Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

MC SYLLABUS 24.1

P.J. VAN DER HOUWEN

NUMERIEKE INTEGRATIE VAN DIFFERENTIAALVERGELIJKINGEN

DEEL 1 : EENSTAPSMETHODEN

MATHEMATISCH CENTRUM AMSTERDAM 1974

AMS (MOS) subject classification scheme (1970): 65L05 65M20

ISBN 90 6196 106 8

VOORWOORD

De in deze syllabus behandelde stof stemt overeen met een aan de Universiteit van Amsterdam gegeven college over numerieke integratiemethoden voor differentiaalvergelijkingen en wel in het bijzonder de *eenstapsmethoden* met een omvang van ongeveer één jaaruur.

De veronderstelde voorkennis is inleidende analyse, lineaire algebra en functionaalanalyse. Meer geavanceerde onderwerpen worden in hoofdstuk I behandeld.

In hoofdstuk II komen dan de eenstapsintegratietechnieken aan de orde waarbij vooral op de *constructie* van dergelijke methoden ingegaan wordt. Drie klassen van methoden zullen onderscheiden worden: Taylor-methoden, Runge-Kuttamethoden en gegeneraliseerde Runge-Kuttamethoden (dit zijn Runge-Kuttamethoden waarvan de parameters door matrixoperatoren vervangen zijn). Uit elk van deze drie klassen worden een of meer formules gelicht die ook als ALGOL 60 procedure aanwezig zijn in de bibliotheek NUMAL van het Academisch Rekencentrum te Amsterdam (SARA). De theoretische achtergronden en het gebruik van deze procedures zullen uitvoerig besproken worden en toegelicht worden met voorbeelden.

Tenslotte wil ik mijn dank betuigen aan mevr. van Gelderen voor de zeer accurate wijze waarop deze syllabus getypt is.

P.J. v.d. H.

INHOUD

HOOFDSTUK I

INLEIDING

In dit inleidende hoofdstuk zal het gebied van de differentiaalvergelijkingen enigszins verkend worden om later, wanneer overgegaan wordt tot de constructie van numerieke integratietechnieken, richtlijnen te hebben volgens welke we te werk moeten gaan. Voorts zal een schets gegeven worden van de belangrijkste karakteristieken van een differentieschema.

1.1 BEGINWAARDEPROBLEMEN

De integratietechnieken (of differentieschemas) die in deze syllabus aan de orde zullen komen, zullen ten doel hebben het beginwaardeprobleem

$$(1.1.1) \qquad \frac{d\vec{y}}{dx} = \vec{f}(\vec{y}) \quad , \quad \vec{y}(x_0) = \vec{y}_0$$

te integreren. Hierin zijn $\vec{y}$ en $\vec{f}$ functies met definitiegebied in respectievelijk $\mathbb{R}_1$ en $\mathbb{R}_r$, en beelden in $\mathbb{R}_r$ (een m-dimensionale vectorruimte zal met $\mathbb{R}_m$ aangegeven worden). De componenten van de vectorfunctie $\vec{y}$ en $\vec{f}$ worden met y_j en f_j, $j=1,2,\ldots,r$, aangegeven. In het vervolg zullen we aannemen dat er precies één oplossing van probleem (1.1.1) bestaat. Voor uniciteits- en existentievoorwaarden van differentiaalvergelijkingen zij verwezen naar de literatuur (b.v. PETROVSKI [1966]).

De problemen die men in de practijk tegenkomt zijn niet altijd geformuleerd in de vorm (1.1.1), maar dikwijls wel te schrijven of te reduceren tot deze vorm. In de volgende paragraaf zullen we laten zien hoe een aantal belangrijke klassen van problemen neerkomen op het oplossen van een beginwaardeprobleem voor zo'n stelsel *autonome, eerste orde differentiaalvergelijkingen.*

1.2 REDUCTIE TOT AUTONOME EERSTE ORDE VORM

De volgende klassen van problemen zullen aan de orde komen:

(1) niet-autonome stelsels
(2) hogere orde stelsels
(3) tweepunts-randwaardeproblemen
(4) partiele differentiaalvergelijkingen
(5) impliciete differentiaalvergelijkingen.

1.2.1 Niet-autonome stelsels

Stel dat gevraagd wordt, in plaats van probleem (1.1.1), het probleem

$$\frac{d\vec{y}}{dx} = \vec{f}(x,\vec{y}) \quad , \quad \vec{y}(x_0) = \vec{y}_0 \tag{1.2.1}$$

op te lossen, waarin $\vec{y} \in \mathbb{R}_1 \to \mathbb{R}_{r-1}$ en $\vec{f} \in \mathbb{R}_1 * \mathbb{R}_{r-1} \to \mathbb{R}_{r-1}$. Een dergelijk niet-autonoom stelsel eerste orde differentiaalvergelijkingen is tot een autonoom stelsel te reduceren door de invoering van de variabele

$$y_r = x.$$

Dit levert het stelsel

$$\begin{aligned} \frac{d\vec{y}}{dx} &= \vec{f}(y_r,\vec{y}) \\ \frac{dy_r}{dx} &= f_r \equiv 1 \end{aligned} \quad , \tag{1.2.1'}$$

hetgeen van de vorm (1.1.1) is.

1.2.2 Hogere orde stelsels

Beschouw de r^{de} orde differentiaalvergelijking

$$\frac{d^r y_1}{dx^r} = f(y_1, \frac{dy_1}{dx}, \frac{d^2 y_1}{dx^2}, \ldots, \frac{d^{r-1} y_1}{dx^{r-1}}), \; r > 1. \tag{1.2.2}$$

Door invoering van de variabelen

$$y_j = \frac{d^{j-1} y_1}{dx^{j-1}} \; , \; j = 2,3,\ldots,r$$

gaat (1.2.2) over in het stelsel

$$\begin{aligned}
&\frac{dy_1}{dx} = y_2,\\
&\frac{dy_2}{dx} = y_3,\\
&\qquad \cdots \qquad\qquad (1.2.2')\\
&\frac{dy_{r-1}}{dx} = y_r,\\
&\frac{dy_r}{dx} = f(y_1,y_2,\ldots,y_r).
\end{aligned}$$

Dit stelsel is van de vorm (1.1.1).

1.2.3 Tweepunts-randwaardeproblemen

Stel dat in probleem (1.1.1) de beginvoorwaarde $\vec{y}(x_0) = \vec{y}_0$ vervangen is door de "gereduceerde" beginvoorwaarde

$$y_j(x_0) = a_j \quad , \quad j \in J_0, \qquad (1.2.3)$$

waarin de a_j gegeven getallen zijn en J_0 een deelverzameling is van de getallen $\{j\}_{j=1}^{r}$. In het algemeen zal de differentiaalvergelijking meerdere oplossingen hebben, die aan (1.2.3) voldoen. We zoeken nu onder deze oplossingen de oplossing die aan de "eindvoorwaarden"

$$y_j(x_e) = b_j \quad , \quad j \in J_e \qquad (1.2.4)$$

voldoet, waarin de b_j gegeven getallen zijn en J_e weer een deelverzameling van $\{j\}_{j=1}^{r}$. Dit probleem wordt een tweepunts-randwaardeprobleem genoemd. De voorwaarden (1.2.3) en (1.2.4) vormen de randvoorwaarden.

Een methode om tweepunts-randwaardeproblemen op te lossen is de zogenaamde *schietmethode*. Deze methode reduceert het randwaardeprobleem tot een reeks van beginwaardeproblemen waarvan de oplossingen naar de gezochte oplossing convergeren, als tenminste aan een aantal voorwaarden voldaan is. In de schietmethode wordt de parametervector

$$\vec{p} = (p_j) \quad , \quad j \in \{j\}_{j=1}^{r} \setminus J_0$$

geïntroduceerd, welke als het ware de "gereduceerde" voorwaarden (1.2.3) aanvullen tot een "complete" beginvoorwaarde:

$$\begin{aligned} & y_j(x_0) = p_j \quad , \quad j \in \{j\}_{j=1}^{r} \setminus J_0, \\ & y_j(x_0) = a_j \quad , \quad j \in J_0. \end{aligned} \tag{1.2.5}$$

Laten we aannemen dat deze beginvoorwaarde precies één oplossing $\vec{y} = \vec{y}(x;\vec{p})$ bepaalt. Wanneer we nu een vector $\vec{p}_0$ kunnen vinden zodanig dat aan de "eindvoorwaarden"

$$y_j(x_e;\vec{p}_0) = b_j \quad , \quad j \in J_e \tag{1.2.4'}$$

voldaan is, dan is hiermee het randwaardeprobleem opgelost. Om de oplossing $\vec{p}_0$ (als deze bestaat) van (1.2.4') te vinden, kan men een iteratieve methode gebruiken; de evaluatie van de functies $y_j(x_e;\vec{p})$ betekent dan voor elke nieuwe approximant $\vec{p}$ de oplossing van een beginwaardeprobleem. Dus het tweepunts-randwaardeprobleem is vervangen door een reeks beginwaardeproblemen.

1.2.4 Partiële differentiaalvergelijkingen

Beschouw de vergelijking

$$\frac{\partial \vec{y}}{\partial x} = F(\vec{y}), \tag{1.2.6}$$

waarin F een differentiaaloperator is in variabelen $z_1, z_2, \ldots$.
Bijvoorbeeld

$$F(y) = \frac{\partial^2}{\partial z^2}\,\vec{y} \ . \tag{1.2.7}$$

De vectorfunctie $\vec{y}$ hangt nu dus niet alleen van x, maar ook van $z_1, z_2, \ldots$ af. Door $\vec{y}$ voor $x = x_0$ voor te schrijven definiëren we een beginwaardeprobleem voor vergelijking (1.2.6):

$$\vec{y}(x_0;z_1,z_2,\ldots) = \vec{y}_0(z_1,z_2,\ldots). \tag{1.2.8}$$

In tegenstelling tot beginwaardeproblemen voor gewone differentiaalvergelijkingen, heeft het beginwaardeprobleem voor de partiële differentiaalvergelijking (1.2.6) niet altijd een unieke oplossing (als er tenminste überhaupt een oplossing bestaat). Bijvoorbeeld wanneer F door (1.2.7) gedefinieerd is dan heeft (1.2.6), (1.2.8) meer dan één oplossing. Door echter $\vec{y}$ op de randen van het gebied in de $(z_1,z_2,\ldots)$-ruimte voor te schrijven wordt het probleem in het algemeen eenduidig oplosbaar. We spreken dan van *begin-randwaardeproblemen*. Ook randvoorwaarden, die afgeleiden van $\vec{y}$ naar $z_1,z_2,\ldots$ bevatten, kunnen eenduidigheid verzekeren (zie de MC Syllabi 3.1, 3.2 en 3.3).

Begin-randwaardeproblemen voor partiële differentiaalvergelijkingen kunnen met behulp van partiële discretisatie tot stelsels eerste orde, gewone differentiaalvergelijkingen gereduceerd worden. Daartoe vervangt men het gebied in de $(z_1,z_2,\ldots)$-ruimte waarop F gedefinieerd is door een eindige verzameling roosterpunten $\{(z_1^{(j)},z_2^{(j)},\ldots)\}_j$ en benadert men F door middel van differentiequotiënten op deze punten. Op deze manier vindt men bij elk roosterpunt een differentiaalvergelijking in z. Men noemt deze methode ook de *"method of lines"*. We zullen dit toelichten aan de hand van een tweetal voorbeelden.

Voorbeeld 1.2.1

Beschouw het niet-lineaire diffusie-probleem

$$\frac{\partial y}{\partial x} = d(z,y)\,\frac{\partial^2 y}{\partial z^2}\,,$$

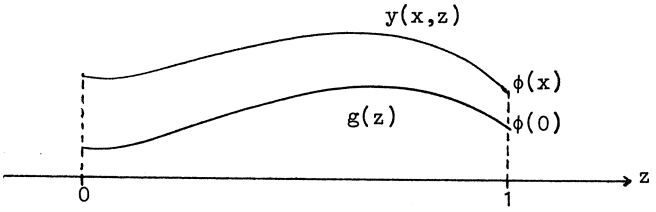
$$y(0,z) = g(z) \quad , \quad 0 \le z \le 1, \tag{1.2.9}$$

$$\frac{\partial y}{\partial z} = 0 \ , \ z = 0 \quad ; \quad y(x,1) = \phi(x),$$

waarin g en ϕ gegeven functies zijn die voldoen aan (zie figuur 1.2.1):

$$g'(0) = 0 \quad , \quad g(1) = \phi(0);$$

verder stelt d(z,y) de diffusiecoëfficiënt voor welke positief verondersteld mag worden. We maken nu de volgende discretiseringen:



Figuur 1.2.1. De functies g = y(0,z) en y = y(x,z), z > 0

$$0 \le z \le 1 \quad \rightarrow \quad \{0,\Delta z,2\Delta z,\ldots,(r-1)\Delta z,r\Delta z=1\},$$

$$y(x,z),\ 0 \le z \le 1 \quad \rightarrow \quad \vec{y}(x) = \{y_0(x),y_1(x),\ldots,y_{r-1}(x),y_r(x)=\phi(x)\},$$

$$g(z),\ 0 \le z \le 1 \quad \rightarrow \quad \vec{g} = \{g(0),g(\Delta z),\ldots,g(r\Delta z)=\phi(0)\},$$

waarin $\Delta z = 1/r$ de roosterpuntafstand voorstelt (zie figuur 1.2.1). De differentiaalvergelijking (1.2.9) discretiseren we door het rechterlid te vervangen door een differentiequotiënt met betrekking tot de roosterpunten $j\Delta z$. Uitgedrukt in $\vec{y}$ en $\vec{g}$ vinden we het stelsel eerste orde, gewone differentiaalvergelijkingen

$$(1.2.10) \qquad \frac{dy_j}{dx} = d(j\Delta z,y_j)\,\frac{y_{j+1}-2y_j+y_{j-1}}{(\Delta z)^2} \quad , \quad j = 0,1,\ldots,r-1$$

met de beginvoorwaarde

$$\vec{y}(0) = \vec{g} \ .$$

Stelsel (1.2.10) bestaat uit r vergelijkingen in r + 2 onbekende functies y_j. Twee onbekenden moeten nog geëlimineerd worden. Hiertoe dienen de randvoorwaarden uit (1.2.9); deze impliceren dat

$$y_{-1} = y_1 \quad , \quad y_r = \phi \ ,$$

zodat (1.2.10) geschreven kan worden als (eliminatie van y_{-1} en y_r)

$$\frac{dy_0}{dx} = (\Delta z)^{-2} d(0,y_0)(2y_1-2y_0) ,$$

$$\frac{dy_j}{dx} = (\Delta z)^{-2} d(\frac{j}{r},y_j)(y_{j+1}-2y_j+y_{j-1}) , \quad j = 1,2,\ldots,r-2 , \tag{1.2.10'}$$

$$\frac{dy_{r-1}}{dx} = (\Delta z)^{-2} d(\frac{r-1}{r},y_{r-1})(\phi(x)-2y_{r-1}+y_{r-2}) .$$

Voorbeeld 1.2.2

Beschouw het Cauchy-probleem

$$\frac{\partial y}{\partial x} = a(z,y)\frac{\partial y}{\partial z} ,$$

$$y(0,z) = g(z) \quad , \quad -\infty \leq z \leq \infty . \tag{1.2.11}$$

Analoog aan voorbeeld 1.2.1 geeft vervanging van $\partial y/\partial z$ door een differentiequotiënt een (oneindig groot) stelsel van eerste orde, gewone differentievergelijkingen:

$$\frac{dy_j}{dx} = a(j\Delta z,y_j)\frac{y_{j+1}-y_{j-1}}{2\Delta z} \quad , \quad j = 0,\pm 1,\pm 2,\ldots . \tag{1.2.12}$$

We merken op, dat wanneer y op een eindig gebied van het (x,z)-vlak gevraagd wordt, volstaan kan worden met het oplossen van een eindig stelsel differentiaalvergelijkingen. In paragraaf 2.7 zullen we hier nog op terugkomen.

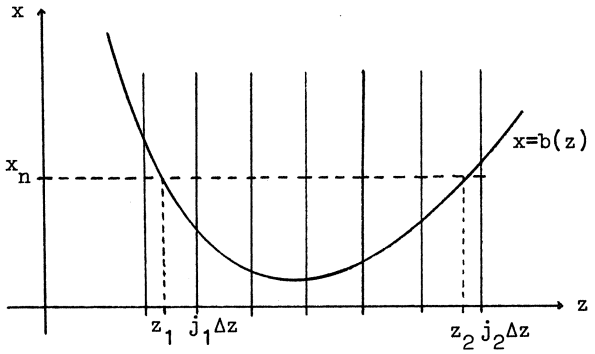
Voorbeeld 1.2.3

Beschouw het diffusieprobleem

$$\frac{\partial y}{\partial x} = \frac{\partial^2 y}{\partial z^2} ,$$

$$y(x,z) = g(b(z),z) \quad \text{langs de lijn } x = b(z) . \tag{1.2.12}$$

In dit probleem zijn de begin-randwaarden samengesmolten tot randvoorwaarden langs een kromme $x = b(z)$. Evenals in de voorgaande voorbeelden vervangen we de z-variabele door de discrete waarden $\{j\Delta z\}_j$. Partiële discretisatie van de vergelijking op deze verzameling zal nu echter een systeem



Figuur 1.2.2 Non-standaard begin-randwaarden

gewone differentiaalvergelijkingen geven waarvan het aantal r afhangt van x.

We zullen de partiële discretisatie van dit soort problemen illustreren aan de hand van de in figuur 1.2.2 geschetste kromme x = b(z). Laat zowel z_1 als z_2 oplossing zijn van de vergelijking

$$x_n - b(z) = 0$$

en stel dat

$$(j_1-1)\Delta z \le z_1 < j_1\Delta z \ , \ j_2\Delta z < z_2 \le (j_2+1)\Delta z \ .$$

Dan vinden we het volgende stelsel vergelijkingen:

$$\frac{dy_{j_1}}{dx} = \frac{2g(b(z_1),z_1) - 2(1+c_1)y_{j_1} + 2c_1y_{j_1+1}}{c_1(1+c_1)(\Delta z)^2}$$

$$\frac{dy_j}{dx} = \frac{y_{j-1} - 2y_j + y_{j+1}}{(\Delta z)^2} \ , \ j = j_1+1,\ldots,j_2-1, \tag{1.2.13}$$

$$\frac{dy_{j_2}}{dx} = \frac{2c_2y_{j_2-1} - 2(1+c_2)y_{j_2} + 2g(b(z_2),z_2)}{c_2(1+c_2)(\Delta z)^2}$$

waarin

$$c_1 = \frac{j_1\Delta z - z_1}{\Delta z} \quad , \quad c_2 = \frac{z_2 - j_2\Delta z}{\Delta z} \; .$$

Dit stelsel is niet gedefinieerd voor c_1 of c_2 gelijk nul. Voor kleine waarden van c_1 of c_2 liggen y_{j_1} of y_{j_2} echter nagenoeg op de kromme $x = b(z)$, zodat men van de randvoorwaarden gebruik kan maken.

1.2.5 Impliciete differentiaalvergelijkingen

Tenslotte gaan we nog even in op *impliciete* vergelijkingen, dat wil zeggen de afgeleide dy/dx kan niet expliciet uitgedrukt worden in $\vec{y}$ (en x):

$$(1.2.14) \qquad \vec{F}(\vec{y}, \frac{d\vec{y}}{dx}) = \vec{0} \quad , \quad \vec{y}(x_0) = \vec{y}_0 \; .$$

Twee voor de hand liggende methoden om dit systeem terug te brengen tot de standaardvorm (1.1.1) zijn ten eerste het invoeren van de formele inverse van $\vec{F}$ ten opzichte van $d\vec{y}/dx$, dus

$$(1.2.14') \qquad \frac{d\vec{y}}{dx} = \vec{f}(\vec{y}) \quad , \quad F(\vec{y}, \vec{f}(\vec{y})) = \vec{0} \; ,$$

waarbij men dan elke keer wanneer de rechterlidfunctie $\vec{f}(\vec{y})$ gevraagd wordt, een stelsel niet-lineaire algebraïsche of transcendente vergelijkingen zal moeten oplossen, en ten tweede de introductie van een nieuwe variabele $\vec{z} = d\vec{y}/dx$, waarmee (1.2.14) geschreven kan worden als het eerste orde stelsel

$$(1.2.14'') \qquad \begin{aligned} \frac{d\vec{y}}{dx} &= \vec{z} \\ \frac{d\vec{z}}{dx} &= -J_2^{-1}(\vec{y},\vec{z})J_1(\vec{y},\vec{z})\vec{z}. \end{aligned}$$

In (1.2.14") stellen J_1 en J_2 de Jacobianen van $\vec{F}$ voor respectievelijk ten opzichte van het eerste en tweede argument (zie de volgende paragraaf voor de definitie van de Jacobiaan van een functie).

1.3 KLASSIFICATIE VAN EERSTE ORDE STELSELS

Een numerieke integratiemethode kan soms aanzienlijk geeconomiseerd worden door de mogelijkheid in te bouwen de methode aan te passen aan de te integreren differentiaalvergelijking. In deze "adaptiviteit" van de methode speelt het eigenwaardespectrum van de Jacobiaan van de rechterlidfunctie, dat wil zeggen de eigenwaarden van de matrix

$$(1.3.1) \qquad J = \left(\frac{\partial f_i}{\partial y_j}\right) \quad , \quad i = \text{rijindex} \quad , \quad j = \text{kolomindex}$$

een belangrijke rol. We zullen drie klassen van differentiaalvergelijkingen bespreken waarvoor het de moeite loont om de integratiemethode aan het probleem aan te passen:

(1) stijve differentiaalvergelijkingen: spectrum bestaat uit ver uiteenliggende clusters in het negatieve halfvlak;
(2) "parabolische"stelsels: spectrum bestaat uit een smalle strook langs de negatieve as;
(3) "hyperbolische" stelsels: spectrum bestaat uit een smalle strook langs de imaginaire as.

1.3.1 Stijve differentiaalvergelijkingen

In veel physische problemen komt men vergelijkingen tegen waarvan de oplossingen bestaan uit langzaam en zeer snel varierende componenten, waarvan laatstgenoemde echter in grootte zeer klein zijn. Zulke vergelijkingen worden *stijve* differentiaalvergelijkingen genoemd. Omdat stijve vergelijkingen zo veelvuldig optreden (netwerk-analyse, circuitsimulatie, chemische kinetica, biomathematica, proces-dynamica, geleide projectielen, enz.) zal er de nodige aandacht aan besteed worden in deze syllabus.

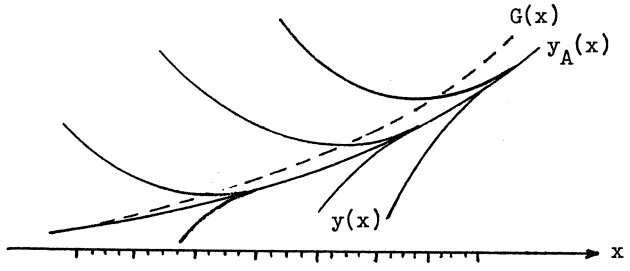
De eerste numerieke behandeling van stijve differentiaalvergelijkingen vinden we bij Curtiss en Hirschfelder [1952], maar pas in de laatste 10 jaar heeft deze belangrijke klasse de aandacht gekregen die zij verdient; we noemen hier Pope [1963], Treanor [1966], Lawson [1967], Calahan [1968], Dahlquist [1968], Gear [1968,'69], Liniger en Willoughby [1970] en Lindberg [1971].

We zullen eerst een meetkundige definitie geven van een enkele, stijve differentiaalvergelijking. Stel dat alle integraalcurven van de vergelij-

king zeer snel naar één bepaalde integraalcurve $y_A = y_A(x)$ convergeren, (zie figuur 1.3.1), dan is de convergentiesnelheid ten opzichte van gegeven referentiepunten op de x-as een maat voor de stijfheid. Dit effect, dat de oplossingen gedwongen worden naar één asymptotische oplossing te convergeren, wordt wel het "wandelstok" effect genoemd. In navolging van Curtiss en Hirschfelder beschouwen we de niet-autonome vergelijking

$$(1.3.1) \qquad \frac{dy}{dx} = \frac{y-G(x)}{a(x,y)} \quad ,$$

waarin G een zich ordentelijk gedragende functie is (zie figuur 1.3.1) en a negatief en klein in absolute waarde is. Voor grote waarden van y zijn de hellingen van de integraalkrommen van (1.3.1) zeer groot in negatieve zin; in de buurt van de kromme y = G(x) verspringen deze hellingen echter van groot negatief naar groot positief. Dit is nu precies het in de figuur weergegeven gedrag.



Figuur 1.3.1 Het "wandelstok-effect"

Voorbeeld 1.3.1

Zij gegeven de vergelijking

$$(1.3.2) \qquad \frac{dy}{dx} = -1000y + x^2 \quad ,$$

dan is de algemene oplossing

$$y(x) = -\frac{1}{1000}\left(x^2 - \frac{2}{1000}x + \frac{2}{1000000}\right) + e^{-1000x}\left(y_0 - \frac{2}{10^9}\right) \quad ,$$

waarin y_0 de integratieconstante voorstelt (zo gekozen dat $y(0) = y_0$). Kennelijk is

$$y_A(x) = -\frac{1}{1000}(x^2 - \frac{2}{1000}x + \frac{2}{1000000}) \simeq -\frac{1}{1000}x^2.$$

Verder wordt G gegeven door

$$G(x) = -\frac{1}{1000}x^2.$$

G gedraagt zich dus nagenoeg als de asymptotische oplossing.

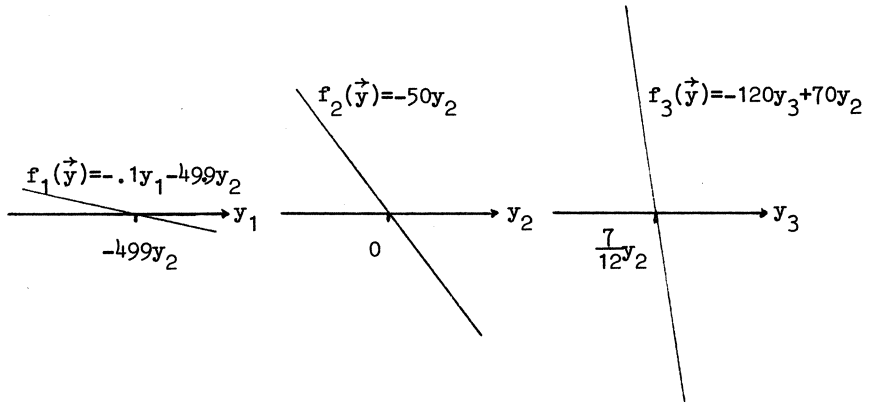
Bovenstaand meetkundig beeld voor een enkele, stijve differentiaalvergelijking laat zich eenvoudig uitbreiden tot stelsels vergelijkingen. Een stelsel is stijf wanneer een of meer componenten f_i van de rechterlidfunctie $\vec{f}$ waarden aanneemt die snel van groot positief naar groot negatief springen als y_i toeneemt terwijl $y_1, y_2, \ldots, y_{i-1}, y_{i+1}, y_{i+1}, \ldots, y_r$ constant gehouden worden. De helling van de functie y_i heeft dan de neiging klein te worden in absolute waarde als $x \to \infty$, zodat y_i na een aanvankelijk snelle variatie, spoedig een langzaam varierende functie wordt.

Voorbeeld 1.3.2

We beschouwen eerst het *lineaire* stelsel (ontleend aan Lapidus en Seinfeld [1971])

$$(1.3.3) \qquad \frac{d}{dx}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} -.1 & -49.9 & 0 \\ 0 & -50 & 0 \\ 0 & 70 & -120 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} .$$

In figuur 1.3.2 is het gedrag van de rechterlidcomponenten f_i als functie van y_i weergegeven. Kennelijk hebben we te maken met een stijf stelsel differentiaalvergelijkingen.



Figuur 1.3.2 Componenten f_i als functie van y_i.

Voorbeeld 1.3.3

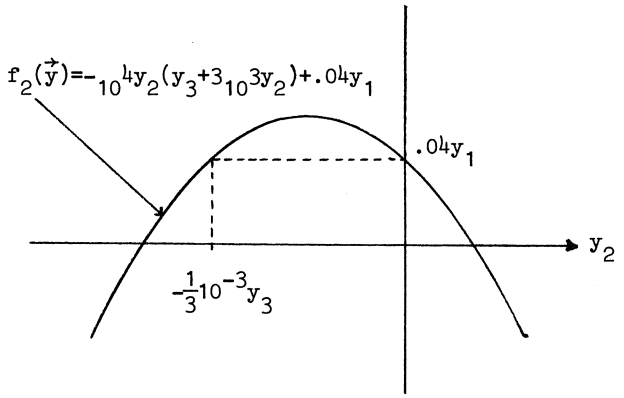
Vervolgens beschouwen we een *niet-lineair* systeem (probleem van Robertson [1967])

$$
\begin{aligned}
y_1' &= -.04y_1 + 10^4 y_2 y_3 \,, \\
y_2' &= .04y_1 - 10^4 y_2 y_3 - 3\,10^7 y_2^2 \,, \\
y_3' &= 3\,10^7 y_2^2 .
\end{aligned}
\tag{1.3.4}
$$

Dit systeem beschrijft de reactiesnelheden in een of ander chemisch proces. De helling van de 2^e component y_2 is weergegeven in figuur 1.3.3; hierin zijn y_1 en y_3 positief gekozen omdat negatieve waarden physisch niet relevant zijn. We zien uit deze figuur dat y_2 een toenemend stijf karakter vertoont.

Een strengere definitie van een stijve differentiaalvergelijking kan gegeven worden met behulp van de Jacobiaan van de rechterlidfunctie. Laten $\vec{y}$ en $\tilde{\vec{y}}$ twee naburige oplossingen van de differentiaalvergelijking zijn, dan geldt

$$(1.3.5) \qquad \frac{d}{dx}[\vec{y}(x)-\tilde{\vec{y}}(x)] \simeq J(\vec{y}(x)) \cdot [\vec{y}(x)-\tilde{\vec{y}}(x)] \ ,$$



Figuur 1.3.3 Component f_2 als functie van y_2

waarin J de Jacobiaan voorstelt. Deze relatie kan formeel geintegreerd worden, waarmee we krijgen

$$(1.3.6) \qquad \begin{aligned} &\vec{y}(x) - \tilde{\vec{y}}(x) \simeq \exp[(x-x_0)A(x)] \cdot [\vec{y}(x_0)-\tilde{\vec{y}}(x_0)] \ , \\ &A(x) = \frac{1}{x-x_0}\int_{x_0}^{x} J(y(\xi))d\xi \ . \end{aligned}$$

Hierin is x_0 het punt waarin de differentiaalvergelijking beschouwd wordt. De matrix A(x) stelt als het ware de "gemiddelde" Jacobiaan voor langs de kromme $\vec{y} = \vec{y}(\xi)$, $\xi \in [x_0,x]$. Merk op dat voor lineaire stelsels de matrix A juist de Jacobiaan J is.

Veronderstel dat voor elke x in een interval $[x_0,x_1]$ de eigenwaarden van de matrix A(x) onder te verdelen zijn in twee groepen met respectievelijk kleine moduli en groot negatieve reële delen. Uit (1.3.5) volgt dan direct dat voor een voldoend groot interval $[x_0,x_1]$, de onderlinge afstand van de integraalcurven $\{x,\vec{y}(x)\}$ en $\{x,\tilde{\vec{y}}(x)\}$ eerst snel varieert om tenslotte een langzaam varierende functie van x te worden. Dit is juist het gedrag van de integraalkrommen van een stijve differentiaalvergelijking.

In de practijk is het echter eenvoudiger om niet het spectrum van de matrix A(x), maar van de Jacobiaan $J(\vec{y}(x))$, als criterium te nemen voor het al of niet stijf zijn van een differentiaalvergelijking. Nu zullen bij niet te snelle variaties van de Jacobiaan in het interval $[x_0,x_1]$, de spectra van A(x) en $J(\vec{y}(x))$ veel overeenkomst vertonen, zodat we tot de volgende definitie van een stijve differentiaalvergelijking komen:

Definitie 1.3.1

Vergelijking (1.1.1) wordt in het punt (x_0,y_0) ten opzichte van het interval $[x_0,x_1]$ stijf genoemd wanneer in de omgeving van y_0 de eigenwaarden δ van $J(\vec{y})$ in twee groepen C_0 en C_1 liggen zodanig dat voor elk tweetal punten (δ_0,δ_1) met $\delta_0 \in C_0$ en $\delta_1 \in C_1$ de relatie

$$\exp[\mathrm{Re}\ \delta_1(x_1-x_0)] \ll \exp[\mathrm{Re}\ \delta_0(x_1-x_0)] \tag{1.3.7}$$

geldt. Als maat voor de stijfheid van een vergelijking nemen we de waarde van

$$\underset{\delta_0\in C_0,\delta_1\in C_1}{\text{maximum}} \ |\ \mathrm{Re}(\delta_1-\delta_0)\ |\ (x_1-x_0)\ . \tag{1.3.8}$$

Opgaven 1.3.1

(1) Onderzoek de stijfheid in de zin van definitie 1.3.1 van de vergelijkingen (1.3.3) en (1.3.4).

1.3.2 Partieel gediscretiseerde parabolische differentiaalvergelijkingen

Zoals we in paragraaf 1.2.4 gezien hebben kunnen beginrandwaardeproblemen voor partiele differentiaalvergelijkingen door discretisatie van de "plaats-variabelen" in vele gevallen tot (grote) stelsels van gewone differentiaalvergelijkingen gereduceerd worden. Behalve dat dergelijke partieel gediscretiseerde vergelijkingen een relatief groot aantal componentvergelijkingen tellen, is ook het spectrum van de Jacobiaan in het algemeen zeer uitgestrekt. In het geval van *parabolische* vergelijkingen liggen de eigenwaarden van de Jacobiaan meestal in een smalle, langgerekte strook

langs de *negatieve* as. We zullen dit illustreren aan de hand van de diffusievergelijking (1.2.9). We hebben al laten zien dat discretisatie van de variabele z tot het stelsel (1.2.10') leidt. De Jacobiaan van dit stelsel is eenvoudig te bepalen; we vinden

$$(1.3.9) \qquad J(\vec{y}) = (\Delta z)^{-2}[D'+DT] \ ,$$

waarin D' de diagonaalmatrix

$$\begin{pmatrix} 2d_0'(y_1-y_0) & & & & 0 \\ 0 & d_1'(y_2-2y_1+y_0) & & & \\ & \ddots & \ddots & & \\ & & d_j'(y_{j+1}-2y_j+y_{j-1}) & & \\ & & & \ddots & \\ 0 & \cdots & & & d_{r-1}'(\phi-2y_{r-1}+y_{r-2}) \end{pmatrix} ,$$

D de diagonaalmatrix $(d(j\Delta z;y_j)$ en T de tridiagonaalmatrix

$$\begin{pmatrix} -2 & 2 & 0 & \cdots & 0 \\ 1 & -2 & 1 & & \\ 0 & 1 & -2 & 1 & \\ \vdots & & \ddots & & \vdots \\ & & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix}$$

voorstelt. Aangezien de componenten y_j voor voldoend kleine waarden van Δz een continue kromme dienen te vormen kan men voor eigenwaardebeschouwingen de matrix D' ten opzichte van de matrix DT wel verwaarlozen. Dit betekent dat de matrix (1.3.9) nagenoeg een tridiagonaalmatrix is met positieve nevendiagonaalelementen. Volgens stelling 1.4.7 (zie paragraaf 1.4) zijn de eigenwaarden van dergelijke matrices reëel. Verder geldt volgens stelling 1.4.3 van Gerschgorin (zie paragraaf 1.4) dat de eigenwaarden gelegen zijn in de vereniging van cirkels

$$|\delta+2d_j| < 2d_j \quad , \quad j = 0,1,\ldots,r-2 \ ,$$

$$|\delta+ 2d_{r-1}| < d_{r-1} \ .$$

Hieruit kan geconcludeerd worden dat het eigenwaardespectrum van J voor $\Delta z \to 0$ gelegen is in het (negatieve) interval

$$(1.3.10) \qquad [-4(\Delta z)^{-2} \max_j d(j\Delta z, y_j), 0] \ .$$

1.3.3 Partieel gediscretiseerde hyperbolische differentiaalvergelijkingen

Het spectrum van een partieel gediscretiseerde hyperbolische vergelijking ligt in het algemeen in een strook langs de *imaginaire* as en wel symmetrisch ten opzichte van de oorsprong. We zullen dit nagaan voor het stelsel (1.2.12). De Jacobiaan van dit stelsel heeft als j^e rij-vector

$$(1.3.11) \qquad \frac{1}{2\Delta z} (\ldots,0,-a_j,a_j'(y_{j+1}-y_{j-1}),a_j,0,\ldots) \ ,$$

waarin $a_j = a(j\Delta z, y_j)$. Voor $\Delta z \to 0$ geldt $y_{j+1} \to y_{j-1}$, zodat de diagonaalelementen van de Jacobiaan nagenoeg nul zijn. Met andere woorden de Jacobiaan van (1.2.12) is "bijna" scheefsymmetrisch en heeft dus "bijna" imaginaire eigenwaarden (zie stelling 1.4.4 in de volgende paragraaf). De lengte van het eigenwaardeinterval bepalen we weer door de stelling van Gerschgorin toe te passen. We vinden dan het (imaginaire) interval

$$(1.3.12) \qquad [-i(\Delta z)^{-1} \max_j |a_j|, i(\Delta z)^{-1} \max_j |a_j|]$$

Opgaven 1.3.2

(1) Pas de methode der lijnen toe op de symmetrisch hyperbolische differentiaalvergelijking

$$\frac{\partial \vec{y}}{\partial x} = A(z,\vec{y}) \frac{\partial \vec{y}}{\partial z} \ , \ A(z,\vec{y}) \text{ symmetrisch} \ ,$$

met beginvoorwaarde $\vec{y}(0,z) = \vec{g}(z)$, $-\infty \le z \le \infty$ en onderzoek het eigenwaardespectrum van de Jacobiaan.

1.4. ENKELE STELLINGEN UIT DE ANALYSE EN ALGEBRA

In deze paragraaf worden zonder bewijs een aantal stellingen gegeven die veel gebruikt worden in de numerieke analyse van differentiaalvergelijkingen. Hierin zal met de matrix $A = (a_{ij})$ steeds een vierkante matrix van de orde r bedoeld worden.

Stelling 1.4.1

De norm van een *normale* matrix $A(A^*A=AA^*, A^*=(\bar{a}_{ji}))$ ten opzichte van de Euclidische vectornorm

$$\|\vec{y}\|_2 = [\sum_{j=1}^{r} y_j^2]^{\frac{1}{2}}$$

is gelijk aan de *spectrale radius* van A:

$$\|A\|_2 = \sigma(A) = \max_j |\delta_j| \ , \ \delta_j \text{ eigenwaarde van } A.$$

Bewijs

Zie Collatz [1968, p.149].

Stelling 1.4.2

Voor iedere matrix A geldt

$$\|A\|_2 = \sqrt{\sigma(AA^*)}$$

$$\|A\|_\infty = \max_i \sum_j |a_{ij}| \ ,$$

waarin $A^* = (\bar{a}_{ji})$ en $\|A\|_\infty$ de matrixnorm ten opzichte van de maximumnorm

$$\|\vec{y}\|_\infty = \max_j |y_j|$$

voorstelt.

Bewijs

Zie Collatz [1968, p.128].

Stelling 1.4.3 (Gerschgorin)

De eigenwaarden van de matrix A liggen in de vereniging van cirkels

$$|z-a_{ii}| = \sum_{j\neq i} |a_{ij}| .$$

Bewijs

Zie Ostrowski [1951].

Stelling 1.4.4

De eigenwaarden van een *hermitische* matrix ($A=A^*$) zijn reëel en die van een *scheefhermitische* ($A=-A^*$) zuiver imaginair.

Stelling 1.4.5

Een normale matrix heeft een orthogonaal stelsel eigenvectoren.

Stelling 1.4.6

Laat q de grootste orde zijn van alle diagonale ondermatrices J_q van de canonische Jordan-voorstelling J van A met $\sigma(J_q) = \sigma(A)$, dan geldt

$$\|A^n\|_2 \sim \nu n^{q-1}[\sigma(A)]^{n-q+1} \quad \text{als } n \to \infty \ ,$$

waarin ν een positieve constante is

Bewijs

Zie Varga [1962, p.64].

Stelling 1.4.7

Een reële tridiagonale matrix met positieve neven-diagonaalelementen heeft reële, onderling verschillende eigenwaarden.

Bewijs

Zie Wilkinson [1965, p.335].

Stelling 1.4.8

Laat A een Fréchet-afgeleide $A'(\vec{y})$ hebben met definitie- en beeldgebied de Banachruimten B_1 en B_2, dan geldt

$$\|A\vec{y}_1 - A\vec{y}_2\| \leq \underset{\substack{\vec{y}=\theta\vec{y}_1+(1-\theta)\vec{y}_2 \\ 0\leq\theta\leq 1}}{\text{Supremum}} \|A'(\vec{y})\| \, \|\vec{y}_1 - \vec{y}_2\| .$$

[$A'(\vec{y})$ is gedefinieerd als de lineaire operator die voldoet aan

$$\|A\vec{y}_1 - A\vec{y}_2 - A'(\vec{y}_1)(\vec{y}_1 - \vec{y}_2)\| \leq \|\vec{y}_1 - \vec{y}_2\| \, o(\|\vec{y}_1 - \vec{y}_2\|)$$

voor elke $\vec{y}_1, \vec{y}_2 \in B_1$.]

Bewijs

Zie Collatz [1968, p.223].

1.5 NUMERIEKE OPLOSSING VAN NIET-LINEAIRE VERGELIJKINGEN

In de bespreking van tweepunts-randwaardeproblemen en impliciete differentiaalvergelijkingen zijn we al geconfontreerd met de noodzaak een, in het algemeen niet-lineair, stelsel algebraische of transcendente vergelijkingen op te lossen. We zullen in deze paragraaf twee standaardmethoden behandelen voor de numerieke oplossing van dergelijke vergelijkingen en wel het Jacobi-iteratieproces en het Newton-Raphson-iteratieproces.

1.5.1 Iteratieproces van Jacobi

Het meest eenvoudige iteratieproces om een stelsel vergelijkingen van de vorm

$$\vec{F}(\vec{y}) = \vec{0} \qquad (1.5.1)$$

op te lossen is gebaseerd op herhaalde substitutie. Hiermee wordt een oplossing $\vec{n}$ van (1.5.1) benaderd door een rij vectoren $\vec{y}^{(j)}$, j=0,1,..., die berekend worden met de recurrente betrekking

(1.5.2) $\quad \vec{y}^{(j+1)} = \vec{y}^{(j)} + \vec{F}(\vec{y}^{(j)})$.

Om dit proces te starten moet men zelf de beginapproximatie $\vec{y}^{(0)}$ kiezen.

Wanneer $\vec{y}^{(j)}$ in de buurt van een oplossing $\vec{\eta}$ komt, kan men voor de fout $\vec{\eta} - \vec{y}^{(j)}$ in eerste benadering schrijven

(1.5.3) $\quad \vec{y}^{(j+1)} - \eta \simeq (I+J(\vec{\eta}))(\vec{y}^{(j)}-\vec{\eta})$,

waarin J de Jacobiaan van $\vec{F}$ voorstelt.
Is dus de beginapproximatie $\vec{y}^{(0)}$ voldoende nauwkeurig, dan geldt

(1.3.4) $\quad \vec{y}^{(j+1)} - \vec{\eta} \simeq (I+J(\vec{\eta}))^{j+1}(\vec{y}^{(0)}-\vec{\eta})$.

Volgens stelling 1.4.6 leidt deze relatie in eerste orde benadering tot de foutschatting

(1.5.5) $\quad \|\vec{y}^{(j+1)}-\vec{\eta}\|_2 \leq \nu j^{q-1}[\sigma(I+J(\vec{\eta}))]^{j-q+1} \|\vec{y}^{(0)}-\vec{\eta}\|_2$

als $j \to \infty$. Een nodige en voldoende voorwaarde voor de convergentie van het Jacobi-proces is blijkbaar

(1.5.6) $\quad \sigma(I+J(\vec{\eta})) < 1$.

Naarmate de spectrale radius van de matrix $I + J(\eta)$ kleiner is, is de convergentiesnelheid van dit iteratieproces groter. Dit suggereert om de functie $\vec{F}$ met een parameter ω voor te vermenigvuldigen; voorwaarde (1.5.6) wordt dan

(1.5.6') $\quad \sigma(I+\omega J(\vec{\eta})) < 1$,

zodat elke ω die de spectrale radius kleiner maakt, de convergentiesnelheid vergroot. De parameter ω wordt *relaxatieparameter* genoemd.

Opgaven 1.5.1

(1) Bepaal de optimale relaxatieparameter als bekend is dat de eigenwaarden van $J(\vec{\eta})$ in een reëel interval [a,b] liggen.

(2) Wat is er over het Jacobi-proces te zeggen wanneer $J(\vec{\eta})$ zuiver imaginaire eigenwaarden heeft ?

1.5.2 Iteratieproces van Newton-Raphson

In plaats van het Jacobi-proces te versnellen met een relaxatieparameter ω, kan men de convergentiesnelheid ook vergroten met een "relaxatiematrix" Ω. Voorwaarde (1.5.6') gaat dan over in

(1.5.6") $\quad \sigma(I+\Omega J(\vec{\eta})) < 1$

en weer geldt dat hoe kleiner de spectrale radius van de matrix $I + \Omega J(\vec{\eta})$ des te sneller de convergentie. Een optimale convergentie wordt bereikt voor

$$\Omega = -J^{-1}(\vec{\eta}) \ .$$

In de practijk is $J(\vec{\eta})$ uiteraard niet beschikbaar en kiest men bijvoorbeeld

(1.5.7) $\quad \Omega = -J^{-1}(\vec{y}^{(j)})$

in de berekening van $\vec{y}^{(j+1)}$, i.e.

