420	8.40	....
☐	CS-R8421	6.--	....
☐	CS-R8422	3.70	....
☐	CS-R8423	3.70	....
☐	CS-R8424	3.70	....
☐	CS-R8501	3.70	....

*) not available on exchange

	Publication code	Price per copy	Number of copies wanted
☐	CS-N8501	3.70	....
☐	OS-R8411	3.70	....
☐	OS-N8402	3.70	....
☐	OS-N8403	6,--	....
☐	OS-N8404	3.70	....
☐	NM-R8414	3.70	....
☐	NM-R8415	3.70	....
☐	NM-R8416	3.70	....
☐	NM-R8501	3.70	....
☐	MS-R8413	6.--	....
☐	MS-R8414	3.70	....
☐	MS-N8401	6.--	....
☐	AM-R8415	3.70	....
☐	AM-R8416	3.70	....
☐	AM-R8417	3.70	....
☐	AM-R8418	3.70	....
☐	AM-R8419	7.20	....
☐	AM-R8420	4.80	....
☐	AM-R8502	4.80	....
☐	AM-R8503	3.70	....
☐	AM-R8504	3.70	....
☐	AM-R8505	3.70	....
☐	PM-R8414	6.--	....
☐	PM-R8415	3.70	....
☐	PM-R8501	3.70	....
☐	PM-R8502	4.80	....

If you wish to order any of the above publications please tick the appropriate boxes and return the completed form to our Sales Department.
Don't forget to add your name and address!
Prices are given in Dutch guilders and are subject to change without notice. Foreign payments are subject to a surcharge per remittance to cover bank, postal and handling charges.

Name ....

Street ....

City ....

Country ....

Contents