(1.5.8) $\quad \vec{y}^{(j+1)} = \vec{y}^{(j)} - J^{-1}(\vec{y}^{(j)})\vec{F}(\vec{y}^{(j)}) \ .$

Dit proces wordt het iteratieproces van Newton-Raphson genoemd.

Indien het evalueren van de Jacobiaan in *iedere* iteratiestap te duur is, dan kiest men ook wel

(1.3.7') $\quad \Omega = -J^{-1}(\vec{v}) \ ,$

waarin $\vec{v}$ "zo nu en dan" gelijk gemaakt wordt aan $\vec{y}^{(j)}$. Deze variant wordt de *gemodificeerde* Newton-Raphson methode genoemd.

1.6 DIFFERENTIESCHEMA'S

De basisbegrippen in de theorie van differentieschema's zijn *consistentie* , *convergentie* en *stabiliteit*. We zullen deze begrippen definieren voor een algemeen differentieschema, waarbij we een differentieschema zullen beschouwen als een afbeelding van een rij (onbekende) vectoren

-de *differentieoplossing*- op een rij gegeven vectoren. Een algemeen differentieschema en een algemeen beginwaardeprobleem worden dan met elkaar in verband gebracht door de consistentievoorwaarde inhoudende dat een oplossing van het beginwaardeprobleem aan het differentieschema voldoet afgezien van een klein residu, de zogenaamde *afbreekfout*. Een kleine afbreekfout betekent niet dat automatisch het verschil tussen de differentieoplossing en de oplossing van het beginwaardeprobleem klein is. Wanneer dit wel het geval is spreken we van convergentie. Tenslotte gaan we in op de stabiliteit van een differentieschema, dat wil zeggen de gevoeligheid van de differentieoplossing voor verstoringen.

1.6.1 Definities

In plaats van de onafhankelijke variabele x met als waardebereik een segment van de reële rechte, voeren we in de theorie van differentieschema's een rij roosterpunten

$$(1.6.1) \qquad X = \{x_n\}_{n=0}^{N} \quad , \quad x_{n+1} > x_n$$

in, waarbij N een positief getal is en x_0 en x_N gegeven getallen zijn die begin- en eindpunt van het integratieinterval voorstellen. De roosterpunten definieren een rij van *integratiestappen*:

$$(1.6.2) \qquad H = \{h_n\}_{n=0}^{N-1} \quad , \quad h_n = x_{n+1} - x_n \ .$$

De maximale integratiestap in H zullen we met h aangeven. Bij een gegeven rij integratiestappen definieren we de productruimte

$$(1.6.3) \qquad P_H = \prod_{n=0}^{N} \mathbb{R}_r \ ,$$

waarin $\mathbb{R}_r$ een Euclidische vectorruimte van dimensie r voorstelt. (In feite hangt P_H bij gegeven $\mathbb{R}_r$ alleen van N af, maar we zullen toch H als onderindex gebruiken (i.p.v. N) voor de uniformiteit van de verdere notatie.) De elementen van P_H zijn rijen $\{\vec{y}_n\}_{n=0}^{N}$ van vectoren uit $\mathbb{R}_r$. Deze rijen geven we aan met Y_H, dus

$$(1.6.4) \qquad Y_H = \{\vec{y}_n\}_{n=0}^{N} \ .$$

De ruimte P_H wordt een lineaire ruimte wanneer we de lineaire operaties

$$
\begin{aligned}
&Y_H + Y_H' = \{\vec{y}_n + \vec{y}_n'\}_{n=0}^{N} , \\
&\alpha Y_H = \{\alpha \vec{y}_n\}_{n=0}^{N}
\end{aligned}
\tag{1.6.5}
$$

definieren, waarin α een scalar is.

Aan H associeren we nu een operator D_H met definitieverzameling en beeldverzameling in P_H. *Een differentieschema* (ook wel *integratieformule* genoemd) wordt nu gedefinieerd door de vergelijking

$$D_H Y_H = G_H , \tag{1.6.6}$$

waarin G_H een gegeven element uit P_H voorstelt. Y_H noemen we de *differentieoplossing*. Laten we aannemen dat voor iedere verzameling H van integratiestappen een operator D_H en een rij G_H gedefinieerd is. Neem verder aan dat D_H zodanig is dat het eerste element van $Y_H = D_H^{-1} G_H$ gelijk is aan het eerste element van G_H, dus

$$\vec{y}_0 = \vec{g}_0 . \tag{1.6.7}$$

Onder deze beginvoorwaarde stelt (1.6.6) een *discreet beginwaardeprobleem* voor. In het vervolg zullen we steeds aannemen dat D_H voor $h \to 0$ bestaat en ongelijk is aan de nul-operator.

Voorbeeld 1.6.1

Beschouw het schema

$$
\begin{aligned}
&y_0 - \tfrac{1}{2} = 0 , \\
&y_1 - y_0 - h_0 y_0^2 = 0 , \\
&y_2 - y_1 - h_1 y_1^2 = 0 , \\
&y_3 - y_2 - h_2 y_2^2 = 0 .
\end{aligned}
\tag{1.6.8}
$$

Dit schema is een discreet beginwaardeprobleem van de vorm (1.6.6)-(1.6.7) wanneer we definieren

$$N = 3 \ , \quad H = \{h_0,h_1,h_2\} \ , \quad Y_H = \{y_0,y_1,y_2,y_3\} \ ,$$

$$G_H = \{\frac{1}{2},0,0,0\}$$

en

$$D_H Y_H = \{y_0, y_1-y_0-h_0y_0^2, y_2-y_1-h_1y_1^2, y_3-y_2-h_2y_2^2\} \ .$$

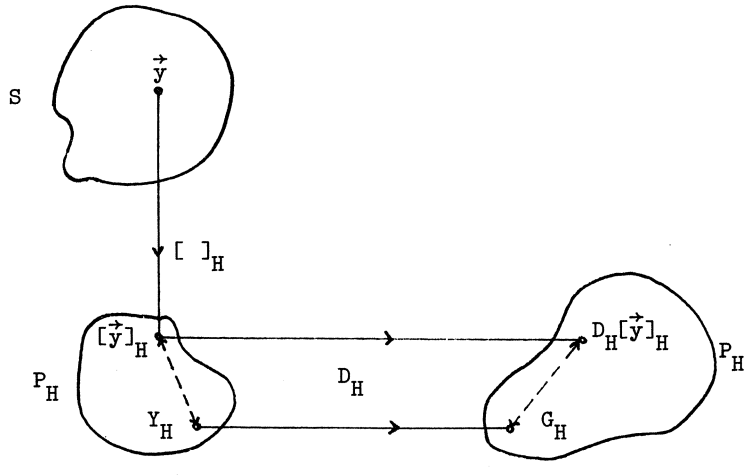
1.6.2 Consistentie

We zullen nu een *analytisch* beginwaardeprobleem van de vorm (1.1.1)-(1.1.2) en een *discreet* beginwaardeprobleem met elkaar in verband brengen en de voorwaarden opstellen waarvoor het discrete probleem het analytisch probleem benadert. Daartoe definieren we de oplossingsruimte S van vergelijking (1.1.1):

$$(1.6.9) \qquad S = \{\vec{y} \mid \frac{d\vec{y}}{dx} - f(\vec{y}) = \vec{0}\} \ .$$

Op S definieren we de *discretiseringsoperator* $[\ ]_H$ die aan een element $\vec{y} \in S$ de rij $\{\vec{y}(x_n)\}_{n=0}^{N} \in P_H$ toevoegt, dus

$$(1.6.10) \qquad [\vec{y}]_H = \{\vec{y}(x_n)\}_{n=0}^{N} \ .$$



Figuur 1.6.1 Illustratie van afbreekfout en discretiseringsfout

Door middel van de operator $[\ ]_H$ kunnen het analytische en het discrete probleem met elkaar worden vergeleken: neem een element $\vec{y} \in S$, discretiseer dit tot het element $[\vec{y}]_H \in P_H$, pas de differentie-operator D_H toe en vergelijk het resultaat $D_H[\vec{y}]_H$ met het gegeven rechterlid G_H van het differentieschema (zie figuur 1.6.1). De verschilrij

$$D_H[\vec{y}]_H - G_H \tag{1.6.11}$$

wordt de rij van *afbreekfouten* genoemd. De elementen van deze rij worden *lokale* afbreekfouten genoemd.

Om de grootte van de afbreekfout te meten, voeren we een norm in voor de ruimte P_H. Deze norm zal met $|||\ |||$ aangegeven worden. Van deze norm zullen we echter eisen dat er in S ook een norm bestaat zodanig dat voor alle $\vec{y} \in S$ geldt

$$|||[\vec{y}]_H|||_P \to |||\vec{y}|||_S \qquad \text{als} \qquad h \to 0. \tag{1.6.12}$$

Wanneer we dit niet zouden eisen dan zou bijvoorbeeld de norm

$$|||Y_H||| = \sum_{n=0}^{N} \|\vec{y}_n\|_2 ,$$

waarin $\|\ \|_2$ de Euclidische norm in $\mathbb{R}_r$ voorstelt, toegestaan zijn; voor grote waarden van N is zo'n norm echter onbruikbaar. Voorbeelden van normen die wel aan (1.6.12) voldoen zijn

$$|||Y_H||| = \frac{1}{N} \sum_{n=0}^{N} \|\vec{y}_n\|_2$$

en

$$|||Y_H||| = \operatorname{Max}_n \|\vec{y}_n\|_2 .$$

Definitie 1.6.1

Differentieschema (1.6.6) is een consistente benadering van de differentiaalvergelijking (1.1.1) wanneer voor alle $\vec{y} \in S$

$$|||D_H[\vec{y}]_H - G_H||| \to 0 \qquad \text{als} \qquad h \to 0 . \tag{1.6.13}$$

Wanneer de afbreekfout van de orde p + 1 in h naar 0 convergeert is de benadering consistent van de orde p ten opzichte van de norm $|||\ |||$.

Voorbeeld 1.6.2

Beschouw voorbeeld 1.6.1 voor willekeurige waarden van N, dus

$$(1.6.8') \qquad y_{n+1} = y_n + h_n y_n^2 \quad , \quad n = 0,1,\ldots,N-1.$$

Verder specificeren we

$$(1.6.14) \qquad x_0 = 0 \quad , \quad x_N = 1 \quad , \quad h_n = h = \frac{1}{N} .$$

We beweren nu dat (1.6.8') een consistente benadering is van de differentiaalvergelijking

$$(1.6.15) \qquad \frac{dy}{dx} = y^2 .$$

Daartoe nemen we een oplossing van (1.6.15) en wel

$$(1.6.16) \qquad y(x) = \frac{y_0}{1-y_0 x} ,$$

waarin y_0 een willekeurige constante is. Toepassing van de discretiseringsoperator $[\ ]_H$ geeft de rij

$$[y]_H = \{\frac{Ny_0}{N-ny_0}\}_{n=0}^{N} .$$

Vervolgens passen we de door (1.6.8') gedefinieerde operator D_H toe:

$$D_H\{y\}_H = \{y_0,\ldots, \frac{Ny_0}{N-(n+1)y_0} - \frac{Ny_0}{N-ny_0} - \frac{Ny_0^2}{(N-ny_0)^2},\ldots\} .$$

Uit (1.6.8') volgt verder dat de rij G_H gegeven wordt door

$$G_H = \{y_0,0,\ldots,0\} .$$

In tabel 1.6.1 zijn de waarden van $|||D_H[y]_H - G_H|||$ voor $y_0 = \frac{1}{2}$ opgenomen, waarbij $|||\ |||$ de maximum norm voorstelt.

Tabel 1.6.1 Maximale afbreekfout voor $y_0 = \frac{1}{2}$

N	h	$\|\|\|D_H[y]_H - G_H\|\|\|$
1	1.00	.75
2	.50	.56
3	.33	.43
4	.25	.36
10	.10	.17

Uit deze tabel volgt duidelijk dat de afbreekfout afneemt naarmate h kleiner wordt.

In de practijk is de juist beschreven wijze om consistentie aan te tonen van weinig waarde omdat de expliciete voorstelling van de elementen uit S gebruikt wordt. Een meer bruikbare methode om consistentie te bewijzen maakt gebruik van het feit dat in de meeste differentieschema's slechts een klein aantal opeenvolgende vectoren $\vec{y}_n$ rechtstreeks gekoppeld zijn. Zo zijn in (1.6.8') slechts y_n en y_{n+1} per vergelijking gekoppeld. Dit maakt het mogelijk om de lokale afbreekfouten in Taylorreeksen te ontwikkelen zonder te veel aan nauwkeurigheid in te boeten. Laten we dit toepassen op (1.6.8'); dus de elementen van de rij

$$D_H[y]_H - G_H = \{0,\ldots,y(x_{n+1})-y(x_n)-hy^2(x_n),\ldots\}$$

ontwikkelen we in Taylorreeksen. Dit geeft

$$D_H[y]_H - G_H = \{0,\ldots,-hy^2(x_n)+hy'(x_n)+\tfrac{1}{2}h^2y''(x_n)+O(h^3),\ldots\} ,$$

of met gebruikmaking van de differentiaalvergelijking (1.6.15)

$$D_H[y]_H - G_H = \{0,\ldots,h^2y^3(x_n),\ldots\} + O(h^3) .$$

We zien dat de analytische oplossing nog steeds nodig is om de grootte van de afbreekfout te bepalen, maar we zien ook dat onafhankelijk van de analytische oplossing, de afbreekfout zich als $O(h^2)$ gedraagt als $h \to 0$. We kunnen dus concluderen dat (1.6.8') een *eerste orde consistente benadering* is van (1.6.15').

1.6.3 Convergentie

Zoals al opgemerkt betekent consistentie nog niet dat de differentie-oplossing naar de analytische oplossing convergeert als $h \to 0$. Het differentieschema is slechts een *formele* benadering van de differentiaalvergelijking. In figuur 1.6.1 betekent dit dat het kleiner worden van de afstand tussen $D_H[\vec{y}]_H$ en G_H niet noodzakelijk het kleiner worden van de afstand tussen $[\vec{y}]_H$ en Y_H impliceert.

Definitie 1.6.2

Een differentieschema is convergent wanneer voor alle $\vec{y} \in S$

$$(1.6.17) \qquad |||[\vec{y}]_H - Y_H||| \to 0 \qquad \text{als} \qquad h \to 0 \ .$$

Wanneer $[\vec{y}]_H - Y_H$ zich gedraagt als de p-de macht van h dan heet het differentieschema p-de orde convergent.

We zullen $[\vec{y}]_H - Y_H$ de *discretiseringsfout* noemen.

Het aantonen van de convergentie van een differentieschema is in het algemeen veel moeilijker dan de consistentie.

Stelling 1.6.1

Laat het differentieschema $D_H Y_H = G_H$ consistent zijn van de orde p en stel dat D_H een eenduidige inverse A_H heeft waarvan de Fréchet-afgeleide $A_H'(G_H)$ gedefinieerd is voor alle $G_H \in P_H$. Er geldt dan voor alle $\vec{y} \in S$

$$(1.6.18) \qquad |||[\vec{y}]_H - Y_H||| \leq ch^{p+1} \, \|A_H'(\bar{G}_H)\| \ ,$$

waarin c uniform begrensd is als $h \to 0$ en $\bar{G}_H$ een element in de omgeving van G_H is.

Bewijs

Voor iedere verzameling van integratiestappen geldt

$$[\vec{y}]_H - Y_H = A_H D_H [\vec{y}]_H - A_H G_H \ .$$

Hierop passen we de "middelwaarde-ongelijkheid" voor niet lineaire operatoren toe (stelling 1.4.8):

$$(1.6.18') \qquad |||[\vec{y}]_H - Y_H||| \leq ||A'_H(\bar{G}_H)|| \cdot |||D_H[\vec{y}]_H - G_H||| \ ,$$

waarin $\bar{G}_H$ een element is op de "verbindingslijn" van $D_H[\vec{y}]_H$ en G_H. Het rechterlid van (1.6.18') kan beschouwd worden als een functie van H. Toepassing van de definitie van consistentie leidt onmiddellijk tot ongelijkheid (1.6.18).

Een gevolg van deze stelling is dat een voldoende voorwaarde voor convergentie gegeven wordt door

$$h^{p+1} \cdot ||A'_H(G_H)|| \to 0 \quad \text{als} \quad h \to 0 \quad \text{voor alle } G_H \in P_H \ .$$

Bepalend voor de convergentie van een differentieschema is dus het gedrag van de Fréchet-afgeleide van de inverse van de differentieoperator D_H als $h \to 0$.

1.6.4 Stabiliteit

In een feitelijke berekening zal men de differentieoplossing Y_H nooit vinden omdat afrondfouten het resultaat zullen beinvloeden. We zullen de in de practijk verkregen oplossing de *numerieke oplossing* noemen en deze met Y_H^* aangeven. Het verschil $Y_H - Y_H^*$ zullen we de *numerieke fout* noemen. Er geldt nu volgens de driehoeksongelijkheid

$$(1.6.19) \qquad |||[\vec{y}]_H - Y_H^*||| \leq |||[\vec{y}]_H - Y_H||| + |||Y_H - Y_H^*||| \ .$$

Met andere woorden het verschil tussen de analytische oplossing en de numerieke oplossing wordt begrensd door de som van de discretiseringsfout en de numerieke fout. In deze paragraaf zullen we de numerieke fout nader bespreken. Dit brengt ons tot het derde basisbegrip in de theorie van differentieschema's, de *stabiliteit*.

We zullen aannemen dat Y_H^* in het definitiegebied van D_H ligt, dus Y_H^* voldoet aan het schema

$$(1.6.20) \qquad D_H Y_H^* = G_H^* \ ,$$

waarin G_H^* in feite door Y_H^* gedefinieerd wordt. Omgekeerd kan men Y_H^* ook beschouwen als de oplossing van het oorspronkelijke schema waarin het rechterlid G_H verstoord is. Om te verzekeren dat de numerieke fout klein is, eisen we dat het differentieschema min of meer ongevoelig is voor verstoringen van het rechterlid. Deze eis is tamelijk vaag en er zijn dan ook vele stabiliteitsdefinities in omloop. We zullen de meest belangrijke, namelijk die van Forsythe en Wasow [1960] en Rjabenki en Filippov [1960], bespreken.

Forsythe en Wasow noemen een differentieschema stabiel wanneer de numerieke fout niet sneller aangroeit dan een of andere lage macht van h^{-1} voor $h \to 0$. Deze definitie rechtvaardigen Forsythe en Wasow door hun ervaring dat in het geval van instabiliteit de numerieke fout *exponentieel* aangroeit in plaats van zich als een *polynoom in* h^{-1} te gedragen. Een meer concrete definitie kan men geven door de ongelijkheid (vergelijk (1.6.18'))

$$(1.6.21) \qquad |||Y_H - Y_H^*||| \leq \|A_H'(\bar{G}_H)\| \cdot |||G_H - G_H^*|||$$

te beschouwen. Kennelijk is een voldoende voorwaarde voor *stabiliteit in de zin van Forsythe en Wasow* dat

$$(1.6.22) \qquad \|A'(\bar{G}_H)\| = O(h^{-q}) \quad , \quad q \geq 1 \quad , \quad \text{als} \quad h \to 0 \ ,$$

voor alle $\bar{G}_H$ in de omgeving van G_H. Merk op dat (1.6.22) convergentie impliceert als $q < p+1$ (zie stelling 1.6.1).

Rjabenki en Filippov formuleerden een veel strengere voorwaarde, namelijk dat het effect van een verstoring van G_H niet harder mag toenemen dan $O(h^{-1})$, met andere woorden dat de operator hA_H uniform continu moet zijn in G_H als $h \to 0$. Uit (1.6.21) volgt dan dat een voldoende voorwaarde voor *stabiliteit in de zin van Rjabenki en Filippov* gegeven wordt door (1.6.22) met $q = 1$.

1.6.5 Interpolatie van de differentieoplossing

Tot dusver is gesproken over de numerieke benadering van de rij $[\vec{y}]_H$. We zullen nu enige opmerkingen maken hoe met behulp van deze discrete benadering van de analytische oplossing $\vec{y}$ een continue benadering van $\vec{y}$ verkregen kan worden. Daartoe gebruiken we *interpolatieformules* op de intervalletjes $[x_n, x_{n+1}]$, $n = 0,1,\ldots,N-1$.

Stel dat als resultaat van een numeriek integratieproces behalve de rij $\{\vec{y}_n\}_{n=0}^N$ ook rijen van numerieke benaderingen van afgeleiden van $\vec{y}$ zijn uitgerekend. Deze rijen geven we aan met $\{\vec{y}'_n\}_{n=0}^N$, $\{\vec{y}''_n\}_{n=0}^N$, enz. Bijvoorbeeld geldt

$$\{\vec{y}'_n\}_{n=0}^N = \{f(\vec{y}_n)\}_{n=0}^N \ .$$

We zoeken nu door middel van een interpolatieformule een benadering van $\vec{y}$ in het punt $x = x_n + h$ met $0 < h < h_n$, waarbij gebruik gemaakt wordt van de vectoren $\vec{y}_n$, $\vec{y}'_n$, $\vec{y}''_n$,... en $\vec{y}_{n+1}$, $\vec{y}'_{n+1}$, $\vec{y}''_{n+1}$, ... Een algemene manier voor de constructie van zo'n benadering gaat als volgt: in de eerste plaats nemen we een functie $\vec{I}_n = \vec{I}_n(x;\vec{\alpha}_1,\vec{\alpha}_2,\ldots)$ die voor geschikte keuze van de parametervectoren $\vec{\alpha}_1,\vec{\alpha}_2,\ldots$ een redelijke voorstelling van het gedrag van $\vec{y}$ geeft in het interval $[x_n,x_n+h_n)$; vervolgens stellen we het stelsel

$$\begin{aligned}
&\vec{I}_n(x_n;\vec{\alpha}_1,\vec{\alpha}_2,\ldots) = \vec{y}_n \ , \\
&\vec{I}_n(x_{n+1};\vec{\alpha}_1,\vec{\alpha}_2,\ldots) = \vec{y}_{n+1} \ , \\
&\vec{I}'_n(x_n;\vec{\alpha}_1,\vec{\alpha}_2,\ldots) = \vec{y}'_n \ , \qquad (1.6.23) \\
&\vec{I}'_n(x_{n+1};\vec{\alpha}_1,\vec{\alpha}_2,\ldots) = \vec{y}'_{n+1} \ , \\
&\qquad\qquad \ldots
\end{aligned}$$

op (hierin betekent het accent differentiatie naar x).
Tenslotte moeten de parametervectoren $\vec{\alpha}_j$ hieruit opgelost worden. Om dit niet te bewerkelijk te maken kiest men meestal polynomen of rationale functies voor de functies $\vec{I}_n$. De parameters $\vec{\alpha}_j$ zijn dan de coefficienten van deze functies en komen dus lineair voor in het stelsel (1.6.23). Dit maakt het mogelijk standaardtechnieken toe te passen voor de berekening van de $\vec{\alpha}_j$'s. Hieronder volgen een aantal voorbeelden van interpolatiepolynomen en rationale interpolatieformules. In deze formules komt de variabele ν voor die gedefinieerd is door

$$\nu = \frac{h}{h_n} \ . \qquad (1.6.24)$$

Verder dienen rationale uitdrukkingen van vectoren geinterpreteerd te worden als componentsgewijze operaties op deze vectoren.

Interpolatiepolynomen

(1.6.25) $$\vec{y}_{n+\nu} = \vec{y}_n + [\vec{y}_{n+1}-\vec{y}_n]\nu \ ;$$

(1.6.26) $$\vec{y}_{n+\nu} = \vec{y}_n + h_n\vec{y}'_n\nu + [\vec{y}_{n+1}-\vec{y}_n-h_n\vec{y}_n]\nu^2 \ ;$$

(1.6.27) $$\vec{y}_{n+\nu} = \vec{y}_n + h_n\vec{y}'_n\nu + [3(\vec{y}_{n+1}-\vec{y}_n) - h_n(2\vec{y}'_n+\vec{y}'_{n+1})]\nu^2 +$$
$$+ [2(\vec{y}_n-\vec{y}_{n+1}) + h_n(\vec{y}'_n+\vec{y}'_{n+1})]\nu^3 \ ;$$

(1.6.28) $$\vec{y}_{n+\nu} = \vec{y}_n + h_n\vec{y}'_n + \tfrac{1}{2}h_n^2\vec{y}''_n\,^2 +$$
$$+ [4(\vec{y}_{n+1}-\vec{y}_n) - h_n(3\vec{y}'_n+\vec{y}'_{n+1}) - h_n^2\vec{y}''_n]\nu^3 +$$
$$+ [3(\vec{y}_n-\vec{y}_{n+1}) + h_n(2\vec{y}'_n+\vec{y}'_{n+1}) + \tfrac{1}{2}h_n^2\vec{y}''_n]\nu^4 \ .$$

Rationale interpolatieformules

(1.6.29) $$\vec{y}_{n+\nu} = \frac{\vec{y}_n\,\vec{y}_{n+1}}{\vec{y}_{n+1} + [\vec{y}_n-\vec{y}_{n+1}]\nu} \ ;$$

(1.6.30) $$\vec{y}_{n+\nu} = \frac{\vec{y}_n[\vec{y}_{n+1}-\vec{y}_n] + [\vec{y}_n(\vec{y}_n-\vec{y}_{n+1}) + h_n\vec{y}'_n\vec{y}_{n+1}]\nu}{[\vec{y}_{n+1}-\vec{y}_n] + [\vec{y}_n-\vec{y}_{n+1}+h_n\vec{y}'_n]\nu}$$

(1.6.31) $$\vec{y}_{n+\nu} = \frac{\vec{y}_n + \vec{\alpha}_1\nu}{1 + \vec{\alpha}_2\nu + \vec{\alpha}_3\nu^2} \ ,$$

$$\vec{\alpha}_2 = \frac{2y_{n-1}(y_{n+1}-y_n) - h_n(y_ny'_{n+1}+y'_ny_{n+1}) - h_n^2y'_ny'_{n+1}}{\vec{y}_{n+1}(\vec{y}_n-\vec{y}_{n+1}) + h_n\vec{y}_n\vec{y}'_{n+1}} \ ,$$

$$\vec{\alpha}_3 = \frac{(\vec{\alpha}_2+1)(\vec{y}_n-\vec{y}_{n+1}) + h_n\vec{y}'_n}{\vec{y}_{n+1}} \ ,$$

$$\vec{\alpha}_1 = \alpha_2\vec{y}_n + h_n\vec{y}'_n \ .$$

We besluiten onze beschouwing van algemene differentieschema's met de opmerking dat een indruk van de nauwkeurigheid van de uiteindelijk gevonden continue benadering van de analytische oplossing verkregen kan worden door per interval $[x_n, x_n+h_n]$ de functies $\vec{I}_n$ in de differentiaalvergelijking te substitueren.

HOOFDSTUK II

EENSTAPSMETHODEN

In het voorgaande hoofdstuk hebben we een aantal belangrijke aspecten van een willekeurig differentieschema besproken. In dit hoofdstuk zullen we wat dieper ingaan op een speciale klasse van differentieschema's, de *eenstapsmethoden*. Deze methoden worden gekarakteriseerd door het feit dat $\vec{y}_{n+1}$ uit $\vec{y}_n$ berekend wordt en niet rechtstreeks met $\vec{y}_{n-1}, \vec{y}_{n-2}, \ldots$ samenhangt. Eenstapsmethoden kunnen dan ook beschreven worden door de recurrente betrekking

$$(2.0.1) \qquad \vec{y}_{n+1} = E_n(\vec{y}_n), n = 0,1,2,\ldots,N-1,$$

waarin E_n in het algemeen een niet-lineaire operator is die van het stapnummer n afhangt.
De operator E_n kan een zeer gecompliceerde operator zijn die niet alleen van het rechterlid $\vec{f}$ van de differentiaalvergelijking afhangt maar ook van de afgeleiden van $\vec{f}$. Bovendien is het ook niet altijd mogelijk de operator E_n expliciet te schrijven; in zulke gevallen zullen we de eenstapsmethode door middel van de impliciete relatie

$$(2.0.2) \qquad I_n(\vec{y}_{n+1}, \vec{y}_n) = \vec{0} \quad , \quad n = 0,1,2,\ldots,N-1$$

beschrijven. Wel zullen we aannemen dat $\vec{y}_{n+1}$ uit deze relatie opgelost kan worden waarmee (2.0.2) overgaat in (2.0.1).

In het bijzonder zal ingegaan worden op de *constructie* van eenstapsmethoden. Hierbij zullen de *consistentie-, convergentie-* en *stabiliteitsvoorwaarden* het uitgangspunt vormen. Globaal gesproken gaan we als volgt te werk. We definieren formeel een klasse van differentieschemas waarin een aantal nog onbepaalde parameters voorkomen, we stellen de consistentie-, convergentie- en stabiliteitsvoorwaarden op en trachten dan de parameters zo te bepalen dat aan deze voorwaarden voldaan is. Drie klassen van eenstapsmethoden zullen behandeld worden:

(1) methoden gebaseerd op *herhaalde differentiatie* van het rechterlid van de differentiaalvergelijking (Taylor-methoden);

(2) methoden gebaseerd op herhaalde *evaluatie* van het rechterlid (Runge-Kuttamethoden);

(3) methoden gebaseerd op een of twee functie-evaluaties per integratiestap en de *Jacobiaan* van het rechterlid (gegeneraliseerde Runge-Kuttamethoden).

De Taylor-methoden doen een zwaar beroep op de analytische eigenschappen van het rechterlid $\vec{f}$ en zijn daarom alleen geschikt voor betrekkelijk eenvoudige vectorfuncties $\vec{f}$.

De Runge-Kuttamethoden vragen alleen naar de waarden van $\vec{f}$ en zijn dan ook toepasbaar bij ingewikkelde rechterlidfuncties. De prijs die hiervoor betaald moet worden is echter een relatief groot aantal functie-evaluaties wanneer een hoge orde formule gewenst wordt.

Voor de gegeneraliseerde Runge-Kuttamethoden is behalve $\vec{f}$, alleen de Jacobiaan van $\vec{f}$ nodig. Wat kennis omtrent de te integreren differentiaalvergelijking betreft, staat deze klasse van methoden tussen de Taylor- en Runge-Kuttamethoden in.

Tenslotte merken we nog op dat vele mengvormen mogelijk zijn van de drie hier genoemde hoofdklassen. Zulke mengvormen worden ook wel *hybride* methoden genoemd.

2.1. TAYLOR-METHODEN

Laten we aannemen dat het rechterlid $\vec{f}$ van vergelijking (1.0.1) zo vaak differentieerbaar is dat de in deze paragraaf voorkomende differentiaties inderdaad mogelijk zijn.
Om een Taylor-methode te kunnen definiëren voeren we het begrip *lokaal analytische oplossing* in, dat wil zeggen de oplossing van de differentiaalvergelijking die in het punt x_n gelijk is aan $\vec{y}_n$, waarin $(x_n,\vec{y}_n)$ het punt voorstelt tot waar de numerieke integratie gevorderd is. De lokaal analytische oplossing geven we aan met $\vec{z}(x_n,\vec{y}_n;x)$. Dus $\vec{z}$ voldoet aan de voorwaarde

$$\vec{z}(x_n,\vec{y}_n;x_n) = \vec{y}_n \ . \tag{2.1.1}$$

De afgeleiden van $\vec{z}$ naar x in het punt x_n zullen eenvoudigheidshalve met

$\vec{y}'_n, \vec{y}''_n, \ldots$ aangegeven worden, dus

$$\vec{y}'_n = \frac{d}{dx}\,\vec{z}(x_n,\vec{y}_n;x)\,\Big|_{x = x_n},$$

$$\vec{y}''_n = \frac{d^2}{dx^2}\,\vec{z}(x_n,\vec{y}_n;x)\,\Big|_{x = x_n}, \tag{2.1.2}$$

$$\ldots .$$

Aangezien $\vec{z}$ een oplossing is van (1.0.1) kunnen we de afgeleiden van $\vec{z}$ uitdrukken in het rechterlid $\vec{f}$. Er geldt bijvoorbeeld

$$\begin{aligned} \vec{y}'_n &= \vec{f}(\vec{y}_n), \\ \vec{y}''_n &= J(\vec{y}_n)\vec{f}(\vec{y}_n), \end{aligned} \tag{2.1.2'}$$

waarin J de Jacobiaan van $\vec{f}$ voorstelt. Door invoering van de gradient-operator

$$\nabla = \left(\frac{\partial}{\partial y_i}\right)$$

kunnen we algemeen schrijven

$$\frac{d^j}{dx^j}\,\vec{z}(x_n,\vec{y}_n;x)\,\Big|_{x = x_n} = (\vec{f}(\vec{y}(x)\cdot\nabla)^{j-1}\vec{f}(\vec{y}(x))\,\Big|_{x = x_n}, \tag{2.1.3}$$

waarin $(\cdot)$ het inwendig product in $\mathbb{R}_r$ voorstelt. Door middel van deze relatie kunnen de afgeleiden in het punt x_n van de lokaal analytische oplossing $\vec{z} = \vec{z}(x_n,\vec{y}_n;x)$ uitgedrukt worden in de afgeleiden van $\vec{f}$ in het punt $\vec{y}_n$.

De algemene (lineaire) *Taylor-methode* wordt nu formeel gedefinieerd door

$$\sum_{j=0}^{m_1} \alpha_j (h_n \frac{d}{dx})^j\,\vec{z}(x_{n+1},\vec{y}_{n+1};x)\,\Big|_{x= x_{n+1}} = \sum_{j=0}^{m_2} \beta_j (h_n \frac{d}{dx})^j\,\vec{z}(x_n,\vec{y}_n;x)\,\Big|_{x = x_n}, \tag{2.1.4}$$

of wat leesbaarder (met behulp van (2.1.2))

$$(2.1.4') \qquad \alpha_0 \vec{y}_{n+1} + \alpha_1 h_n \vec{y}'_{n+1} + \alpha_2 h_x^{2} \vec{y}''_{n+1} + \ldots =$$

$$= \beta_0 \vec{y}_n + \beta_1 h_n \vec{y}'_n + \beta_2 h_n^{2} \vec{y}''_n + \ldots \ ,$$

of uitgedrukt in het rechterlid $\vec{f}$ (met behulp van (2.1.3))

$$(2.1.4'') \qquad \alpha_0 \vec{y}_{n+1} + \sum_{j=1}^{m_1} \alpha_j h_n^j (\vec{f}(\vec{y}(x)) \cdot \nabla)^{j-1} \vec{f}(\vec{y}(x)) \Big|_{x = x_{n+1}} =$$

$$= \beta_0 y_n + \sum_{j=1}^{m_2} \beta_j h_n^j (\vec{f}(\vec{y}(x)) \cdot \nabla)^{j-1} \vec{f}(y(x)) \Big|_{x = x_n} .$$

Hierin zijn de α_j en β_j nog nader te bepalen parameters; ze dienen om te kunnen voldoen aan consistentie-, convergentie- en stabiliteitsvoorwaarden (zie paragrafen 2.4, 2.5 en 2.6). In het vervolg zullen we zonder de algemeenheid prijs te geven, $\alpha_0 = 1$ stellen.

Tenslotte merken we op dat in het geval van een *expliciete* Taylor-formule ($m_1=0$) de vector $\vec{y}_{n+1}$ als functie van h_n een polynoom is in h_n, terwijl in het geval van *impliciete* Taylor-formules $\vec{y}_{n+1}$ in het algemeen geen polynoom in h_n is (zie voorbeeld 2.1.2).

Voorbeeld 2.1.1

Overbekende voorbeelden van eenstapsformules zijn de (expliciete) *Euler-formule*

$$(2.1.5) \qquad \vec{y}_{n+1} = \vec{y}_n + h_n \vec{y}'_n$$

en de (impliciete) *terugwaartse Euler-formule* en *trapezium-regel*

$$(2.1.6) \qquad \vec{y}_{n+1} - h_n \vec{y}'_{n+1} = \vec{y}_n \ ,$$

$$(2.1.7) \qquad \vec{y}_{n+1} - \tfrac{1}{2} h_n \vec{y}'_{n+1} = \vec{y}_n + \tfrac{1}{2} h_n \vec{y}'_n \ .$$

Voorbeeld 2.1.2

Laten we de trapezium-regel eens toepassen op de differentiaalver-

gelijking

$$(2.1.8) \qquad \frac{dy}{dx} = -y^2.$$

Door (2.1.7) uit te drukken in de rechterlidfunctie $f(y) = -y^2$ vinden we

$$(2.1.9) \qquad y_{n+1} + \tfrac{1}{2}h_n y_{n+1}^2 = y_n - \tfrac{1}{2}h_n y_n^2 .$$

Dus

$$(2.1.10) \qquad y_{n+1} = h_n^{-1}\left[-1+\sqrt{1+2h_n y_n - h_n^2 y_n}\right] .$$

In deze uitdrukking voor y_{n+1} moet de integratiestap h_n uiteraard voldoen aan de voorwaarde

$$(2.1.11) \qquad 1 + 2h_n y_n - h_n y_n^2 \geq 0 ,$$

anders wordt geen reële waarde voor y_{n+1} verkregen.

Opgaven 2.1.1

(1) Waarom is van de 2 wortels van (2.1.9) alleen (2.1.10) relevant?

(2) Construeer een *niet-lineaire* Taylor-formule door "vermenigvuldiging" van de expliciete en impliciete Euler-formules.

2.2 RUNGE-KUTTAMETHODEN

In het laatste voorbeeld werd een impliciete Taylor-formule toegepast op een zeer eenvoudige differentiaalvergelijking; de vergelijking voor y_{n+1} kon hierin analytisch opgelost worden. In het algemeen is dit echter niet mogelijk en zal een of ander iteratieproces gebruikt moeten worden. Stel dat we voor een algemene differentiaalvergelijking de trapezium-regel

$$(2.2.1) \qquad \vec{y}_{n+1} - \tfrac{1}{2}h_n \vec{f}(\vec{y}_{n+1}) = \vec{y}_n + \tfrac{1}{2}h_n \vec{f}(\vec{y}_n)$$

met behulp van het Jacobi-proces trachten op te lossen. Stel verder dat $\vec{y}_n$ als beginapproximatie wordt gebruikt en dat het m-de iteratieresultaat als

uiteindelijke benadering voor de oplossing $\vec{y}_{n+1}$ wordt gebruikt. Dan krijgen we het volgende rekenschema

$$
\begin{aligned}
&\vec{y}_{n+1}^{(0)} = \vec{y}_n \ , \\
&\vec{y}_{n+1}^{(j)} = [\vec{y}_n + \tfrac{1}{2}h_n \vec{f}(\vec{y}_n)] + \tfrac{1}{2}h_n \vec{f}(\vec{y}_{n+1}^{(j-1)}) \quad , \quad j = 1,2,\ldots m \ , \\
&\vec{y}_{n+1} = \vec{y}_{n+1}^{(m)} \ .
\end{aligned} \tag{2.2.2}
$$

Dit iteratieproces levert $\vec{y}_{n+1}$ (of exacter gezegd een benadering voor $\vec{y}_{n+1}$) als lineaire combinatie van $\vec{y}_n$ en de functiewaarden $\vec{f}(\vec{y}_{n+1}^{(j)})$, $j = 0,1,\ldots,m-1$, waarbij de argumenten van $\vec{f}$ ook weer lineaire combinaties zijn van $\vec{y}_n$ en $\vec{f}(\vec{y}_{n+1}^{(l)})$, $l = 0,1,\ldots,j-1$. Met andere woorden, (2.2.2) behoort tot de klasse van formules

$$
\begin{aligned}
&\vec{y}_{n+1}^{(0)} = \vec{y}_n \ , \\
&\vec{y}_{n+1}^{(j)} = \vec{y}_n + h_n \sum_{l=0}^{j-1} \lambda_{j,l} \vec{f}(y_{n+1}^{(l)}) \quad , \quad j = 1,2,\ldots,m, \\
&\vec{y}_{n+1} = \vec{y}_{n+1}^{(m)} \ .
\end{aligned} \tag{2.2.3}
$$

In het bijzondere geval van (2.2.2) worden de parameters $\lambda_{j,l}$ gegeven door

$$
\lambda_{j,0} = \frac{1}{2}, \quad \lambda_{j,j-1} = \frac{1}{2}, \quad \lambda_{j,l} = 0, \quad j = 1,2,\ldots,m, \quad l = 1,\ldots,j-2. \tag{2.2.2'}
$$

Schema (2.2.3) stelt de algemene formule van een *m-punts Runge-Kuttamethode* voor. Deze methode, die uitsluitend van evaluaties van het rechterlid $\vec{f}$ gebruikt maakt, werd in 1895 door Runge geïntroduceerd, zij het in een iets gewijzigde, maar in de literatuur meer ingeburgerde vorm, namelijk

$$
\begin{aligned}
&\vec{y}_{n+1} = \vec{y}_n + \sum_{j=0}^{m-1} \lambda_{m,j} \vec{k}_n^{(j)} \ , \\
&\vec{k}_n^{(j)} = h_n \vec{f}(y_n + \sum_{l=0}^{j-1} \lambda_{j,l} \vec{k}_n^{(l)}) \ .
\end{aligned} \tag{2.2.4}
$$

De vectoren $\vec{k}_n^{(j)}$ en $\vec{y}_{n+1}^{(j)}$ voldoen aan de relatie

$$
\vec{k}_n^{(j)} = h_n \vec{f}(\vec{y}_{n+1}^{(j)}) \ . \tag{2.2.5}
$$

Zowel de voorstelling (2.2.3) als (2.2.4) zullen in deze syllabus gebruikt worden. Hiernaast zal dikwijls een Runge-Kuttaformule gekarakteriseerd worden door het vermelden van de zogenaamde *genererende matrix* $(\lambda_{j,l})$, waarin j de rij-index is (j=0,1,...,m) en l de kolom-index (l=0,1,...,m-1). Zo wordt (2.2.2) gekarakteriseerd door de matrix

$$(2.2.2') \qquad (\lambda_{j,l}) = \begin{pmatrix} 0 & & & \dots & 0 \\ \frac{1}{2} & 0 & & \dots & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & \dots & 0 \\ \vdots & & \ddots & & \\ \frac{1}{2} & 0 & \dots & 0 & \frac{1}{2} \end{pmatrix}.$$

Formule (2.2.3) is *expliciet*; *impliciete* Runge-Kuttaformules worden ook wel toegepast. De index l in de definitie van $\vec{y}_{n+1}^{(j)}$ loopt dan van 0 tot m-1. In deze syllabus worden ze echter buiten beschouwing gelaten.

Voorbeeld 2.2.1

De formule van Euler (2.1.5) is het meest eenvoudige voorbeeld van een Runge-Kuttaformule; de genererende matrix luidt

$$(2.1.5') \qquad (\lambda_{j,l}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

De eerste twee-puntsformules werden door Runge [1895] gegeven:

$$(2.2.6) \qquad \begin{pmatrix} 0 & 0 \\ \frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

(de tweede formule wordt ook wel de *verbeterde Euler-formule* genoemd); drie-puntsformules werden door Heun [1900] en Kutta [1901] gegeven:

$$(2.2.7) \qquad \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 \\ 0 & \frac{2}{3} & 0 \\ \frac{1}{4} & 0 & \frac{3}{4} \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ -1 & 2 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{pmatrix}.$$

De meest bekende Runge-Kuttaformule (*standaard Runge-Kuttaformule*) is ook aan Kutta [1901] te danken. Deze wordt gedefinieerd door de matrix

$$(2.2.8) \qquad \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{pmatrix} .$$

Opgaven 2.2.1

(1) Stel de genererende matrices op van de terugwaartse Euler-formule en de trapezium-regel.

(2) Toon aan dat Jacobi-iteratie van een (2,2)-Taylor-formule na m iteraties leidt tot formules uit de klasse

$$(2.2.9) \qquad \begin{aligned} \vec{y}_{n+1}^{(0)} &= \vec{y}_n , \\ \vec{y}_{n+1}^{(j)} &= \vec{y}_n + h_n \sum_{l=0}^{j-1} [\lambda_{j,l} \vec{f}(\vec{y}_{n+1}^{(l)}) + \mu_{j,l} h_n \vec{g}(\vec{y}_{n+1}^{(l)})] , \quad j = 1,2,\ldots,m, \\ \vec{y}_{n+1} &= \vec{y}_{n+1}^{(m)} \end{aligned}$$

waarin $\vec{g}(\vec{y})$ de tweede afgeleide van y naar x is, i.e.

$$\vec{g}(\vec{y}) = \frac{d\vec{f}(\vec{y})}{dx} = J(\vec{y})\vec{f}(\vec{y}).$$

2.3 GEGENERALISEERDE RUNGE-KUTTAMETHODEN

We hebben gezien hoe *Jacobi-iteratie* van de trapezium-regel tot de klasse van Runge-Kuttaformules leidde. Men kan zich nu afvragen tot wat voor formules *Newton-iteratie* zal leiden. Het zal blijken dat Newton-iteratie tot Runge-Kutta-achtige formules leidt, waarin de parameters $\lambda_{j,l}$ dan operatoren zijn in plaats van scalairen.

De trapezium-regel (2.2.1) geeft aanleiding tot het oplossen van de vergelijking

$$(2.3.1) \qquad \vec{y} - \tfrac{1}{2}h_n\vec{f}(\vec{y}) = \vec{y}_n + \tfrac{1}{2}h_n\vec{f}(\vec{y}_n) .$$

Laat $J(\vec{y})$ de Jacobiaan van $\vec{f}$ in $\vec{y}$ voorstellen, dan levert Newton-iteratie het rekenschema

$$
\begin{aligned}
&\vec{y}_{n+1}^{(0)} = \vec{y}_n \ , \\
&\vec{y}_{n+1}^{(j)} = \vec{y}_{n+1}^{(j-1)} - [I-\tfrac{1}{2}h_n J(\vec{y}_{n+1}^{(j-1)})]^{-1} \cdot \\
&\qquad \cdot [\vec{y}_{n+1}^{(j-1)} - \tfrac{1}{2}h_n \vec{f}(\vec{y}_{n+1}^{(j-1)}) - \vec{y}_n - \tfrac{1}{2}h_n \vec{f}(\vec{y}_n)], \quad j = 1,2,\ldots,m, \\
&\vec{y}_{n+1} = \vec{y}_{n+1}^{(m)} \ ,
\end{aligned}
\tag{2.3.2}
$$

waarin $\vec{y}_n$ weer als beginapproximatie is genomen en m het aantal uitgevoerde iteraties voorstelt. Volgens dezelfde redenatie toegepast op de Jacobigeitereerde trapezium-regel (2.2.2), kan men laten zien dat (2.3.2) behoort tot de klasse van formules

$$
\begin{aligned}
&\vec{y}_{n+1}^{(0)} = \vec{y}_n \ , \\
&\vec{y}_{n+1}^{(j)} = \vec{y}_n + h_n \sum_{l=0}^{j-1} \Lambda_{j,l} \vec{f}(\vec{y}_{n+1}^{(l)}) \quad , \quad j = 1,2,\ldots,m, \\
&\vec{y}_{n+1} = \vec{y}_{n+1}^{(m)} \ ,
\end{aligned}
\tag{2.3.3}
$$

waarin de $\Lambda_{j,l}$ matrixoperatoren zijn. Voor het bijzondere geval van (2.3.2) zijn de matrices $\Lambda_{j,l}$ rationale uitdrukkingen in de Jacobianen $J(\vec{y}_{n+1}^{(i)})$, $i = 0,1,\ldots,j-1$. Schema (2.3.3) zullen we de algemene formule van een *m-punts gegeneraliseerde Runge-Kuttaformule* noemen. Evenals bij traditionele Runge-Kuttaformules kan men de voorstelling

$$
\begin{aligned}
&\vec{y}_{n+1} = \vec{y}_n + \sum_{j=0}^{m-1} \Lambda_{m,j} \vec{k}_n^{(j)} \ , \\
&\vec{k}_n^{(j)} = h_n \vec{f}(y_n + \sum_{l=0}^{j-1} \Lambda_{j,l} \vec{k}_n^{(l)})
\end{aligned}
\tag{2.3.4}
$$

kiezen. Het verband tussen beide voorstellingen wordt weer gegeven door (2.2.5).

In de practijk is het rekenen met operatoren $\Lambda_{j,l}$, waarin Jacobianen in verschillende punten $\vec{y}_{n+1}^{(i)}$ voorkomen, in het algemeen te duur (verschillende Jacobiaan-evaluaties per integratiestap en verschillende LU-ontbindingen wanneer deze rationaal voorkomen). We zullen ons dan ook beperken

tot operatoren $\Lambda_{j,l}$ die rationale functies of polynomen in $h_n J(\vec{y}_n)$ zijn. Een generaliseerde Runge-Kuttaformule wordt dan volledig vastgelegd door de matrix van functies $(\Lambda_{j,l}(z))$.

Voorbeelden 2.3.1

Rosenbrock [1963] was de eerste die formules uit de klasse (2.3.3) construeerde. De belangrijkste formule wordt gegeven door de genererende matrix

(2.3.5)
$$\begin{pmatrix} 0 & 0 \\ \dfrac{\sqrt{2}-1}{2-(2-\sqrt{2})z} & 0 \\ 0 & \dfrac{2}{2-(2-\sqrt{2})z} \end{pmatrix}.$$

Een tweede voorbeeld is de formule van Calahan [1968]:

(2.3.6)
$$\begin{pmatrix} 0 & 0 \\ \dfrac{-8\sqrt{3}}{12-(6+2\sqrt{3})z} & 0 \\ \dfrac{9}{12-(6+2\sqrt{3})z} & \dfrac{3}{12-(6+2\sqrt{3})z} \end{pmatrix}.$$

Opgaven 2.3.1

(1) Stel de genererende matrices op voor de formules die verkregen worden door de gemodificeerde Newton-iteratiemethode $(J(\vec{y}^{(j)}_{n+1}) = J(\vec{y}_n))$ op de trapezium-regel toe te passen.

2.4 CONSISTENTIE

Tot dusver zijn de gedefinieerde klassen van eenstapsmethoden slechts *formeel* methoden voor de numerieke integratie van beginwaardeproblemen;

alhoewel de beschouwde integratieformules geformuleerd zijn in termen van de rechterlidfunctie $\vec{f}$ en zijn afgeleiden, heeft de differentie-oplossing voor willekeurige waarden van de parameters in deze formules nauwelijks iets met de analytische oplossing te maken. Zoals al opgemerkt bij de behandeling van algemene differentieschema's zal men op z'n minst willen dat de analytische oplossing min of meer voldoet aan het differentieschema en dat voor afnemende integratiestappen ook het residu afneemt, met andere woorden dat differentiaalvergelijking en differentieschema *consistent* zijn.

Definitie 2.4.1

Een eenstapsmethode heet een consistente benadering van de differentiaalvergelijking (1.1.1) in het punt x_n wanneer voor iedere oplossing $\vec{y}$ van (1.1.1)

$$(2.4.1) \qquad I_n(\vec{y}(x_{n+1}),\vec{y}(x_n)) \to \vec{0} \quad \text{als} \quad h_n \to 0.$$

Het linkerlid van (2.4.1) wordt de *lokale afbreekfout* genoemd. Wanneer deze afbreekfout zich als h_n^{p+1} gedraagt voor $h_n \to 0$, dan heet de eenstapsmethode *consistent van de orde p*.

De analyse van de consistentie van een integratieformule bestaat in het algemeen uit het ontwikkelen van de lokale afbreekfout in een Taylorreeks in het punt x_n.

2.4.1 Consistentie van Taylor-methoden

Substitutie van een oplossing $\vec{y}$ in (2.1.4) en ontwikkeling van $\vec{y}(x_n+h_n)$ in machten van h_n geeft

$$\begin{aligned}(2.4.2) \qquad I_n(\vec{y}(x_n+h_n),\vec{y}(x_n)) = {} & (1-\beta_0)\vec{y}(x_n) + (1+\alpha_1-\beta_1)h_n\vec{y}'(x_n) + \\ & + \tfrac{1}{2}(1+2\alpha_1+2\alpha_2-2\beta_2)h_n^2\vec{y}''(x_n) + \\ & + \tfrac{1}{6}(1+3\alpha_1+6\alpha_2+6\alpha_3-6\beta_3)h_n^3\vec{y}'''(x_n) + \\ & + \tfrac{1}{24}(1+4\alpha_1+12\alpha_2+24\alpha_3+24\alpha_4-24\beta_4)h_n^4\vec{y}''''(x_n) + \\ & + O(h_n^5) .\end{aligned}$$

Dit levert direct de volgende tabel van consistentievoorwaarden

Tabel 2.4.1 Consistentievoorwaarden voor Taylor-formules ($\alpha_0=1$)

$p \geq 0$	$\beta_0 = 1$
$p \geq 1$	$\beta_1 = 1 + \alpha_1$
$p \geq 2$	$\beta_2 = \frac{1}{2} + \alpha_1 + \alpha_2$
$p \geq 3$	$\beta_3 = \frac{1}{6} + \frac{1}{2}\alpha_1 + \alpha_2 + \alpha_3$
$p \geq 4$	$\beta_4 = \frac{1}{24} + \frac{1}{6}\alpha_1 + \frac{1}{2}\alpha_2 + \alpha_3 + \alpha_4$
...	...

Deze tabel dient zo gelezen te worden dat de voorwaarden voor een zekere orde van consistentie p_0 behalve de achter p_0 vermelde voorwaarde ook alle voorgaande voorwaarden omvat. Algemeen zal aan $p + 1$ voorwaarden voldaan moeten worden om p^e orde consistentie te krijgen. Aangezien een (m_1,m_2)-Taylorformule $m_1 + m_2 + 1$ vrije parameters heeft, geldt voor p de ongelijkheid

(2.4.3) $\quad p \leq m_1 + m_2$.

Meetkundig gezien kan men zeggen dat in de $(\vec{\alpha},\vec{\beta})$-ruimte van Taylor-formules ($\vec{\alpha}=(1,\alpha_1,\alpha_2,\ldots)$, $\vec{\beta} = (\beta_0,\beta_1,\ldots)$) de deelruimte van p^e orde consistentie Taylor-formules gevormd wordt door de gemeenschappelijke punten van $p + 1$ hypervlakken waarvan de vergelijkingen in tabel 2.4.1 gegeven zijn.

Opgaven 2.4.1

(1) Bepaal de orde van consistentie van de formules (2.1.5) - (2.1.7).

(2) Bepaal de orde van consistentie van de Jacobi-geïtereerde trapezium-regel na 1, 2 en 3 iteraties.

(3) Toon aan dat elke Jacobi-iteratie van een impliciete Taylor-formule de orde van consistentie met één verhoogt totdat de orde van de Taylor-formule is bereikt.

2.4.2 Consistentie van Runge-Kuttamethoden

De consistentie-analyse van Runge-Kuttaformules is veel moeilijker dan die voor Taylor-formules, omdat de Taylor-ontwikkeling van de afbreekfout niet direct in afgeleiden van $\vec{y}$ uitgedrukt kan worden, maar uitgedrukt moet worden in $\vec{f}$ en zijn afgeleiden. Deze analyse is voor $p \leq 8$ systematisch uitgevoerd door Butcher [1963] zowel voor *expliciete* als *impliciete* Runge-Kuttaformules. In deze syllabus zal de consistentie-analyse voor 2-puntsformules gegeven worden en verder volstaan worden met een opsomming van de consistentievoorwaarden voor $p \leq 4$.

Beschouw de algemene 2-puntsformule

$$(2.4.4) \qquad \vec{y}_{n+1} = E_n(\vec{y}_n) \equiv \vec{y}_n + \theta_0 h_n \vec{f}(\vec{y}_n) + \theta_1 h_n \vec{f}(\vec{y}_n + \lambda h_n \vec{f}(\vec{y}_n)) \ ,$$

waarin we $\lambda_{1,0} = \lambda$, $\lambda_{2,0} = \theta_0$ en $\lambda_{2,1} = \theta_1$ gesteld hebben om dubbel-indices in de formules te vermijden. Wanneer een oplossing $\vec{y}$ in (2.4.4) gesubstitueerd wordt, moeten vervolgens $\vec{y}(x_n+h_n)$ en $E_n(\vec{y}(x_n))$ in machten van h_n ontwikkeld worden, waarbij de coefficienten uitgedrukt dienen te worden in $\vec{f}$ om coefficienten-vergelijking mogelijk te maken. De ontwikkeling van $\vec{y}(x_n+h_n)$ luidt

$$(2.4.5) \qquad \vec{y}(x_n+h_n) = [\vec{y}+h_n\vec{f}(\vec{y})+\tfrac{1}{2}h_n^2(\vec{f}(\vec{y})\cdot\nabla)\vec{f}(\vec{y})+ \ \ldots$$

$$\ldots + \tfrac{1}{j!}h_n^j(\vec{f}(\vec{y})\cdot\nabla)^{j-1}\vec{f}(\vec{y})+\ldots] \ (x)\big|_{x=x_n} \ ,$$

waarin ∇ de reeds eerder ingevoerde gradient-operator voorstelt. De ontwikkeling van $E_n(\vec{y}(x_n))$ in machten van h_n brengt ons tot Taylor-ontwikkelingen van functies van meerdere variabelen; er geldt

$$(2.4.6) \qquad \vec{f}(\vec{y}+\vec{\eta}) = \vec{f}(\vec{y}) + (\vec{\eta}\cdot\nabla)\vec{f}(\vec{y}) + \tfrac{1}{2}(\eta\cdot\nabla)^2\vec{f}(\vec{y}) + \ \ldots$$

$$\ldots + \tfrac{1}{j!}(\vec{\eta}\cdot\nabla)^j\vec{f}(\vec{y}) + \ \ldots \ ,$$

waarin $\vec{\eta}$ een increment-vector onafhankelijk van $\vec{y}$ voorstelt.
Toepassing van deze ontwikkeling met $\vec{\eta} = \lambda h_n \vec{f}(\vec{y}(x_n))$ geeft voor $E_n(\vec{y}(x_n))$ de reeks

$$(2.4.7)\qquad E_n(\vec{y}(x_n)) = [\vec{y}+\theta_0 h_n \vec{f}(\vec{y})+\theta_1 h_n(\vec{f}(\vec{y})+\lambda h_n(\vec{f}(\vec{y}(x_n))\cdot\nabla)\vec{f}(\vec{y})+$$
$$+\tfrac{1}{2}\lambda^2 h_n^2(\vec{f}(\vec{y}(x_n)\cdot\nabla)^2\vec{f}(\vec{y})+\ldots)]\,(x)\Big|_{x=x_n}\ .$$

Uit (2.4.5) en (2.4.7) volgt voor de afbreekfout de ontwikkeling

$$\begin{aligned}\vec{y}(x_n+h_n) - E_n(\vec{y}(x_n)) = {} & (1-\theta_0-\theta_1)h_n\vec{f}(\vec{y}(x_n)) + \\ & + (\tfrac{1}{2}-\theta_1\lambda)h_n^2(\vec{f}(\vec{y}(x_n)\cdot\nabla)\vec{f}(\vec{y})\Big|_{x=x_n} + \\ & + \tfrac{1}{6}h_n^3(\vec{f}(\vec{y})\cdot\nabla)^2\vec{f}(\vec{y})\Big|_{x=x_n} + \\ & - \tfrac{1}{2}\theta_1\lambda^2 h_n^3(\vec{f}(\vec{y}(x_n))\cdot\nabla)^2\vec{f}(\vec{y})\Big|_{x=x_n} + \\ & + O(h_n^4).\end{aligned}$$

Hieruit volgt direct dat formule (2.4.4) 1^e orde consistent is als

$$\theta_0 + \theta_1 = 1$$

en 2^e orde consistent als bovendien

$$\theta_1\lambda = \frac{1}{2}$$

Kennelijk kan een 2-puntsformule niet 3^e orde consistent zijn, tenzij

$$(2.4.8)\qquad (\vec{f}(\vec{y})\cdot\nabla)^2\vec{f}(\vec{y})\Big|_{x=x_n} = (\vec{f}(\vec{y}_n)\cdot\nabla)^2\vec{f}(\vec{y})\Big|_{x=x_n}\ .$$

Wanneer deze relatie geldt geeft de keuze

$$\theta_1\lambda^2 = \frac{1}{3}$$

een 3^e orde consistente benadering. Relatie (2.4.8) is echter alleen geldig wanneer $\vec{f}$ een constante is; dit betekent dat voor langzaam varierende rechterlidfuncties door deze keuze de afbreekfout geminimaliseerd wordt.

Om de consistentievoorwaarden voor m-puntsformules te formuleren definieren we eerst een aantal nieuwe parameters:

$$
\begin{aligned}
\beta_1 &= \sum_{j=0}^{m-1} \lambda_{m,j} \ , \\
\beta_2 &= \sum_{j=1}^{m-1} \lambda_{m,j} \sum_{l=0}^{j-1} \lambda_{j,l} \ , \\
\beta_3 &= \sum_{j=2}^{m-1} \lambda_{m,j} \sum_{l=1}^{j-1} \lambda_{j,l} \sum_{k=0}^{l-1} \lambda_{l,k} \ , \\
\beta_{3,1} &= \sum_{j=1}^{m-1} \lambda_{m,j} \left(\sum_{l=0}^{j-1} \lambda_{j,l} \right)^2 \ , \\
\beta_4 &= \sum_{j=3}^{m-1} \lambda_{m,j} \sum_{l=2}^{j-1} \lambda_{j,l} \sum_{i=1}^{l-1} \lambda_{l,i} \sum_{k=0}^{i-1} \lambda_{i,k} \ , \\
\beta_{4,1} &= \sum_{j=2}^{m-1} \lambda_{m,j} \sum_{l=1}^{j-1} \lambda_{j,l} \left(\sum_{k=0}^{l-1} \lambda_{l,k} \right)^2 \ , \\
\beta_{4,2} &= \sum_{j=2}^{m-1} \lambda_{m,j} \sum_{l=0}^{j-1} \lambda_{j,l} \sum_{k=1}^{j-1} \lambda_{j,k} \sum_{i=0}^{k-1} \lambda_{k,i} \ , \\
\beta_{4,3} &= \sum_{j=1}^{m-1} \lambda_{m,j} \left(\sum_{l=0}^{j-1} \lambda_{j,l} \right)^3 \ .
\end{aligned}
\tag{2.4.9}
$$

Uitgedrukt in deze parameters luiden de consistentievoorwaarden zoals gegeven in tabel 2.4.2. (De parameters β_j en $\beta_{j,l}$ treden op als coefficienten van de j-de orde termen in de reeksontwikkeling van de afbreekfout.)

Tabel 2.4.2 Consistentievoorwaarden voor Runge-Kuttaformules

$p \geq 1$	$\beta_1 = 1$
$p \geq 2$	$\beta_2 = \frac{1}{2}$
$p \geq 3$	$\beta_3 = \frac{1}{6}$, $\beta_{3,1} = \frac{1}{3}$
$p \geq 4$	$\beta_4 = \frac{1}{24}$, $\beta_{4,1} = \frac{1}{12}$, $\beta_{4,2} = \frac{1}{8}$, $\beta_{4,3} = \frac{1}{4}$

Wanneer een Runge-Kuttamethode toegepast wordt op een *lineaire* differentiaalvergelijking reduceert deze tot een Taylor-formule; in de machtreeks voor de afbreekfout verdwijnen dan de termen met de parameters $\beta_{j,l}$ (dubbele index) als coefficienten, zodat de consistentievoorwaarden tot

$$\beta_j = \frac{1}{j!}, \quad j = 1,2,\ldots,p \tag{2.4.10}$$

reduceren (vergelijk de consistentievoorwaarden voor expliciete Taylorformules).

Voor niet-lineaire differentiaalvergelijkingen moet een aantal niet-lineaire vergelijkingen opgelost worden. In het algemeen is het aantal vergelijkingen kleiner dan het aantal Runge-Kuttaparameters voor gegeven m en bijbehorende maximale orde van consistentie (zie tabel 2.4.3). Dit betekent dat er hele klassen van maximaal consistente Runge-Kuttaformules zijn. Merk op dat in het geval van een 6^e orde 7-puntsformule, 37 vergelijkingen in 28 onbekenden opgelost moeten worden!

Tabel 2.4.3 Maximale orde van consistentie bij Runge-Kuttaformules

m	1	2	3	4	5	6	7	$m \geq 8$
maximale orde	1	2	3	4	4	5	6	$p \geq m-2$
aantal parameters	1	3	6	10	15	21	28	$\frac{1}{2}m(m+1)$
aantal vergelijkingen	1	2	4	8	8	17	37	

Voorbeeld 2.4.1

Kutta [1901] gaf een klasse van 4^e orde formules gegenereerd door

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ \frac{\lambda-1}{2\lambda} & \frac{1}{2\lambda} & 0 & 0 \\ 0 & 1-\lambda & \lambda & 0 \\ \frac{1}{6} & \frac{2-\lambda}{3} & \frac{\lambda}{3} & \frac{1}{6} \end{pmatrix},$$

waarin λ vrij te kiezen is. Uiteraard is deze klasse een deelklasse van de algemene 4^e orde 4-puntsformules.

Opgaven 2.4.2

(1) Leid de algemene formule af voor de 2-punts, 2^e orde en 3-punts, 3^e orde Runge-Kuttamethoden.

(2) Bewijs dat de 2-puntsformule

$$\vec{y}_{n+1} = \vec{y}_n + h_n\vec{f}(\vec{y}_n) + \tfrac{1}{2}h_n^2\vec{g}(\vec{y}_n + \tfrac{1}{3}h_n\vec{f}(\vec{y}_n))$$

uit de klasse (2.2.9) derde orde consistent is.

2.4.3 Consistentie van gegeneraliseerde Runge-Kuttamethoden

Op het Mathematisch Centrum is onlangs de gegeneraliseerde 2-puntsformule

$$\vec{y}_{n+1} = \vec{y}_n + h_n\theta_0(h_nJ(\vec{y}_n))\vec{f}(\vec{y}_n) + h_n\theta_1(h_nJ(\vec{y}_n))\vec{f}(\vec{y}_n + h_n\Lambda(h_nJ(\vec{y}_n))\vec{f}(\vec{y}_n))$$

Tabel 2.4.4 Consistentievoorwaarden voor gegeneraliseerde Runge-Kuttaformules

$p \geq 1$	$\theta_0 + \theta_1 = 1$
$p \geq 2$	$\theta_0' + \theta_1' + \theta_1\lambda = \frac{1}{2}$
$p \geq 3$	$\frac{1}{2}(\theta_0'' + \theta_1'') + \theta_1\lambda' + \theta_1'\lambda = \frac{1}{6}$
	$\frac{1}{2}\theta_1\lambda^2 = \frac{1}{6}$
$p \geq 4$	$\frac{1}{6}(\theta_0''' + \theta_1''') + \frac{1}{2}(\theta_1\lambda'' + 2\theta_1'\lambda' + \theta_1''\lambda) = \frac{1}{24}$
	$\theta_1\lambda\lambda' = \frac{1}{8}$
	$\frac{1}{2}\theta_1'\lambda^2 = \frac{1}{24}$
	$\frac{1}{6}\theta_1\lambda^3 = \frac{1}{24}$

Tabel 2.4.5 Consistentievoorwaarden voor gegeneraliseerde Runge-Kuttaformules in expliciete vorm

p = 1	$\theta_0 = 1 - \theta_1$
p = 2	$\theta_0 = 1 - \frac{1}{2\lambda}[1-2\theta_0'-2\theta_1']$ $\theta_1 = \frac{1}{2\lambda}[1-2\theta_0'-2\theta_1']$
p = 3	$\theta_0 = 1 - \frac{1}{3\lambda^2}$ $\theta_1 = \frac{1}{3\lambda^2}$ $\theta_0' = \frac{1}{2} - \frac{1}{2\lambda}[1-\theta_0''-\theta_1''] + \frac{\lambda'}{3\lambda^3}$ $\theta_1' = \frac{1}{6\lambda}\theta[1-3\theta_0''-3\theta_1''] - \frac{\lambda'}{3\lambda^3}$
p = 4	$\theta_0 = \frac{11}{27}$ $\theta_1 = \frac{16}{27}$ $\theta_0' = -\frac{5}{54}$ $\theta_1' = \frac{4}{27}$ $\theta_0'' = \frac{4}{81}[9\theta_0'''+9\theta_1'''+16\lambda''] - \frac{2}{9}$ $\theta_1'' = -\frac{4}{81}[9\theta_0'''+9\theta_1'''+16\lambda'']$ $\lambda = \frac{3}{4}$ $\lambda' = \frac{9}{32}$

grondig onderzocht. We zullen wat de consistentie betreft hier volstaan met het noemen van de consistentievoorwaarden. Een gedetailleerde analyse kan men vinden in van der Houwen [1975].

In tabel 2.4.4 zijn de consistentievoorwaarden uitgedrukt in de waarden $\theta_0,\theta_0',\theta_0'',\ldots$, $\theta_1,\theta_1',\theta_1'',\ldots$ en $\lambda,\lambda',\lambda'',\ldots$ van de afgeleiden van respectivelijk de functies Θ_0, Θ_1 en Λ in $z = 0$.
Het is eenvoudig te verifieren dat deze vergelijkingen de in tabel 2.4.5 gegeven oplossingen hebben; hierin zijn alle afgeleiden in het rechterlid nog vrij te kiezen.

Voorbeeld 2.4.2

We zullen met behulp van tabel 2.4.5 een 4^e orde formule construeren waarin de functies θ_0, θ_1 en Λ polynomen zijn. Volgens de tabel mogen voor $p = 4$ de afgeleiden θ_0''', θ_1''' en λ'' vrij gekozen worden; stel dat we kiezen

$$\theta_0''' = \theta_1''' = \lambda'' = 0,$$

dan geldt

$$\theta_0 = \frac{11}{27} \ , \ \theta_0' = -\frac{5}{54} \ , \ \theta_0'' = -\frac{2}{9}$$
$$\theta_1 = \frac{16}{27} \ , \ \theta_1' = \frac{4}{27} \ , \ \theta_1'' = 0$$
$$\lambda = \frac{3}{4} \ , \ \lambda' = \frac{9}{39}$$

zodat

$$\theta_0(z) = \frac{11}{27} - \frac{5}{54} z - \frac{1}{9} z^2 + z^4 g_0(z)$$
$$\theta_1(z) = \frac{16}{27} + \frac{4}{27} z + z^4 g_1(z) \ ,$$
$$\Lambda(z) = \frac{3}{4} + \frac{9}{32} z + z^3 g_2(z) \ ,$$

waarin g_1, g_2 en g_3 willekeurige reguliere functies mogen zijn. Bijvoorbeeld $g_0 \equiv g_1 \equiv g_2 \equiv 0$ leidt tot de integratieformule

$$\vec{y}_{n+1} = \vec{y}_n + h_n[\frac{11}{27} - \frac{5}{54}h_n J_n - \frac{1}{9}h_n^2 J_n^2]\vec{f}(\vec{y}_n) +$$
$$+ h_n[\frac{16}{27} + \frac{4}{27}h_n J_n]\vec{f}(\vec{y}_n + h_n[\frac{3}{4} + \frac{9}{32}h_n J_n]\vec{f}(\vec{y}_n)) \ ,$$

waarin $J_n = J(\vec{y}_n)$.

Opgaven 2.4.3

(1) Bepaal de orde van consistentie van de Rosenbrock-formule (2.3.5) en van de Calahan-formule (2.3.6).

(2) Bepaal de orde van consistentie van de Newton-geïtereerde trapezium regel na 1 en 2 iteraties met $J(\vec{y}_{n+1}^{(j)}) = J(\vec{y}_n)$.

2.4.4 Methoden met integratiestap-afhankelijke parameters

In het voorgaande is stilzwijgend aangenomen dat de parameters α_j, β_j in de Taylor-formules, $\lambda_{j,l}$ in de Runge-Kuttaformules en de parameters $\theta_0,\theta_0',\ldots,\ \theta_1,\theta_1',\ldots,\ \lambda,\lambda',\ldots$ in de gegeneraliseerde 2-puntformule niet van h_n afhangen. Wanneer deze parameters wel van h_n afhangen veranderen de consistentievoorwaarden in de tabellen 2.4.1, 2.4.2, en 2.4.4 in zoverre dat de voorwaarden genoemd voor $p \geq j$ moeten gelden op een term van de orde $p + 1 - j$ in h_n na. Dus de consistentievoorwaarden voor een 3^e orde Taylor-formule met h_n-afhankelijke parameters worden (vergelijk tabel 2.4.1)

$$\beta_0 = 1 + O(h_n^4)$$
$$\beta_1 = 1 + \alpha_1 + O(h_n^3)$$
$$\beta_2 = \frac{1}{2} + \alpha_1 + \alpha_2 + O(h_n^2)$$
$$\beta_3 = \frac{1}{6} + \frac{1}{2}\alpha_1 + \alpha_2 + \alpha_3 + O(h_n) \ .$$

2.5 CONVERGENTIE

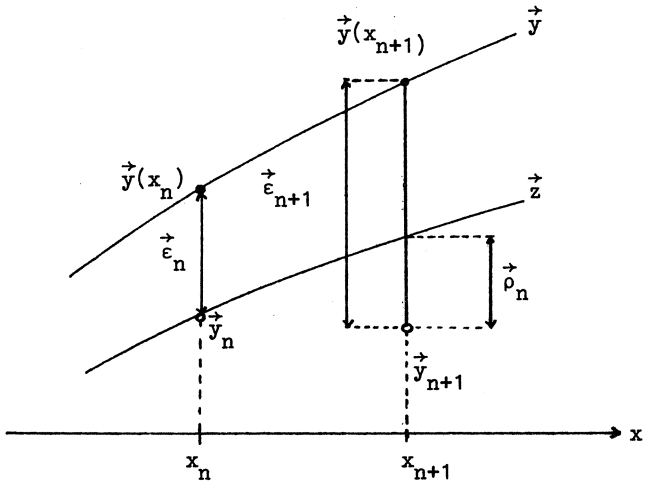
We hebben gezien dat de consistentievoorwaarden een deelruimte definieren in de parameterruimte waarop de eenstapsmethode is gedefinieerd. We zullen nu laten zien dat de convergentievoorwaarden deze deelruimte niet verder inperken, maar alleen nog (milde) eisen aan de rechterlidfunctie $\vec{f}$ stellen.

In de convergentie-analyse definieert men de *lokale discretiseringsfout* (zie figuur 2.5.1)

$$\vec{\rho}_n = \vec{z}(x_n,\vec{y}_n;x_{n+1}) - \vec{y}_{n+1} \qquad (2.5.1)$$

en de *globale discretiseringsfout*

$$\vec{\varepsilon}_n = \vec{y}(x_n) - \vec{y}_n \ . \qquad (2.5.2)$$



Figuur 2.5.1 Lokale en globale discretiseringsfout

De convergentievoorwaarden kunnen afgeleid worden met de algemene convergentiestelling uit hoofdstuk I. In dit speciale geval van eenstapsformules is het echter eenvoudiger om een directe afleiding te geven. Het uitgangspunt in deze afleiding is de volgende stelling:

Stelling 2.5.1

Stel dat de eenstapsoperator E_n in de omgeving van de analytische oplossing y in $x = x_n$ voldoet aan de Lipschitzvoorwaarde

$$(2.5.3) \qquad \|E_n(\vec{u})-E_n(\vec{v})\| \leq L_n \|\vec{u}-\vec{v}\| \ ,$$

dan voldoet de globale discretiseringsfout aan de recurrente betrekking

$$(2.5.4) \qquad \|\vec{\varepsilon}_{n+1}\| \leq L_n \|\vec{\varepsilon}_n\| + \|\vec{y}(x_{n+1})-E_n(\vec{y}(x_n))\| \ .$$

Bewijs

De globale discretiseringsfout $\vec{\varepsilon}_{n+1}$ kan geschreven worden als

$$\vec{\varepsilon}_{n+1} = \vec{y}(x_{n+1}) - \vec{y}_{n+1} = [E_n(\vec{y}(x_n)) - E_n(\vec{y}_n)] +$$

$$+ [\vec{y}(x_{n+1}) - E_n(\vec{y}(x_n))] .$$

Toepassing van (2.5.3) levert dan direct ongelijkheid (2.5.4).

Uit (2.5.4) kan een bovengrens voor de globale fout afgeleid worden, die alleen afhangt van de Lipschitz-constante L_n en de lokale afbreekfout van de integratieformule. Daartoe definieren we de maximale integratiestap na n stappen

$$h = \max_{\nu} h_\nu ,$$

de maximale foutconstante van de lokale afbreekfouten na n stappen

$$C_1 = \max_{\nu} \frac{\|\vec{y}(x_{\nu+1}) - E_\nu(\vec{y}(x_\nu))\|}{h_\nu^{p+1}}$$

en de maximale waarde van $(L_\nu - 1)/h_\nu$ na n stappen

$$C_2 = \max_{\nu} \frac{L_\nu - 1}{h} .$$

Er geldt nu

Stelling 2.5.2

De globale discretiseringsfout $\vec{\varepsilon}_{n+1}$ voldoet aan de ongelijkheid

$$\|\vec{\varepsilon}_{n+1}\| \leq \begin{cases} C_1(x_{n+1}-x_0)h^p & \text{als } C_2 \leq 0 \\ \frac{C_1}{C_2}[\exp[C_2(x_{n+1}-x_0)]-1]h^p & \text{als } C_2 > 0 \end{cases} .$$

Bewijs

Voor $C_2 \leq 0$ $(L_\nu \leq 1)$ volgt uit (2.5.4) dat

$$\|\vec{\varepsilon}_{n+1}\| \leq \|\vec{\varepsilon}_n\| + C_1 h_n^{p+1} \leq \|\vec{\varepsilon}_{n-1}\| + C_1 h_{n-1}^{p+1} + C_1 h_n^{p+1} \leq \ldots$$

$$\ldots \leq \sum_{\nu=0}^{n} C_1 h_\nu^{p+1} \leq C_1 h^p \sum_{\nu=0}^{n} h_\nu = C_1 h^p (x_{n+1} - x_0) ,$$

en voor $C_2 > 0$ ($L_\nu > 1$) vinden we

$$\|\vec{\varepsilon}_{n+1}\| \leq (1+C_2h_n)\,\|\vec{\varepsilon}_n\| + C_1h_n^{p+1} \leq \ldots$$

$$\ldots \leq C_1[h_n^{p+1}+h_{n-1}^{p+1}(1+C_2h_n) + \ldots h_0^{p+1}\prod_{\nu=1}^{n}(1+C_2h_\nu)]$$

$$\leq C_1h^p[h_n+h_{n-1}\exp(C_2h_n)+\ldots+h_0\exp[C_2(h_1+\ldots+h_n)]]$$

$$\leq C_1h^p\int_0^{x_{n+1}-x_0}\exp[C_2x]dx = \frac{C_1}{C_2}h^p[\exp[C_2(x_{n+1}-x_0)]-1].$$

Opgaven 2.5.1

(1) Toon aan dat de in stelling 2.5.2 gegeven bovengrens voor $\vec{\varepsilon}_{n+1}$ een continue functie is van C_2.

(2) Bewijs dat de globale discretiseringsfout van de (0,2)-Taylorformule voor de vergelijking $y' = y$ zich gedraagt als

$$\frac{1}{6}y(x_{n+1})x_{n+1}h^2 \qquad \text{voor } h \to 0\ ,$$

waarin $x_n = nh$. Toon verder aan dat de in stelling 2.5.2 gegeven bovengrens zich gedraagt als

$$\frac{1}{6}y_{n+1}[e^{x_{n+1}}-1]h^2 \qquad \text{voor } h \to 0\ .$$

2.6 STABILITEIT

In paragraaf 1.6.4 is aangetoond dat het verschil tussen de oplossing die we zoeken (analytische oplossing $\vec{y}$) en de oplossing die numeriek berekend wordt (numerieke oplossing $\{\vec{y}_n^*\}$) begrensd wordt door de som van de globale discretiseringsfout en de numerieke fout. Van de discretiseringsfout weten we nu dat elke eenstapsmethode die consistent is en waarvoor de operator E_n een Lipschitz-constante heeft die op orde h na ≤ 1 is, convergeert. Voor de in deze syllabus beschouwde methoden legt deze voorwaarde slechts beperkingen op aan de rechterlidfunctie $\vec{f}$ van de differentiaalvergelijking. Om de numerieke fout onder controle te houden zullen we echter strengere voorwaarden aan de eenstapsoperator E_n moeten opleggen

welke ook consequenties hebben voor de nog vrije parameters in E_n. We zul len het begrip *stabiliteitsfunctie* van een integratieproces invoeren. Deze functie karakteriseert het stabiliteitsgedrag van het proces wanneer dit op *lineaire* differentiaalvergelijkingen wordt toegepast. In geval van *niet-lineaire* vergelijkingen geeft de stabiliteitsfunctie slechts ruwe informatie over het stabiliteitsgedrag, maar voor de meeste practische toepassingen is deze functie toch gebleken een bevredigende indicator te zijn of afrondingsfouten al of niet accumuleren.

2.6.1 De Fréchet-afgeleide van de inverse differentie-operator

Analoog aan de lokaal analytische oplossing $\vec{z}$ definieren we de *lokale differentieoplossing* $\vec{w} = \vec{w}(x_n,\vec{y}_n;x)$ als de functie $E_n(\vec{y}_n)$ waarin h_n vervangen wordt door $x - x_n$. Voorts definieren we de *lokale numerieke fout*

$$\vec{\rho}^{**}_n = \vec{w}(x_n,\vec{y}^{**}_n;x_{n+1}) - \vec{y}^{**}_{n+1} \qquad (2.6.1)$$

en de *globale numerieke fout*

$$\vec{\varepsilon}^{**}_n = \vec{y}_n - \vec{y}^{**}_n \; . \qquad (2.6.2)$$

Het differentieschema voor de numerieke oplossing $\{\vec{y}^{**}_n\}$ luidt nu

$$\vec{y}^{**}_{n+1} = E_n(\vec{y}^{**}_n) - \vec{\rho}^{**}_n \; , \quad n = 0,1,\ldots,N-1. \qquad (2.6.3)$$

In paragraaf 1.6.4 zijn stabiliteitsdefinities gegeven voor algemene differentieschema's $D_H Y_H = G_H$, of in expliciete vorm

$$Y_H = A_H G_H \; . \qquad (2.6.4)$$

Om te zien hoe deze definities luiden voor het bijzondere geval van een-stapsmethoden schrijven we schema (2.0.1) in de vorm (2.6.4):

$$Y_H = \{y_n\}_{n=0}^N \; , \quad G_H = \{g_n\}_{n=0}^N = \{\vec{g}_0,0,\ldots,0\} \; , \qquad (2.6.5)$$

$$\vec{y}_{n+1} = E_n(\vec{y}_n) + \vec{g}_{n+1} \; .$$

Vergelijken we (2.6.5) met (2.6.3) dan zien we dat (2.6.3) opgevat kan worden als schema (2.6.5) waarin de rij $G_H = \{\vec{g}_n\}$ verstoord is met de rij van verstoringen

$$\Delta G_H = \{0, -\vec{\rho}^{**}_0, -\vec{\rho}^{**}_1, \ldots, -\vec{\rho}^{**}_{N-1}\} .$$

Volgens relatie (1.6.21) geldt nu

$$|||Y^*_H - Y_H||| \leq ||A'_H(\bar{G}_H)|| \cdot |||\Delta G_H||| ,$$

waarin $A'_H(\bar{G}_H)$ de Fréchet-afgeleide is van de door (2.6.5) gedefinieerde operator A_H in het punt $\bar{G}_H$ ($\bar{G}_H$ in de omgeving van G_H). We laten het aan de lezer over om aan te tonen dat $A'_H(G_H)$ geschreven kan worden als

(2.6.6)
$$\begin{pmatrix} I & 0 & \cdots & 0 \\ E'_0 & I & & \\ E'_1E'_0 & E'_1 & I & \\ E'_2E'_1E'_0 & E'_2E'_1 & E'_2 & I \\ \vdots & \vdots & & \ddots \\ \prod_{N-1}^{0} E'_n & \prod_{N-1}^{1} E'_n & \cdots & I \end{pmatrix} ,$$

waarin met E'_n de Fréchet-afgeleide van de operator E_n in het punt $\vec{y}_n$ bedoeld wordt. Voor de norm van $||A'_H(G_H)||$ geldt nu

$$||A'_h(G_H)|| = \operatorname*{Sup}_{\Delta G_H} \frac{|||V_H|||}{|||\Delta G_H|||}$$

waarin G_H de rij van verstoringen $-\vec{\rho}^{**}_n$ en V_H de rij

$$\{\vec{v}_n\}_{n=0}^{N} = -\{\vec{\rho}^{**}_{n-1} + E'_{n-1}\vec{\rho}_{n-2} + E'_{n-1}E'_{n-2}\vec{\rho}^{**}_{n-3} + \ldots + \prod_{n-1}^{1} E'_\nu \vec{\rho}^{**}_0\}_0^N$$

voorstelt.

Om een indruk te krijgen wat in het algemeen de orde van grootte is van $||A'_H(G_H)||$, zullen we een bovengrens afleiden voor de *maximum-norm* van $A'_H(G_H)$. De maximum-norm van een matrix $A = (a_{i,j})$ wordt aangegeven met

$\| \ \|_\infty$ en gedefinieerd volgens

$$\|A\|_\infty = \max_i \sum_j |a_{i,j}| .$$

Stel dat (vergelijk de voorgaande paragraaf)

(2.6.7) $$\|E'_n\|_\infty \leq L = 1 + Ch,$$

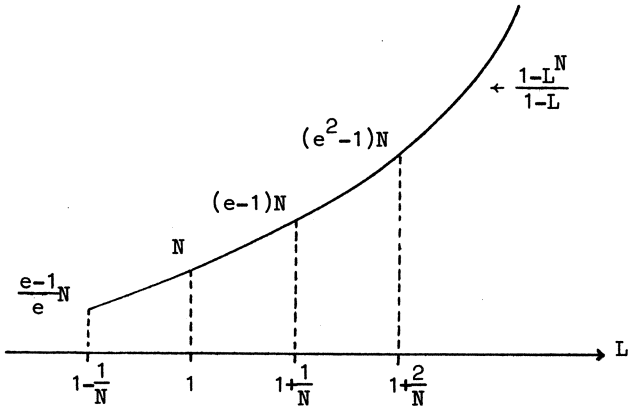
dan is eenvoudig in te zien dat

$$\begin{aligned} \|A'_H(G_H)\|_\infty &\leq \max_n [\| \prod_n^0 E'_\nu \|_\infty + \| \prod_n^1 E'_\nu \|_\infty + \ldots + \|E'_n\|_\infty + 1] \\ &\leq \max_n [1 + L + L^2 + \ldots + L^n] \\ &= \max_n \frac{1-L^{n+1}}{1-L} \\ &\leq \frac{1-L^N}{1-L} = \frac{1-(1+Ch)^N}{-Ch} . \end{aligned}$$

Voor $h \to 0$ geldt $h = O(N^{-1})$, zodat

$$\|A'_H(G_H)\|_\infty \leq O(N) = O(h^{-1}) .$$

Met andere woorden, voorwaarde (2.6.7) garandeert *stabiliteit in de zin van Rjabenki en Filippov* (zie paragraaf 1.6.4). Merk op dat de *orde* van grootte van de bovengrens voor $\|A'_H(G_H)\|_\infty$ steeds h^{-1} is als $h \to 0$, of L nu kleiner of groter dan 1 is ($C<0$ respectievelijk $C>0$). De *waarde* van deze bovengrens voor gegeven h en N, hangt echter wel degelijk van L af en neemt "bijna" exponentieel toe als L groter wordt (zie figuur (2.6.1). In de practijk zal men er dan ook meestal naar streven integratieformules te ontwerpen waarin $\|E'_n\|_\infty$, of algemener de een of andere norm van E'_n, kleiner dan of hoogstens gelijk 1 is. Hierbij moet wel opgemerkt worden dat E'_n met het rechterlid van de differentiaalvergelijking samenhangt zodat in het geval van een instabiele differentiaalvergelijking (*inherente instabiliteit*), een stabiel differentieschema een niet realistische eis is.



Figuur 2.6.1 Sup $\|A'_H(G_H)\|_\infty$ als functie van L voor gegeven N >> 1

Bovenstaande beschouwing suggereert de volgende definitie van stabiliteit.

Definitie 2.6.1

Een eenstapsmethode wordt stabiel genoemd in het punt $\vec{y}_n$, wanneer

$$\|E'_n(\vec{y}_n)\| < 1 .$$

In de literatuur wordt ook wel gesproken van *zwakke* en *sterke* stabiliteit en zoals er vele definities van stabiliteit zijn, zo komen de begrippen zwakke en sterke stabiliteit met verschillende betekenissen voor. In combinatie met bovenstaande stabiliteitsdefinitie is het gebruikelijk onder zwakke stabiliteit de voorwaarde

$$\|E'_n(\vec{y}_n)\| \leq 1$$

te verstaan en sterke stabiliteit met definitie 2.6.1 te identificeren.

Tot slot nog een opmerking over de verstoringen $\vec{\rho}^*_n$ in (2.6.3) waarmee de numerieke oplossing $\vec{y}^*_n$ gedefinieerd werd. Stel dat men weet dat bepaalde vectoren uit de een of andere basis voor $\mathbb{R}_r$ niet voorkomen in de analytische oplossing $\vec{y} = \vec{y}(x)$, dan zou men deze componenten ook niet in de

numerieke oplossing $\vec{y}_n^{**}$ willen laten doordringen; de verstoringen $\vec{\rho}_n^{*}$ zijn in principe echter willekeurig en zullen deze componenten wel degelijk bevatten, zodat ze via (2.6.3) in de $\vec{y}_n^{**}$ terecht komen. Het effect hiervan wordt beschreven door de operator $A'_H(G_H)$. Uit (2.6.6) is af te leiden dat dit effect beperkt kan worden door er voor te zorgen dat de operatoren

$$\prod_{n}^{m} E'_\nu \quad ; \quad m \leq n \quad ; \quad m,n = 0,1,\ldots,N-1$$

een dempende werking hebben op de ongewenste componenten. Op dit aspect komen we nog terug in paragraaf 2.9.

2.6.2 De stabiliteitsfunctie

Het is duidelijk dat in het algemeen de afgeleide (Jacobiaan) van de operator E_n moeilijk te vinden is of erg veel rekenwerk kost wanneer we deze numeriek berekenen. Daarom stelt men zich in de practijk tevreden met een benadering van E'_n. Uitgangspunt is hierbij de *lokale linearisering* van de differentiaalvergelijking, dat wil zeggen men benadert de rechterlidfunctie $\vec{f}$ door een eerste orde Taylor-ontwikkeling in het punt $\vec{y}_n$, dus

$$\frac{d\vec{y}}{dx} = J(\vec{y}_n)\vec{y} + [f(\vec{y}_n) - J(\vec{y}_n)\vec{y}_n] \; . \tag{2.6.8}$$

(In een gegeven omgeving van $\vec{y}_n$ is deze benadering van de differentiaalvergelijking beter naarmate J minder hard varieert met $\vec{y}$.) Vergelijking (2.6.8) kan geschreven worden als

$$\frac{d\vec{v}}{dx} = J(\vec{y}_n)\vec{v} \; , \tag{2.6.8'}$$

waarin

$$\vec{v} = \vec{y} - \vec{y}_n + J^{-1}(\vec{y}_n)\vec{f}(\vec{y}_n) \; .$$

Passen we nu op (2.6.8') een van de eenstapsmethoden toe uit de in dit hoofdstuk gedefinieerde hoofdklassen, dan vinden we de relatie

$$\vec{v}_{n+1} = R(h_n J(\vec{y}_n))\vec{v}_n \; , \tag{2.6.9'}$$

of uitgedrukt in $\vec{y}_n$ en $\vec{y}_{n+1}$

(2.6.9) $$\vec{y}_{n+1} = \vec{y}_n + [R(h_n J(\vec{y}_n))-I]J^{-1}(\vec{y}_n)f(\vec{y}_n) \ .$$

Hierin is R een polynoom (wanneer de eenstapsmethode expliciet is) of een rationale functie (wanneer de eenstapsmethode impliciet is), waarin de coefficienten uitsluitend afhangen van de parameters van de toegepaste integratiemethode.

Voorbeelden 2.6.1

Laten we eerst de *algemene Taylor-methode* (2.1.4') toepassen op vergelijking (2.6.8'). Aangezien

$$\vec{v}' = J(\vec{y}_n)\vec{v} \quad , \quad \vec{v}'' = J^2(\vec{y}_n)\vec{v} \quad , \quad \ldots$$

kan (2.1.4') geschreven worden als

$$[\alpha_0+\alpha_1 h_n J(\vec{y}_n)+\alpha_2 h_n^2 J^2(\vec{y}_n)+\ldots]\vec{v}_{n+1} =$$

$$= [\beta_0+\beta_1 h_n J(\vec{y}_n)+\beta_2 h_n^2 J^2(\vec{y}_n)+\ldots]\vec{v}_n \ ,$$

ofwel

$$\vec{v}_{n+1} = R(h_n J(\vec{y}_n))\vec{v}_n \ ,$$

met

(2.6.10) $$R(z) = \frac{\beta_0+\beta_1 z+\beta_2 z^2+\ldots+\beta_{m_2} z^{m_2}}{\alpha_0+\alpha_1 z+\alpha_2 z^2+\ldots+\alpha_{m_1} z^{m_1}} \ .$$

Vervolgens passen we de *2-punts Runge-Kuttaformule* (2.4.4) toe op (2.6.8'). We vinden

$$\vec{v}_{n+1} = \vec{v}_n + \theta_0 h_n J(\vec{y}_n)\vec{v}_n + \theta_1 h_n J(\vec{y}_n)\,[\vec{v}_n+\lambda h_n J(\vec{y}_n)\vec{v}_n] \ .$$

Voor (2.4.4) wordt de functie R blijkbaar gegeven door het polynoom

(2.6.11') $$R(z) = 1 + (\theta_0+\theta_1)z + \theta_1\lambda z^2 \ .$$

Voor de *algemene m-punts Runge-Kuttaformule* zijn de coefficienten in het polynoom betrekkelijk ingewikkelde uitdrukkingen van de Runge-Kuttaparameters $\lambda_{j,l}$. Schrijven we formeel R als

$$R(z) = 1 + \beta_1 z + \beta_2 z^2 + \ldots + \beta_m z^m \quad , \tag{2.6.11}$$

dan kan aangetoond worden dat β_1, β_2, β_3 en β_4 juist de in (2.4.9) gedefinieerde functies van de $\lambda_{j,l}$ zijn. Voor de verdere coefficienten β_j kunnen dergelijke uitdrukkingen afgeleid worden, maar aangezien we ons later zullen beperken tot een deelklasse van de algemene klasse van Runge-Kuttaformules, stellen we de berekening van deze coefficienten nog even uit.

Tenslotte beschouwen we de toepassing van de *gegeneraliseerde 2-puntsformule* (2.4.11) op vergelijking (2.6.11'). We vinden

$$R(z) = 1 + [\theta_0(z)+\theta_1(z)]z + \theta_1(z)\Lambda(z)z^2 \quad . \tag{2.6.12}$$

Het belang van de functie R is gelegen in het feit dat voor Jacobianen $J(\vec{y})$ die zeer langzaam met $\vec{y}$ varieren -het geval dus waarvoor (2.6.8) de differentiaalvergelijking benadert- de operator $E_n'(\vec{y}_n)$ benaderd wordt door $R(h_n J(\vec{y}_n))$, dus

$$E_n'(y_n) \simeq R(h_n J(\vec{y}_n)). \tag{2.6.13}$$

Dit volgt direct door differentiatie van het rechterlid van (2.6.9) naar $\vec{y}_n$ waarbij dan $J(\vec{y}_n)$ als een constante matrix beschouwd wordt. De functie R zullen we de *stabiliteitsfunctie* van de eenstapsmethode noemen. Deze functie beschrijft voor lineaire differentiaalvergelijkingen exact de stabiliteit van het integratieproces. Voor niet-lineaire vergelijkingen met sterk varierende Jacobianen geldt (2.6.13) alleen voor $h_n \to 0$; R beschrijft dan als het ware *lokaal* de stabiliteit van het proces.

Definitie 2.6.1'

Een eenstapsmethode wordt lokaal stabiel genoemd in het punt $\vec{y}_n$, wanneer

$$\|R(h_n J(y_n))\| < 1 \, . \tag{2.6.14}$$

Stelling 2.6.1

De stabiliteitsfunctie van een p^e orde consistente integratieformule voldoet aan de relaties

$$(2.6.15) \qquad \frac{d^j}{dh_n^j} R(h_n J) = J^j \quad \text{voor } h_n \to 0 \quad \text{en} \quad j = 0,1,\ldots,p.$$

Bewijs

Passen we een p^e orde formule toe op de modelvergelijking

$$\frac{d\vec{y}}{dx} = J\vec{y} \ ,$$

waarin J een matrix is met constante matrixelementen, dan geldt

$$\vec{y}_{n+1} = R(h_n J)\vec{y}_n \ .$$

Ontwikkelen we $R(h_n J)$ in machten van h_n, dan vinden we voor $\vec{y}_{n+1}$ de Taylor-ontwikkeling

$$\vec{y}_{n+1} = \left[\sum_{j=0}^{\infty} \frac{1}{j!} \frac{d^j}{dh_n^j} R(h_n J) \Big|_{h_n=0} h_n^j \right] \vec{y}_n \ .$$

Volgens de p^e orde consistentie geldt echter ook

$$\vec{y}_{n+1} = \left[\sum_{j=0}^{p} \frac{1}{j!} J^j h_n^j \right] \vec{y}_n + O(h_n^{p+1}) \ .$$

Coefficienten-vergelijking van deze twee ontwikkelingen leidt tot (2.6.15).

Definitie 2.6.2

Een stabiliteitsfunctie wordt p^e orde consistent genoemd wanneer aan de voorwaarden (2.6.15) voldaan is.

In het algemeen zal een integratieformule waarvan de stabiliteits-functie p^e orde consistent is niet noodzakelijk zelf consistent van de orde p zijn. Bijvoorbeeld een 4-punts Runge-Kuttaformule heeft volgens (2.6.11) een stabiliteitsfunctie van de vorm

$$R(z) = 1 + \beta_1 z + \beta_2 z^2 + \beta_3 z^3 + \beta_4 z^4 \ .$$

Kiezen we de Runge-Kuttaparameters $\lambda_{j,1}$ zodanig dat $\beta_j = 1/j!$ voor $j = 1,2,3$ en 4 dan is $R(z)$ 4^e orde consistent, maar de formule zelf is dat in het algemeen niet omdat volgens tabel 2.4.2 slechts 4 van de 8 consistentievoorwaarden vervuld zijn. De volgende stelling is eenvoudig te bewijzen voor de eenstapsmethoden die in dit hoofdstuk gedefinieerd zijn.

Stelling 2.6.2

(a) Wanneer een Taylor-formule een p^e orde stabiliteitsfunctie heeft dan is de formule ook p^e orde consistent.

(b) Wanneer een eenstapsformule een 2^e of hogere orde consistente stabiliteitsfunctie heeft dan is de formule zelf minstens 2^e orde consistent.

(c) Wanneer een eenstapsformule een p^e orde consistente stabiliteitsfunctie heeft dan is de formule p^e orde consistent voor alle lineaire differentiaalvergelijkingen.

Opgaven 2.6.2

(1) Leid de stabiliteitsvoorwaarde af(zowel volgens definitie 2.6.1 als 2.6.1') voor de volgende methoden en differentiaalvergelijkingen

verbeterde Euler-formule	toegepast op	$y' = y^2$
2^e orde (0,2)-Taylorformule	" "	$y' = -1/y^2$
verbeterde Euler-formule	" "	$y' = -1/y^2$

(2) Bewijs stelling 2.6.2 voor de drie hoofdklassen van eenstapsformules uit dit hoofdstuk.

(3) Stel de stabiliteitsfunctie van formule (2.2.9) op voor $m = 2$ en $m = 3$.

(4) Welke stabiliteitsfunctie heeft een oneindig grote orde van consistentie?

2.6.3 Stabiliteitsgebieden

Uit definitie 2.6.1' volgt dat een noodzakelijke voorwaarde voor lokale stabiliteit is:

(2.6.16) $\quad |R(z)| < 1 \quad$ voor $\quad z \in h_n\Delta_n$,

waarin Δ_n het eigenwaardespectrum van de Jacobiaan $J(\vec{y}_n)$ voorstelt. De punten in het complexe z-vlak waar R in modulus kleiner dan 1 is, wordt het *stabiliteitsgebied* S genoemd dus

(2.6.17) $\quad S = \{z \mid |R(z)| < 1\}$.

Het stabiliteitsgebied hangt uitsluitend van de *integratieformule* af en *niet* van de *differentiaalvergelijking*. Omgekeerd zal men echter voor een gegeven differentiaalvergelijking een integratieformule wensen waarvan het stabiliteitsgebied S voor realistische waarden van de integratiestap h_n de verzameling punten $h_n\Delta_n$ nog bevat. Bijvoorbeeld een partieel gediscretiseerde parabolische differentiaalvergelijking (zie paragraaf 1.3.2) heeft zeer grote en zeer kleine negatieve eigenwaarden; voor integratistappen h_n die voldoende klein zijn om de gevraagde nauwkeurigheid te halen kan de verzameling $h_n\Delta_n$ toch nog een zeer groot interval van de negatieve as beslaan. Dit betekent dat alleen integratieformules in aanmerking komen waarvan het stabiliteitsgebied S een flink stuk van de negatieve as bevat. Evenzo zal men voor partieel gediscretiseerde hyperbolische differentiaalvergelijkingen wensen dat S een flink stuk van de imaginaire as bevat, enz. Men definieert wel de *reële stabiliteitsgrens* β_{real} en de *imaginaire stabiliteitsgrens* β_{imag} als de lengte van respectievelijk de intervallen $(-\beta_{real},0)$ en $(i\beta_{imag},0)$ die nog juist binnen S liggen. Indien de Jacobiaan negatieve respectievelijk imaginaire eigenwaarden heeft dan is een nodige voorwaarde voor lokale stabiliteit

(2.6.18) $\quad h_n < \dfrac{\beta}{\sigma(J(\vec{y}_n))}$,

waarin $\sigma(J(\vec{y}_n))$ de spectraalradius van de Jacobiaan voorstelt en β respectievelijk de reële en imaginaire stabiliteitsgrens is.

Uit het voorgaande volgt dat we in de practijk de stabiliteitsfunctie R zelf zouden willen kiezen afhankelijk van de te integreren differentiaalvergelijking. Deze eis zullen we de *adaptiviteitsvoorwaarde* noemen. Hierbij zij wel opgemerkt dat de gekozen stabiliteitsfunctie aan (2.6.15) moet voldoen om niet met de consistentievoorwaarden in conflict te komen.

Opgaven 2.6.3

(1) Bewijs dat (2.6.16) een *voldoende* voorwaarde voor lokale stabiliteit t.o.v. de spectrale norm is indien $J(\vec{y}_n)$ een *normale* matrix is.

(2) Bepaal de lengte van het reële en imaginaire interval dat nog juist ligt binnen het stabiliteitsgebied van de functie

$$R(r) = 1 + z + \frac{1}{2}z^2 + \ldots + \frac{1}{p!}z^p \quad \text{voor} \quad p = 1,2,3 \text{ en } 4 .$$

2.7 CONSTRUCTIE VAN INTEGRATIEFORMULES MET ADAPTIEVE STABILITEITSFUNCTIE EN TOEPASSINGEN

We zijn nu zover dat we de consistentievoorwaarden voor drie hoofdklassen van eenstapsmethoden uitgedrukt hebben in de parameters van deze methoden en dat we de stabiliteit door middel van de adaptiviteitsvoorwaarde enigszins kunnen controleren. Rest ons de constructie van integratieformules uit deze drie hoofdklassen die aan de consistentie- en adaptiviteitsvoorwaarden voldoen. Bij de Taylor- en Runge-Kuttamethoden gaan we als volgt te werk: we leiden een formule af van de gewenste orde van consistentie waarin nog een voldoend aantal vrije parameters zitten, vervolgens stellen we de stabiliteitsfunctie van deze formule op en identificeren deze met de gegeven stabiliteitsfunctie R. Op deze manier trachten we de vrije parameters uit te drukken in de coefficienten van de gegeven functie R. Deze functie zullen we steeds schrijven in de vorm

$$(2.7.1) \qquad R(r) = \frac{1 + \beta_1 z + \beta_2 z^2 + \ldots + \beta_{m_2} z^{m_2}}{1 + \alpha_1 z + \alpha_2 z^2 + \ldots + \alpha_{m_1} z^{m_1}} .$$

In het geval van de gegeneraliseerde Runge-Kuttamethoden is het handiger één van de coefficientfuncties in de stabiliteitsfunctie R en de andere coefficientfuncties uit te drukken en daarna de vrije parameters te gebruiken om aan de consistentievoorwaarden te voldoen.

2.7.1 Expliciete Taylor-methoden, de procedure *modified taylor*

Bij Taylor-formules waar een éénduidig verband tussen stabiliteitsfunctie en integratieformule bestaat (stelling 2.6.2), genereert een gegeven p^e orde consistent stabiliteitspolynoom direct de p^e orde Taylor-formule

$$(2.7.2) \qquad \vec{y}_{n+1} = R(h_n \frac{d}{dx})\vec{z}(x_n,\vec{y}_n;x)\Big|_{x=x_n} .$$

De procedure *modified taylor*

Een ALGOL 60-versie van formule (2.7.2) waarin R een vrij te kiezen polynoom is, is aanwezig in de bibliotheek NUMAL (Numerieke procedures in ALGOL 60); deze bibliotheek is beschikbaar bij het Academisch Rekencentrum Amsterdam (SARA). De Taylor-algorithme is geimplementeerd onder de naam *modified taylor* en gedocumenteerd in section 5.2.1.1.1.3. van de NUMAL-manual. Een listing van de gebruiksaanwijzingen voor *modified taylor* volgt aan het eind van deze paragraaf. We zullen hier nog enige toelichting geven.

In de eerste plaats benadrukken we nog eens dat Taylor-methoden en dus de procedure *modified taylor* gebaseerd is op *herhaalde differentiatie* van het rechterlid van de differentiaalvergelijking. Dus in eerste instantie komt de procedure alleen in aanmerking wanneer de rechterlidfunctie eenvoudig te differentieren is. Toch kan *modified taylor* ook gebruikt worden wanneer alleen de Jacobiaan beschikbaar is, zij het dat de orde van consistentie dan niet groter dan 2 is. We zullen hier nog op terugkomen bij de constructie van gegeneraliseerde Runge-Kuttamethoden (paragraaf 2.7.8).

In de tweede plaats is de procedure ook direct toepasbaar op niet-autonome stelsels zodat de introductie van een additionele afhankelijke variabele (zie paragraaf 1.2.1) niet nodig is. Verder merken we op dat de gebruiksaanwijzing van *modified taylor* refereert naar de vergelijking $d\vec{u}/dt = H(\vec{u},t)$ in plaats van naar $d\vec{y}/dx = f(x,\vec{y})$.

Tenslotte de parameters van de procedure. De parameters *t*, *te*, *mo*, *m*, *u* en *sigma* spreken voor zich. De parameter *taumin* is opgenomen om de stapkeuzestrategie onder controle te kunnen houden: de procedure kiest zelf zijn integratiestappen op grond van de opgegeven absolute en relatieve toleranties (*aeta* en *reta*) voor de lokale discretiseringsfout $\vec{\rho}_n$; om nu te voorkomen dat de stappen onrealistisch klein worden kan de gebruiker via *taumin* een benedengrens opgeven. Deze benedengrens mag echter ook weer niet in conflict komen met de stabiliteitsvoorwaarde die een maximale stap (vergelijk (2.6.18))

$$taust = \frac{data\ [0]}{sigma}$$

voorschrijft (*data* [*0*] is de stabiliteitsgrens van het gebruikte stabiliteitspolynoom); het stapkeuzemechanisme levert dan ook altijd een stap *tau* af die voldoet aan

$$\min(taust, taumin) \leq tau \leq taust \ .$$

Wil men met constante stappen h integreren ongeacht de nauwkeurigheid en de stabiliteit, dan kieze men de parameters zodanig dat

$$taumin = h \quad , \quad h\ sigma = data\ [0] \ .$$

De parameter *derivative* wordt voldoende geillustreerd door de procedure *der* in "example of use" van de gebruiksaanwijzing. Het array *data* karakteriseert het gebruikte stabiliteitspolynoom en vormt dus het adaptive element in de procedure. In tabel 2.7.1 zijn een aantal polynomen gedefinieerd samen met de klassen van differentiaalvergelijkingen waarvoor ze geschikt zijn. Voor een completer overzicht van stabiliteitspolynomen verwijzen we naar van der Houwen [1972]. De parameter *alpha* heeft evenals *taumin* de functie om het stapkeuzemechanisme onder controle te houden. Er geldt altijd dat de stap niet groter is dan *alpha* maal de voorgaande stap. Tenslotte de parameters *eta* en *rho*: *eta* wordt gedefinieerd door

$$eta = aeta + reta * \|\vec{u}\| \ ,$$

waarin $\| \ \|$ door de parameter *norm* wordt gedefinieerd. Het stapkeuzemechanisme berekent nu een staplengte zodanig dat de hierbij behorende lokale discretiseringsfout $\vec{\rho}$ in norm ongeveer gelijk is aan *eta*. In hoeverre het mechanisme hierin slaagt kan tijdens het rekenproces nagegaan worden door *eta* en *rho* door middel van *out* op te vragen. Er worden geen stappen verworpen om geheugenruimte te sparen. Voor details van het stapkeuzemechanisme verwijzen we naar paragraaf 2.8.

Tabel 2.7.1 Stabiliteitspolynomen en bijbehorende klassen van differentiaalvergelijkingen

A : Δ_n bestaat uit een strook langs de negatieve as (parabolische differentiaalvergelijkingen)

B : Δ_n bestaat uit een strook langs de imaginaire as (hyperbolische vergelijkingen)

C : Δ_n onbekend

Klasse	*data[-2]*	*data[-1]*	*data[0]*	*data[1]*	*data[2]*	*data[3]*	*data[4]*
	m	p	β	β_1	β_2	β_3	β_4
A	1	1	2	1			
A	2	1	8	1	$^1/_8$		
B,C	2	1	1	1	1		
A	2	2	2	1	$^1/_2$		
A	3	1	18	1	$^4/_{27}$	$^4/_{729}$	
A	3	2	6.26	1	$^1/_2$	$^1/_{16}$	
B	3	2	2	1	$^1/_2$	$^1/_4$	
A,C	3	3	2.51	1	$^1/_2$	$^1/_6$	
A	4	1	32	1	$^5/_{32}$	$^1/_{128}$	$^1/_{8192}$
A	4	2	12	1	$^1/_2$	(-7)780845	(-7)36085
A	4	3	6	1	$^1/_2$	$^1/_6$	(-7)184557
A,B,C	4	4	$2\sqrt{2}$	1	$^1/_2$	$^1/_6$	$^1/_{24}$

In de volgende paragrafen zullen we een tweetal toepassingen op partiele differentiaalvergelijkingen bespreken. Een toepassing op een enkele gewone differentiaalvergelijking vindt men in de gebruiksaanwijzing (onder "example of use") van de procedure *modified taylor*. Deze toepassingen zijn ontleend aan de MC-publicaties TW 130/71 en TN 58/70.

Deze paragraaf wordt besloten met de gebruiksaanwijzingen voor de procedure *modified taylor*.

AUTHORS: P.J. VAN DER HOUWEN AND P.A.BEENTJES.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 730616.

BRIEF DESCRIPTION:

MODIFIED TAYLOR SOLVES AN INITIAL (BOUNDARY) VALUE PROBLEM, GIVEN AS A SYSTEM OF FIRST ORDER DIFFERENTIAL EQUATIONS , BY MEANS OF A ONE-STEP TAYLOR-METHOD.
IN PARTICULAR THIS METHOD IS SUITABLE FOR THE INTEGRATION OF LARGE SYSTEMS ARISING FROM PARTIAL DIFFERENTIAL EQUATIONS , PROVIDED THAT HIGHER ORDER DERIVATIVES CAN BE EASILY OBTAINED.

KEYWORDS:

DIFFERENTIAL EQUATIONS,
INITIAL (BOUNDARY) VALUE PROBLEMS,
ONE-STEP TAYLOR-METHOD.

```
CALLING SEQUENCE:

    THE HEADING OF THE PROCEDURE READS:
    "PROCEDURE" MODIFIED TAYLOR (T,TE,M0,M,U,SIGMA,TAUMIN,I,DERIVATIVE,
                                 K,DATA,ALFA,NORM,AETA,RETA,ETA,RHO,OUT);
    "INTEGER" M0,M,I,K,NORM;
    "REAL" T,TE,SIGMA,TAUMIN,ALFA,AETA,RETA,RHO;
    "ARRAY" U,DATA;
    "PROCEDURE" DERIVATIVE,OUT;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    T:       <VARIABLE>;
             THE INDEPENDENT VARIABLE T;
             MAY BE USED IN DERIVATIVE, SIGMA ETC.;
             ENTRY: THE INITIAL VALUE T0;
             EXIT : THE FINAL VALUE TE;
    TE:      <ARITHMETIC EXPRESSION>;
             THE FINAL VALUE OF T (TE >= T);
    M0,M:    <ARITHMETIC EXPRESSION>;
             INDICES OF THE FIRST AND LAST EQUATION OF THE SYSTEM TO  BE
             SOLVED;
    U:       <ARRAY IDENTIFIER>;
             "ARRAY" U[M0:M];
             THE DEPENDENT VARIABLE;
             ENTRY: THE INITIAL VALUES OF THE SOLUTION OF THE SYSTEM  OF
                    DIFFERENTIAL EQUATIONS AT T = T0;
             EXIT : THE VALUES OF THE SOLUTION AT T = TE;
    SIGMA:   <ARITHMETIC EXPRESSION>;
             THE SPECTRAL RADIUS OF THE JACOBIAN MATRIX WITH RESPECT
             TO THOSE EIGENVALUES WHICH ARE LOCATED IN THE LEFT
             HALFPLANE;
             IF  SIGMA  TENDS  TO  INFINITY , PROCEDURE  MODIFIED TAYLOR
             TERMINATES;
    TAUMIN:  <ARITHMATIC EXPRESSION>;
             MINIMAL STEP LENGTH BY WHICH THE INTEGRATION IS  PERFORMED;
             IF  TAUMIN EXCEEDS  THE STEPLENGTH TAUST = DATA[0] / SIGMA,
             PRESCRIBED BY STABILITY CONSIDERATIONS,THEN TAUMIN:= TAUST;
    I:       <VARIABLE>;
             A JENSEN PARAMETER FOR PROCEDURE DERIVATIVE;
             MAY BE USED IN M0 AND M;
    DERIVATIVE: <PROCEDURE IDENTIFIER>;
             THE HEADING OF THIS PROCEDURE READS:
             "PROCEDURE" DERIVATIVE(I,A); "INTEGER" I; "ARRAY" A;
             WHEN  THIS  PROCEDURE  IS  CALLED , ARRAY  A  CONTAINS  THE
             COMPONENTS OF THE (I-1)-ST DERIVATIVE OF U  AT THE POINT T;
             UPON COMPLETION OF DERIVATIVE, ARRAY A SHOULD  CONTAIN  THE
             COMPONENTS  OF THE I-TH  DERIVATIVE  OF U AT  THE POINT  T;
    K:       <VARIABLE>;
             INDICATES  THE  NUMBER  OF  INTEGRATION  STEPS  PERFORMED;
             ENTRY: K = 0;
```

```
DATA:   <ARRAY IDENTIFIER>;
        "ARRAY" DATA[-2 : DATA[-2]];
        ENTRY;
        DATA[-2]: THE  ORDER OF THE  HIGHEST DERIVATIVE  UPON WHICH
                  THE TAYLOR METHOD IS BASED;
        DATA[-1]: ORDER OF ACCURACY OF THE METHOD;
        DATA[0] : STABILITY PARAMETER;
        DATA[1] , ... , DATA[DATA[-2]] : POLYNOMIAL  COEFFICIENTS;
        FOR FURTHER EXPLANATION AND POSSIBLE VALUES OF THE ELEMENTS
        OF ARRAY DATA SEE REFERENCES [2] AND [3];
ALFA:   <ARITHMETIC EXPRESSION>;
        GROWTH FACTOR FOR THE INTEGRATION STEP LENGTH;
NORM:   <ARITHMETIC EXPRESSION>;
        IF NORM = 1 DISCREPANCY AND TOLERANCE ARE ESTIMATED IN  THE
        MAXIMUM NORM, OTHERWISE IN THE EUCLIDIAN NORM;
AETA,RETA: <ARITHMETIC EXPRESSION>;
        DESIRED ABSOLUTE AND RELATIVE ACCURACY;
        IF BOTH  AETA AND RETA  ARE NEGATIVE , ACCURACY  CONDITIONS
        WILL BE IGNORED;
ETA,RHO: <VARIABLE>;
        COMPUTED TOLERANCE AND DISCREPANCY;
OUT:    <PROCEDURE IDENTIFIER>;
        THE  HEADING  OF  THIS  PROCEDURE  READS : "PROCEDURE" OUT;
        THROUGH  THIS  PROCEDURE  THE VALUES AFTER EACH INTEGRATION
        STEP  OF  FOR INSTANCE  T, U, ETA AND RHO  ARE  ACCESSIBLE.
```

DATA AND RESULTS:

FOR FURTHER EXPLANATION OF THE PARAMETERS AETA, RETA, ETA, RHO, M0, M AND THE ARRAY DATA SEE REFERENCES [2] AND [3].
AS FOR THE INDICES M0 AND M THE FOLLOWING MAY BE REMARKED: WHEN THE METHOD OF LINES IS APPLIED TO HYPERBOLIC DIFFERENTIAL EQUATIONS THE NUMBER OF RELEVANT ORDINARY DIFFERENTIAL EQUATIONS DECREASES DURING THE INTEGRATION PROCESS.
IN PROCEDURE MODIFIED TAYLOR , THIS MAY BE REALIZED BY INTEGER PROCEDURES M0 AND M WHICH ARE DEFINED AS FUNCTIONS OF I, K AND DATA[-2].

PROCEDURES USED: VECVEC = CP34010.

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: CIRCA 75 + M - M0.

RUNNING TIME:

DEPENDS STRONGLY ON THE DIFFERENTIAL EQUATION TO BE SOLVED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE: SEE REFERENCES.

REFERENCES:

[1].P.J. VAN DER HOUWEN.
ONE-STEP METHODS FOR LINEAR INITIAL VALUE PROBLEMS I,
POLYNOMIAL METHODS,TW REPORT 119,
MATHEMATICAL CENTRE, AMSTERDAM (1970).

[2].P.J. VAN DER HOUWEN, P.BEENTJES, K.DEKKER AND E.SLAGT
ONE-STEP METHODS FOR LINEAR INITIAL VALUE PROBLEMS III,
NUMERICAL EXAMPLES, TW REPORT 130/71,
MATHEMATICAL CENTRE, AMSTERDAM (1971).

[3].P.J. VAN DER HOUWEN, J.KOK.
NUMERICAL SOLUTION OF A MINIMAX PROBLEM, TW REPORT 123/71,
MATHEMATICAL CENTRE, AMSTERDAM (1971).

EXAMPLE OF USE:

THE SOLUTION AT T=EXP(1) AND T=EXP(2) OF THE DIFFERENTIAL EQUATION
DU/DT=-EXP(T)*(U-LN(T)) + 1/T WITH INITIAL CONDITION U(.01)=LN(.01)
AND ANALYTICAL SOLUTION U(T) = LN(T), MAY BE OBTAINED AS FOLLOWS:

```
"BEGIN" "INTEGER" I,K;"REAL" T,TE,ETA,RHO,EXPT,LNT,C0,C1,C2,C3;
    "ARRAY" U[0:0],DATA[-2:4];
    "PROCEDURE" OP;"IF" T=TE "THEN"
    OUTPUT(61,"("("("NUMBER OF STEPS:")",3ZD,/,
                  "("SOLUTION:  T= ")",+D.5D,
                  "("    U(T) = ")",+D.7D,//")",K,T,U[0]);
    "PROCEDURE" DER(I,A);"INTEGER" I;"ARRAY" A;
    "BEGIN" "IF" I=1 "THEN"
        "BEGIN" EXPT:=EXP(T);LNT:=LN(T);C0:=A[0];
            C1:=A[0]:=-EXPT*C0+1/T+EXPT*LNT
        "END";
        "IF" I=2 "THEN" C2:=A[0]:=EXPT*(LNT+1/T-C0-C1)-1/T/T;
        "IF" I=3 "THEN" C3:=A[0]:=
        EXPT*(LNT+2/T-C0-2*C1-C2-1/T/T)+2/T/T/T;
        "IF" I=4 "THEN" A[0]:=C3-2*(1+3/T)/T/T/T+
        EXPT*((1-(2-2/T)/T)/T-C1-C2*2-C3)
    "END";
    "PROCEDURE" MODIFIED TAYLOR(T,TE,M0,M,U,SIGMA,TAUMIN,I,
                DERIVATIVE,K,DATA,ALFA,NORM,AETA,RETA,ETA,RHO,OUT);
    "CODE" 33040;
    I:=-2;"FOR" T:=4,3,6.025,1,.5,1/6,.018455702 "DO"
    "BEGIN" DATA[I]:=T;I:=I+1 "END";
    T:=U[0]:="-2;K:=0;"FOR" TE:=EXP(1),TE*TE "DO"
    MODIFIED TAYLOR(T,TE,0,0,U,EXP(T),"-4,I,DER,K,DATA,1,5,1,"-5,
                    "-4,ETA,RHO,OP)
"END"
```

THIS PROGRAM DELIVERS:

```
NUMBER OF STEPS:  46
SOLUTION:  T= +2.71828     U(T) = +1.0000285

NUMBER OF STEPS: 424
SOLUTION:  T= +7.38906     U(T) = +1.9999967
```

2.7.2 Een lineair diffusieprobleem

Beschouw het beginwaardeprobleem

$$\frac{\partial U}{\partial x} = \frac{\partial^2 U}{\partial z^2} + \exp(-x)\,.\,(z^{10}+90z^8-z) \quad , \quad 0 \le z \le 1 \ , \quad x \ge 0 \ ,$$

$$U = 1 + z(1-z^9) \quad \text{voor} \quad x = 0 \ , \tag{2.7.3}$$

$$U = 1 \quad \text{voor} \quad z = 0,\ 1 \quad \text{en} \quad x \ge 0 \ .$$

Dit probleem kan met behulp van partiele discretisatie gereduceerd worden tot een beginwaardeprobleem voor een stelsel eerste orde, gewone differentiaalvergelijkingen waar de procedure *modified taylor* een geschikte oplossingsmethode voor is.

Volgens de beschouwingen in paragraaf 1.2.4 definieren we in het interval $0 \le z \le 1$ de punten $j\Delta z, j = 0,1,\ldots,(\Delta z)^{-1}$. We willen nu in (2.7.3) de operator $\partial^2/\partial z^2$ vervangen door de differentie-operator

$$D = a(Z_+^2+Z_-^2) + b(Z_+ + Z_-) + c \ , \tag{2.7.4}$$

waarin a, b en c nog nader te bepalen gewichten zijn en $Z_\pm$ translaties voorstellen over een afstand $\pm\,\Delta z$ langs de z-as, dus

$$Z_\pm U(x,j\Delta z) = U(x,(j\pm 1)\Delta z) \ .$$

Wanneer we in (2.7.4) stellen

$$a = 0 \quad , \quad b = (\Delta z)^{-2} \quad , \quad c = -2(\Delta z)^{-2} \ , \tag{2.7.5}$$

dan stelt DU juist het 2^e orde differentiequotient voor dat in voorbeeld 1.2.1 gebruikt werd voor de discretisering van $\partial^2/\partial z^2$. Het is eenvoudig na te gaan dat in geval van (2.7.5)

$$\frac{\partial^2}{\partial z^2} = D + O((\Delta z)^2) \ . \tag{2.7.6}$$

Nu varieren de inhomogene term en ook de beginvoorwaarde in (2.7.3) nogal sterk met z, zodat een vrij nauwkeurige benadering van de operator $\partial^2/\partial z^2$ wenselijk kan zijn. Dit is mogelijk door in (2.7.4) a gunstig te kiezen.

Daartoe ontwikkelen we de differentie-operator D in een Taylor-reeks:

$$(2.7.7)\qquad D = (2a+2b+c) + (4a+b)\,(\Delta z\frac{\partial}{\partial z})^2 + \frac{1}{12}(16a+b)\,(\Delta z\frac{\partial}{\partial z})^4 + \\ + \frac{1}{360}(64a+b)\,(\Delta z\frac{\partial}{\partial z})^6 + \ldots$$

Stellen we

$$(2a+2b+c) = 0 \quad , \quad (4a+b)(\Delta z)^2 = 1 \quad , \quad (16a+b) = 0$$

of wel

$$(2.7.8)\qquad a = -\frac{1}{12}(\Delta z)^{-2} \quad , \quad b = \frac{4}{3}(\Delta z)^{-2} \quad , \quad c = -\frac{5}{2}(\Delta z)^{-2} \quad ,$$

dan geldt

$$(2.7.9)\qquad \frac{\partial^2}{\partial z^2} = D + O((\Delta z)^4) \quad .$$

In dit geval is de operator D echter alleen gedefinieerd in de roosterpunten $j\Delta z, j = 2,3,\ldots,(\Delta z)^{-1}-2$. In het punt $z = \Delta z$ definieren we de differentie-operator

$$(2.7.10)\qquad D' = a'Z_- + b' + c'Z_+ + d'Z_+^2 + e'Z_+^3 + f'Z_+^4 \quad .$$

en iets analoogs in het punt $1 - \Delta z$.
Ontwikkeling in machten van Δz geeft

$$D' = (a'+b'+c'+d'+e'+f') + (-a'+b'+2d'+3e'+4f')\Delta z\frac{\partial}{\partial z} + \\ + \frac{1}{2}(a'+c'+4d'+9e'+16f')(\Delta z)^2\,\frac{\partial^2}{\partial z^2} + \frac{1}{6}(-a'+c'+8d'+27e'+64f')(\Delta z\frac{\partial}{\partial z})^3 + \\ + \frac{1}{24}(a'+c'+16d'+81e'+256f')(\frac{\Delta z\partial}{\partial z})^4 + \\ + \frac{1}{120}(-a'+c'+32d'+243e'+1024f')(\frac{\Delta z\partial}{\partial z})^5 + O((\Delta z)^4) \quad .$$

Stellen we

$$(2.7.8') \qquad a' = \frac{5}{6}(\Delta z)^{-2}, b' = -\frac{5}{4}(\Delta z)^{-2}, c' = -\frac{1}{3}(\Delta z)^{-2}, e' = -\frac{1}{2}(\Delta z)^{-2},$$

$$f' = \frac{1}{12}(\Delta z)^{-2},$$

dan is D' nauwkeurig op een term $O(\Delta z)^4$ na.

Met behulp van deze differentie-operatoren kan (2.7.3) geschreven worden als het stelsel

$$(2.7.11) \qquad \begin{aligned} &\frac{d\vec{y}}{dx} = J\vec{y} + \vec{g}\ , \\ &\vec{y}(0) = \left\{1+j\Delta z-(j\Delta z)^{10}\right\}_{j=1}^{(\Delta z)^{-1}-1}, \end{aligned}$$

waarin $y_j(x)$ een benadering voor $U(x,j\Delta z)$ voorstelt en $\vec{g}$ en J gegeven worden door

$$\vec{g}(x) = e^{-x}\{(\Delta z)^{10}+90(\Delta z)^{8}-\Delta z+a'e^{x},(2\Delta z)^{10}+90(2\Delta z)^{8}-2\Delta z+ae^{-x},$$

$$\ldots,(j\Delta z)^{10}+90(j\Delta z)^{8}-j\Delta z,\ldots,$$

$$,(1-2\Delta z)^{10}+90(1-2\Delta z)^{8}-1+2\Delta z+ae^{t},(1-\Delta z)^{10}+90(1-\Delta z)^{8}-1+\Delta z+a'e^{x}\}$$

en

$$J = \begin{pmatrix} b' & c' & a' & e' & f' & & & & & \\ b & c & b & a & 0 & & & & & \\ a & b & c & b & a & & & & & \\ & \ddots & & & & & \ddots & & & \\ & & & & & a & b & c & b & a \\ & & & & & 0 & a & b & c & b \\ & & & & & f' & e' & d' & c' & b' \end{pmatrix}$$

De discretisering (2.7.11) geldt ook voor het geval (2.7.5) wanneer we definieren

(2.7.5') $\quad a' = (\Delta z)^{-2},\ b = -2(\Delta z)^{-2},\ c' = (\Delta z)^{-2},\ d' = e' = f' = 0.$

De volgende stap is het selecteren van een geschikte integratieformule voor (2.7.11). Aangezien het rechterlid eenvoudig te differentieren is komt een formule uit de Taylor-klasse in aanmerking. Dus de procedure *modified taylor* lijkt inderdaad één van de aangewezen integratietechnieken voor dit probleem. Het stabiliteitspolynoom dat voor dit probleem wenselijk is, moet een relatief groot negatief stabiliteitsinterval hebben, omdat de Jacobiaan "bijna" symmetrisch is -dus "bijna" reële eigenwaarden- zodat het spectrum van J op grond van de Gerschgorin-cirkels

$$|\delta+2(\Delta z)^{-2}| \leq 2(\Delta z)^{-2}$$

(voor het geval (2.7.5)) en

$$|\delta+\tfrac{5}{2}(\Delta z)^{-2}| \leq \tfrac{17}{6}(\Delta z)^{-2}\ ,\ |\delta+\tfrac{5}{4}(\Delta z)^{-2}| \leq \tfrac{25}{12}(\Delta z)^{-2}$$

(voor het geval (2.7.8)), een groot deel van de negatieve as beslaat en wel de respectieve intervallen

(2.7.12) $\quad [-4(\Delta z)^{-2},0] \quad$ en $\quad [-\tfrac{16}{3}(\Delta z)^{-2},0]\ .$

We hebben gekozen de stabiliteitspolynomen

$$\begin{aligned}
&P_2(z) = 1 + z + \tfrac{1}{2}z^2 \quad , \quad \beta = 2\ , \\
&P_3(z) = 1 + z + \tfrac{1}{2}z^2 + \tfrac{1}{16}z^3 \quad , \quad \beta = 6.26\ , \\
&P_4(z) = 1 + z + \tfrac{1}{2}z^2 + .0780845z^3 + .00360845z^4 \quad , \quad \beta = 12\ , \\
&P_{10}(z) = T_{10}(1+\tfrac{z}{100}) \quad , \quad \beta = 200\ ,
\end{aligned} \tag{2.7.13}$$

waarin β de reële stabiliteitsgrens voorstelt.
De eerste 3 polynomen zijn tweede orde consistent, terwijl P_{10} eerste orde consistent is. De afbreekfout zal dan $O(h_n^3)$ resp. $O(h_n^2)$ zijn. We willen deze fout in *orde* gelijk maken aan de door de partiele discretisatie geintroduceerde fout. Partiele discretisatie kan geinterpreteerd worden als het vervangen van een "analytische" rechterlid door een "algebraïsche" rechterlid met een zekere onnauwkeurigheid. In dit voorbeeld is deze onnauwkeurigheid

$O((\Delta z)^2)$ voor de driepuntsformule (zie (2.7.6)) en $O((\Delta z)^4)$ voor de vijfpuntsformule (zie (2.7.9)). Past men nu de Taylor-formule toe op het gediscretiseerde systeem dan treedt het rechterlid met een factor h_n in de uitdrukking voor $\vec{y}_{n+1}$ op. Dus partiele discretisatie introduceert een fout $O(h_n(\Delta z)^2)$ respectievelijk $O(h_n(\Delta z)^4)$. Aangezien volgens (2.7.12) de stabiliteitsvoorwaarde (2.6.18) hier voor de drie- en vijfpuntsformule respectievelijk wordt,

$$(2.7.14) \qquad h_n \leq \frac{\beta}{4}(\Delta z)^2 \quad \text{en} \quad h_n \leq \frac{3\beta}{16}(\Delta z)^2 \ ,$$

zijn bovengenoemde fouten van de orde 2 respectievelijk 3 in h_n. Hieruit concluderen we dat de vijfpuntsformule met partiele discretiseringsfout $O(h_n^3)$ gekozen moet worden ingeval van 2^e orde consistente stabiliteitspolynomen. Evenzo zien we dat de driepuntsformule in combinatie met eerste orde consistente stabiliteitspolynomen gebruikt moet worden.

In tabel 2.7.2 zijn de relatieve fouten van de oplossing (in de maximum norm) weergegeven, welke met de polynomen (2.7.13) in het interval [0,.3] verkregen werden. Hierbij was de integratiestap gelijk aan de maximale stap die voldeed aan (2.7.14).
Verder is de "bewerkelijkheid" van de Taylor-formule in de tabel opgenomen door middel van de formule

$$\text{bewerkelijkheid} = \frac{Nmc}{100\Delta z} \ ,$$

waarin N het aantal integratiestappen en m de graad van het polynoom voorstelt; c neemt de waarden 1 en 2 aan voor de driepunts- respectievelijk vijfpuntsformule.

Tabel 2.7.2 Relatieve fouten en bewerkelijkheid voor Taylor-formules toegepast op (2.7.3)

ε_{rel}	P_2	P_3	P_4	P_{10}
1 $^0/_{00}$	30	15	11	80
.6 $^0/_{00}$	40	20	15	160
.2 $^0/_{00}$	67	33	24	700

2.7.3 Diffusieproblemen met discontinue beginvoorwaarden

Het volgende begin-randwaardeprobleem is afkomstig van het F O M - laboratorium:

$$\frac{\partial T}{\partial x} = (b-aT)\left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r}\right), \quad 0 \le r \le \infty, \quad x \ge 0,$$

$$T(r,0) = \begin{cases} 1 & \text{voor } 0 \le r \le 1 \\ 0 & \text{voor } 1 < r \le \infty \end{cases}, \tag{2.7.15}$$

$$\frac{\partial T}{\partial r} = 0 \quad \text{voor} \quad r = 0, \quad x \ge 0,$$

$$T = 0 \quad \text{voor} \quad r = \infty, \quad x \ge 0.$$

Hierin stellen a en b positieve constanten voor (b>1).
Gevraagd werd de temperatuur T in een rechthoek $0 \le r \le r_0$, $0 \le t \le t_0$ te berekenen met een nauwkeurigheid van 1% voor een groot aantal waarden van a en b (a<<b).

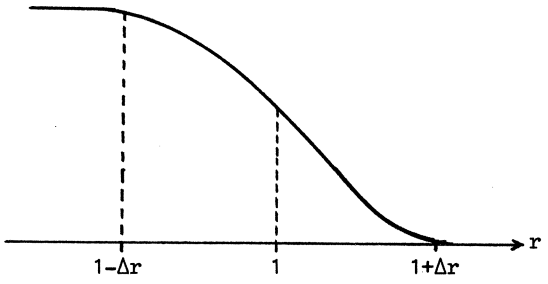
In de herleiding van dit probleem tot een "fatsoenlijk" stelsel eerste orde gewone differentiaalvergelijkingen zullen de volgende punten aan de orde komen:

(1) *Discontinue* beginvoorwaarde vervangen door *continue* beginvoorwaarde;
(2) Variabele r vervangen door variabele z zodanig dat het rechterlid niet te snel met z varieert;
(3) Partiele discretisatie, i.h.b. in het punt r = 0;
(4) Reductie van het *oneindige* stelsel tot een *eindig* stelsel;
(5) Het spectrum van de Jacobiaan;
(6) Het stabiliteitspolynoom.

(1) De beginvoorwaarden

Voor numerieke berekeningen is het wenselijk dat de beginfunctie T(r,0) continu is. Daarom benaderen we T(r,0) door (zie figuur 2.7.1)

$$T(r,0) = \begin{cases} 1 & \text{voor} \quad r \le 1 - \Delta r \\ \frac{1}{2}\left[1+\cos\left(\frac{r-1+\Delta r}{2\Delta r}\pi\right)\right] & \text{voor} \quad |r-1| \le \Delta r \\ 0 & \text{voor} \quad r \ge 1 + \Delta r \end{cases}. \tag{2.7.16}$$



Figuur 2.7.1 Beginfunctie T(r,0)

Deze benadering is des te nauwkeuriger naarmate Δr kleiner is.

(2) Transformatie z = z(r)

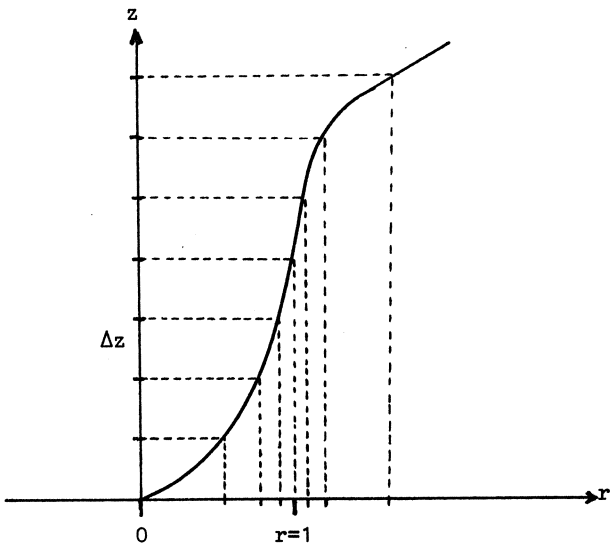
Het rechterlid van de differentiaalvergelijking zal voor kleine waarden van x snel met r varieren, althans in de buurt van r = 1. Dit betekent dat bij de partiele discretisatie de roosterpunten op de r-as in de buurt van r = 1 dichterbij elkaar gekozen moeten worden dan op enige afstand van r = 1, anders zou het stelsel differentiaalvergelijkingen óf de partiele differentiaalvergelijking te onnauwkeurig benaderen óf veel te groot ten opzichte van de partiele discretiseringsfout zijn. Een alternatief voor zo'n niet-uniform rooster is echter de transformatie van de r-as zodanig dat het hiermee getransformeerde rechterlid wel langzaam varieert met de nieuwe variabele. Dus als z = z(r), dan moet het rechterlid van de getransformeerde vergelijking

$$\frac{\partial T}{\partial x} = A(z,T)\,\frac{\partial^2 T}{\partial z^2} + B(z,T)\,\frac{\partial T}{\partial z}\ ,$$

(2.7.17)
$$A(z,T) = (b-aT)\left(\frac{dz}{dr}\right)^2\ ,$$

$$B(z,T) = (b-aT)\left(\frac{d^2z}{dr^2} + \frac{1}{r}\,\frac{dz}{dr}\right)\ ,$$

langzaam met z veranderen. De functie z = z(r) zal zich ongeveer moeten gedragen zoals aangegeven in figuur 2.7.2.



Figuur 2.7.2 Transformatie z = z(r)

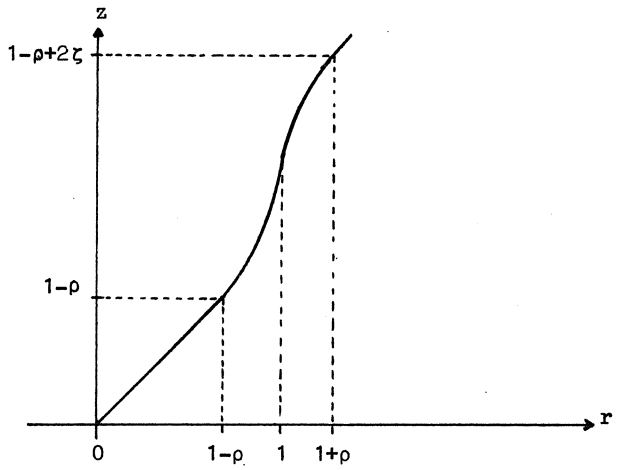
Bovendien zullen we ervoor moeten zorgen dat dz/dr en d^2z/dr^2 *continu* zijn in r, anders zouden de functies A en B uit (2.7.17) niet continu zijn. Wanneer we zo'n functie construeren, kan vergelijking (2.7.17) op efficiente wijze gediscretiseerd worden op een uniform rooster $\{j\Delta z\}_{j=0}^{\infty}$.
Voor de functie z = z(r) kiezen we

$$z = \begin{cases} r & \text{voor } r \le 1-\rho \\ r\,\frac{\zeta}{\rho} - (1-\rho)\,\frac{\zeta-\rho}{\rho} - \frac{1}{\pi}(\zeta-\rho)\,\sin\left(\frac{r-1+\rho}{\rho}\pi\right) & \text{voor } |r-1| < \rho \\ r + 2(\zeta-\rho) + 1 & \text{voor } r \ge 1+\rho \end{cases} \tag{2.7.18}$$

waarin ζ en ρ nog te kiezen parameters zijn (zie figuur 2.7.3).
De keuze van ρ en ζ wordt bepaald door de eis dat het rechterlid van (2.7.17) voldoende langzaam met z varieert in het interval $z(1-\Delta r) \le z \le z(1+\Delta r)$; we zullen eisen dat voor x = 0 de norm

$$V = \int_{z(1-\Delta r)}^{z(1+\Delta r)} \left| \frac{d}{dz}\left[\left(\frac{dz}{dr}\right)^2 \frac{\partial^2 T}{\partial z^2} + \left(\frac{d^2z}{dr^2} + \frac{1}{r}\frac{dz}{dr}\right)\frac{\partial}{\partial z} T\right]\right|^2 dz \tag{2.7.19}$$

voldoende klein is. Het is eenvoudig na te gaan dat (2.7.19) benaderd kan



Figuur 2.7.3 De functie (2.7.18) voor $\rho << \zeta$

worden door

$$(2.7.19') \qquad V \cong \int_{1-\Delta r}^{1+\Delta r} \left[\frac{\partial^3 T}{\partial r^3} + \frac{\partial^2 T}{\partial r^2}\right]^2 \left[\frac{dz}{dr}\right]^{-2} dr \ ,$$

als we in (2.7.19) de coefficient $1/r$ door 1 vervangen.

In tabel 2.7.3 zijn een aantal waarden van V/1000 opgenomen voor $\Delta r = .1$.

Tabel 2.7.3 Waarden van $_{10}$-3 V voor $\Delta r = 10^{-1}$

ζ	ρ=.1	ρ=.2	ρ=.3	ρ=.4	ρ=.5
.1	377				
.2	73	377			
.3	34	103	377		
.4	21	47	139	377	
.5	14	27	72	170	377
.6	11	17	44	96	194
.7	8	12	30	62	118
.8	7	9	21	43	79
.9	6	7	16	32	57
1.0	5	5	13	24	43

Hieruit volgt dat de lijnen van gelijke variatie ongeveer lopen langs ζ/ρ = constant.

(3) Partiele discretisatie

In de roosterpunten $\{j\Delta z\}_{j=1}^{\infty}$ vervangen we de operator $A\partial^2/\partial z^2 + B\partial/\partial z$ door de differentie-operator

$$(2.7.20) \qquad A_j \frac{Z_+ - 2 + Z_-}{(\Delta z)^2} + B_j \frac{Z_+ - Z_-}{2\Delta z} \quad ,$$

waarin A_j en B_j de functiewaarden van A en B in $z = j\Delta z$ voorstellen. In het punt $z = 0$ is de functie B singulier, maar met behulp van de linkerrand-voorwaarde kunnen we schrijven:

$$B(z,T)\frac{\partial T}{\partial z}\Big|_{z=0} = (b-aT)\left(\frac{d^2 z}{dr^2} + \frac{1}{r}\frac{dz}{dr}\right)\frac{\partial T}{\partial z}\Big|_{z=0} =$$

$$= (b-aT)\frac{\frac{\partial T}{\partial z}}{z}\Big|_{z=0} = (b-aT)\frac{\partial^2 T}{\partial z^2}\Big|_{z=0} \quad ,$$

zodat in $z = 0$

$$A\frac{\partial^2 T}{\partial z^2} + B\frac{\partial T}{\partial z} = 2A\frac{\partial^2 T}{\partial z^2} = 2A_0 \frac{Z_+ - 2 + Z_-}{(\Delta z)^2} T = 4A_0 \frac{Z_+ - 1}{(\Delta z)^2} T \quad .$$

Aldus kan (2.7.17) gediscretiseerd worden tot het stelsel gewone differentiaalvergelijkingen:

$$(2.7.21) \qquad \frac{d\vec{y}}{dx} = D\vec{y} \quad ,$$

waarin $\vec{y}(x)$ de vector $(T(j\Delta z,x))_{j=0}^{\infty}$ benadert en D de (oneindige) matrix

$$\frac{1}{(\Delta z)^2}\begin{pmatrix} -4A_0 & 4A_0 & 0 & & \cdots \\ A_1 - \frac{1}{2}\Delta z B_1 & -2A_1 & A_1 + \frac{1}{2}\Delta z B_1 & 0 & \cdots \\ 0 & A_2 - \frac{1}{2}\Delta z B_2 & -2A_2 & A_2 + \frac{1}{2}\Delta z B_2 & \\ \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \end{pmatrix}$$

voorstelt.

(4) Reductie tot een eindig stelsel

Omdat de beginfunctie nul is voor $r \geq 1 + \Delta r$ zal de kromme $x = x(z)$, die in het (x,z)-vlak het gebied met $T > 0$ scheidt van het gebied waar $T \equiv 0$, een positieve helling hebben, met andere woorden het "warmtefront" beweegt zich in de tijd x van links naar rechts. Het is duidelijk dat in het gebied $T \equiv 0$ niet gerekend hoeft te worden, zodat het aantal relevante differentiaalvergelijkingen in het stelsel (2.7.21) eindig is, maar wel toeneemt met x. Bij een daadwerkelijke berekening neemt men al die vergelijkingen mee waarvoor $y_j(x) = T(j\Delta z,x)$ een zekere ondergrens (stel 10^{-12}) overschrijdt.

(5) Het spectrum van de Jacobiaan

Wanneer we de afhankelijkheid van de coefficientfuncties A en B van de temperatuur T verwaarlozen, dan geldt voor de Jacobiaan

$$J(\vec{y}) \cong D.$$

Aangezien in probleem (2.7.15) steeds $a << b$, is deze benadering voor J gerechtvaardigd (zie formule (2.7.17)), te meer daar het hier slechts gaat om een indruk te krijgen van het spectrum van J voor de keuze van een geschikt stabiliteitspolynoom.

Stel nu dat Δz voldoet aan de voorwaarde

$$\Delta z \leq 2 \max_j \frac{A_j}{|B_j|}, \tag{2.7.22}$$

dan heeft D positieve nevendiagonaalelementen, zodat volgens stelling 1.4.7 de eigenwaarden reëel zijn. Passen we vervolgens stelling 1.4.3 van Gerschgorin toe, dan blijken de eigenwaarden van D in het interval

$$[-4(\Delta z)^{-2} \max_{j\geq 1}(2A_0,A_j),0]$$

te liggen. Substitutie van de functie A geeft tenslotte het "veilige" eigenwaarde-interval

$$[-4(\Delta z)^{-2}b\ \max\{2,(\frac{2\zeta-\rho}{\rho})^2\},0] \ .$$

De eigenwaarden van de Jacobiaan J zullen "bijna" in dit interval liggen.

(6) Het stabiliteitspolynoom

Omdat de eigenwaarden van de Jacobiaan van het stelsel differentiaalvergelijkingen langs de negatieve reële as verwacht moeten worden, is het raadzaam een stabiliteitspolynoom met een grote reële stabiliteitsgrens te kiezen. Volgens dezelfde redenering als in de voorgaande paragraaf, is eerste orde consistentie voldoende; derhalve zoeken we binnen de klasse van eerste orde consistente polynomen die polynomen die het langst tussen ±1 blijven, als het argument van deze polynomen langs de negatieve naar $-\infty$ loopt. Dit probleem is een bekend minimax-probleem en wordt opgelost door de verschoven Chebyshev-polynomen

$$(2.7.23) \qquad T_m(1 + \frac{z}{m^2}) \quad , \quad \beta_{\text{reëel}} = 2m^2.$$

De hierbij behorende stabiliteitsvoorwaarde wordt:

$$(2.7.24) \qquad h_n \leq \frac{(m\Delta z)^2}{2b\ \max(2,(\frac{2\zeta-\rho}{\rho})^2)} \ .$$

Voor verdere details van de oplossing van probleem (2.7.15) verwijzen we naar de betreffende MC-publicatie TN 58/70.

2.7.4 Impliciete Taylor-methoden, de procedures *liniger 1vs* en *2vs*

Formeel kan een impliciete Taylor-methode gedefinieerd worden door formule (2.7.2); R is dan een rationale functie. Wanneer R consistent van de orde p is, dan is de gegenereerde methode ook p^e orde consistent.

In de bibliotheek NUMAL is een ALGOL 60-versie van (2.7.2) aanwezig voor de specifieke gevallen waarin de stabiliteitsfunctie R gegeven wordt door respectievelijk

$$(2.7.25) \qquad R(z) = \frac{1+(1+\alpha_1)z}{1+\alpha_1 z}$$

en

$$(2.7.26) \qquad R(z) = \frac{1+(1+\alpha_1)z+(\frac{1}{2}+\alpha_1+\alpha_2)z^2}{1+\alpha_1 z+\alpha_2 z^2} \; ;$$

hierin worden de parameter α_1 (en α_2) bepaald door de relatie(s)

$$(2.7.27) \qquad R(z_j) = \exp(z_j) \quad , \quad z_j \text{ gegeven.}$$

Deze relaties noemt men *exponentiele aanpassing van de orde 0*: in z_j lijkt de stabiliteitsfunctie namelijk wat functiewaarde betreft op de exponentiele functie. Indien ook de eerste afgeleide van de stabiliteitsfunctie met die van de exponentiele functie in een punt overeenstemt, dan spreekt men van *eerste orde exponentiele aanpassing* in dat punt, enz. In verband met het begrip exponentiele aanpassing merken we op dat de "stabiliteitsfunctie" van differentiaalvergelijkingen juist exp(z) is, immers de relatie tussen $\vec{y}(x_{n+1})$ en $\vec{y}(x_n)$ wordt voor lineaire vergelijkingen van de vorm (2.6.8') gegeven door

$$\vec{y}(x_{n+1}) = \exp(h_n J(y_n))\vec{y}(x_n).$$

Wat de aanpassingspunten z_1 en z_2 betreft, in het geval van (2.7.25) moet z_1 reëel zijn wil men tenminste complex rekenen vermijden en in het geval (2.7.26) moeten z_1 en z_2 óf reëel óf toegevoegd complex zijn. Overigens mag z_j vrij gekozen worden, waarbij de keuze $z_1 = z_2$ in (2.7.26) geinterpreteerd moet worden als *eerste orde aanpassing in* z_1.

De stabiliteitsfuncties (2.7.25) en (2.7.26) zijn in het algemeen 1^e en 2^e orde consistent. Men kan eenvoudig nagaan dat de keuze $z_1 = 0$ de orde van consistentie met 1 verhoogt; kiest men in (2.7.26) bovendien $z_2 = 0$ dan wordt R consistent van de orde 4.

Het stabiliteitsgebied van (2.7.25) beslaat het gehele linker halfvlak (zogenaamde *A-stabiliteit*) zolang z_1 in het linker halfvlak ligt. Voor (2.7.26) geldt hetzelfde maar z_1 en z_2 moeten bovendien niet te dicht bij de imaginaire as liggen.

De procedures *liniger 1vs* en *2vs*.

De door (2.7.25) en (2.7.26) gegenereerde impliciete Taylor-formules werden voorgesteld door Liniger en Willoughby [1970] en door Dekker geimplementeerd voor de SARA-computer onder de namen *liniger 1vs* en *liniger 2vs*. De gebruiksaanwijzingen, overgenomen uit de NUMAL-manual, section 5.2.1.1.1.2, vindt men aan het eind van deze paragraaf. We zullen hier nog enige toelichting geven.

Van de in de parameterlijst voorkomende parameters is de betekenis van *x*, *xe*, *m*, *y*, *aeta*, *reta* en *out* min of meer analoog aan die genoemd in de parameterlijst van *modified taylor*.
De parameter *sigma*, welke alleen in *liniger 1vs* voorkomt, is op een factor h_n^{-1} na gelijk aan de absolute waarde van het aanpassingspunt z_1, dus

$$|z_1| = sigma * h_n \ .$$

Kiest men voor *sigma* de absolute waarde van die (negatieve) eigenwaarde van de Jacobiaan waarvan de eigenvector een belangrijke rol speelt in de gezochte oplossing, dan zal *liniger 1vs* deze component in het geval van lineaire differentiaalvergelijkingen exact representeren; voor niet-lineaire vergelijkingen heeft exponentiele aanpassing alleen effect wanneer de verandering van de Jacobiaan klein is ten opzichte van de integratiestap h_n. Voor *sigma* = 0 en *sigma* $\to \infty$ gaat *liniger 1vs* over in respectievelijk de trapezium-regel en de terugwaartse Euler-formule. Deze keuzen worden aanbevolen voor niet-stijve respectievelijk stijve differentiaalvergelijkingen waarvan men het eigenwaardespectrum niet kent. De parameters *sigma 1* en *sigma 2* uit de parameterlijst van *liniger 2vs* hangen als volgt met de punten z_1 en z_2 samen:

	$z_1 \le z_2 \le 0$	$z_1 = \bar{z}_2$
sigma 1	$\lvert z_1 \rvert / h_n$	$\lvert z_1 \rvert / h_n$
sigma 2	$\lvert z_2 \rvert$	$-\arg(z_1)$

Dus door geschikte keuze van *sigma 1* en *sigma 2* kan men òf in twee op de negatieve as gelegen punten òf in twee toegevoegd complexe punten aanpassing verkrijgen. Ook hier geldt de aanbeveling om in $-\infty$ en 0 aan te passen

voor stijve respectievelijk niet-stijve differentiaalvergelijkingen met onbekend spectrum. De parameters *derivative* en *second derivative* zijn procedures die respectievelijk de rechterlidfunctie f en zijn afgeleide (naar x) $\vec{g}$ moeten leveren (zie "example of use" van de gebruiksaanwijzing). De parameters *itmax*, *jacobian*, en *j* doen hun intrede omdat in elke integratiestap een stelsel, in het algemeen niet-lineaire, vergelijkingen opgelost moet worden om $\vec{y}_{n+1}$ te verkrijgen. Dit stelsel heeft de vorm

$$\vec{y} + \alpha_1 h_n \vec{f}(\vec{y}) + \alpha_2 h_n^2 \vec{g}(\vec{y}) + \vec{b}_n \ ,$$

waarin

$$\vec{g}(\vec{y}) = J(\vec{y})\vec{f}(\vec{y}) \ ,$$

$$\vec{b}_n = \vec{y}_n + \beta_1 h_n \vec{f}(\vec{y}_n) + \beta_2 h_n^2 \vec{g}(\vec{y}_n) \ ,$$

$$\alpha_2 = \beta_2 = 0 \quad , \quad \beta_1 = 1 + \alpha_1 \quad \text{in geval van (2.7.25)} \ ,$$

$$\beta_1 = 1 + \alpha_1 \quad , \quad \beta_2 = \frac{1}{2} + \alpha_1 + \alpha_2 \quad \text{in geval van (2.7.26)} \ .$$

In navolging van Liniger en Willoughby wordt dit stelsel opgelost met een gemodificeerd Newton-Raphoon-proces met beginapproximatie $\vec{y}_n$:

$$\vec{y}_{n+1}^{(0)} = \vec{y}_n \ ,$$

$$\vec{y}_{n+1}^{(j+1)} = \vec{y}_{n+1}^{(j)} - \left[I+\alpha_1 h_n J(\vec{y}_{n+1}^{(j)})+\alpha_2 h_n^2 J^2(\vec{y}_{n+1}^{(j)})\right]^{-1} \cdot \tag{2.7.28}$$

$$\cdot \left[\vec{y}_{n+1}^{(j)}+\alpha_1 h_n \vec{f}(\vec{y}_{n+1}^{(j)})+\alpha_2 h_n^2 \vec{g}(\vec{y}_{n+1}^{(j)})-\vec{b}_n\right] \ ,$$

$$\vec{y}_{n+1} = \vec{y}_{n+1}^{(m)} \ .$$

(Voor $\alpha_2 \neq 0$ verschilt dit proces van het ongemodificeerde iteratieproces door het feit dat de Jacobiaan van de functie $\vec{g}$ (2^e afgeleide) vervangen is door het kwadraat van de Jacobiaan van de functie $\vec{f}$ (1^e afgeleide).)
Het iteratieproces (2.7.28) is echter zo geimplementeerd dat eerst geprobeerd wordt of het proces convergeert met de matrix $J(\vec{y}_n)$ in plaats van $J(\vec{y}_{n+1}^{(j)})$. Pas wanneer blijkt dat de convergentie te traag is worden J en $\vec{g}$ door middel van de procedures *jacobian* en *second derivative* gereëvalueerd.

Omdat deze evaluaties in het algemeen nogal duur zijn vergeleken bij rechterlidevaluaties kan hiermee veel rekentijd uitgespaard worden. Het voert te ver om de criteria volgens welke dit gebeurt en de strategie om het iteratieproces te stoppen hier uiteen te zetten (zie Dekker [1974]). In elk geval zal het proces echter nooit meer dan *itmax* iteraties uitvoeren.
Dit afbreken van het iteratieproces roept de vraag op wat er van het stabiliteitsgedrag van de impliciete formule overblijft, met andere woorden: wat is de stabiliteitsfunctie van schema (2.7.28)? Om dit te beantwoorden passen we (2.7.28) toe op de testvergelijking, dat wil zeggen een homogene, lineaire differentiaalvergelijking met Jacobiaan J. We vinden

$$\vec{y}_{n+1}^{(j+1)} = \{\vec{y}_{n+1}^{(j)} - [I+\alpha_1 h_n J+\alpha_2 h_n^2 J^2]^{-1} \cdot$$

$$\cdot [(I+\alpha_1 h_n J+\alpha_2 h_n^2 J^2)\vec{y}_{n+1}^{(j)} - (I+\beta_1 h_n J+\beta_2 h_n^2 J^2)\vec{y}_n]\} =$$

$$= [I+\alpha_1 h_n J+\alpha_2 h_n^2 J^2]^{-1} [I+\beta_1 h_n J+\beta_2 h^2 J^2]\vec{y}_n$$

voor j = 0,1,..., m-1. Hieruit volgt dat waar we ook afbreken, steeds wordt dezelfde stabiliteitsfunctie verkregen en deze is identiek met die van de impliciete formule. Dit is een direct gevolg van het feit dat het (gemodificeerde) Newton-Raphoon-process lineaire stelsels in één iteratieslag oplost (bij Jacobi-iteratie is dit niet het geval!).
Tenslotte verwijzen we voor de parameters *hmin*, *hmax* en *info* naar de nu volgende gebruiksaanwijzingen van de procedures *liniger 1vs* en *2vs*.

AUTHOR: K.DEKKER.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 730901.

BRIEF DESCRIPTION:

LINIGER1VS SOLVES INITIAL VALUE PROBLEMS , GIVEN AS AN AUTONOMOUS SYSTEM OF FIRST ORDER DIFFERENTIAL EQUATIONS , BY MEANS OF AN IMPLICIT,FIRST ORDER ACCURATE, EXPONENTIALLY FITTED ONESTEP METHOD. AUTOMATIC STEPSIZE CONTROL IS PROVIDED.
IN PARTICULAR THIS METHOD IS SUITABLE FOR THE INTEGRATION OF STIFF DIFFERENTIAL EQUATIONS.

KEYWORDS:

DIFFERENTIAL EQUATIONS,
INITIAL VALUE PROBLEMS,
STIFF EQUATIONS,
EXPONENTIAL FITTING,
IMPLICIT ONESTEP METHODS.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE LINIGER1VS READS;

```
"PROCEDURE" LINIGER1VS (X,XE,M,Y,SIGMA,DERIVATIVE,J,JACOBIAN,
                        ITMAX,HMIN,HMAX,AETA,RETA,INFO,OUTPUT);
"VALUE" M;
"INTEGER" M,ITMAX;
"REAL" X,XE,SIGMA,HMIN,HMAX,AETA,RETA;
"ARRAY" Y,J,INFO;
"PROCEDURE" DERIVATIVE,JACOBIAN,OUTPUT;
```

THE MEANING OF THE FORMAL PARAMETERS IS;

```
X;      <VARIABLE>;
        THE INDEPENDENT VARIABLE X;
        ENTRY; THE INITIAL VALUE X0;
        EXIT ; THE FINAL VALUE XE;
XE;     <ARITHMETIC EXPRESSION>;
        THE FINAL VALUE OF X (XE>=X);
M;      <ARITHMETIC EXPRESSION>;
        THE NUMBER OF EQUATIONS;
Y;      <ARRAY IDENTIFIER>;
        "ARRAY" Y[1:M];
        THE DEPENDENT VARIABLE;
        ENTRY; THE INITIAL VALUES OF THE SYSTEM OF DIFFERENTIAL
               EQUATIONS; Y[I] AT X=X0;
        EXIT ; THE FINAL VALUES OF THE SOLUTION; Y[I] AT X=XE;
```

```
SIGMA:  <ARITHMETIC EXPRESSION>;
        THE  MODULUS  OF  THE POINT AT WHICH EXPONENTIAL FITTING IS
        DESIRED, FOR EXAMPLE THE LARGEST NEGATIVE EIGENVALUE OF THE
        JACOBIAN OF THE SYSTEM OF DIFFERENTIAL EQUATIONS;
DERIVATIVE: <PROCEDURE IDENTIFIER>;
        THE HEADING OF THIS PROCEDURE READS;
        "PROCEDURE" DERIVATIVE(Y); "ARRAY" Y;
        THIS  PROCEDURE  SHOULD  DELIVER THE RIGHT HAND SIDE OF THE
        I-TH DIFFERENTIAL EQUATION AT THE POINT (Y) AS Y[I];
 J:     <ARRAY IDENTIFIER>;
        "ARRAY" J[1:M,1:M];
        THE JACOBIAN MATRIX OF THE SYSTEM;
        THE ARRAY J SHOULD BE UPDATED IN THE PROCEDURE JACOBIAN;
JACOBIAN: <PROCEDURE IDENTIFIER>;
        THE HEADING OF THIS PROCEDURE READS;
        "PROCEDURE" JACOBIAN(J,Y); "ARRAY" J,Y;
        IN THIS PROCEDURE (AN APPROXIMATION OF) THE JACOBIAN HAS TO
        BE ASSIGNED TO THE ARRAY J;
ITMAX:  <ARITHMETIC EXPRESSION>;
        AN  UPPERBOUND  FOR  THE  NUMBER  OF ITERATIONS IN NEWTON'S
        PROCESS, USED TO SOLVE THE IMPLICIT EQUATIONS;
HMIN:   <ARITHMETIC EXPRESSION>;
        MINIMAL  STEPSIZE  BY WHICH  THE INTEGRATION  IS PERFORMED;
HMAX:   <ARITHMETIC EXPRESSION>;
        MAXIMAL  STEPSIZE  BY WHICH  THE INTEGRATION  IS PERFORMED;
AETA:   <ARITHMETIC EXPRESSION>;
        REQUIRED  ABSOLUTE  PRECISION  IN THE INTEGRATION  PROCESS;
RETA:   <ARITHMETIC EXPRESSION>;
        REQUIRED  RELATIVE  PRECISION  IN THE INTEGRATION  PROCESS;
        IF BOTH AETA  AND RETA  HAVE  NEGATIVE VALUES , INTEGRATION
        WILL BE PERFORMED WITH A STEPSIZE EQUAL TO HMAX , WHICH MAY
        BE  VARIATED BY USER ; IN THIS CASE  THE ABSOLUTE VALUES OF
        AETA AND RETA WILL CONTROL THE NEWTON ITERATION;
INFO:   <ARRAY IDENTIFIER>;
        "ARRAY" INFO[1:9];
        DURING INTEGRATION  AND UPON EXIT  THIS ARRAY CONTAINS  THE
        FOLLOWING INFORMATION;
        INFO[1]: NUMBER OF INTEGRATION STEPS TAKEN;
        INFO[2]: NUMBER OF DERIVATIVE EVALUATIONS USED;
        INFO[3]: NUMBER OF JACOBIAN EVALUATIONS USED;
        INFO[4]: NUMBER OF INTEGRATION STEPS  EQUAL TO HMIN TAKEN ;
        INFO[5]: NUMBER OF INTEGRATION STEPS  EQUAL TO HMAX TAKEN ;
        INFO[6]: MAXIMAL NUMBER OF  ITERATIONS TAKEN  IN THE NEWTON
                 PROCESS;
        INFO[7]: LOCAL ERROR TOLERANCE;
        INFO[8]: ESTIMATED LOCAL ERROR;
        INFO[9]: MAXIMUM VALUE OF THE ESTIMATED LOCAL ERROR;
OUTPUT: <PROCEDURE IDENTIFIER>;
        THE HEADING OF THIS PROCEDURE READS;
        "PROCEDURE" OUTPUT;
        THIS  PROCEDURE  IS  CALLED  AT THE END OF EACH INTEGRATION
        STEP ; THE USER CAN ASK FOR OUTPUT  OF THE PARAMETERS , FOR
        EXAMPLE X, Y, INFO.
```

DATA AND RESULTS: SEE EXAMPLE OF USE, AND REF[2].

PROCEDURES USED:

INIVEC= CP31010,
MULVEC= CP31020,
MULROW= CP31021,
DUPVEC= CP31030,
MATVEC= CP34011,
ELMVEC= CP34020,
VECVEC= CP34010,
DEC = CP34300,
SOL = CP34051.

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: CIRCA 20 + M * (5+M).

RUNNING TIME: DEPENDS STRONGLY ON THE DIFFERENTIAL EQUATION TO SOLVE.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

LINIGER1VS: INTEGRATES THE SYSTEM OF DIFFERENTIAL EQUATIONS FROM X0 UNTIL XE, BY MEANS OF A FIRST ORDER FORMULA.

THE INTEGRATION METHOD IS BASED ON THE F(1) FORMULA DESCRIBED BY LINIGER AND WILLOUGHBY (SEE REF[1]). ERROR ESTIMATES AND A STEPSIZE STRATEGY FOR THIS METHOD ARE DESCRIBED IN [2] , AND A VARIABLE STEP METHOD IS CONSTRUCTED FOR THE CONVENIENCE OF THE USER. HOWEVER, THE STEPSIZE STRATEGY REQUIRES MANY EXTRA ARRAY OPERATIONS. THE USER MAY AVOID THIS EXTRA WORK BY GIVING AETA AND RETA A NEGATIVE VALUE AND PRESCRIBING A STEPSIZE (HMAX) HIMSELF.

REFERENCES:

[1]. W.LINIGER AND R.A.WILLOUGHBY.
EFFICIENT INTEGRATION METHODS FOR STIFF SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS.
SIAM J. NUM. ANAL. 7 (1970) PAGE 47.

[2]. K.DEKKER.
ERROR ESTIMATES AND STEPSIZE STRATEGIES FOR THE LINIGER-WILLOUGHBY FORMULAE.
(TO APPEAR IN 1974).

```
EXAMPLE OF USE:

    CONSIDER THE SYSTEM OF DIFFERENTIAL EQUATIONS;
    DY[1]/DX = -Y[1] + Y[1] * Y[2] + .99 * Y[2]
    DY[2]/DX = -1000 * ( -Y[1] + Y[1] * Y[2] + Y[2] )
    WITH THE INITIAL CONDITIONS AT X = 0;
    Y[1] = 1 AND Y[2] = 0.
    THE SOLUTION AT X = 50 IS APPROXIMATELY;
    Y[1] = .765 878 320 2487  AND  Y[2] = .433 710 353  5768.
    THE FOLLOWING  PROGRAM  SHOWS  INTEGRATION  OF  THIS  PROBLEM  WITH
    VARIABLE AND CONSTANT STEPSIZES;

"BEGIN" "COMMENT" TEST LINIGER1VS;
    "PROCEDURE" LINIGER1VS(X,XE,M,Y,SIGMA,F,J,JACOBIAN,
        ITMAX,HMIN,HMAX,AETA,RETA,INFO,OUTPUT);
    "CODE" 300;

    "INTEGER" ITMAX;
    "REAL" X,SIGMA,RETA,TIME;
    "REAL" "ARRAY" Y[1:2],J[1:2,1:2],INFO[1:9];

    "PROCEDURE" F(A); "ARRAY" A;
    "BEGIN" "REAL" A1,A2; A1:=A[1]; A2:=A[2];
        A[1]:=(A1+.99)*(A2-1)+.99;
        A[2]:=1000*((1+A1)*(1-A2)-1);
    "END";

    "PROCEDURE" JACOBIAN(J,Y); "ARRAY" J,Y;
    "BEGIN" J[1,1]:=Y[2]-1; J[1,2]:=.99+Y[1];
        J[2,1]:=1000*(1-Y[2]); J[2,2]:=-1000*(1+Y[1]);
        SIGMA:=ABS(J[2,2]+J[1,1]-SQRT((J[2,2]-J[1,1])**2+
              4*J[2,1]*J[1,2]))/2;
    "END" JACOBIAN;

    "PROCEDURE" OUT;
    "IF" X=50 "THEN"
    OUTPUT(61,"("6(3ZDB),2BD"-ZD,2(2B+,3DB3D),-3ZD,3D,/")",
    INFO[1],INFO[2],INFO[3],INFO[4],INFO[5],INFO[6],INFO[9],Y[1],
    Y[2],CLOCK-TIME);

    "FOR" RETA:="-2,"-4,"-6 "DO"
    "BEGIN" X:=Y[2]:=0; Y[1]:=1; TIME:=CLOCK;
        LINIGER1VS(X,50,2,Y,SIGMA,F,J,JACOBIAN,10,.1,50,RETA,
            RETA,INFO,OUT);
    "END"; OUTPUT(61,"("//")");
    "FOR" RETA:=-"-2,-"-4,-"-6 "DO"
    "BEGIN" X:=Y[2]:=0; Y[1]:=1; TIME:=CLOCK;
        LINIGER1VS(X,50,2,Y,SIGMA,F,J,JACOBIAN,10,.1,1,RETA,
            RETA,INFO,OUT);
    "END";
"END"
```

AUTHOR: K.DEKKER.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 730901.

BRIEF DESCRIPTION:

LINIGER2VS SOLVES INITIAL VALUE PROBLEMS , GIVEN AS AN AUTONOMOUS SYSTEM OF FIRST ORDER DIFFERENTIAL EQUATIONS , BY MEANS OF AN IMPLICIT EXPONENTIALLY FITTED ONESTEP METHOD.
AUTOMATIC STEPSIZE CONTROL IS PROVIDED.
IN PARTICULAR THIS METHOD IS SUITABLE FOR THE INTEGRATION OF STIFF DIFFERENTIAL EQUATIONS.

KEYWORDS:

DIFFERENTIAL EQUATIONS,
INITIAL VALUE PROBLEMS,
STIFF EQUATIONS,
EXPONENTIAL FITTING,
IMPLICIT ONESTEP METHODS.

```
CALLING SEQUENCE:

    THE HEADING OF THE PROCEDURE LINIGER2VS READS:
    "PROCEDURE" LINIGER2VS (X,XE,M,Y,SIGMA1,SIGMA2,DERIVATIVE,SECOND
       DERIVATIVE,J,JACOBIAN,ITMAX,HMIN,HMAX,AETA,RETA,INFO,OUTPUT);
    "VALUE" M;
    "INTEGER" M,ITMAX;
    "REAL" X,XE,SIGMA1,SIGMA2,HMIN,HMAX,AETA,RETA;
    "ARRAY" Y,J,INFO;
    "PROCEDURE" DERIVATIVE,SECOND DERIVATIVE,JACOBIAN,OUTPUT;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    X:      <VARIABLE>;
            THE INDEPENDENT VARIABLE X;
            ENTRY: THE INITIAL VALUE X0;
            EXIT : THE FINAL VALUE XE;
    XE:     <ARITHMETIC EXPRESSION>;
            THE FINAL VALUE OF X (XE>=X);
    M:      <ARITHMETIC EXPRESSION>;
            THE NUMBER OF EQUATIONS;
    Y:      <ARRAY IDENTIFIER>;
            "ARRAY" Y[1:M];
            THE DEPENDENT VARIABLE;
            ENTRY: THE  INITIAL  VALUES  OF  THE SYSTEM OF DIFFERENTIAL
                   EQUATIONS: Y[I] AT X=X0;
            EXIT : THE FINAL VALUES OF THE SOLUTION: Y[I] AT X=XE;
    SIGMA1: <ARITHMETIC EXPRESSION>;
            THE  MODULUS  OF  THE POINT AT WHICH EXPONENTIAL FITTING IS
            DESIRED;  THIS POINT  MAY BE  COMPLEX OR REAL AND NEGATIVE;
    SIGMA2: <ARITHMETIC EXPRESSION>;
            SIGMA2  MAY DEFINE  THREE  DIFFERENT TYPES  OF  EXPONENTIAL
            FITTING: FITTING IN TWO COMPLEX CONJUGATED POINTS , FITTING
            IN  TWO  REAL  NEGATIVE  POINTS , OR  FITTING  IN ONE POINT
            COMBINED WITH THIRD ORDER ACCURACY;
            IF THIRD ORDER ACCURACY IS DESIRED , SIGMA2 SHOULD HAVE THE
            VALUE 0;
            IF FITTING IN  A SECOND NEGATIVE POINT  IS DESIRED , SIGMA2
            SHOULD BE THE VALUE OF THE MODULUS OF THIS POINT;
            IF FITTING IN  TWO COMPLEX CONJUGATED  POINTS  IS  DESIRED,
            THEN SIGMA SHOULD BE MINUS THE VALUE OF THE ARGUMENT OF THE
            POINT IN THE SECOND QUADRANT (THUS A VALUE BETWEEN - PI AND
            - PI/2);
    DERIVATIVE: <PROCEDURE IDENTIFIER>;
            THE HEADING OF THIS PROCEDURE READS:
            "PROCEDURE" DERIVATIVE(Y); "ARRAY" Y;
            THIS  PROCEDURE  SHOULD  DELIVER THE RIGHT HAND SIDE OF THE
            I-TH DIFFERENTIAL EQUATION AT THE POINT (Y) AS Y[I];
```

```
SECOND DERIVATIVE: <PROCEDURE IDENTIFIER>;
        THE HEADING OF THIS PROCEDURE READS:
        "PROCEDURE" SECOND DERIVATIVE(Y,YACC);
        "VALUE" YACC; "ARRAY" Y,YACC;
        THIS PROCEDURE SHOULD DELIVER THE SECOND DERIVATIVES OF THE
        FUNCTIONS Y[I] AT THE POINT (Y) AS Y[I] ; IN THIS PROCEDURE
        THE FIRST DERIVATIVES, GIVEN IN ARRAY YACC, MAY BE USED BUT
        NOT MODIFIED;
 J:     <ARRAY IDENTIFIER>;
        "ARRAY" J[1:M,1:M];
        THE JACOBIAN MATRIX OF THE SYSTEM;
        THE ARRAY J SHOULD BE UPDATED IN THE PROCEDURE JACOBIAN;
JACOBIAN: <PROCEDURE IDENTIFIER>;
        THE HEADING OF THIS PROCEDURE READS:
        "PROCEDURE" JACOBIAN(J,Y); "ARRAY" J,Y;
        IN THIS PROCEDURE (AN APPROXIMATION OF) THE JACOBIAN HAS TO
        BE ASSIGNED TO THE ARRAY J;
ITMAX:  <ARITHMETIC EXPRESSION>;
        AN UPPERBOUND FOR THE NUMBER OF ITERATIONS IN NEWTON:S
        PROCESS, USED TO SOLVE THE IMPLICIT EQUATIONS;
HMIN:   <ARITHMETIC EXPRESSION>;
        MINIMAL STEPSIZE BY WHICH THE INTEGRATION IS PERFORMED;
HMAX:   <ARITHMETIC EXPRESSION>;
        MAXIMAL STEPSIZE BY WHICH THE INTEGRATION IS PERFORMED;
AETA:   <ARITHMETIC EXPRESSION>;
        REQUIRED ABSOLUTE PRECISION IN THE INTEGRATION PROCESS;
RETA:   <ARITHMETIC EXPRESSION>;
        REQUIRED RELATIVE PRECISION IN THE INTEGRATION PROCESS;
        IF BOTH AETA AND RETA HAVE NEGATIVE VALUES , INTEGRATION
        WILL BE PERFORMED WITH A STEPSIZE EQUAL TO HMAX , WHICH MAY
        BE VARIATED BY USER ; IN THIS CASE THE ABSOLUTE VALUES OF
        AETA AND RETA WILL CONTROL THE NEWTON ITERATION;
INFO:   <ARRAY IDENTIFIER>;
        "ARRAY" INFO[1:10];
        DURING INTEGRATION AND UPON EXIT THIS ARRAY CONTAINS THE
        FOLLOWING INFORMATION;
        INFO[1]: NUMBER OF INTEGRATION STEPS TAKEN;
        INFO[2]: NUMBER OF DERIVATIVE EVALUATIONS USED;
        INFO[3]: NUMBER OF JACOBIAN EVALUATIONS USED;
        INFO[4]: NUMBER OF INTEGRATION STEPS EQUAL TO HMIN TAKEN ;
        INFO[5]: NUMBER OF INTEGRATION STEPS EQUAL TO HMAX TAKEN ;
        INFO[6]: MAXIMAL NUMBER OF ITERATIONS TAKEN IN THE NEWTON
                 PROCESS;
        INFO[7]: LOCAL ERROR TOLERANCE;
        INFO[8]: ESTIMATED LOCAL ERROR;
        INFO[9]: MAXIMUM VALUE OF THE ESTIMATED LOCAL ERROR;
        INFO[10]:NUMBER OF SECOND DERIVATIVE EVALUATIONS USED;
OUTPUT: <PROCEDURE IDENTIFIER>;
        THE HEADING OF THIS PROCEDURE READS;
        "PROCEDURE" OUTPUT;
        THIS PROCEDURE IS CALLED AT THE END OF EACH INTEGRATION
        STEP , THE USER CAN ASK FOR OUTPUT OF THE PARAMETERS , FOR
        EXAMPLE X, Y, INFO.
```

DATA AND RESULTS: SEE EXAMPLE OF USE, AND REF[2].

PROCEDURES USED:

INIVEC= CP31010,
MULVEC= CP31020,
MULROW= CP31021,
DUPVEC= CP31030,
MATVEC= CP34011,
ELMVEC= CP34020,
VECVEC= CP34010,
DEC = CP34300,
SOL = CP34051.

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: CIRCA 30 + M * (6+M).

RUNNING TIME: DEPENDS STRONGLY ON THE DIFFERENTIAL EQUATION TO SOLVE.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

LINIGER2VS: INTEGRATES THE SYSTEM OF DIFFERENTIAL EQUATIONS FROM X0 UNTIL XE , BY MEANS OF A SECOND ORDER (IF SIGMA2=0 EVEN THIRD ORDER) FORMULA.

THE INTEGRATION METHOD IS BASED ON THE F(2) AND F(3) FORMULA DESCRIBED BY LINIGER AND WILLOUGHBY (SEE REF[1]) , ERROR ESTIMATES AND A STEPSIZE STRATEGY FOR THESE METHODS ARE DESCRIBED IN [2], AND A VARIABLE STEP METHOD IS CONSTRUCTED FOR THE CONVENIENCE OF THE USER . HOWEVER , THE STEPSIZE STRATEGY REQUIRES MANY EXTRA ARRAY OPERATIONS. THE USER MAY AVOID THIS EXTRA WORK BY GIVING AETA AND RETA A NEGATIVE VALUE, AND PRESCRIBING A STEPSIZE (HMAX) HIMSELF.

REFERENCES:

[1]. W.LINIGER AND R.A.WILLOUGHBY,
EFFICIENT INTEGRATION METHODS FOR STIFF SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS,
SIAM J. NUM. ANAL. 7 (1970) PAGE 47.

[2]. K.DEKKER,
ERROR ESTIMATES AND STEPSIZE STRATEGIES FOR THE LINIGER-WILLOUGHBY FORMULAE,
(TO APPEAR IN 1974).

EXAMPLE OF USE:

CONSIDER THE SYSTEM OF DIFFERENTIAL EQUATIONS:
DY[1]/DX = -Y[1] + Y[1] * Y[2] + .99 * Y[2]
DY[2]/DX = -1000 * (-Y[1] + Y[1] * Y[2] + Y[2])
WITH THE INITIAL CONDITIONS AT X = 0:
Y[1] = 1 AND Y[2] = 0.
THE SOLUTION AT X = 50 IS APPROXIMATELY:
Y[1] = .765 878 320 2487 AND Y[2] = .433 710 353 5768.
THE FOLLOWING PROGRAM SHOWS INTEGRATION OF THIS PROBLEM WITH
VARIABLE AND CONSTANT STEPSIZES:

```
"BEGIN" "COMMENT" TEST LINIGER2;
    "PROCEDURE" LINIGER2VS(X,XE,M,Y,SIGMA1,SIGMA2,F,G,J,JACOBIAN,
        ITMAX,HMIN,HMAX,AETA,RETA,INFO,OUTPUT);
    "CODE" 300;

    "INTEGER" ITMAX;
    "REAL" X,SIGMA,RETA,TIME;
    "REAL" "ARRAY" Y[1:2],J[1:2,1:2],INFO[1:9];

    "PROCEDURE" F(A); "ARRAY" A;
    "BEGIN" "REAL" A1,A2; A1:=A[1]; A2:=A[2];
        A[1]:=(A1+.99)*(A2-1)+.99;
        A[2]:=1000*((1+A1)*(1-A2)-1);
    "END";
    "PROCEDURE" JACOBIAN(J,Y); "ARRAY" J,Y;
    "BEGIN" J[1,1]:=Y[2]-1; J[1,2]:=.99+Y[1];
        J[2,1]:=1000*(1-Y[2]); J[2,2]:=-1000*(1+Y[1]);
        SIGMA:=ABS(J[2,2]+J[1,1]-SQRT((J[2,2]-J[1,1])**2+
               4*J[2,1]*J[1,2]))/2;
    "END" JACOBIAN;
    "PROCEDURE" G(Y,YACC); "ARRAY" Y, YACC;
    "BEGIN" "REAL" Y1,Y2; Y1:=Y[1]; Y2:=Y[2];
        Y[1]:=(Y2-1)*YACC[1]+(Y1+.99)*YACC[2];
        Y[2]:=1000*((1-Y2)*YACC[1]-(1+Y1)*YACC[2])
    "END";
    "PROCEDURE" OUT;
    "IF" X=50 "THEN"
    OUTPUT(61,"("7(3ZDB),2BD"-ZD,2(2B+.3DB3D),-3ZD,3D,/")",
    INFO[1],INFO[2],INFO[3],INFO[10],INFO[4],INFO[5],INFO[6],INFO[9],
    Y[1],Y[2],CLOCK-TIME);

    "FOR" RETA:="-2,"-4,"-6 "DO"
    "BEGIN" X:=Y[2]:=0; Y[1]:=1; TIME:=CLOCK;
        LINIGER2VS(X,50,2,Y,SIGMA,0,F,G,J,JACOBIAN,10,.1,50,RETA,
            RETA,INFO,OUT);
    "END";  OUTPUT(61,"("//")");
    "FOR" RETA:=-"-2,-"-4,-"-6 "DO"
    "BEGIN" X:=Y[2]:=0; Y[1]:=1; TIME:=CLOCK;
        LINIGER2VS(X,50,2,Y,SIGMA,0,F,G,J,JACOBIAN,10,.1,1,RETA,
            RETA,INFO,OUT);
    "END";
"END"
```

Opgaven 2.7.1

(1) Toon aan dat (2.7.26) 4^e orde consistent wordt als $z_1 = z_2 = 0$ gesteld wordt.

(2) Aan welke voorwaarden moeten de parameters α_1 en α_2 in (2.7.26) voldoen opdat R A-stabiel is.

(3) Bewijs dat een p^e orde consistente stabiliteitsfunctie een p^e orde exponentiele aanpassing heeft in z = 0.

2.7.5 Eerste en tweede orde Runge-Kuttaformules

In paragraaf 2.2 hebben we laten zien dat de algemene expliciete m-punts Runge-Kuttaformule $\frac{1}{2}m(m-1)$ parameters bevat. Identificatie van het stabiliteitspolynoom van zo'n formule met een gegeven eerste of tweede orde consistent polynoom (de Runge-Kuttaformule is dan ook eerste of tweede orde consistent) geeft aanleiding tot m relaties zodat verwacht mag worden dat er vele eerste en tweede orde formules bestaan. We zullen hiervan gebruik maken door uit te gaan van formules met *beperkt geheugengebruik*, namelijk formules van de vorm

$$
\begin{aligned}
&\vec{y}_{n+1}^{(0)} = \vec{y}_n \ , \\
&\vec{y}_{n+1}^{(j)} = \vec{y}_n + \lambda_{j,j-1} h_n \vec{f}(\vec{y}_{n+1}^{(j-1)}) \ , \quad j = 1,2,\ldots,m \ , \\
&\vec{y}_{n+1} = \vec{y}_{n+1}^{(m)} \ .
\end{aligned} \tag{2.7.28}
$$

Toepassing op de modelvergelijking

$$\frac{d\vec{y}}{dx} = J\vec{y}$$

geeft het stabiliteitspolynoom

$$1 + \lambda_{m,m-1} z(1+\lambda_{m-1,m-2} z(1+\lambda_{m-2,m-3}(1+\ldots$$

$$+\lambda_{2,1} z(1+\lambda_{1,0} z))\ldots) \ .$$

Identificatie met een gegeven polynoom

$$R(z) = 1 + \beta_1 z + \beta_2 z^2 + \ldots + \beta_m z^m \tag{2.7.29}$$

leidt tot relaties tussen de Runge-Kuttaparameters $\lambda_{j,j-1}$ en de coefficienten van de gegeven functie R, namelijk

$$(2.7.30) \qquad \lambda_{j,j-1} = \frac{\beta_{m+1-j}}{\beta_{m-j}} \quad , \quad j = 1,2,\ldots,m \ ,$$

waarin $\beta_0 = 1$.

Indien $\beta_1 = 1$ dan is (2.7.29) 1^e *orde consistent* en daarmee (2.7.28). We zullen deze formules *gestabiliseerde Euler-formules* noemen. Is bovendien $\beta_2 = \frac{1}{2}$ dan is (2.7.29) en dus (2.7.28) 2^e *orde consistent*. De corresponderende formules noemen we *gestabiliseerde Runge-formules*.

Voorbeeld 2.7.1

Enkele gestabiliseerde Euler-formules worden gegenereerd door de parametermatrices

$$\begin{pmatrix} 0 & 0 \\ \frac{1}{8} & 0 \\ 0 & 1 \end{pmatrix}, \quad R(z) = 1 + z + \frac{1}{8}z^2 \quad , \quad \beta_{\text{reëel}} = 8 \ ;$$

$$\begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{27} & 0 & 0 \\ 0 & \frac{4}{27} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad R(z) = 1 + z + \frac{4}{27}z^2 + \frac{4}{729}z^3 \quad , \quad \beta_{\text{reëel}} = 18 \ ;$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{64} & 0 & 0 & 0 \\ 0 & \frac{1}{20} & 0 & 0 \\ 0 & 0 & \frac{5}{32} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad R(z) = 1 + z + \frac{5}{32}z^2 + \frac{1}{128}z^3 + \frac{1}{8192}z^4 \ ,$$

$$\beta_{\text{reëel}} = 32 \ ;$$

$$\begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad R(z) = 1 + z + z^2 \quad , \quad \beta_{\text{imag}} = 1 \ ;$$

Voorbeeld 2.7.2

Voorbeelden van gestabiliseerde Runge-formules zijn

$$\begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{8} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad R(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{16}z^3, \quad \beta_{\text{reëel}} = 6.26 ;$$

$$\begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad R(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{4}z^3, \quad \beta_{\text{imag}} = 2 .$$

2.7.6 Derde orde Runge-Kuttaformules

Wanneer we voor formule (2.7.28) de voorwaarden voor derde orde consistentie, gegeven door tabel 2.4.2, uitwerken dan vinden we de relaties

$$\lambda_{m,m-1} = 1, \quad \lambda_{m-1,m-2} = \frac{1}{2}, \quad \lambda_{m-2,m-3} = \frac{1}{3}, \quad \lambda^2_{m-2,m-3} = \frac{1}{3} .$$

Dit betekent dat formule (2.7.28) niet derde orde consistent kan zijn. Daarom introduceren we een nieuwe parameter in de genererende matrix. In het bijzonder beschouwen we matrices van de vorm

$$\begin{pmatrix} 0 & 0 & & \cdots & 0 \\ \lambda_{1,0} & 0 & & & \\ \lambda_{2,0} & \lambda_{2,1} & & & \\ \lambda_{2,0} & 0 & \lambda_{3,2} & & \\ \vdots & \vdots & & \ddots & \\ \lambda_{2,0} & 0 & \cdots & 0 & \lambda_{m,m-1} \end{pmatrix} . \tag{2.7.31}$$

Deze vorm is gekozen omdat het corresponderende Runge-Kuttaschema geheugenbesparend is; het kan namelijk geschreven worden als:

$$\vec{y}_n^{(0)} = \vec{y}_n + \lambda_{2,0} h_n \vec{f}(\vec{y}_n) ,$$

$$\vec{y}_{n+1}^{(1)} = \vec{y}_n^{(0)} + (\lambda_{1,0} - \lambda_{2,0}) h_n \vec{f}(\vec{y}_n) ,$$

$$\vec{y}_{n+1}^{(j)} = \vec{y}_n^{(0)} + \lambda_{j,j-1} h_n \vec{f}(\vec{y}_{n+1}^{(j-1)}) \quad , \quad j = 2,3,\ldots,m ,$$

$$\vec{y}_{n+1} = \vec{y}_{n+1}^{(m)} .$$

Volgens de formules (2.4.9) kunnen we de parameters $\lambda_{2,0}$, $\lambda_{m-2,m-3}$, $\lambda_{m-1,m-2}$ en $\lambda_{m,m-1}$ in de parameters β_1, β_2, β_3 en $\beta_{3,1}$ uitdrukken; we vinden dan

$$\lambda_{2,0} = \beta_1 - \frac{\beta_2^2}{\beta_{3,1}} \quad , \quad \lambda_{m,m-1} = \frac{\beta_2^2}{\beta_{3,1}}$$

(2.7.32)

$$\lambda_{m-1,m-2} = \frac{\beta_2}{\lambda_{m,m-1}} - \lambda_{2,0} \quad , \quad \lambda_{m-2,m-3} = \frac{\beta_3}{\beta_2}\left(1 + \frac{\lambda_{2,0}}{\lambda_{m-1,m-2}}\right) - \lambda_{2,0} .$$

Kiezen we vervolgens voor β_1, β_2, β_3 en $\beta_{3,1}$ de waarde 1, $\frac{1}{2}$, $\frac{1}{6}$ en $\frac{1}{3}$ uit tabel 2.4.2, of nog iets algemener de waarden $1 + O(h_n^3)$, $\frac{1}{2} + O(h_n^2)$, $\frac{1}{6} + O(h_n)$ en $\frac{1}{3} + O(h_n)$ volgens de overwegingen uit paragraaf 2.4.4, dan definieren (2.7.31) en (2.7.32) een *derde orde consistente Runge-Kuttaformule* met stabiliteitsfunctie

$$R(z) = 1 + \beta_1 z + \ldots + \beta_m z^m .$$

2.7.7 De procedure *ark*

Een ALGOL 60-versie van de formules (2.7.28) en (2.7.31) waarin R een vrij te kiezen polynoom is, kan men vinden in de bibliotheek NUMAL; deze formules zijn geïmplementeerd onder de naam *ark* (Adaptive Runge Kutta) en gedocumenteerd in section 5.2.1.1.1.1. van de bijbehorende manual. De gebruiksaanwijzingen zijn aan het eind van deze paragraaf opgenomen. De parameters zijn vrijwel analoog aan die van *modified taylor* met uitzondering

van de parameter *derivative* waarmee nu uitsluitend de rechterlidfunctie (eerste afgeleide van $\vec{y}$), en geen hogere afgeleiden, gegeven wordt. De organisatie van de parameterlijst is echter anders: zo worden de parameters *i*, *sigma*, *taumin*, *aeta*, *reta*, *k* en *rho* nu gegeven door *data* [*1*] en *data* [*4*] tot en met *data* [*9*], terwijl het array *data* uit *modified taylor* nu correspondeert met de array-elementen *data* [*1*] tot en met *data* [*3*] en *data* [*11*] tot en met *data* [*10 + data* [*1*]]. De parameters *alfa*, *norm* en *eta* komen niet meer in *ark* voor; de groeiparameter *alfa* is in de procedure zelf gelijk aan 2 gemaakt, waar vectornormen in de procedure nodig zijn wordt altijd de Euclidische norm gekozen en *eta* kan de gebruiker zelf laten uitrekenen volgens de in paragraaf 2.7.1 gegeven formule.

AUTHOR: P.A. BEENTJES.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 740510.

BRIEF DESCRIPTION:

ARK SOLVES AN INITIAL VALUE PROBLEM, GIVEN AS A SYSTEM OF FIRST ORDER (NON-LINEAR) DIFFERENTIAL EQUATIONS BY MEANS OF A STABILIZED RUNGE KUTTA METHOD WITH LIMITED STORAGE REQUIREMENTS.

KEYWORDS:

DIFFERENTIAL EQUATIONS,
INITIAL VALUE PROBLEM,
EXPLICIT ONE-STEP METHOD,
STABILIZED RUNGE KUTTA METHOD.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE READS:
"PROCEDURE" ARK (T, TE, M0, M, U, DERIVATIVE, DATA, OUT);
"INTEGER" M0, M; "REAL" T, TE; "ARRAY" U, DATA;
"PROCEDURE" DERIVATIVE, OUT;

THE MEANING OF THE FORMAL PARAMETERS IS:
T: <VARIABLE>;
THE INDEPENDENT VARIABLE T; CAN BE USED IN DERIVATIVE;
ENTRY: THE INITIAL VALUE T0;
EXIT : THE FINAL VALUE TE;
TE: <ARITHMETIC EXPRESSION>;
THE FINAL VALUE OF T (TE >= T);
M0,M: <ARITHMETIC EXPRESSION>;
INDICES OF THE FIRST AND LAST EQUATION OF THE SYSTEM;
U: <ARRAY IDENTIFIER>;
"ARRAY" U[M0 : M];
ENTRY: THE INITIAL VALUES OF THE SOLUTION OF THE SYSTEM OF DIFFERENTIAL EQUATIONS AT T = T0;
EXIT : THE VALUES OF THE SOLUTION AT T = TE;
DERIVATIVE: <PROCEDURE IDENTIFIER>;
THE HEADING OF THIS PROCEDURE READS:
"PROCEDURE" DERIVATIVE(T, V); "REAL" T; "ARRAY" V;
THIS PROCEDURE PERFORMS AN EVALUATION OF THE RIGHT HAND SIDE OF THE SYSTEM WITH DEPENDENT VARIABLES V[M0 : M] AND INDEPENDENT VARIABLE T; UPON COMPLETION OF DERIVATIVE, THE RIGHT HAND SIDE SHOULD BE OVERWRITTEN ON V[M0 : M];

```
DATA:   <ARRAY IDENTIFIER>;
        "ARRAY" DATA[1 : 10 + DATA[1]];
        IN ARRAY DATA ONE SHOULD GIVE:
        DATA[1]: THE NUMBER OF EVALUATIONS OF H(U, T) PER
                 INTEGRATION STEP(DATA[1] >= DATA[2]);
        DATA[2]: THE ORDER OF ACCURACY OF THE METHOD (DATA[2]<=3);
        DATA[3]: STABILITY BOUND(SEE REFERENCE [3]);
        DATA[4]: THE SPECTRAL RADIUS OF THE JACOBIAN MATRIX WITH
                 RESPECT TO THOSE EIGENVALUES, WHICH ARE LOCATED
                 IN THE NON-POSITIVE HALF PLANE;
        DATA[5]: THE MINIMAL STEPSIZE;
        DATA[6]: THE ABSOLUTE TOLERANCE;
        DATA[7]: THE RELATIVE TOLERANCE;
                 IF BOTH DATA[6] AND DATA[7] ARE NEGATIVE, THE
                 INTEGRATION IS PERFORMED WITH A CONSTANT STEP
                 DATA[5];
        DATA[8]: DATA[8] SHOULD BE 0 IF ARK IS CALLED FOR
                 A FIRST TIME; FOR CONTINUED INTEGRATION (E.G.
                 FROM TE TO TE-NEW) DATA[8] SHOULD NOT BE CHANGED;
        DATA[11], ..., DATA[10 + DATA[1]]: POLYNOMIAL COEFFICIENTS
                 (SEE REFERENCE [3]);
        AFTER EACH STEP THE FOLLOWING BY-PRODUCTS ARE DELIVERED:
        DATA[8]: THE NUMBER OF INTEGRATION STEPS PERFORMED;
        DATA[9]: AN ESTIMATION OF THE LOCAL ERROR LAST MADE;
        DATA[10]: INFORMATIVE MESSAGES:
                 DATA[10] = 0: NO DIFFICULTIES;
                 DATA[10] = 1: MINIMAL STEPLENGTH EXCEEDS THE
                          STEPLENGTH PRESCRIBED BY STABILITY
                          THEORY, I.E. DATA[5] > DATA[3] / DATA[4];
                          (TERMINATION OF ARK);
                          DECREASE MINIMAL STEPLENGTH;
        IF NECESSARY, DATA[I],I = 4(1)7, CAN BE UPDATED(AFTER EACH
        STEP) BY MEANS OF PROCEDURE OUT;
OUT:    <PROCEDURE IDENTIFIER>;
        THE HEADING OF THIS PROCEDURE READS:
        "PROCEDURE" OUT;
        AFTER EACH INTEGRATION STEP PERFORMED INFORMATION CAN BE
        OBTAINED OR UPDATED BY THIS PROCEDURE, E.G. THE VALUES OF
        T, U[M0 : M] AND DATA[I], I = 4(1)10.
```

DATA AND RESULTS:

FOR THE INDICES M0 AND M THE FOLLOWING REMARKS CAN BE MADE: WHEN THE METHOD OF LINES IS APPLIED TO HYPERBOLIC DIFFERENTIAL EQUATIONS THE NUMBER OF RELEVANT ORDINARY DIFFERENTIAL EQUATIONS DECREASES DURING THE INTEGRATION PROCESS; IN PROCEDURE ARK THIS MAY BE REALIZED BY INTEGERS M_0 AND M, WHICH ARE DEFINED AS FUNCTIONS OF THE NUMBER OF RIGHT HAND SIDE EVALUATIONS.

PROCEDURES USED:

INIVEC = CP31010,
MULVEC = CP31020,
DUPVEC = CP31030,
VECVEC = CP34010,
ELMVEC = CP34020,
DECSOL = CP34301.

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: CIRCA 75 + 2 * (M - M0).

RUNNING TIME: DEPENDS STRONGLY ON THE PROBLEM TO BE SOLVED.

LANGUAGE: ALGOL60.

METHOD AND PERFORMANCE:

ARK IS AN IMPLEMENTATION OF LOW ORDER STABILIZED RUNGE KUTTA METHODS (SEE REFERENCE [1]);
AUTOMATIC STEPSIZE CONTROL IS PROVIDED BUT STEP-REJECTION HAS BEEN EXCLUDED IN ORDER TO SAVE STORAGE;
BECAUSE OF ITS LIMITED STORAGE REQUIREMENTS AND ADAPTIVE STABILITY FACILITIES THE METHOD IS WELL SUITED FOR THE SOLUTION OF INITIAL BOUNDARY VALUE PROBLEMS FOR PARTIAL DIFFERENTIAL EQUATIONS;
NUMERICAL RESULTS, OBTAINED WITH A SLIGHTLY DIFFERENT IMPLEMENTATION CAN BE FOUND IN REFERENCE [2].

REFERENCES:

[1]. P.J. VAN DER HOUWEN,
STABILIZED RUNGE KUTTA METHOD WITH LIMITED STORAGE REQUIREMENTS,
MATH. CENTR. REPORT TW 124/71;

[2]. P.A. BEENTJES,
AN ALGOL 60 VERSION OF STABILIZED RUNGE KUTTA METHODS (DUTCH),
MATH. CENTR. REPORT NR 23/72;

[3]. P.J. VAN DER HOUWEN, J. KOK,
NUMERICAL SOLUTION OF A MINIMAX PROBLEM,
MATH. CENTR. REPORT TW 123/71;

EXAMPLE OF USE:

THE VALUES OF

1, Y(1) AND Y(2) OF THE INITIAL VALUE PROBLEM
DY / DX = Y - 2 * X / Y, Y(0) = 1

AND

2, U(.6, 0) OF THE CAUCHY PROBLEM (SEE REFERENCE [2]):
DU / DT = .5 * DU / DX, U(0, X) = EXP(-X * X)

MAY BE OBTAINED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "INTEGER" M0, M, I; "REAL" T, TE, DAT;
    "ARRAY" Y[1 : 1], U[-150 : 150], DATA[1 : 14];

    "PROCEDURE" ARK
    (T, TE, M0, M, U, DERIVATIVE, DATA, OUT); "CODE" 33061;

    "PROCEDURE" DER1(T, V); "REAL" T; "ARRAY" V;
    V[1]:= V[1] - 2 * T / V[1];

    "PROCEDURE" DER2(T, V); "REAL" T; "ARRAY" V;
    "BEGIN" "INTEGER" J; "REAL" V1, V2, V3;
       V2:= V[M0]; M0:= M0 + 1; M:= M - 1; V3:= V[M0];
       "FOR" J:= M0 "STEP" 1 "UNTIL" M "DO"
       "BEGIN" V1:= V2; V2:= V3; V3:= V[J + 1];
           V[J]:= 250 * (V3 - V1) / 3
       "END"
    "END" DER2;

    "PROCEDURE" OUT1;
    "IF" T = TE "THEN"
    "BEGIN" "IF" T = 1 "THEN" OUTPUT(61, "("/, "(" PROBLEM 1")", //,
        "(" X  NUMBER OF INTEGRATION STEPS Y(COMPUTED)  Y(EXACT)")",
        //")");
        OUTPUT(61, "("ZD, 13ZD,12B,2(-3ZD,7D),"(",,,")", /")",
        T, DATA[8], Y[1], SQRT(2 * T + 1));
        TE:= 2
    "END" OUT1;
```

```
    "PROCEDURE" OUT2;
    "IF" T = ,6 "THEN"
    OUTPUT(61, "("//, "(" PROBLEM 2")", //,
       "(" NUMBER OF DERIVATIVE CALLS")",
       "(" U(,6, 0)COMPUTED   U(,6, 0)EXACT")", //, 13ZD, 4D,
       2(-10Z,7D), "("...")""")", DATA[1] * DATA[8], U[0], EXP(-,09));

    I:= 1;
    "FOR" DAT:= 3, 3, 1, 1, "-3, "-6, "-6, 0, 0, 0, 1, ,5, 1 / 6 "DO"
    "BEGIN" DATA[I]:= DAT; I:= I + 1 "END";
    T:= 0; Y[1]:= 1; TE:= 1;
    ARK(T, TE, 1, 1, Y, DER1, DATA, OUT1);
    I:= 1;
    "FOR" DAT:= 4, 3, SQRT(8), 500 / 3, DATA[3] / DATA[4], -1, -1,
       0, 0, 0, 1, ,5, 1 / 6, 1 / 24 "DO"
    "BEGIN" DATA[I]:= DAT; I:= I + 1 "END";
    M0:= -150; M:= 150; T:= 0; U[0]:= 1;
    "FOR" I:= 1 "STEP" 1 "UNTIL" M "DO"
    U[I]:= U[-I]:= EXP(-(,003 * I) ** 2);
    ARK(T, ,6, M0, M, U, DER2, DATA, OUT2)
"END"
```

THIS PROGRAM DELIVERS:

PROBLEM 1

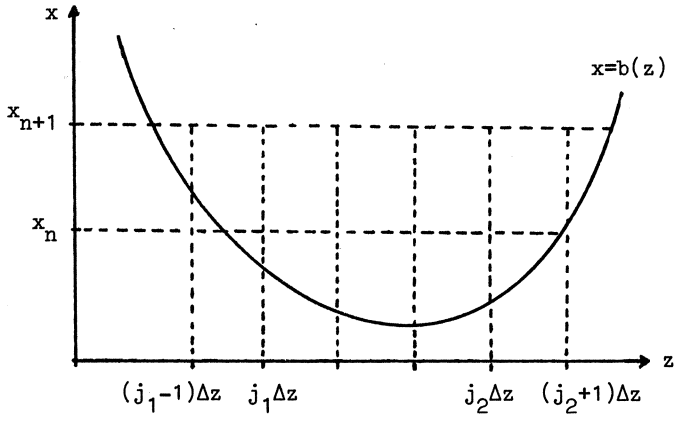
X	NUMBER OF INTEGRATION STEPS	Y(COMPUTED)	Y(EXACT)
1	38	1,7320535	1,7320508...
2	56	2,2360928	2,2360680...

PROBLEM 2

NUMBER OF DERIVATIVE CALLS	U(,6, 0)COMPUTED	U(,6, 0)EXACT
144	,9139326	,9139312...

2.7.8 Een diffucieprobleem met "opschuivende" randvoorwaarden

In paragraaf 1.2.4, voorbeeld 1.2.3 hebben we een diffusieprobleem met samengesmolten begin- en randvoorwaarden beschouwd en door middel van partiele discretisatie herleid tot een stelsel gewone differentiaalvergelijkingen waarvan het aantal toeneemt wanneer x toeneemt (zie formule (1.2.13)). Dit betekent dat de dimensie van de differentieoplossing in x_{n+1} (de vector $\vec{y}_{n+1}$) *groter* kan zijn dan de dimensie van de differentieoplossing in x_n (de vector $\vec{y}_n$). Het is duidelijk dat een integratieformule $\vec{y}_{n+1} = E_n(\vec{y}_n)$, zoals gedefinieerd in de voorgaande paragrafen, niet zonder meer toegepast kan worden; aan de randen zullen aanvullende differentieformules gedefinieerd moeten worden. We zullen dit illustreren voor het geval waarin stelsel (1.2.13) er in elke integratiestap links en rechts een differentiaalvergelijking bij krijgt (zie figuur 2.7.4).



Figuur 2.7.4 Opschuivende randvoorwaarden

Stel dat de vector $\vec{y}_n$, waarvan de componenten een benadering voor $\vec{y}(x_n)$ in de punten $j_1\Delta z,\ldots,j_2\Delta z$ voorstellen, berekend is. Toepassing van een Runge-Kuttaoperator E_n op $\vec{y}_n$ levert de $j_1^{\,e}$ tot en met $j_2^{\,e}$ component van $\vec{y}_{n+1}$. De $(j_1-1)^e$ en $(j_2+1)^e$ componeqnt van y_{n+1} moeten op andere wijze verkregen worden. Stel dat E_n eerste orde nauwkeurig is, dan zijn de volgende additionele formules bruikbaar

$$(\vec{y}_{n+1})_{j_1-1} = g(b((j_i-1)\Delta z,(j_1-1)\Delta z)) + (x_{n+1}-b((j_1-1)\Delta z)) \cdot$$
$$\cdot (\vec{f}(\vec{y}_{n+1}))_{j_1-1}$$
;
$$(\vec{y}_{n+1})_{j_2+1} = g(b((j_2+1)\Delta z,(j_2+1)\Delta z)) + (x_{n+1}-b((j_2+1)\Delta z)) \cdot$$
$$\cdot (\vec{f}(\vec{y}_{n+1}))_{j_2+1}$$

hierin stelt $\vec{f}(\vec{y})$ het door (2.1.13) gedefinieerde rechterlid voor.
Merk op dat deze formules impliciet zijn in respectievelijk $(y_{n+1})_{j_1-1}$ en $(y_{n+1})_{j_2+1}$, maar omdat deze componenten lineair voorkomen zijn ze direct te berekenen.

2.7.9 Gegeneraliseerde Runge-Kuttaformules van eerste en tweede orde

In paragraaf 2.4.3 zijn de consistentievoorwaarden afgeleid voor de algemene 2-puntsformule. Het is mogelijk om 2^e orde consistentie te verkrijgen voor de deelklasse der 1-puntsformules, dus formules van de vorm (vergelijk (2.4.11) met $\Theta_1 \equiv 0$)

$$(2.7.33) \qquad \vec{y}_{n+1} = \vec{y}_n + h_n\Theta_0(h_nJ(\vec{y}_n))\vec{f}(\vec{y}_n) \ .$$

Volgens stelling 2.6.2 is (2.7.33) respectievelijk 1^e en 2^e orde consistent wanneer de stabiliteitsfunctie

$$1 + z\Theta_0(z)$$

geidentificeerd wordt met een gegeven functie R die respectievelijk 1^e en 2^e orde consistent is, dat wil zeggen wanneer we Φ_0 definieren volgens de relatie

$$(2.7.34) \qquad \Theta_0(z) \equiv \frac{R(z)-1}{z} \ .$$

Formule (2.7.33) gaat dan over in

$$(2.7.33') \qquad \vec{y}_{n+1} = \vec{y}_n + J^{-1}(\vec{y}_n) \cdot [R(h_nJ(y_n))-I]\vec{f}(\vec{y}_n) \ .$$

Wanneer R een polynoom is dan kan deze formule gesimuleerd worden met behulp van de procedure *modified taylor*. Daartoe wijzige men de definitie van de parameter *derivative* in de parameterlijst van *modified taylor* als volgt:

```
DERIVATIVE : <PROCEDURE IDENTIFIER>;
             THE HEADING OF THIS PROCEDURE READS:
             PROCEDURE DERIVATIVE (I,A); INTEGER I; ARRAY A;
             WHEN THIS PROCEDURE IS CALLED, ARRAY A CONTAINS THE
             COMPONENTS OF IF I = 2 THEN THE RIGHT HAND SIDE IF
             I > 2 THEN THE RIGHT HAND SIDE MULTIPLIED BY THE
             (I-2)-ND POWER OF THE JACOBIAN MATRIX OF THE RIGHT
             HAND SIDE;
             UPON COMPLETION OF DERIVATIVE ARRAY A SHOULD CONTAIN
             THE COMPONENTS OF IF I = 1 THEN THE RIGHT HAND SIDE
             ELSE THE RIGHT HAND SIDE MULTIPLIED BY THE (I-1)-ST
             POWER OF THE JACOBIAN MATRIX;
```

Verder moet de voorwaarde gesteld worden dat het te integreren stelsel autonoom is. Indien hieraan voldaan is voert *modified taylor* de volgende algorithme uit:

$$(2.7.35)\quad \vec{y}_{n+1} = \vec{y}_n + \beta_1 h_n \vec{f}(\vec{y}_n) + \beta_2 h_n^2 J(\vec{y}_n)\vec{f}(\vec{y}_n) + \ldots + \beta_m h_n^m J^{m-1}(\vec{y}_n)\vec{f}(\vec{y}_n),$$

waarin de coefficienten β_j de coefficienten van het voorgeschreven stabiliteitspolynoom R zijn. We kunnen (2.7.35) nu schrijven als

$$\vec{y}_{n+1} = \vec{y}_n + J^{-1}(\vec{y}_n)[\beta_1 h_n J(\vec{y}_n) + \beta_2 (h_n J(\vec{y}_n))^2 + \ldots + \beta_m (h_m J(\vec{y}_n))^m]\vec{f}(\vec{y}_n)\ ,$$

hetgeen identiek is met de gegeneraliseerde 1-puntsformule (2.7.33').

Opgaven 2.7

(1) Toon aan dat formule (2.7.33') maximaal 2^e orde consistent kan zijn.

(2) Toon aan dat formule (2.7.33') 1^e orde consistent blijft als $J(\vec{y}_n)$ vervangen wordt door een willekeurige niet-singuliere matrix en R een willekeurige 1^e orde consistente stabiliteitsfunctie is.

2.7.10 Gegeneraliseerde Runge-Kuttaformules van derde orde, de procedures *eferk* en *efsirk*

Voor derde orde consistentie zijn minstens twee functie-evaluaties per integratiestap nodig; beschouw de twee-puntsformule

$$(2.7.36)\qquad \vec{y}_{n+1} = \vec{y}_n + h_n\Theta_0(h_nJ(\vec{y}_n))\vec{f}(\vec{y}_n) + h_n\Theta_1(h_nJ(\vec{y}_n))\vec{f}(\vec{y}_n+h_n\Lambda(h_nJ(\vec{y}_n))\vec{f}(\vec{y}_n))$$

met de stabiliteitsfunctie

$$1 + z\Theta_0(z) + z\Theta_1(z)(1+z\Lambda(z)) .$$

Door deze functie te identificeren met een gegeven functie R kan de coefficientfunctie Θ_0 uitgedrukt worden in R, Θ_1 en Λ:

$$(2.7.37)\qquad \Theta_0(z) = \frac{R(z)-1}{z} - \Theta_1(z) - z\Theta_1(z)\Lambda(z) .$$

Wanneer R consistent van de orde 3 is, dan volgt uit deze relatie

$$\theta_0 = 1 - \theta_1 , \quad \theta_0' = \frac{1}{2} - \theta_1' - \theta_1\lambda , \quad \theta_0'' = \frac{1}{3} - \theta_1'' - z(\theta_1\lambda)' .$$

Vervolgens substitueren deze uitdrukkingen voor de parameters θ_0, θ_0' en θ_0'' in de consistentievoorwaarden uit tabel 2.4.4 met $p \geq 3$. Het is eenvoudig na te gaan dat aan alle consistentievoorwaarden voldaan is behalve de vierde relatie:

$$(2.7.38)\qquad 3\theta_1\lambda^2 = 1 .$$

Dus de relaties (2.7.37) en (2.7.38) maken formule (2.7.36) *derde orde consistent* mits R ook derde orde consistent is.

Van de vele formules gedefinieerd door (2.7.36)-(2.7.38) is de deelklasse gedefinieerd door de relatie

$$(2.7.39)\qquad 2\theta_1\lambda = 1$$

van bijzondere interesse. Men kan namelijk bewijzen (zie van der Houwen [1975]) dat deze deelklasse van formules niet alle betekenis verliezen wanneer de Jacobiaan $J(\vec{y}_n)$ onnauwkeurig bepaald wordt maar nog altijd 2^e orde consistent blijft hoe slecht de benadering van $J(\vec{y}_n)$ ook is.
Uit (2.7.38) en (2.7.39) vinden we

$$\theta_1 = \frac{3}{4} \quad , \quad \lambda = \frac{2}{3} .$$

Het is eenvoudig te verifieren dat de coefficientfuncties

$$(2.7.40) \qquad \theta_0(z) = \frac{1}{4} \ , \ \theta_1(z) = \frac{3}{4} \ , \ \Lambda(z) = \frac{4}{3}\frac{R(z)-1-z}{z^2}$$

aan alle voorwaarden voldoen zodat we uiteindelijk afgeleid hebben de formule

$$(2.7.40') \qquad \vec{y}_{n+1} = \vec{y}_n + \frac{1}{4}h_n\vec{f}(\vec{y}_n) + \frac{3}{4}h_n\vec{f}(\vec{y}_n + \frac{4}{3h_n}[J^*]^{-2}[R(h_nJ^*)-I-h_nJ^*]\vec{f}(\vec{y}_n)).$$

Deze formule is 2^e orde consistent als R 2^e orde consistent is en 3^e orde consistent als R dat ook is en $J^* = J(\vec{y}_n)$.

De procedure *eferk*

In de NUMAL-bibliotheek is onder de naam *eferk* de algorithme (2.7.40) aanwezig met stabiliteitsfunctie

$$(2.7.41) \qquad R(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{(z_2-z_1)(F(z_2)+F(z_1))-(z_2+z_1)(F(z_2)-F(z_1))}{2(z_2-z_1)}z^4 + \frac{F(z_2)-F(z_1)}{z_2-z_1}z^5 ,$$

waarin

$$F(z) = \frac{e^z-1-z-\frac{1}{2}z^2-\frac{1}{6}z^3}{z^4} .$$

Deze stabiliteitsfunctie heeft de eigenschap dat

$$R(z_i) = \exp(z_j) \quad , \quad j = 1,2;$$

met andere woorden R is exponentieel gefit in de punten z_1 en z_2.

We zullen het stabiliteitsgebied van dit polynoom afleiden voor grote waarden van $|z_1|$ en $|z_2|$. Uit de definitie van F(z) volgt direct dat

$$F(z) \sim -\frac{1}{6z} \quad \text{als Re } z \to -\infty ,$$

zodat

$$(2.7.42) \qquad R(z) \sim 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 - \frac{1}{6}\frac{z_1+z_2}{z_1z_2}z^4 + \frac{1}{6}\frac{1}{z_1z_2}z^5 .$$

Het stabiliteitsgebied in de buurt van de oorsprong zal dus ongeveer gelijk zijn aan dat van het polynoom (zie tabel 2.7.1)

$$1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 .$$

In de buurt van z_1 en z_2 moet R(z) echter ook nog stabiliteitsgebieden hebben omdat $R(z_j) \sim 0$ als Re $z_j \to -\infty$. Blijkbaar is R(z) te schrijven in de vorm

$$R(z) \sim (1+z+\frac{1}{2}z^2+\frac{1}{6}z^3)(\frac{z_1-z}{z_1})(\frac{z_2-z}{z_2}) ,$$

zodat in de omgeving van z_j

$$(2.7.42') \qquad R(z) \sim \frac{1}{6}z_j^3\frac{z_2-z}{z_2}\cdot\frac{z_2-z}{z_2} \quad \text{als} \quad \text{Re } z_1 \to -\infty , \quad \text{Re } z_2 \to -\infty .$$

Indien $z_1 \neq z_2$ dan volgt hieruit dat de stabiliteitsgebieden in de omgeving van z_1 en z_2 respectievelijk gegeven worden door

$$(2.7.43a) \qquad |z_1-z| < \frac{6|z_2|}{|z_1|^2|z_2-z_1|}$$

en

$$(2.7.43b) \qquad |z_2-z| < \frac{6|z_1|}{|z_2|^2|z_1-z_2|} .$$

Indien $z_1 \to z_2$ vinden we het stabiliteitsgebied

$$(2.7.44) \qquad |z_1-z| < \sqrt{\frac{6}{|z_1|}} .$$

Schrijven we tenslotte $z_1 = h_n\delta_1$ en $z_2 = h_n\delta_2$, waarin δ_1 en δ_2 de eigenwaarden zijn van de eigenvectoren die men goed wil representeren, dan verkrijgen we de stabiliteitsvoorwaarden:

$$h_n < \sqrt[3]{\frac{6|\delta_1\delta_2|}{|\delta_2-\delta_1|}} \cdot \min\left\{\frac{1}{|\delta_1|\sqrt[3]{\rho_1}}, \frac{1}{|\delta_2|\sqrt[3]{\rho_2}}\right\}$$

en

$$h_n < \sqrt[3]{6\left|\frac{\delta_1}{\rho_1}\right|} \cdot \frac{1}{|\delta_1|}$$

voor $z_1 \neq z_2$ respectievelijk $z_1 \to z_2$.

Hieronder volgen de gebruiksaanwijzingen van de procedure *eferk*; deze zijn overgenomen uit de NUMAL-manual, section 5.2.1.1.1.2. De parameterlijst is nagenoeg gelijk aan die van de *liniger*-procedures.

AUTHOR: K.DEKKER.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 1973/07/31.

BRIEF DESCRIPTION:

EFERK SOLVES INITIAL VALUE PROBLEMS , GIVEN AS AN AUTONOMOUS SYSTEM OF FIRST ORDER DIFFERENTIAL EQUATIONS, BY MEANS OF AN EXPONENTIALLY FITTED, EXPLICIT RUNGE KUTTA METHOD OF THIRD ORDER, WHICH INVOLVES THE USE OF THE JACOBIAN MATRIX. AUTOMATIC STEP CONTROL IS PROVIDED. IN PARTICULAR THIS METHOD IS SUITABLE FOR THE INTEGRATION OF STIFF DIFFERENTIAL EQUATIONS.

KEYWORDS:

DIFFERENTIAL EQUATIONS,
INITIAL VALUE PROBLEMS,
STIFF EQUATIONS,
EXPONENTIAL FITTING,
EXPLICIT RUNGE KUTTA METHODS.

```
CALLING SEQUENCE:

    THE HEADING OF THE PROCEDURE EFERK READS:
    "PROCEDURE" EFERK(X,XE,M,Y,SIGMA,PHI,DERIVATIVE,J,JACOBIAN,
                      K,L,AUT,AETA,RETA,HMIN,HMAX,LINEAR,OUTPUT);
    "VALUE" L;
    "INTEGER" M,K,L;
    "REAL" X,XE,SIGMA,PHI,AETA,RETA,HMIN,HMAX;
    "ARRAY" Y,J;
    "BOOLEAN" AUT,LINEAR;
    "PROCEDURE" DERIVATIVE,JACOBIAN,OUTPUT;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    X:      <VARIABLE>;
            THE INDEPENDENT VARIABLE X;
            CAN BE USED IN DERIVATIVE, JACOBIAN, OUTPUT, ETC.;
            ENTRY: THE INITIAL VALUE X0;
            EXIT : THE FINAL VALUE XE;
    XE:     <ARITHMETIC EXPRESSION>;
            THE FINAL VALUE OF X (XE>=X);
    M:      <ARITHMETIC EXPRESSION>;
            THE NUMBER OF EQUATIONS;
    Y:      <ARRAY IDENTIFIER>;
            "REAL" "ARRAY" Y[1:M];
            THE DEPENDENT VARIABLE;
            ENTRY: THE INITIAL VALUES OF THE SYSTEM OF DIFFERENTIAL
                   EQUATIONS: Y[I] AT X=X0;
            EXIT : THE FINAL VALUES OF THE SOLUTION: Y[I] AT X=XE;
    SIGMA:  <ARITHMETIC EXPRESSION>;
            THE MODULUS OF THE POINT AT WHICH EXPONENTIAL FITTING IS
            DESIRED, FOR EXAMPLE THE LARGEST NEGATIVE EIGENVALUE OF THE
            JACOBIAN MATRIX OF THE SYSTEM OF DIFFERENTIAL EQUATIONS;
    PHI:    <ARITHMETIC EXPRESSION>;
            THE ARGUMENT OF THE COMPLEX POINT AT WHICH EXPONENTIAL
            FITTING IS DESIRED;
    DERIVATIVE: <PROCEDURE IDENTIFIER>;
            THE HEADING OF THIS PROCEDURE READS;
            "PROCEDURE" DERIVATIVE(Y); "ARRAY" Y;
            THIS PROCEDURE SHOULD DELIVER THE RIGHT HAND SIDE OF THE
            I=TH DIFFERENTIAL EQUATION AT THE POINT (Y) AS Y[I];
    J:      <ARRAY IDENTIFIER>;
            "REAL" "ARRAY" J[1:M,1:M];
            THE JACOBIAN MATRIX OF THE SYSTEM;
            THE ARRAY J SHOULD BE UPDATED IN THE PROCEDURE JACOBIAN;
```

```
JACOBIAN: <PROCEDURE IDENTIFIER>;
          THE HEADING OF THIS PROCEDURE READS;
          "PROCEDURE" JACOBIAN(J,Y); "ARRAY" J,Y;
          IN THIS PROCEDURE  THE JACOBIAN AT THE POINT (Y)  HAS TO BE
          ASSIGNED TO THE ARRAY J;
K:        <VARIABLE>;
          COUNTS THE NUMBER OF INTEGRATION STEPS TAKEN;
          FOR EXAMPLE, MAY BE USED IN THE EXPRESSION FOR XE;
L:        <ARITHMETIC EXPRESSION>;
          ENTRY;
          IF PHI = 4*ARCTAN(1): THE ORDER OF THE EXPONENTIAL FITTING,
          ELSE TWICE THE ORDER OF THE EXPONENTIAL FITTING;
AUT:      <BOOLEAN EXPRESSION>;
          IF THE SYSTEM HAS BEEN WRITTEN IN AUTONOMOUS FORM BY ADDING
          THE EQUATION DY[M]/DX = 1 TO THE SYSTEM , THEN AUT MAY HAVE
          THE VALUE "FALSE", ELSE AUT SHOULD HAVE THE VALUE "TRUE";
AETA:     <ARITHMETIC EXPRESSION>;
          REQUIRED  ABSOLUTE  PRECISION  IN THE  INTEGRATION PROCESS;
          AETA HAS TO BE POSITIVE;
RETA:     <ARITHMETIC EXPRESSION>;
          REQUIRED  RELATIVE  PRECISION  IN THE  INTEGRATION PROCESS;
          RETA HAS TO BE POSITIVE;
HMIN:     <ARITHMETIC EXPRESSION>;
          THE STEPLENGTH CHOSEN WILL BE AT LEAST EQUAL TO HMIN;
HMAX:     <ARITHMETIC EXPRESSION>;
          THE STEPLENGTH CHOSEN WILL BE AT MOST EQUAL TO HMAX;
LINEAR:   <ARITHMETIC EXPRESSION>;
          THE PROCEDURE JACOBIAN IS CALLED  ONLY IF LINEAR="FALSE" OR
          K=0; SO IF THE SYSTEM IS LINEAR , LINEAR MAY HAVE THE VALUE
          "TRUE";
OUTPUT:   <PROCEDURE IDENTIFIER>;
          THE HEADING OF THIS PROCEDURE READS;
          "PROCEDURE" OUTPUT;
          THIS  PROCEDURE  IS  CALLED  AT THE END OF EACH INTEGRATION
          STEP , THE USER CAN ASK FOR OUTPUT OF SOME PARAMETERS , FOR
          EXAMPLE X, K, Y.
```

DATA AND RESULTS: SEE EXAMPLE OF USE, AND REF[4].

PROCEDURES USED:

```
VECVEC= CP34010,
MATVEC= CP34011,
DEC   = CP34300,
SOL   = CP34051.
```

REQUIRED CENTRAL MEMORY:
EXECUTION FIELD LENGTH: CIRCA 30 + 4 * M + L * (5+L).

RUNNING TIME: DEPENDS STRONGLY ON THE DIFFERENTIAL EQUATION TO SOLVE.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE PROCEDURE EFERK IS AN EXPONENTIALLY FITTED, SEMI-EXPLICIT RUNGE KUTTA METHOD OF THIRD ORDER (SEE REF [1] AND [3]) . THE ALGORITHM USES FOR EACH STEP TWO FUNCTION EVALUATIONS AND IF LINEAR = "FALSE" ONE EVALUATION OF THE JACOBIAN MATRIX.THE STEPSIZE IS DETERMINED BY AN ESTIMATION OF THE LOCAL TRUNCATION ERROR BASED ON THE RESIDUAL FUNCTION (SEE REF[3]). INTEGRATION STEPS ARE NOT REJECTED.

REFERENCES:

[1]. P.J.VAN DER HOUWEN,
ONE-STEP METHODS WITH ADAPTIVE STABILITY FUNCTIONS FOR THE INTEGRATION OF DIFFERENTIAL EQUATIONS,
LECTURES NOTES OF THE CONFERENCE ON NUMERISCHE, INSBESONDERE APPROXIMATIONSTHEORETISCHE BEHANDLUNG VON FUNKTIONAL-GLEICHUNGEN,
OBERWOLFACH, DECEMBER, 3 -12, 1972.

[2]. T.J.DEKKER, P.W.HEMKER AND P.J.VAN DER HOUWEN,
COLLOQUIUM STIFF DIFFERENTIAL EQUATIONS 1 (DUTCH),
MC SYLLABUS 15.1, (1972) MATHEMATICAL CENTRE,

[3]. P.A.BEENTJES, K.DEKKER, H.C.HEMKER, S.P.N.VAN KAMPEN AND G.M.WILLEMS,
COLLOQUIUM STIFF DIFFERENTIAL EQUATIONS 2 (DUTCH),
MC SYLLABUS 15.2, (1973) MATHEMATICAL CENTRE,

[4]. (TO APPEAR),
COLLOQUIUM STIFF DIFFERENTIAL EQUATIONS 3 (DUTCH),
MC SYLLABUS 15.3, (1973) MATHEMATICAL CENTRE.

EXAMPLE OF USE:

CONSIDER THE SYSTEM OF DIFFERENTIAL EQUATIONS:
DY[1]/DX = -Y[1] + Y[1] * Y[2] + .99 * Y[2]
DY[2]/DX = -1000 * (-Y[1] + Y[1] * Y[2] + Y[2])
WITH THE INITIAL CONDITIONS AT X = 0:
Y[1] = 1 AND Y[2] = 0, (SEE REF[2], PAGE 11),
THE SOLUTION AT X = 50 IS APPROXIMATELY:
Y[1] = .765 878 320 487 AND Y[2] = .433 710 353 5768.
THE FOLLOWING PROGRAM SHOWS SOME DIFFERENT CALLS OF THE PROCEDURE EFERK,AND ILLUSTRATES THE ACCURACIES WHICH MAY BE OBTAINED BY THEM:

```
"BEGIN"
    "PROCEDURE" EFERK(X,XE,M,Y,SIGMA,PHI,DERIVATIVE,J,JACOBIAN,
                  K,L,AUT,AETA,RETA,HMIN,HMAX,LINEAR,OUTPUT);
    "CODE" 33120;

    "INTEGER" K,PASSES,PASJAC;
    "REAL" X,SIGMA,PHI,TIME,TOL;
    "REAL" "ARRAY" Y[1:2],J[1:2,1:2];

    "PROCEDURE" DER(Y); "ARRAY" Y;
    "BEGIN" "REAL" Y1,Y2; Y1:=Y[1]; Y2:=Y[2];
        Y[1]:=(Y1+.99)*(Y2-1)+.99;
        Y[2]:=1000*((1+Y1)*(1-Y2)-1);
        PASSES:=PASSES+1
   "END";

    "PROCEDURE" JACOBIAN(J,Y); "ARRAY" J,Y;
    "BEGIN" J[1,1]:=Y[2]-1; J[1,2]:=.99+Y[1];
        J[2,1]:=1000*(1-Y[2]); J[2,2]:=-1000*(1+Y[1]);
        SIGMA:=ABS(J[2,2]+J[1,1]-SQRT((J[2,2]-J[1,1])**2+
                4*J[2,1]*J[1,2]))/2;
        PASJAC:=PASJAC+1
    "END" JACOBIAN;

    "PROCEDURE" OUT;
    "IF" X=50 "THEN"
   OUTPUT(61,"("3(-5ZD),2(4B+.3DB3DB3D),-5ZD.3D,/")",K,PASSES,
    PASJAC,Y[1],Y[2],CLOCK-TIME);

    OUTPUT(61,"(""("    THIS LINE AND THE FOLLOWING TEXT IS ")"
    "("PRINTED BY THIS PROGRAM")",//,
    "("    THE RESULTS WITH EFERK -FIRST ORDER FIT- ARE:")",/,
    "("     K   DER.EV. JAC.EV.       Y[1]              Y[2]")"
    "("         TIME")",/")");
    PHI:=4*ARCTAN(1);
    "FOR" TOL:=1,"-1,"-2,"-3 "DO"
    "BEGIN" PASSES:=PASJAC:=0; X:=Y[2]:=0; Y[1]:=1; TIME:=CLOCK;
        EFERK(X,50,2,Y,SIGMA,PHI,DER,J,JACOBIAN,K,1,"TRUE",TOL,
                                          TOL,"-6,50,"FALSE",OUT);
    "END";
"END";
```

```
THIS LINE AND THE FOLLOWING TEXT IS PRINTED BY THIS PROGRAM:

THE RESULTS WITH EFERK -FIRST ORDER FIT- ARE:
  K   DER.EV. JAC.EV.       Y[1]              Y[2]          TIME
 93     186      93    +.765 883 211    +.428 752 781      1.170
105     210     105    +.765 878 445    +.433 569 561      1.296
147     294     147    +.765 878 317    +.433 708 489      1.834
266     532     266    +.765 878 320    +.433 710 229      3.297
```

De procedure *efsirk*

Naast de *expliciete* formule gegenereerd door het polynoom (2.7.41) is ook een *semi-impliciete* formule in NUMAL aanwezig: de procedure *efsirk*. Evenals *eferk* is deze algorithme van de vorm (2.7.40) maar nu met een *rationale* stabiliteitsfunctie en wel de functie van Liniger en Willoughby gedefinieerd door (2.7.26) en (2.7.27) met $z_1 = z_2$. De procedure *efsirk* heeft dus dezelfde (lokale) stabiliteitseigenschappen als *liniger 2vs* voor het geval $z_1 = z_2$, maar is in het algemeen veel "goedkoper" per integratiestap (vergelijk tabel 2.7.3). Anderzijds hebben experimenten uitgewezen (zie Beentjes en Dekker [1974]) dat *liniger 2vs* betrouwbaarder is voor sterk niet-lineaire problemen. Voor zwak niet-lineaire problemen is *efsirk* te verkiezen boven *liniger 2vs* te meer ook daar incidentele evaluatie van de Jacobiaan toegestaan is.

We besluiten de discussie van gegeneraliseerde derde orde formules met de gebruiksaanwijzingen van *efsirk*. De parameterlijst bevat geen nieuwe grootheden zodat met een reproductie van de documentatie uit de NUMAL-manual, section 5.2.1.1.1.2 volstaan wordt.

AUTHOR: S.P.N. VAN KAMPEN.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 730529.

BRIEF DESCRIPTION:

EFSIRK SOLVES AN INITIAL VALUE PROBLEM, GIVEN AS AN AUTONOMOUS SYSTEM OF FIRST ORDER DIFFERENTIAL EQUATIONS DY/DX = F(Y), BY MEANS OF AN EXPONENTIALLY FITTED, SEMI-IMPLICIT RUNGE-KUTTA METHOD; IN PARTICULAR THIS PROCEDURE IS SUITABLE FOR THE INTEGRATION OF STIFF EQUATIONS.

KEYWORDS:

DIFFERENTIAL EQUATIONS,
INITIAL VALUE PROBLEM,
AUTONOMOUS SYSTEM,
STIFF EQUATIONS,
SEMI-IMPLICIT RUNGE-KUTTA METHOD,
EXPONENTIAL FITTING.

CALLING SEQUENCE:

```
HEADING:
"PROCEDURE" EFSIRK(X, XE, M, Y, DELTA, DERIVATIVE, JACOBIAN, J,
                   N, AETA, RETA, HMIN, HMAX, LINEAR, OUTPUT);
"VALUE" M; "INTEGER" M, N;
"REAL" X, XE, DELTA, AETA, RETA, HMIN, HMAX;
"PROCEDURE" DERIVATIVE, JACOBIAN, OUTPUT;
"ARRAY" Y, J;
"BOOLEAN" LINEAR;

THE MEANING OF THE FORMAL PARAMETERS IS:
X:      <VARIABLE>;
        THE INDEPENDENT VARIABLE X;
        ENTRY: THE INITIAL VALUE X0;
        EXIT : THE END VALUE XE;
XE:     <ARITHMETIC EXPRESSION>;
        THE END VALUE OF X;
M:      <ARITHMETIC EXPRESSION>;
        THE NUMBER OF DIFFERENTIAL EQUATIONS;
Y:      <ARRAY IDENTIFIER>;
        "ARRAY" Y[1 : M];
        THE DEPENDENT VARIABLE;
        DURING THE INTEGRATION PROCESS THE COMPUTED SOLUTION
        AT THE POINT X IS ASSIGNED TO THE ARRAY Y;
        ENTRY: THE INITIAL VALUES OF THE SOLUTION OF THE SYSTEM;
```

```
DELTA:  <ARITHMETIC EXPRESSION>;
        DELTA DENOTES THE REAL PART OF THE POINT AT WHICH
        EXPONENTIAL FITTING IS DESIRED;
        ALTERNATIVES:
        DELTA = (AN ESTIMATE OF) THE REAL PART OF THE, IN  ABSOLUTE
        VALUE, LARGEST EIGENVALUE OF THE JACOBIAN MATRIX OF THE
        SYSTEM;
        DELTA < -10**14, IN ORDER TO OBTAIN ASYMPTOTIC
        STABILITY;
        DELTA = 0, IN ORDER TO OBTAIN A HIGHER ORDER OF ACCURACY IN
        CASE OF LINEAR OR ALMOST LINEAR EQUATIONS;
DERIVATIVE: <PROCEDURE IDENTIFIER>;
        "PROCEDURE" DERIVATIVE(A); "ARRAY" A;
        WHEN IN EFSIRK DERIVATIVE IS CALLED, A[I] CONTAINS THE
        VALUES OF Y[I];
        UPON COMPLETION OF A CALL OF DERIVATIVE, THE ARRAY A
        SHOULD CONTAIN THE VALUES OF F(Y);
        NOTE THAT THE VARIABLE X SHOULD NOT BE USED IN DERIVATIVE,
        BECAUSE THE DIFFERENTIAL EQUATION IS SUPPOSED TO BE
        AUTONOMOUS;
JACOBIAN: <PROCEDURE IDENTIFIER>;
        "PROCEDURE" JACOBIAN(J, Y); "ARRAY" J, Y;
        WHEN IN EFSIRK JACOBIAN IS CALLED THE ARRAY Y CONTAINS
        THE VALUES OF THE DEPENDENT VARIABLE;
        UPON COMPLETION OF A CALL OF JACOBIAN THE ARRAY J SHOULD
        CONTAIN THE VALUES OF THE JACOBIAN MATRIX OF F(Y);
J:      <ARRAY IDENTIFIER>;
        J[1 : M, 1 : M];
        J IS AN AUXILLIARY ARRAY WHICH IS USED IN THE PROCEDURE
        JACOBIAN;
N:      <VARIABLE>;
        AN INTEGER WHICH COUNTS THE INTEGRATION STEPS;
AETA, RETA:
        <ARITHMETIC EXPRESSION>;
        REQUIRED ABSOLUTE AND RELATIVE LOCAL ACCURACY;
HMIN, HMAX:
        <ARITHMETIC EXPRESSION>;
        MINIMAL  AND MAXIMAL  STEPSIZE BY WHICH  THE INTEGRATION IS
        PERFORMED;
LINEAR: <BOOLEAN EXPRESSION>;
        IF LINEAR = "TRUE" THE PROCEDURE JACOBIAN WILL ONLY BE
        CALLED IF N = 1; THE INTEGRATION WILL THEN BE PERFORMED
        WITH A STEPSIZE HMAX; THE CORRESPONDING REDUCTION
        OF COMPUTING TIME CAN BE EXPLOITED IN CASE OF LINEAR OR
        ALMOST LINEAR EQUATIONS;
OUTPUT: <PROCEDURE IDENTIFIER>;
        "PROCEDURE" OUTPUT;
        IN OUTPUT ONE MAY PRINT THE VALUES OF E,G, X,
        Y[I], J[K, L] AND N.
```

DATA AND RESULTS: SEE REF[2] AND [3].

PROCEDURES USED:

VECVEC = CP34010,
MATVEC = CP34011,
MATMAT = CP34013,
GSSELM = CP34231,
SOLELM = CP34061.

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: CIRCA M * M + 5 * M.

RUNNING TIME:

DEPENDS STRONGLY ON THE DIFFERENTIAL EQUATION TO BE SOLVED

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE PROCEDURE EFSIRK IS AN EXPONENTIALLY FITTED, A-STABLE,
SEMI-IMPLICIT RUNGE-KUTTA METHOD OF THIRD ORDER (SEE REF[1]
AND [2]). THE ALGORITHM USES FOR EACH STEP TWO FUNCTION EVALUATIONS
AND IF LINEAR = "FALSE" ONE EVALUATION OF THE JACOBIAN MATRIX.
THE STEPSIZE IS NOT DETERMINED BY THE ACCURACY OF THE NUMERICAL
SOLUTION, BUT BY THE AMOUNT BY WHICH THE GIVEN DIFFERENTIAL
EQUATION DIFFERS FROM A LINEAR EQUATION (SEE REF[2]).
THE PROCEDURE DOES NOT REJECT INTEGRATION STEPS.

REFERENCES:

[1].P.J. VAN DER HOUWEN,
ONE-STEP METHODS WITH ADAPTIVE STABILITY FUCNTIONS
FOR THE INTEGRATION OF DIFFERENTIAL EQUATIONS,
LECTURE NOTES OF THE CONFERENCE ON
NUMERISCHE, INSBESONDERE APPROXIMATIONSTHEORETISCHE
BEHANDLUNG VON FUNKTIONALGLEICHUNGEN,
OBERWOLFACH, DECEMBER, 3 - 12, 1972.

[2].SYLLABUS COLLOQUIUM STIFF DIFFERENTIAL EQUATIONS 2 (DUTCH),
MATH.CENTR. SYLLABUS 15.2/73.

[3].SYLLABUS COLLOQUIUM STIFF DIFFERENTIAL EQUATIONS 3 (DUTCH),
MATH.CENTR. SYLLABUS 15.3/73,
TO APPEAR IN 1973.

EXAMPLE OF USE:

```
    WE CONSIDER THE DIFFERENTIAL EQUATION
    DY / DX = -EXP(X) * (Y - LN(X)) + 1 / X,
    ON THE INTERVAL  [0,01, 8], WITH  INITIAL VALUE  Y(0,01) = LN(0,01)
    AND ANALYTICAL SOLUTION Y(X) = LN(X);
    FOR THE FIT POINT WE USE THE  EIGENVALUE  OF THE JACOBIAN MATRIX,
    I,E, DELTA = -EXP(X);

    "BEGIN"
        "PROCEDURE" EFSIRK(X, XE, M, Y, DELTA, DERIVATIVE, JACOBIAN, J,
                           N, AETA, RETA, HMIN, HMAX, LINEAR, OUTPUT);
        "CODE" 33160;
        "PROCEDURE" DER(Y); "ARRAY" Y;
        "BEGIN" "REAL" Y2; Y2:= Y[2];
            DELTA:= -EXP(Y2); LNX:= LN(Y2);
            Y[1]:= (Y[1] - LNX) * DELTA + 1 / Y2;
            Y[2]:= 1
        "END" DER;
        "PROCEDURE" JAC(J, Y); "ARRAY" J, Y;
        "BEGIN" "REAL" Y2; Y2:= Y[2];
            J[1, 1]:= DELTA;
            J[1, 2]:= (Y[1] - LNX - 1 / Y2) * DELTA - 1 / (Y2 * Y2);
            J[2, 1]:= J[2, 2]:= 0
        "END" JAC;
        "PROCEDURE" OUTP;
        "IF" X = XE "THEN"
        "BEGIN" "REAL" Y1; Y1:= Y[1]; LNX:= LN(X);
            OUTPUT(61, "("("N = ")", 2ZD,
                       "("   X = ")", +D,D,
                       "("  Y(X) = ")", +D,5D,
                       "("  DELTA = ")", +3ZD,2D, /,
                       "("ABS, ERR, = ")", ,2D"+2D,
                       "("  REL, ERR, = ")", ,2D"+2D, //")",
                       N, X, Y1, DELTA,
                       ABS(Y1 - LNX), ABS((Y1 - LNX) / LNX));
            "IF" X = 0,4 "THEN" XE:= 8
        "END" OUTP;
        "INTEGER" N;
        "REAL" X, XE, DELTA, LNX;
        "ARRAY" Y[1 : 2], J[1 : 2, 1 : 2];

        XE:= 0,4; X:= 0,01; Y[1]:= LN(0,01); Y[2]:= X;
        EFSIRK(X, XE, 2, Y, DELTA, DER, JAC, J,
               N, "-2, "-2, 0,005, 1,5, "FALSE", OUTP)
    "END"
```

THIS PROGRAM DELIVERS:

```
    N =  10    X = +0,4   Y(X) = -0,91099  DELTA =    -1,44
    ABS, ERR, = ,53"-02   REL, ERR, = ,58"-02

    N = 98     X = +8,0   Y(X) = +2,07911  DELTA = -2980,02
    ABS, ERR, = ,33"-03   REL, ERR, = ,16"-03
```

2.7.11 Hogere orde integratieformules, de procedure *rke*

Integratieformules van de 4^e orde met aanpasbare stabiliteitsfunctie zijn wel bekend maar niet aanwezig in de NUMAL-bibliotheek (afgezien van de reeds besproken Taylor-formules, die echter vanwege de vereiste afgeleiden-evaluatie veelal voor de practijk onbruikbaar zijn). Een discussie van 4^e orde Runge-Kuttaformules met adaptief stabiliteitspolynoom en gegeneraliseerde formules van de 4^e orde kan men vinden in van der Houwen [1972]. Formules uit de Runge-Kuttaklasse van de orde 5 en hoger zijn al zó moeilijk te construeren, dat de constructie van dergelijke formules met adaptief stabiliteitspolynoom, zo al mogelijk, vermoedelijk tot onhanteerbare algorithmen zal leiden. (Voor gegeneraliseerde Runge-Kuttaformules geldt hetzelfde zij het in iets mindere mate.) Vijfde orde formules met vast stabiliteitspolynoom zijn op het Mathematisch Centrum ontwikkeld door Zonneveld [1964], door Beentjes en Dekker [1972] en onlangs door Beentjes [1974]. De formule van Zonneveld is in NUMAL beschikbaar onder de naam *rk1n* en gedocumenteerd in section 5.2.1.1.1.1. van de manual. Het ligt in de bedoeling om de door Beentjes ontwikkelde formule ook op te nemen, aangezien bij de eerste experimenten aanzienlijk betere resultaten verkregen werden dan met *rk1n*. We zullen dan ook volstaan met een bespreking van de Beentjes-procedure.

De procedure *rke*

De genererende matrix van *rke* wordt gegeven door het array (2.7.45):

$$
(2.7.45)\qquad
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5-\sqrt{5}}{15} & 0 & 0 & 0 & 0 & 0 \\
\frac{5-\sqrt{5}}{40} & -\frac{15-3\sqrt{5}}{40} & 0 & 0 & 0 & 0 \\
\frac{3}{16} & -\frac{3\sqrt{5}}{16} & \frac{5+3\sqrt{5}}{16} & 0 & 0 & 0 \\
\frac{9+\sqrt{5}}{40} & -\frac{15+3\sqrt{5}}{40} & \frac{5+3\sqrt{5}}{20} & \frac{2}{5} & 0 & 0 \\
-\frac{3}{4} & \frac{3\sqrt{5}}{4} & \frac{5-\sqrt{5}}{4} & -2 & \frac{5-\sqrt{5}}{2} & 0 \\
\frac{1}{12} & 0 & \frac{5}{12} & 0 & \frac{5}{12} & \frac{1}{12}
\end{pmatrix}.
$$

Deze formule behoort tot een klasse van 5^e orde formules welke door England [1969] afgeleid werd. Door minimalisering van de afbreekfout van de formules uit de England-familie kwam Beentjes tot (2.7.45). Het stabiliteitspolynoom wordt gegeven door

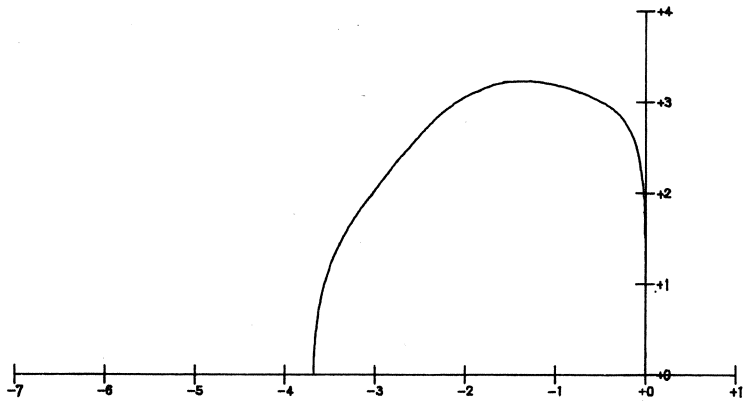
$$(2.7.46) \qquad R(z) = 1 + z + \frac{1}{2} z^2 + \frac{1}{6} z^3 + \frac{1}{24} z^4 + \frac{1}{120} z^5 + \frac{1}{777} z^6 \ ,$$

waaruit de stabiliteitsgrenzen

$$(2.7.47) \qquad \beta_{\text{reëel}} \cong 3.7 \quad , \quad \beta_{\text{imag}} \cong 1.8$$

volgen (zie ook figuur 2.7.5).

De gebruiksaanwijzing uit de NUMAL-manual, section 5.2.1.1.1.1, bevat geen nieuwe aspecten die niet al besproken zijn in voorgaande paragrafen.



Figuur 2.7.5 Stabiliteitsgebied van formule (2.7.45)

AUTHOR: P.A. BEENTJES.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 740520.

BRIEF DESCRIPTION:

RKE INTEGRATES A SYSTEM OF FIRST ORDER DIFFERENTIAL EQUATIONS (INITIAL VALUES BEING GIVEN) BY MEANS OF A FIFTH ORDER EXPLICIT RUNGE KUTTA METHOD.

KEYWORDS:

RUNGE KUTTA METHODS,
DIFFERENTIAL EQUATIONS,
INITIAL VALUE PROBLEMS.

```
CALLING SEQUENCE:

    THE HEADING OF THE PROCEDURE READS:

    "PROCEDURE" RKE (X, XE, N, Y, DER, DATA, FI, OUT);
    "VALUE" N, FI; "INTEGER" N; "REAL" X, XE; "BOOLEAN" FI;
    "ARRAY" Y, DATA;
    "PROCEDURE" DER, OUT;

    THE MEANING OF THE FORMAL PARAMETERS IS:

    X:      <VARIABLE>;
            THE INDEPENDENT VARIABLE;
            ENTRY: THE INITIAL VALUE X0;
    XE:     <ARITHMETIC EXPRESSION>;
            ENTRY: THE FINAL VALUE OF X;
    N:      <ARITHMETIC EXPRESSION>;
            THE NUMBER OF EQUATIONS OF THE SYSTEM;
    Y:      <ARRAY IDENTIFIER>;
            "ARRAY" Y[1 : N];
            THE DEPENDENT VARIABLES;
            ENTRY: THE INITIAL VALUES OF Y AT X = X0;
            EXIT : THE VALUES OF THE SOLUTION AT X = XE;
    DER:    <PROCEDURE IDENTIFIER>;
            THE HEADING OF THIS PROCEDURE READS;
            "PROCEDURE" DER (T, V); "REAL" T; "ARRAY" V;
            THIS PROCEDURE PERFORMS AN EVALUATION OF THE RIGHT HAND
            SIDE OF THE SYSTEM WITH DEPENDENT VARIABLES V[1 : N] AND
            INDEPENDENT VARIABLE T; UPON COMPLETION OF DERIVATIVE,
            THE RIGHT HAND SIDE SHOULD BE OVERWRITTEN ON V[1 : N];
    DATA:   <ARRAY IDENTIFIER>;
            "ARRAY" DATA[1 : 6];
            IN ARRAY DATA ONE SHOULD GIVE:
                DATA[1]: THE RELATIVE TOLERANCE;
                DATA[2]: THE ABSOLUTE TOLERANCE;
            AFTER EACH STEP THE FOLLOWING BY-PRODUCTS ARE DELIVERED:
                DATA[3]: THE STEPLENGTH USED FOR THE LAST STEP;
                DATA[4]: THE NUMBER OF INTEGRATION STEPS PERFORMED;
                DATA[5]: THE NUMBER OF INTEGRATION STEPS REJECTED;
                DATA[6]: THE NUMBER OF INTEGRATION STEPS SKIPPED;
                         IF UPON COMPLETION OF RKE DATA[6] > 0 ,
                         RESULTS SHOULD BE CONSIDERED MOST CRITICALLY;
    FI:     <BOOLEAN EXPRESSION>;
            IF FI = "TRUE" THE INTEGRATION STARTS AT X0 WITH A
            TRIAL STEP XE - X0; IF FI = "FALSE" THE INTEGRATION IS
            CONTINUED WITH A STEPLENGTH DATA[3] * SIGN(XE - X0);
    OUT:    <PROCEDURE IDENTIFIER>;
            THE HEADING OF THIS PROCEDURE READS:
            "PROCEDURE" OUT;
            AFTER EACH INTEGRATION STEP PERFORMED, INFORMATION CAN BE
            OBTAINED OR UPDATED BY THIS PROCEDURE, E.G. THE VALUES OF
            X, Y[1 : N] AND DATA[3 : 6].
```

DATA AND RESULTS:

SEE REFERENCES [1] AND [3].

PROCEDURES USED:

DUPVEC = CP31030.

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: CIRCA 5 * N.

RUNNING TIME:

DEPENDS STRONGLY ON THE SYSTEM OF DIFFERENTIAL EQUATIONS TO BE SOLVED.

LANGUAGE: ALGOL60.

METHOD AND PERFORMANCE:

THE SCHEME UPON WHICH THE METHOD IS BASED, IS A MEMBER OF A CLASS OF FIFTH ORDER RUNGE KUTTA FORMULAS PRESENTED IN REFERENCE [1].
AUTOMATIC STEPSIZE CONTROL IS IMPLEMENTED IN A WAY AS PROPOSED IN REFERENCE [2].
FOR TESTRESULTS AND FURTHER INFORMATION SEE REFERENCE [3].

REFERENCES:

[1]. R. ENGLAND,
ERROR ESTIMATES FOR RUNGE KUTTA TYPE SOLUTIONS TO SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS,
THE COMPUTER JOURNAL , VOLUME 12, P 166 - 169, 1969.

[2]. J.A. ZONNEVELD,
AUTOMATIC NUMERICAL INTEGRATION,
MATH. CENTRE TRACT 8(1970).

[3]. P.A. BEENTJES,
SOME SPECIAL FORMULAS OF THE ENGLAND-CLASS OF FIFTH ORDER RUNGE KUTTA SCHEMES,
(TO APPEAR).

EXAMPLE OF USE:

THE SOLUTION AT T = 1 AND T = -1 OF THE SYSTEM

DX / DT = Y - Z,
DY / DT = X * X + 2 * Y + 4 * T,
DZ / DX = X * X + 5 * X + Z * Z + 4 * T,

WITH X = Y = 0 AND Z = 2 AT T = 0,

CAN BE OBTAINED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "REAL" T, TE; "ARRAY" Y[1 : 3], DATA[1 : 6];

   "PROCEDURE" RKE(X, XE, N, Y, DER, DATA, FI, OUT);
   "CODE" 33033;

   "PROCEDURE" RHS(T, Y); "VALUE" T; "REAL" T; "ARRAY" Y;
   "BEGIN" "REAL" XX, YY, ZZ;
      XX:= Y[1]; YY:= Y[2]; ZZ:= Y[3];
      Y[1]:= YY - ZZ;
      Y[2]:= XX * XX + 2 * YY + 4 * T;
      Y[3]:= XX * (XX + 5) + 2 * ZZ + 4 * T
   "END" RHS;

   "PROCEDURE" INFO;
   "IF" T = TE "THEN"
   "BEGIN" "REAL" ET, T2, AEX, AEY, AEZ, REX, REY, REZ;
      ET:= EXP(T); T2:= 2 * T;
      REX:= -ET * SIN(T2); AEX:= REX - Y[1]; REX:= ABS(AEX / REX);
      REY:= ET * ET * (8 + 2 * T2 - SIN(2 * T2)) / 8 - T2 - 1;
      REZ:= ET * (SIN(T2) + 2 * COS(T2)) + REY;
      AEY:= REY - Y[2]; REY:= ABS(AEY / REY); AEZ:= REZ - Y[3];
      REZ:= ABS(AEZ / REZ);
      OUTPUT(61, "("")(" T = ")", +D, //,
      "(" RELATIVE AND ABSOLUTE ERRORS IN X, Y AND Z :")", //,
      "("   RE(X)   RE(Y)   RE(Z)   AE(X)   AE(Y)   AE(Z)")", //,
      6(B,-,2D"+D), //,
      "(" NUMBER OF INTEGRATION STEPS PERFORMED :")",4ZD,/,
      "(" NUMBER OF INTEGRATION STEPS SKIPPED   :")",4ZD,/,
      "(" NUMBER OF INTEGRATION STEPS REJECTED  :")",4ZD,///")",
      T, REX, REY, REZ, ABS(AEX), ABS(AEY), ABS(AEZ),
      DATA[4], DATA[6], DATA[5])
   "END" INFO;

   TE:= 1;
LEFT:
   Y[1]:= Y[2]:= 0; Y[3]:= 2; T:=0;
   DATA[1]:= DATA[2]:= "-5;
   RKE(T, TE, 3, Y, RHS, DATA, "TRUE", INFO);
   "IF" TE = 1 "THEN" "BEGIN" TE:= -1; "GOTO" LEFT "END"
"END"
```

```
THIS PROGRAM DELIVERS:

T = +1

RELATIVE AND ABSOLUTE ERRORS IN X, Y AND Z :

  RE(X)    RE(Y)    RE(Z)    AE(X)    AE(Y)    AE(Z)

 ,37"-6   ,15"-5   ,13"-5   ,91"-6   ,13"-4   ,11"-4

NUMBER OF INTEGRATION STEPS PERFORMED :    9
NUMBER OF INTEGRATION STEPS SKIPPED   :    0
NUMBER OF INTEGRATION STEPS REJECTED  :    5


T = -1

RELATIVE AND ABSOLUTE ERRORS IN X, Y AND Z :

  RE(X)    RE(Y)    RE(Z)    AE(X)    AE(Y)    AE(Z)

 ,22"-6   ,52"-7   ,19"-6   ,75"-7   ,55"-7   ,77"-7

NUMBER OF INTEGRATION STEPS PERFORMED :   10
NUMBER OF INTEGRATION STEPS SKIPPED   :    0
NUMBER OF INTEGRATION STEPS REJECTED  :    7
```

2.7.12 Conclusies

We hebben in de vorige paragrafen een twaalftal integratie-technieken besproken; we zullen nu de voornaamste kenmerken samenvatten. In tabel 2.7.4 zijn de waarden opgenomen van de volgende grootheden:

p	:	orde van nauwkeurigheid;
f.ev.	:	aantal evaluaties van de rechterlidfunctie per integratiestap;
a.ev.	:	aantal evaluaties van de afgeleide(n) van de rechterlidfunctie; in het geval van *modified taylor* zijn dit successieve afgeleiden en in het geval van *liniger 2vs* de eerste afgeleide per ingetratiestap;
J.ev.	:	aantal Jacobiaan-evaluaties;
LU-dec.	:	aantal LU-decomposities;
g.r.	:	vereiste geheugenruimte in woorden voor grote waarden van r (r>100);

De procedures *liniger** *1vs* en *2vs*, *eferk** en *efsirk** zijn "afgeleide" procedures in de zin dat de Jacobiaan minder vaak bepaald wordt als in de oorspronkelijke procedures; evenzo is *modified taylor** afgeleid van *modified taylor* door de successieve differentiatie te vervangen door vermenigvuldiging met de Jacobiaan.

Tabel 2.7.4 Overzicht van de belangrijkste kenmerken van een aantal procedures uit NUMAL

	p	f.ev.	a.ev.	J.ev.	LU-dec.	g.r.
modified taylor	≤m	1	m-1	0	0	2r
liniger 2vs	≤3	m	m	m	m	$5r^2$
liniger 2vs*	≤3	m	m	<m	<m	$5r^2$
ark	≤3	m	0	0	0	3r
rke	5	6	0	0	0	
*modified taylor**	≤2	1	0	1	0	$2r \leq g.r. \leq r^2$
liniger 1vs	1	m	0	m	m	$4r^2$
liniger 1vs*	1	m	0	<m	<m	$4r^2$
eferk	3	2	0	1	0	r^2
*eferk**	2	2	0	<1	0	r^2
efsirk	3	2	0	1	1	$2r^2$
*efsirk**	2	2	0	<1	<1	$2r^2$

Tabel 2.7.5 Aanbeveling (onder voorbehoud) van procedures afhankelijk van beschikbare informatie
(⌡ = niet beschikbaar; ⌊ = beschikbaar

partiele d.v.	⌡	afg.r.l.f.			*ark*
	⌊	jac.r.l.f.			*modified taylor**
	⌊	afg.r.l.f.			*modified taylor*
stijve d.v.	⌊	jac.r.l.f.	⌡ Δ_n	sterk niet-lin.	*liniger 1vs, 2vs*
				zwak niet-lin.	*efsirk*
			⌊ Δ_n	sterk niet-lin.	*liniger 1vs, 2vs*
				zwak niet-lin.	*efsirk, efsirk**
					*eferk, eferk**
					liniger 1vs, 2vs*
niet-stijve d.v.	⌡	afg.r.l.f.			*rke*
	⌊	afg.r.l.f.			*modified taylor*

2.8. STAPKEUZESTRATEGIEN

2.8.1 Discrepantiefuncties

Wat men zou wensen bij numerieke integratie, is de controle over de grootte van de globale discretiseringsfout $\vec{\varepsilon}_n$. Volgens stelling 2.5.1 geldt voor $\vec{\varepsilon}_n$ de ongelijkheid

$$\|\vec{\varepsilon}_{n+1}\| \leq L_n \|\varepsilon_n\| + \|\vec{y}(x_{n+1}) - E_n(\vec{y}(x_n))\| , \tag{2.8.1}$$

waarin L_n een Lipschitz-constante is voor de operator E_n. Stel nu dat het mogelijk is de afbreekfout in het punt $(x_n, \vec{y}(x_n))$, dat wil zeggen de vector

$$\vec{y}(x_{n+1}) - E_n(\vec{y}(x_n)) , \tag{2.8.2}$$

te berekenen als functie van h_n en stel dat men de integratiestap h_n zo kiest dat

$$\|\vec{y}(x_{n+1}) - E_n(y(x_n))\| = \eta_n , \tag{2.8.3}$$

waarin η_n een gegeven tolerantie is; laten we het geval beschouwen waarin de toegestane afbreekfout per eenheid van interval constant is, dus

$$\eta_n = \tau h_n ,$$

dan geldt

$$\|\vec{\varepsilon}_{n+1}\| \leq \left[h_n + \sum_{j=1}^{n} h_{j-1} \prod_{\nu=j}^{n} L_\nu \right] \tau .$$

Evenals bij het bewijs van de convergentiestelling 2.5.2 onderscheiden we de gevallen $L_\nu \leq 1$ en $L_\nu \leq 1 + ch_\nu$ (c uniform begrensde constante). We vinden dan (vergelijk het bewijs van stelling 2.5.2)

$$\|\vec{\varepsilon}_{n+1}\| \leq \begin{cases} (x_{n+1}-x_0)\,\tau & \text{als } L_\nu \leq 1 \\ \dfrac{\exp[c(x_{n+1}-x_0)]-1}{c}\,\tau & \text{als } L_\nu > 1 \end{cases} . \tag{2.8.4}$$

Op grond van deze relatie zou men de nauwkeurigheid van de numerieke oplossing kunnen sturen door in (2.8.3) voor $\eta_n = h_n\tau$ een geschikte tolerantie te kiezen en vervolgens de bijbehorende integratiestap h_n uit te rekenen. In de practijk kent men de afbreekfout (2.8.2) echter niet, maar in vele gevallen is een goede benadering voor deze afbreekfout de lokale discretiseringsfout $\vec{\rho}_n$, ofwel de afbreekfout in het punt $(x_n,\vec{y}_n)$; in principe is $\vec{\rho}_n$ te benaderen door met een referentieformule een referentieoplossing $\tilde{\vec{y}}_{n+1}$ te bepalen, die nauwkeuriger is dan de numerieke oplossing $\vec{y}_{n+1}$, zodat geldt

$$\vec{\rho}_n \cong \tilde{\vec{y}}_{n+1} - \vec{y}_{n+1} \ . \tag{2.8.5}$$

Soms is het inderdaad mogelijk, zonder veel extra rekenwerk, zo'n referentieoplossing uit te rekenen (bijvoorbeeld, bij gestabiliseerde Taylorformules); vraagt de referentieoplossing echter een substantiele hoeveelheid extra rekenwerk dan stelt men zich vaak tevreden met een referentieoplossing van gelijke of zelfs lagere orde van nauwkeurigheid (*ingebedde* referentieformules).

Bovenstaande is gebaseerd op de aanname dat $\vec{\rho}_n$ dezelfde orde van grootte heeft als de afbreekfout (2.8.2). Met name voor stijve differentiaalvergelijkingen geldt deze aanname echter niet omdat de afgeleiden van de analytische oplossing in de asymptotische phase totaal verschillen van de afgeleiden van de lokaal analytische oplossing door $(x_n,\vec{y}_n)$; in het algemeen zal de lokale discretiseringsfout dan een wat pessimistische benadering voor (2.8.2) zijn. Wat echter nog erger is, de benadering van $\vec{\rho}_n$ door middel van referentieoplossingen is op zich zelf al zeer dubieus voor stijve differentiaalvergelijkingen, zodat naar andere discrepantie-functies uitgekeken moet worden. Een alternatief is om in plaats van de lokaal *analytische oplossing* $\vec{z} = \vec{z}(x_n,\vec{y}_n;x)$ te substitueren in de integratieformule (hetgeen $\vec{\rho}_n$ oplevert), de lokale *differentieoplossing* $\vec{w} = \vec{w}(x_n,\vec{y}_n;x)$ te substitueren in de differentiaalvergelijking en het residu

$$\vec{\zeta}_n = \vec{f}(\vec{y}_{n+1}) - \frac{d}{dx}\vec{w}(x_n,\vec{y}_n;x)\Big|_{x=x_{n+1}} \tag{2.8.6}$$

als maat voor de nauwkeurigheid te nemen.

Het verband tussen $\vec{\rho}_n$ en $\vec{\zeta}_n$ wordt gegeven door de volgende stelling

Stelling 2.8.1

Laat p de orde van consistentie zijn dan geldt

$$\vec{\rho}_n = \int_0^{h_n} [\vec{f}(\vec{w}(x_n,\vec{y}_n;x_n+h) - \frac{d}{dh}\vec{w}(x_n,\vec{y}_n;x_n+h)]dh +$$

$$+ O(h_n^{p+2}) \quad \text{als} \quad h_n \to 0.$$

Bewijs

De lokale discretiseringsfout kan als volgt herleid worden:

$$\vec{\rho}_n = \vec{z}(x_n,\vec{y}_n;x_{n+1}) - \vec{w}(x_n,\vec{y}_n;x_{n+1}) =$$

$$= [\vec{z}(x_n,\vec{y}_n;x_{n+1}) - \vec{z}(x_n,\vec{y}_n;x_n)] + [\vec{z}(x_n,\vec{y}_n;x_n) - \vec{w}(x_n,\vec{y}_n,x_{n+1})] =$$

$$= \int_{x_n}^{x_{n+1}} \vec{z}'(x_n,\vec{y}_n;x)dx - \int_{x_n}^{x_{n+1}} \vec{w}'(x_n,\vec{y}_n;x)dx =$$

$$= \int_{x_n}^{x_{n+1}} [\vec{z}'(x_n,\vec{y}_n,x) - \vec{f}(\vec{w}(x_n,\vec{y}_n;x))]dx +$$

$$+ \int_{x_n}^{x_{n+1}} [\vec{f}(\vec{w}(x_n,\vec{y}_n;x)) - \vec{w}'(x_n,\vec{y}_n;x)]dx .$$

De eerste integraal in deze uitdrukking wordt begrensd door

$$h_n M_n \underset{x_n \le x \le x_{n+1}}{\text{maximum}} \|\vec{z}(x_n,\vec{y}_n;x) - \vec{w}(x_n,\vec{y}_n;x)\| ,$$

waarin M_n weer de Lipschitz-constante voor $\vec{f}$ is. Deze bovengrens is duidelijk van de orde p + 2 in h_n waarmee de stelling bewezen is.

Uit deze stelling volgt dat voor voldoend kleine h_n

$$(2.8.7) \qquad \vec{\rho}_n \cong \frac{1}{p+1} h_n \vec{\zeta}_n \ .$$

Deze benadering is (voor vaste h_n) des te beter naarmate de Lipschitz-constante M_n voor de rechterlidfunctie kleiner is. Dit betekent dat voor stijve differentiaalvergelijkingen, gekenmerkt door grote waarden van M_n, relatie (2.8.7) ook weer heel twijfelachtig is. Desalniettemin wijst de practijk uit dat voor stijve differentiaalvergelijking de nauwkeurigheid op aanvaardbare wijze met $\vec{\zeta}_n$ (of om de orde in h_n aan te passen met $h_n\vec{\zeta}_n$) gestuurd kan worden.

Tenslotte is voor exponentieel aangepaste integratieformules wel geëxperimenteerd met het onder controle houden van de niet-lineariteit ten opzichte van de staplengte van de differentiaalvergelijking. De gedachte hierbij is dat, wanneer de integratiestap h_n zo klein is dat de vergelijking zich min of meer lineair gedraagt in het interval $[x_n, x_n+h_n]$, een exponentieel aangepaste formule dan vrijwel exact integreert. Dit suggereert dat de nauwkeurigheid evenredig is met de mate van lineariteit van de vergelijking. Een indicatie van de mate van lineariteit kan verkregen worden door een referentieformule toe te passen waarvan de stabiliteitsfunctie identiek is met die van de gebruikte integratieformule. Voor lineaire differentiaalvergelijkingen geeft dit dan

$$\tilde{\vec{y}}_{n+1} = \vec{y}_{n+1}$$

en voor niet-lineaire vergelijkingen is de discrepantie

$$(2.8.8) \qquad \vec{\nu}_n = \tilde{\vec{y}}_{n+1} - \vec{y}_{n+1}$$

een aanwijzing hoe niet-lineair de vergelijking is ten opzichte van h_n.

2.8.2 Berekening van de integratiestap

We hebben nu een drietal discrepantiefuncties besproken:

$$\vec{d}(x_n,\vec{y}_n;h_n) = \begin{cases} \vec{y}_{n+1} - \tilde{\vec{y}}_{n+1} & \text{voor niet-stijve problemen} \\ \frac{1}{p+1} h_n \vec{\zeta}_n & \text{voor stijve problemen} \\ \vec{\nu}_n & \text{voor exponentieel aangepaste formules.} \end{cases} \tag{2.8.9}$$

De integratiestap h_n moet nu voldoen aan de relatie (vergelijk (2.8.3))

$$\|\vec{d}(x_n,y_n;h_n)\| = \eta_n \ . \tag{2.8.10}$$

Indien men $\vec{d}$ als functie van h_n zou kennen, volgt h_n direct door een of andere numerieke nulpuntszoeker op (2.8.10) los te laten. In het algemeen is $\vec{d}$ echter alleen bekend in de punten $(x_j,\vec{y}_j;h_j)$, $j = n-1,n-2,\ldots$; in zulke gevallen wordt h_n berekend door $\vec{d}$ voor te stellen door een interpolatieformule op deze punten en te substitueren in (2.8.10), waarna h_n of expliciet of numeriek berekend kan worden. Bijvoorbeeld leidt de interpolatieformule

$$\tilde{\vec{d}}(x,y;h) = (\vec{B}_n x+\vec{C}_n)h^q \ ,$$

$$\vec{B}_n = h_{n-2}^{-1}h_{n-1}^{-q}\vec{d}_{n-1} - h_{n-2}^{-q-1}\vec{d}_{n-2} \ ,$$

$$\vec{C}_n = h_{n-2}^{-q}\vec{d}_{n-2} - x_{n-2}\vec{B}_n \ , \tag{2.8.11}$$

$\vec{d}_{n-1},\vec{d}_{n-2}$ berekende discrepanties in $(x_{n-1},\vec{y}_{n-1};h_{n-1})$ en $(x_{n-2},\vec{y}_{n-2},h_{n-2})$,

q orde van $\vec{d}_{n-1}$ en $\vec{d}_{n-2}$ in h_{n-1} en h_{n-2} ,

tot de staplengte (η_n is onafhankelijk van h_n verondersteld)

$$h_n = \sqrt[q]{\frac{\eta_n}{\|\vec{B}_n x_n+\vec{C}_n\|}} \ . \tag{2.8.12}$$

Als controle kan men nu de discrepantie (2.8.9) uitrekenen en vergelijken met de tolerantie η_n. Indien de tolerantie niet gehaald wordt en men wil

kunnen garanderen dat in elke stap de discrepantie kleiner dan of gelijk aan aan de tolerantie is, dan kan een nieuwe staplengte berekend worden door interpolatie van de discrepanties $\vec{d}(x_n,\vec{y}_n;h_n^*)$, $\vec{d}(x_{n-1},\vec{y}_{n-1};h_{n-1})$,... waarin h_n^* de verworpen stap voorstelt, enz.
We merken nog op dat een strategie met stapverwerping een grotere geheugenruimte in het systeem vergt. Daarom ziet men in algorithmen voor grote stelsels differentiaalvergelijkingen wel af van stapverwerping bij de implementatie.

2.8.3 Stapkeuzestrategieën bij Taylor-methoden

Bij *expliciete* Taylor-methoden zijn de discrepantiefuncties gebaseerd op een *referentieoplossing* en de *residufunctie*, eenvoudig te verkrijgen; we vinden respectievelijk

$$\vec{d}(x_n,\vec{y}_n;h_n) = [\tilde{R}(h_n \tfrac{d}{dx}) - R(h_n \tfrac{d}{dx})]\vec{z}(x_n,\vec{y}_n;x)\Big|_{x=x_n} \qquad (2.8.13)$$

en

$$\vec{d}(x_n,\vec{y}_n;h_n) = \frac{1}{p+1} h_n\Big[\vec{f}(\vec{y}_{n+1}) - (\frac{d}{dh_n} R(h_n \tfrac{d}{dx}))\vec{z}(x_n,\vec{y}_n;x)\Big|_{x=x_n}\Big] \qquad (2.8.14)$$

Hierin stelt $\tilde{R}$ de stabiliteitsfunctie van de referentieoplossing voor. Merk op dat de discrepantie van de lineariteit $\vec{\nu}_n$ voor Taylor-methoden geen informatie geeft omdat identificatie van stabiliteitsfuncties identificatie van de integratieformules geeft.

De stapkeuzestrategie van de in paragraaf 2.7.1 besproken procedure *modified taylor* is gebaseerd op (2.8.13) met

$$\tilde{R}(z) = 1 + z + \frac{1}{2}z^2 + \ldots + \frac{1}{\tilde{m}!}z^{\tilde{m}}\ ,\ \tilde{m} = \begin{cases} m & \text{als } p < m \\ m-1 & \text{als } p = m \end{cases}$$

Dit geeft voor $\vec{d}$ de uitdrukking

$$\vec{d}(x_n,\vec{y}_n;h_n) = h_n^{p+1} \cdot \sum_{j=p+1}^{m} (\beta_j - \frac{1}{j!})h_n^{j-p+1} \frac{d^j}{dx^j}\vec{z}(x_n,y_n;x)\Big|_{x=x_n} \qquad (2.8.13')$$

als $p < m$ en

(2.8.13") $\quad \vec{d}(x_n,\vec{y}_n;h_n) = -\frac{1}{m!}(h_n \frac{d}{dx})^m \vec{z}(x_n,\vec{y}_n;x)\Big|_{x=x_n}$

als p = m.

2.8.4 Stapkeuzestrategieën bij Runge-Kuttamethoden

Van de drie beschouwde discrepantiefuncties is degene gebaseerd op een *referentieoplossing* voor Runge-Kuttaformules de meest realistische. We zullen ons beperken tot referentieformules gegenereerd door matrices van de vorm

(2.8.15)
$$\left(\begin{array}{cccc} 0 & \cdots & & 0 \\ \lambda_{1,0} & & & 0 \\ \vdots & \ddots & & \vdots \\ \lambda_{m,0} & \cdots & \lambda_{m,m-1} & 0 \\ \hline \tilde{\lambda}_{m,0} & \cdots & \tilde{\lambda}_{m,m-1} & \tilde{\lambda}_{m,m} \end{array}\right) \begin{matrix} \\ \\ \\ \\ \vec{y}_{n+1} \\ \tilde{\vec{y}}_{n+1} \end{matrix} ,$$

waarin de matrix bestaande uit de eerste m + 1 rijen de gebruikte Runge-Kuttaformule voorstelt. De referentieformule bevat dus nog m + 1 vrij te kiezen parameters $\tilde{\lambda}_{m,j}$. Merk op dat alhoewel de formule voor $\tilde{\vec{y}}_{n+1}$ een (m+1)-puntsformule is, geen extra functie-evaluatie vereist is; immers de $(m+1)^e$ functie-evaluatie in $\tilde{\vec{y}}_{n+1}$ is gelijk aan de eerste functieevaluatie in $\vec{y}_{n+2}$, tenzij uiteraard de voorspelde staplengte h_n verworpen wordt. De consistentievoorwaarden voor $\tilde{\vec{y}}_{n+1}$ luiden (vergelijk tabel 2.4.2)

Deze voorwaarden zijn lineair in de parameters $\tilde{\lambda}_{m,j}$. Hieruit volgt dat voor een $\tilde{p}^e$ orde referentieformule, de waarde van m + 1 minstens gelijk aan het aantal consistentievoorwaarden $\mu(\tilde{p})$ moet zijn, waarin $\mu(\tilde{p})$ gedefinieerd is volgens (vergelijk tabel 2.4.3)

$\tilde{p}$	1	2	3	4	5	6
$\mu(\tilde{p})$	1	2	4	8	17	37

Tabel 2.8.1 Consistentievoorwaarden voor $\tilde{\tilde{y}}_{n+1}$

$\tilde{p} \geq 1$	$m \geq 0$	$\tilde{\beta}_1 = \sum_{j=0}^{m} \tilde{\lambda}_{m,j} = 1$
$\tilde{p} \geq 2$	$m \geq 1$	$\tilde{\beta}_2 = \sum_{j=1}^{m} \tilde{\lambda}_{m,j} \sum_{l=0}^{j-1} \lambda_{j,l} = \frac{1}{2}$
$\tilde{p} \geq 2$	$m \geq 3$	$\tilde{\beta}_3 = \sum_{j=2}^{m} \tilde{\lambda}_{m,j} \sum_{l=1}^{j-1} \lambda_{j,l} \sum_{k=0}^{l-1} \lambda_{l,k} = \frac{1}{6}$ $\tilde{\beta}_{3,1} = \sum_{j=1}^{m} \tilde{\lambda}_{m,j} \left(\sum_{l=0}^{j-1} \lambda_{j,l} \right)^2 = \frac{1}{3}$
...	...	...

Indien m + 1 groter is dan $\mu(\tilde{p})$, maar kleiner dan $\mu(\tilde{p}+1)$, dan beschikken we over een hele klasse van p^e orde referentieformules; men kan deze vrijheid benutten door een referentieformule te kiezen waarin de parameters $\tilde{\lambda}_{m,j}$ zo klein mogelijk zijn in absolute waarde. Het stelsel voor de $\tilde{\lambda}_{m,j}$ heeft namelijk de neiging slecht geconditionneerd te zijn zodat sterk alternerende waarden gevonden worden. Een andere mogelijkheid is om een referentieformule te kiezen die voor lineaire differentiaalvergelijkingen een orde nauwkeuriger is.

De stapkeuzestrategie in de procedure *ark* (zie paragraaf 2.7.7) is gebaseerd op (2.8.15) met zo groot mogelijke waarde voor $\tilde{p}$ en zo klein mogelijke waarden voor de $\tilde{\lambda}_{m,j}$.

De in paragraaf 2.7.11 besproken 5^e orde formule van Beentjes is dusdanig geconstrueerd dat de gewichten

$$(\tilde{\lambda}_{6,j}) = (0,0, \tfrac{5}{6}, -\tfrac{2}{3}, \tfrac{5}{6}, 0, 0) \qquad (2.8.16)$$

een 4^e orde referentieformule definieren. Bij de implementatie van deze formule, de procedure *rke*, is de stapkeuzestrategie op deze referentieformule gebaseerd.

We besluiten deze paragraaf met een aantal voorbeelden van ingebedde referentieformules.

Voorbeelden 2.8.1

Euler's formule (2.1.5')

$$\left(\begin{array}{cc} 0 & 0 \\ 1 & 0 \\ \hline \frac{1}{2} & \frac{1}{2} \end{array}\right) \begin{array}{c} \vec{y}_{n+1} \\ \tilde{\vec{y}}_{n+1} \end{array} \quad , \; p = 1 \; , \; \tilde{p} = 2$$

Gestabiliseerde Euler-formule

$$\left(\begin{array}{ccc} 0 & 0 & 0 \\ \frac{1}{8} & 0 & 0 \\ 0 & 1 & 0 \\ \hline \frac{19}{3} & -\frac{20}{3} & \frac{4}{3} \end{array}\right) \begin{array}{c} \vec{y}_{n+1} \\ \tilde{\vec{y}}_{n+1} \end{array} \quad , \; p = 1 \; , \; \tilde{p} = \begin{array}{l} 2 \text{ voor niet-lineaire vgln} \\ 3 \text{ voor lineaire vgln} \end{array}$$

Runge's formule (2.2.6)

$$\left(\begin{array}{ccc} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ \hline \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{array}\right) \begin{array}{c} \vec{y}_{n+1} \\ \tilde{\vec{y}}_{n+1} \end{array} \quad , \; p = 2 \; , \; \tilde{p} = \begin{array}{l} 2 \text{ voor niet-lineaire vgln} \\ 3 \text{ voor lineaire vgln} \end{array}$$

Gestabiliseerde Runge-formule

$$\left(\begin{array}{cccc} 0 & 0 & 0 & 0 \\ \frac{1}{8} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline -\frac{37}{30} & \frac{64}{30} & -\frac{8}{30} & \frac{11}{30} \end{array}\right) \begin{array}{c} \vec{y}_{n+1} \\ \tilde{\vec{y}}_{n+1} \end{array} \quad , \; p = 2 \; , \; \tilde{p} = 3 \; .$$

2.8.5 Stapkeuzestrategieën bij gegeneraliseerde Runge-Kuttamethoden

We beperken ons hier tot de uitwerking van twee strategieën voor de respectieve procedures *eferk* en *efsirk*.

In de procedure *eferk* is de stapcontrole gebaseerd op een benadering van de residufunctie $\vec{\zeta}_n$; de discrepantiefunctie is namelijk gedefinieerd door (vergelijk (2.8.9))

$$\vec{d}(x_n,\vec{y}_n;h_n) = \frac{1}{p+1}\, h_n[\vec{f}(\vec{y}_{n+1})-J^{-1}(\vec{y}_n)\,\frac{d}{dh_n}\,R(h_nJ(\vec{y}_n))\vec{f}(\vec{y}_n)] \quad . \tag{2.8.17}$$

Voor lineaire differentiaalvergelijkingen is deze discrepantiefunctie precies $h_n\vec{\zeta}_n/(p+1)$, omdat volgens formule (2.6.9) de functie $\vec{w}$ dan gegeven wordt door

$$\vec{w}(x_n,\vec{y}_n,x_n+h) = \vec{y}_n + [R(hJ(\vec{y}_n))-I]J^{-1}(\vec{y}_n)\vec{f}(\vec{y}_n) \quad .$$

Voor zwak niet-lineaire vergelijkingen blijkt (2.8.17) echter ook nog bevredigende resultaten te geven en aangezien de procedure *eferk* vanwege zijn beperkte stabiliteitsgebieden hoofdzakelijk gebruikt dient te worden voor problemen met langzaam veranderende Jacobianen, is de beperking tot zwak niet-lineaire problemen niet wezenlijk. We wijzen er nog op dat (2.8.17) gebruikt kan worden voor een willekeurige éénstapsintegratieformule met stabiliteitsfunctie R mits de differentiaalvergelijking maar voldoende lineair is.

De procedure *efsirk* is ontwikkeld om ook sterker niet-lineaire problemen te kunnen integreren (*efsirk* is A-stabiel!).Om de exponentiele aanpassing echter tot z'n recht te laten komen moet de integratiestap zo klein zijn dat de vergelijking zich in het interval $[x_n,x_n+h_n]$ voldoende lineair gedraagt. Daartoe construeren we een referentieoplossing $\tilde{\vec{y}}_{n+1}$ met dezelfde stabiliteitsfunctie als *efsirk* en definieren de discrepantiefunctie

$$\vec{v}_n = \tilde{\vec{y}}_{n+1} - \vec{y}_{n+1} \quad .$$

We zoeken de referentieoplossing in de vorm (nagenoeg geen extra rekenwerk!)

$$\tilde{\vec{y}}_{n+1} = \vec{y}_n + \tilde{\theta}_0h_n\vec{f}(\vec{y}_n) + \tilde{\theta}_1h_n\vec{f}(\vec{y}_n+\Lambda(h_nJ(\vec{y}_n))h_n\vec{f}(\vec{y}_n)) + \tag{2.8.18}$$

$$+ \tilde{\theta}_2h_n\Lambda(h_nJ(\vec{y}_n))\vec{f}(\vec{y}_n) + \tilde{\theta}_3h_n\vec{f}(\vec{y}_{n+1}) \quad .$$

Deze formule heeft de stabiliteitsfunctie

$$\tilde{R}(z) = 1 + \tilde{\theta}_0 z + \tilde{\theta}_1 z(1+z\Lambda(z)) + \tilde{\theta}_2 z\Lambda(z) + \tilde{\theta}_3 zR(z).$$

Identificatie met R geeft

$$(2.8.19) \qquad 1 + (\tilde{\theta}_0+\tilde{\theta}_1)z + (\tilde{\theta}_1 z+\tilde{\theta}_2)z\Lambda(z) + (\tilde{\theta}_3 z-1)R(z) = 0 .$$

Substitutie van de bij *efsirk* behorende functies Λ en R (zie (2.7.40)) leidt tot 3 lineaire betrekkingen tussen de coefficienten $\tilde{\theta}_j$. Kiezen we $\tilde{\theta}_0 = 0$ dan vinden we

$$\tilde{\theta}_1 = \frac{6+9\alpha_1}{8+6\alpha_1} \quad , \quad \tilde{\theta}_2 = \frac{9}{16+12\alpha_1} \quad , \quad \tilde{\theta}_3 = -\frac{1+3\alpha_1}{8+6\alpha_1} \quad ,$$

waarin α_1 de aanpassingsparameter is uit de Liniger-Willoughbystabiliteitsfunctie gedefinieerd door (2.7.26) en (2.7.27).
De discrepantiefunctie gebaseerd op (2.8.18) meet niet alleen de mate van niet-lineariteit ten opzichte van het interval $[x_n, x_{n+1}]$, maar is bovendien een (conservatieve) maat voor de nauwkeurigheid, omdat $\vec{\tilde{y}}_{n+1}$ volgens stelling 2.6.2 tweede orde consistent is.

2.9 ENKELE NIEUWE ONTWIKKELINGEN BIJ DE CONSTRUCTIE VAN EENSTAPSINTEGRATIEFORMULES

Deze syllabus wordt besloten met de bespreking van enkele nieuwe ideeën over de numerieke integratie van differentiaalvergelijkingen en wel in het bijzonder wanneer deze toegepast worden op eenstapsformules.

2.9.1 Taylor-Runge-Kuttamethoden

In opgave 2.2.1/(2) hebben we al een integratieformule gegeven waarin $\vec{y}_{n+1}$ gedefinieerd werd als $\vec{y}_n$ plus een lineaire combinatie van evaluaties van het rechterlid $\vec{f}$ en van de eerste afgeleide $\vec{g}$ van de rechterlidfunctie. De punten waarin de functies $\vec{f}$ en $\vec{g}$ werden geëvalueerd waren op zich ook op deze manier gedefinieerd, enz. We hebben hier dus te maken met een mengvorm van Taylor- en Runge-Kuttaformules (ook wel een *hybride* formule genoemd). Men kan dit uitbreiden tot formules waarin $\vec{y}_{n+1}$ gedefinieerd wordt

als $\vec{y}_n$ plus een lineaire combinatie van evaluaties van $\vec{f}$ en zijn eerste, tweede, ... afgeleiden. Of dit een zinvolle klasse van formules is, moet echter nog onderzocht worden. Één voordeel is wel direct aan te geven, namelijk dat in gevallen waarin slechts enkele opvolgende afgeleiden van $\vec{f}$ beschikbaar zijn, te weinig om een hierop gebaseerde Taylor-formule voldoende nauwkeurig te maken, men niet zijn toevlucht hoeft te nemen tot Runge-Kuttaformules waarin de informatie omtrent $\vec{f}$ helemaal niet meer aan bod komt, maar dat men de extra informatie als het ware in de Runge-Kuttaformule kan pompen.

We beperken ons hier tot een eerste onderzoek van formules die uitsluitend van $\vec{f}$ en $\vec{g}$ gebruik maken (de reeds gegeven formule (2.2.9))

$$
\begin{aligned}
&\vec{y}_{n+1}^{(0)} = \vec{y}_n \ , \\
&\vec{y}_{n+1}^{(j)} = \vec{y}_n + h_n \sum_{l=0}^{j-1} [\lambda_{j,l}\vec{f}(\vec{y}_{n+1}^{(l)}) + \mu_{j,l} h_n \vec{g}(\vec{y}_{n+1}^{(l)})], \quad j = 1,2,\ldots,m, \\
&\vec{y}_{n+1} = \vec{y}_{n+1}^{(m)} \ ,
\end{aligned}
\tag{2.9.1}
$$

waarin $\vec{g}(\vec{y}(x)) = d\vec{f}(\vec{y}(x))/dx$. Merk op dat de relatie

$$
\vec{g}(\vec{y}) = J(\vec{y})\vec{f}(\vec{y})
$$

formule (2.9.1) overvoert in een formule uit de klasse van gegeneraliseerde Runge-Kuttaformules (2.3.3). Een bijzonder geval van (2.9.1) is de derde orde consistente formule (zie opgave 2.4.2/(2))

$$
\vec{y}_{n+1} = \vec{y}_n + h_n\vec{f}(\vec{y}_n) + \tfrac{1}{2}h_n^2\vec{g}(\vec{y}_n + \tfrac{1}{3}h_n\vec{f}(\vec{y}_n)) \ . \tag{2.9.2}
$$

Deze formule vinden we al bij Henrici [1962]; het toont aan dat het mengen van Taylor- en Runge-Kuttaformules inderdaad zinvol kan zijn: met *twee evaluaties* wordt *derde orde consistentie* verkregen!

We zullen nu de algemene formule (2.9.1) analyseren voor het geval van eerste en tweede orde consistentie. Daartoe is het voldoende dat de stabiliteitsfunctie van (2.9.1) respectievelijk eerste en tweede orde consistent is. Het is eenvoudig te verifieren dat de stabiliteitsfunctie R recursief gegeven wordt door

$$R^{(0)}(z) = 1 ,$$

$$(2.9.3) \qquad R^{(j)}(z) = 1 + z \sum_{l=0}^{j-1} [\lambda_{j,l}+\mu_{j,l}z]R^{(l)}(z) , \quad j = 1,2,\ldots,m ,$$

$$R(z) = R^{(m)}(z) .$$

Schrijven we R(z) als

$$(2.9.4) \qquad 1 + \beta_1 z + \beta_2 z^2 + \ldots + \beta_j z^j + \ldots ,$$

dan vinden we na enig rekenwerk

$$(2.9.5) \qquad \beta_1 = \sum_{j=0}^{m-1} \lambda_{m,j} \quad , \quad \beta_2 = \sum_{j=0}^{m-1} [\mu_{m,j}+\lambda_{m,j} \sum_{l=0}^{j-1} \lambda_{j,l}] .$$

De consistentievoorwaarden zijn dus $\beta_1 = 1$ voor eerste orde consistentie en $\beta_1 = 2\beta_2 = 1$ voor tweede orde consistentie.

De graad van het polynoom R is maximaal 2m, namelijk wanneer alle parameters $\mu_{j,j-1}$ ongelijk nul zijn. Aangezien het vanuit het oogpunt van stabiliteit wenselijk is om een stabiliteitspolynoom van zo hoog mogelijke graad te hebben,zullen we in elk geval

$$\mu_{j,j-1} \neq 0 \quad , \quad j = 1,2,\ldots,m$$

kiezen. Verder willen we natuurlijk het aantal functie-evaluaties beperken. Dit kan alleen door de parameters $\lambda_{j,l}$ nul te kiezen. Volgens (2.9.5) zal echter minstens één parameter $\lambda_{m,j}$ ongelijk nul moeten zijn, anders zou $\beta_1 = 0$ worden waarmee de consistentievoorwaarde $\beta_1 = 1$ geschonden wordt. Laten we

$$\lambda_{j,0} \neq 0 \quad , \quad \lambda_{j,l} = 0 \quad , \quad j = 1,2,\ldots,m \quad , \quad l = 1,\ldots,j-1$$

kiezen; we krijgen dan de parametermatrices

$$(2.9.6)\quad (\lambda_{j,l}) = \begin{pmatrix} 0 & & \cdots & 0 \\ \lambda_{1,0} & & & \\ \lambda_{2,0} & 0 & & \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m,0} & 0 & \cdots & 0 \end{pmatrix}, \quad (\mu_{j,l}) = \begin{pmatrix} 0 & & & 0 \\ \mu_{1,0} & & & \\ \vdots & & \ddots & \vdots \\ & & & \\ \mu_{m,0} & \cdots & & \mu_{m,m-1} \end{pmatrix}.$$

Dit kost één evaluatie van $\vec{f}$ en m evaluaties van $\vec{g}$. Er zijn m(m+3)/2 vrije parameters, zodat identificatie met een voorgeschreven stabiliteitspolynoom (dit betekent 2m voorwaarden) heel goed mogelijk lijkt. Dit impliceert dat *als één evaluatie van $\vec{g}$ minder dan (2m-1)/m maal de rekentijd kost van een evaluatie van $\vec{f}$*, we er wat betreft de graad van het stabiliteitspolynoom op vooruitgaan vergeleken bij de Runge-Kuttaformules; voor dezelfde hoeveelheid rekenwerk per integratiestap zal men over grotere stabiliteitsgebieden kunnen beschikken. Of formule (2.9.6) nog verdere voordelen biedt ten opzichte van Runge-Kuttaformules is een open vraag.

Voorbeeld 2.9.1

Laten we de formule gegenereerd door (2.9.6) met m = 2 eens aanpassen aan een gegeven stabiliteitspolynoom

$$(2.9.7)\qquad R(z) = 1 + \beta_1 z + \beta_2 z^2 + \beta_3 z^2 + \beta_4 z^4.$$

Om het rekenwerk te vereenvoudigen stellen we direct al

$$\mu_{2,0} = 0.$$

Uit (2.9.3) volgt dan dat het stabiliteitspolynoom gegeven wordt door

$$R(z) = 1 + \lambda_{2,0} z + \mu_{2,1} z^2 + \mu_{2,1}\lambda_{1,0} z^3 + \mu_{2,1}\mu_{1,0} z^4.$$

Identificatie met (2.9.7) geeft

$$\lambda_{2,0} = \beta_1 \quad , \quad \mu_{2,1} = \beta_2 \quad , \quad \lambda_{1,0} = \frac{\beta_3}{\beta_2} \quad , \quad \mu_{1,0} = \frac{\beta_4}{\beta_2}.$$

De parametermatrices (2.9.6) worden dus

$$(\lambda_{j,l}) = \begin{pmatrix} 0 & 0 \\ \beta_3/\beta_2 & 0 \\ \beta_1 & 0 \end{pmatrix} \quad , \quad (\mu_{j,l}) = \begin{pmatrix} 0 & 0 \\ \beta_4/\beta_2 & 0 \\ 0 & \beta_2 \end{pmatrix} \quad .$$

Als numeriek voorbeeld kiezen we het polynoom

$$R(z) = 1 + z + \frac{5}{32} z^2 + \frac{1}{128} z^3 + \frac{1}{8192} z^4 \ ,$$

dat een reële stabiliteitsgrens $\beta = 32$ heeft. We vinden dan de formule

$$(2.9.8) \qquad \vec{y}_{n+1} = \vec{y}_n + h_n \vec{f}(\vec{y}_n) + \frac{5}{32} h_n^2 \vec{g}(\vec{y}_n + \frac{1}{20} h_n \vec{f}(\vec{y}_n) + \frac{1}{1280} h_n^2 \vec{g}(\vec{y}_n)) \ .$$

2.9.2 Ingebedde Runge-Kuttamethoden

In de formulering (2.2.3) van de algemene m-punts Runge-Kuttamethode hebben de tussenresultaten $\vec{y}_{n+1}^{(j)}$ voor $j = 2,3,\ldots,m-1$ in eerste instantie niets met de analytische oplossing te maken, ook al zijn de consistentievoorwaarden voor een zekere orde p vervuld. Alleen $\vec{y}_{n+1}^{(m)} = \vec{y}_{n+1}$ en $\vec{y}_{n+1}^{(1)}$ geven een benadering voor $\vec{y}$ in de punten $x_n + h_n$ respectievelijk $x_n + \lambda_{1,0} h_n$. Nu hebben we bij de constructie van gestabiliseerde formules gezien (paragrafen 2.7.4 en 2.7.5) dat er meer dan voldoende parameters zijn om aan consistentie- en adaptiviteitsvoorwaarden te voldoen. Deze extra vrijheidsgraden hebben we gebruikt om de *geheugenruimte* nodig om de formules te implementeren, te minimaliseren. Een zinvol alternatief lijkt de extra parameters zó te kiezen dat de *tussenresultaten* $\vec{y}_{n+1}^{(j)}$, *$j = 2,3,\ldots,m-1$, ook benaderingen zijn voor $\vec{y}$ in punten $x_n + q_j h_n$ uit het interval* $[x_n, x_n + h_n]$. Met andere woorden de formules voor de $\vec{y}_{n+1}^{(j)}$ moeten ook aan een aantal consistentievoorwaarden voldoen. Laten we de nieuwe parameters

$$\lambda_{i,l}^{(j)} = \frac{\lambda_{i,l}}{q_j} \quad , \quad i = 1,2,\ldots,j \quad ,$$

voor $j = 1,2,\ldots,m$ invoeren, dan luidt de formule voor $\vec{y}_{n+1}^{(j)}$:

$$\vec{y}_{n+1}^{(0)} = \vec{y}_n$$

$$\vec{y}_{n+1}^{(i)} = \vec{y}_n + (q_j h_n) \sum_{l=0}^{i-1} \lambda_{i,l}^{(j)} \vec{f}(\vec{y}_{n+1}^{(l)}) \quad , \qquad i = 1,2,\ldots,j \ .$$

Blijkbaar moeten de parameters $\lambda_{i,l}^{(j)}$ voor elke j voldoen aan een aantal voorwaarden uit tabel 2.4.2, waarbij dan $\lambda_{i,l}$ vervangen wordt door $\lambda_{i,l}^{(j)}$. Wanneer $\vec{y}_{n+1}^{(j)}$ op deze manier geconstrueerd wordt, spreekt men van een *ingebedde Runge-Kuttaformule*.

Ingebedde Runge-Kuttaformules kunnen gebruikt worden om schattingen te verkrijgen voor de lokale afbreekfout (zie Lapidus en Seinfeld [1971]). Voor gestabiliseerde Runge-Kuttaformules is dit echter nog niet onderzocht.

Voorbeeld 2.9.2

We beschouwen een vierpunts Runge-Kuttaformule van de tweede orde met gegeven stabiliteitspolynoom

$$(2.9.9) \qquad R(z) = 1 + z + \frac{1}{2} z^2 + \beta_3 z^3 + \beta_4 z^4 \ .$$

Dit geeft aanleiding tot 4 relaties voor de parameters $\lambda_{j,l}$:

$$\lambda_{4,0} + \lambda_{4,1} + \lambda_{4,2} + \lambda_{4,3} = 1 \ ,$$

$$\lambda_{1,0}\lambda_{4,1} + (\lambda_{2,0}+\lambda_{2,1})\lambda_{4,2} + (\lambda_{3,0}+\lambda_{3,1}+\lambda_{3,2})\lambda_{4,3} = \frac{1}{2} \ ,$$

$$\lambda_{1,0}\lambda_{2,1}\lambda_{4,2} + \lambda_{1,0}\lambda_{3,1}\lambda_{4,3} + (\lambda_{2,0}+\lambda_{2,1})\lambda_{3,2}\lambda_{4,3} = \beta_3 \ ,$$

$$\lambda_{1,0}\lambda_{2,1}\lambda_{3,2}\lambda_{4,3} = \beta_4 \ .$$

Hieraan voegen we de relaties toe, die $\vec{y}_{n+1}^{(1)}$ eerste orde en $\vec{y}_{n+1}^{(2)}$, $\vec{y}_{n+1}^{(3)}$ tweede orde consistent maken:

$$\lambda_{1,0} = q_1 \ ,$$

$$\lambda_{2,0} + \lambda_{2,1} = q_2 \ ,$$

$$\lambda_{1,0}\lambda_{2,1} = \frac{1}{2} q_2^2 \ ,$$

$$\lambda_{3,0} + \lambda_{3,4} + \lambda_{3,2} = q_3 \ ,$$

$$\lambda_{1,0}\lambda_{3,1} + (\lambda_{2,0}+\lambda_{2,1})\lambda_{3,2} = \frac{1}{2} q_3^2 \ ,$$

We hebben nu 9 relaties in 10 onbekenden als we de q_j als voorgeschreven beschouwen. Stel dat we

$$\lambda_{3,1} = 0$$

kiezen, dan vinden we

$$\lambda_{1,0} = q_1 \quad , \quad \lambda_{2,1} = \frac{q_2^2}{2q_1} \quad , \quad \lambda_{2,0} = q_2(1- \frac{q_2}{2q_1}) \ ,$$
$$\lambda_{3,2} = \frac{q_3^2}{2q_2} \quad , \quad \lambda_{3,0} = q_3(1- \frac{q_3}{2q_2}) \ ,$$

terwijl de vector $\vec{\lambda}_4 = (\lambda_{4,j})$ gegeven wordt door het lineaire stelsel

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & q_1 & q_2 & q_3 \\ 0 & 0 & \frac{1}{2}q_2^2 & \frac{1}{2}q_3^2 \\ 0 & 0 & 0 & \frac{1}{4}q_2 q_3^2 \end{pmatrix} \vec{\lambda}_4 = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \beta_3 \\ \beta_4 \end{pmatrix}$$

We geven twee numerieke voorbeelden van dit soort ingebedde formules door middel van hun genererende matrices:

(2.9.10)
$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2}-2\beta_3 & 2(\beta_3-2\beta_4) & 4\beta_4 \end{pmatrix} \qquad \text{met } q_1 = q_2 = q_3 = 1 \ ,$$

$$(2.9.11)\qquad \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -8\beta_4 & 1-8(\beta_3-4\beta_4) & 8(\beta_3-4\beta_4) & 8\beta_4 \end{pmatrix} \text{ met } q_1=q_2=\frac{1}{2}\ ,\ q_3=1\ .$$

In het eerste geval geeft vergelijking van $\vec{y}^{(2)}_{n+1}$, $\vec{y}^{(3)}_{n+1}$ en $\vec{y}_{n+1}$ informatie over de fout in x_{n+1}, terwijl in het tweede geval iets over de fout in $x_n + \frac{1}{2}h_n$ en x_{n+1} gezegd kan worden, namelijk door vergelijking van $\vec{y}^{(1)}_{n+1}$ met $\vec{y}^{(2)}_{n+1}$ en $\vec{y}^{(3)}_{n+1}$ met $\vec{y}_{n+1}$.

2.9.3 Rationale Taylor- en Runge-Kuttamethoden

Tot dusver waren alle in deze syllabus onderzochte formules *lineaire* combinaties van evaluaties van het rechterlid en zijn afgeleiden. Voor de *expliciete* Taylor-methoden betekende dit dat $\vec{y}_{n+1}$ als functie van h_n benaderd werd door een *polynoom* in h_n. Nu is een polynoom voor wat grotere intervallen niet altijd geschikt en doen rationale functies het soms veel beter. Dit suggereert om integratieformules van de vorm

$$(2.9.12)\qquad \vec{y}_{n+1} = \frac{\vec{a}_0+h_n\vec{a}_1+h_n^2\vec{a}_2+\ldots}{\vec{1}+h_n\vec{b}_1+h_n^2\vec{b}_2+\ldots}$$

te beschouwen, waarin de deling van vectoren componentsgewijs uitgevoerd dienen te worden; de vectoren $\vec{a}_j$ en $\vec{b}_j$ worden bepaald door relaties van de vorm

$$\vec{y}_{n+1}(0) = \vec{y}_n\ ,$$

$$(2.9.13)\qquad \frac{d}{dh_n}\vec{y}_{n+1}(h_n)\Big|_{h_n=0} = \vec{y}'_n \equiv \vec{f}(\vec{y}_n)\ ,$$

$$\frac{d^2}{dh_n^2}\vec{y}_{n+1}(h_n)\Big|_{h_n=0} = \vec{y}''_n \equiv \vec{g}(\vec{y}_n)\ ,$$

$$\ldots \qquad .$$

Formule (2.9.12) stelt een *rationale Taylor-formule* voor. Dergelijke methoden werden onderzocht door Lambert en Shaw [1965], maar het laatste woord is hier nog zeker niet over gezegd.

Men verkrijgt *rationale Runge-Kuttaformules* wanneer de afgeleiden $\vec{y}_n''$, $\vec{y}_n'''$ benaderd worden door lineaire combinaties van rechterlidevaluaties.

De reden om dergelijke rationale formules te onderzoeken is het over het algemeen *aanzienlijk betere stabiliteitsgedrag* voor grote waarden van h_n.

Voorbeeld 2.9.3

In de formule

$$\vec{y}_{n+1} = \frac{2\vec{y}_n\vec{y}_n' + h_n(2(\vec{y}_n')^2 - \vec{y}_n''\vec{y}_n)}{2\vec{y}_n' - h_n\vec{y}_n''} \tag{2.9.14}$$

stemmen de eerste en tweede afgeleide van $\vec{y}_{n+1} = \vec{y}_{n+1}(h_n)$ voor $h_n = 0$ overeen met $\vec{y}_n'$ en $\vec{y}_n''$. Deze formule is dus consistent van de orde 2.

Vervangt men in deze formule $\vec{y}_n''$ door een lineaire combinatie van $\vec{f}(\vec{y}_n)$ en $\vec{f}(\vec{y}_n + \frac{1}{\mu}h_n\vec{f}(\vec{y}_n))$, dan is eenvoudig na te gaan dat de formule

$$\vec{y}_{n+1} = \frac{[(2+\mu)\vec{f}(\vec{y}_n) - \mu\vec{f}(\vec{y}_n + \frac{1}{\mu}h_n\vec{f}(\vec{y}_n)]y_n + 2h_n\vec{f}(\vec{y}_n)^2}{(2+\mu)\vec{f}(\vec{y}_n) - \mu\vec{f}(\vec{y}_n + \frac{1}{\mu}h_n\vec{f}(\vec{y}_n))} \tag{2.9.15}$$

consistent van de orde 2 is; hierin is μ een nog vrije parameter.

We zullen de stabiliteit van deze laatste formule onderzoeken. Daartoe verstoren we $\vec{y}_n$ met een "kleine" vector $\vec{\rho}_n^*$, zodat $\vec{y}_{n+1}$ overgaat in $\vec{y}_{n+1}^*$. We vinden

$$\vec{y}_{n+1}^* = \vec{y}_n + \vec{\rho}_n^* + \frac{2h_n\vec{f}^2(\vec{y}_n + \vec{\rho}_n^*)}{(2+\mu)\vec{f}(\vec{y}_n + \vec{\rho}_n^*) - \mu\vec{f}(\vec{y}_n + \vec{\rho}_n^* + \frac{1}{\mu}h_n\vec{f}(\vec{y}_n + \vec{\rho}_n^*))} =$$

$$\cong \vec{y}_n + \vec{\rho}_n^* + \frac{2h_n\vec{f}^2(\vec{y}_n) + 4h_n\vec{f}(\vec{y}_n)J_n\vec{\rho}_n^* + 2h_n(J_n\vec{\rho}_n^*)^2}{(2+\mu)\vec{f}(\vec{y}_n) - \mu\vec{f}(\vec{y}_n + \frac{1}{\mu}h_n\vec{f}(\vec{y}_n)) + J_n(2 - h_nJ_n)\vec{\rho}_n^*},$$

waarin $J_n = J(\vec{y}_n)$ en waarin termen van de orde $(\vec{\rho}_n^*)^2$ verwaarloosd zijn. Verdere herleiding geeft

$$\vec{y}^{**}_{n+1} \tilde{=} \vec{y}^{*}_{n+1} + [I+h_n D_n(2-D_n)J_n + \tfrac{1}{2}h_n^2(D_nJ_n)^2]\vec{\rho}^{**}_n \ ,$$

waarin D_n een diagonaalmatrix is waarvan de diagonaalelementen gegeven worden door:

$$d_{jj} = \frac{2f_j(\vec{y}_n)}{(2+\mu)f_j(\vec{y}_n)-\mu f_j(\vec{y}_n+\frac{1}{\mu}h_n\vec{f}(\vec{y}_n))} \ .$$

Voor een enkele lineaire vergelijking

$$\frac{dy}{dx} = \delta y$$

geeft deze analyse

$$y^*_{n+1} = y_{n+1} + \frac{2+h_n\delta}{2-h_n\delta}\rho^*_n \ ,$$

waaruit volgt dat voor $Re\delta < 0$ de fout ρ^*_n niet aangroeit hoe groot h_n ook is. Hoe het voor algemene (lineaire) vergelijkingen gesteld is met de stabiliteit is nog niet theoretisch onderzocht; de ervaringen zijn echter bijzonder gunstig (zie Fiolet [1973]).

Tenslotte merken we op dat de parameter μ gebruikt kan worden om eventuele polen van de noemer in (2.9.15) te ontlopen. Stel namelijk dat voor de gekozen waarden van h_n en μ een component van de vector

$$(2+\mu)\vec{f}(\vec{y}_n) - \mu\vec{f}(\vec{y}_n+\frac{1}{\mu}h_n\vec{f}(\vec{y}_n))$$

nul wordt , dan kan men h_n en μ zodanig wijzigen dat h_n/μ constant blijft -dus geen nieuwe functie-evaluatie- terwijl de bewuste component niet meer nul is.

REFERENTIES

BEENTJES, P.A. [1974]: *Some special formulas of the England class of fifth order Runge-Kutta schemes*, NW report, Mathematisch Centrum, Amsterdam (te publiceren).

BEENTJES, P.A. & K. DEKKER [1974]: *Colloquium stijve differentiaalvergelijkingen*, MC Syllabus 15.3, Mathematisch Centrum, Amsterdam.

BUTCHER, J.C. [1963]: *On the integration processes of A. Huta*, J. Austr. Math. Soc. 3, 202-206.

CALAHAN, D.A. [1968]: *A stable, accurate method of numerical integration for non-linear systems*, Proc. IEEE 56, 744-747.

COLLATZ, L. [1968]: *Funktionalanalysis und numerische Mathematik*, Springer-Verlag, Berlin.

CURTISS, C.F. & J.O. HIRSCHFELDER [1952]: *Integration of stiff equations*, Proc. Nat. Acad. Sci. U.S. 38, 235-243.

DAHLQUIST, G. [1968]: *A numerical method for some ordinary differential equations with large Lipschitz constants*, Information proceeding 68, ed. A.J.H. Morrell, North Holland Publishing Co., Amsterdam, 183-186.

DEKKER, K. [1973]: *NUMAL, a library of ALGOL 60 procedures in numerical mathematics*, section 5.2.1.1.1.2, Mathematisch Centrum, Amsterdam.

FIOLET, H. [1973]: *Numerieke integratie van differentiaalvergelijkingen door middel van Padé-benaderingen*, NN 1/73, Mathematisch Centrum, Amsterdam.

FORSYTHE, G.E. & W.R. WASOW [1960]: *Finite difference method for partial differential equations*, John Wiley & Sons, New York.

GEAR, C.W. [1968]: *The automatic integration of stiff ordinary differential equations*, Information Processing 68, 187-194, ed. A.J.H. Morrell, North-Holland Publishing Co., Amsterdam.

GEAR, C.W. [1971]: *Numerical initial value problems in ordinary differential equations*, Prentice Hall, Englewood Cliffs, New Jersey.

HENRICI, P. [1962]: *Discrete variable methods in ordinary differential equations*, John Wiley & Sons, New York.

HEUN, K. [1900]: *Neue Methode zur approximativen Integration der Differential-gleichungen einer unabhängigen Veränderlichen*, Z. Math. Phys. 45, 23-38.

HOUWEN, P.J. VAN DER [1972]: *Explicit Runge-Kutta formulas with increased stability boundaries*, Numer. Math. 20, 149-164.

HOUWEN, P.J. VAN DER [1975]: *Construction of integration formulas for initial value problems*, North-Holland Publishing Co., Amsterdam (te verschijnen).

HOUWEN, P.J. VAN DER, P.A. BEENTJES, K. DEKKER & E. SLAGT [1971]: *One step methods for linear initial value problems III, Numerical results*, TW 130/71, Mathematisch Centrum, Amsterdam.

HOUWEN, P.J. VAN DER & C. DE VREUGD [1970]: *A diffusion problem with a discontinuous initial condition*, TN 58/70, Mathematisch Centrum, Amsterdam.

KAMPEN, S.P.N. VAN [1973]: *NUMAL, a library of ALGOL 60 procedures in numerical mathematics, section 5.2.1.1.1.2*, Mathematisch Centrum, Amsterdam.

KUTTA, W. [1901]: *Beitrag zur näherungsweise Integration totaler Differentialgleichungen*, Z. Math. Phys. 46, 435-453.

LAMBERT, J.D. [1973]: *Computational methods in ordinary differential equations*, John Wiley & Sons, London.

LAMBERT, J.D. & B. SHAW [1965]: *On the numerical solution of* y'=f(x,y) *by a class of formulae based on rational approximations*, Math. Comp. 19, 456-462.

LAPIDUS, L. & J.H. SEINFELD [1971]: *Numerical solution of ordinary differential equations*, Academic Press, New York.

LAUWERIER, H.A. [1967]: *Randwaardeproblemen*, MC Syllabi 3.1, 3.2 en 3.3, Mathematisch Centrum, Amsterdam.

LAWSON, J.D. [1967]: *Generalized Runge-Kutta processes for stable systems with large Lipschitz constants*, SIAM J. Numer. Anal. 4, 372-380.

LINDBERG, B. [1971]: *On smoothing and extrapolation for the trapezoidal rule*, BIT 11, 29-52.

LINNIGER, W. & R.A. WILLOUGHBY [1970]: *Efficient numerical integration of stiff systems of ordinary differential equations*, SIAM J. Numer. Anal. 7, 47-66.

OSTROWSKI, A.M. [1951]: *Sur les conditions générales pour la regularité des matrices*, Rend. di. Math. e delle sue Appl.[V] 10, 156-161.

PETROVSKI, I.C. [1966]: *Ordinary differential equations*, Prentice Hall, Englewood Cliffs, New Jersey.

POPE, D.A. [1963]: *An exponential method of numerical integration of ordinary differential equations*, Comm. ACM 6, 491-493.

RJABENKI, W.S. & A.F. FILIPPOV [1960]: *Über die Stabilität von Differenzengleichungen*, Deutscher Verlag der Wissenschaften, Berlin.

ROBERTSON, H.H. [1967]: *The solution of a set of reaction-rate equations in numerical analysis*, ed. J. Walsh, Thompson Book C., Washington.

ROSENBROCK, H.H. [1963]: *Some general implicit processes for the numerical solution of differential equations*, Comput. J. 5, 329-330.

RUNGE, C. [1895]: *Über die numerische Auflösung von Differentialgleichungen*, Math. Ann. 46, 167-178.

TREANOR, C.E. [1966]: *A method for the numerical integration of coupled first-order differential equations with greatly different time constants*, Math. Comp. 20, 39-45.

VARGA, R.S. [1962]: *Matrix iterative analysis*, Prentice Hall, Englewood cliffs, New Jersey.

WILKINSON, J.H. [1965]: *The algebraic eigenvalue problem*, Clarendon Press, Oxford.

ZONNEVELD, J.A. [1964]: *Automatic numerical integration*, MC tract 8, Mathematisch Centrum, Amsterdam.

UITGAVEN IN DE SERIE MC SYLLABUS

Onderstaande uitgaven zijn verkrijgbaar bij het Mathematisch Centrum, 2e Boerhaavestraat 49 te Amsterdam-1005, tel. 020-947272.

MCS 1.1 F. GOBEL & J. VAN DE LUNE, *Leergang Besliskunde, deel 1: Wiskundige basiskennis*, 1965.

MCS 1.2 J. HEMELRIJK & J. KRIENS, *Leergang Besliskunde, deel 2: Kansberekening*, 1965.

MCS 1.3 J. HEMELRIJK & J. KRIENS, *Leergang Besliskunde, deel 3: Statistiek*, 1966.

MCS 1.4 G. DE LEVE & W. MOLENAAR, *Leergang Besliskunde, deel 4: Markovketen, en wachttijden*, 1966.

MCS 1.5 G. DE LEVE & J. KRIENS, *Leergang Besliskunde, deel 5: Inleiding tot de mathematische besliskunde*, 1966.

MCS 1.6a B. DORHOUT & J. KRIENS, *Leergang Besliskunde, deel 6a: Wiskundige programmering 1*, 1968.

MCS 1.7a G. DE LEVE, *Leergang Besliskunde, deel 7a: Dynamische programmering 1*, 1968.

MCS 1.7b G. DE LEVE & H.C. TIJMS, *Leergang Besliskunde, deel 7b: Dynamische programmering 2*, 1970.

MCS 1.7c G. DE LEVE & H.C. TIJMS, *Leergang Besliskunde, deel 7c: Dynamische programmering 3*, 1971.

MCS 1.8 J. KRIENS, F. GOBEL & W. MOLENAAR, *Leergang Besliskunde, deel 8: Minimaxmethode, netwerkplanning, simulatie*, 1968.

MCS 2.1 G.J.R. FORCH, P.J. VAN DER HOUWEN & R.P. VAN DE RIET, *Colloquium stabiliteit van differentieschema's, deel 1*, 1967.

MCS 2.2 L. DEKKER, T.J. DEKKER, P.J. VAN DER HOUWEN & M.N. SPIJKER, *Colloquium stabiliteit van differentieschema's, deel 2*, 1968.

MCS 3.1 H.A. LAUWERIER, *Randwaardeproblemen, deel 1*, 1967.

MCS 3.2 H.A. LAUWERIER, *Randwaardeproblemen, deel 2*, 1968.

MCS 3.3 H.A. LAUWERIER, *Randwaardeproblemen, deel 3*, 1968.

MCS 4 H.A. LAUWERIER, *Representaties van groepen*, 1968.

MCS 5 J.H. VAN LINT, J.J. SEIDEL, P.C. BAAYEN, *Colloquium discrete wiskunde*, 1968.

MCS 6 K.K. KOKSMA, *Cursus ALGOL 60*, 1969.

MCS 7.1 *Colloquium moderne rekenmachines, deel 1*, 1969.

MCS 7.2 *Colloquium moderne rekenmachines, deel 2*, 1969.

MCS 8 H. BAVINCK & J. GRASMAN, *Relaxatietrillingen*, 1969.

MCS 9.1 T.M.T. COOLEN, G.J.R. FORCH, E.M. DE JAGER & H.G.J. PIJLS, *Colloquium elliptische differentiaalvergelijkingen, deel 1*, 1969.

MCS 9.2 W.P. VAN DE BRINK, T.M.T. COOLEN, B. DIJKHUIS, P.P.N. DE GROEN, P.J. VAN DER HOUWEN, E.M. DE JAGER, N.M. TEMME & R.J. DE VOGELAERE, *Colloquium elliptische differentiaalvergelijkingen, deel 2*, 1970.

MCS 10 J. FABIUS & W.R. VAN ZWET, *Grondbegrippen van de waarschijnlijkheidsrekening*, 1970.

MCS 11 H. BART, M.A. KAASHOEK, H.G.J. PIJLS, W.J. DE SCHIPPER & J. DE VRIES, *Colloquium halfalgebra's en positieve operatoren*, 1971.

MCS 12 T.J. DEKKER, *Numerieke algebra*, 1971.

MCS 13 F.E.J. KRUSEMAN ARETZ, *Programmeren voor rekenautomaten; De MC ALGOL 60 vertaler voor de EL X8*, 1971.

MCS 14 H. BAVINCK, W. GAUTSCHI & G.M. WILLEMS, *Colloquium approximatietheorie*, 1971.

MCS 15.1 T.J. DEKKER, P.W. HEMKER & P.J. VAN DER HOUWEN, *Colloquium stijve differentiaalvergelijkingen, deel 1*, 1972.

MCS 15.2 P.A. BEENTJES, K. DEKKER, H.C. HEMKER, S.P.N. VAN KAMPEN & G.M. WILLEMS, *Colloquium stijve differentiaalvergelijkingen, deel 2*, 1973.

* MCS 15.3 P.A. BEENTJES e.a., *Colloquium stijve differentiaalvergelijkingen, deel 3.*

MCS 16.1 L. GEURTS, *Cursus programmeren, deel 1: De elementen van het programmeren*, 1973.

MCS 16.2 L. GEURTS, *Cursus programmeren, deel 2: De programmeertaal ALGOL 60*, 1973.

MCS 17.1 P.S. STOBBE, *Lineaire algebra, deel 1*, 1974.

MCS 17.2 P.S. STOBBE, *Lineaire algebra, deel 2*, 1974.

MCS 18 F. VAN DER BLIJ, H. FREUDENTHAL, J.J. DE IONGH, J.J. SEIDEL & A. VAN WIJNGAARDEN, *Een kwart eeuw wiskunde 1946-1971, Syllabus van de Vakantiecursus 1971*, 1974.

MCS 19 A. HORDIJK, R. POTHARST & J. TH. RUNNENBURG, *Optimaal stoppen van Markovketens*, 1974.

* MCS 20 T.M.T. COOLEN, P.W. HEMKER, P.J. VAN DER HOUWEN & E. SLAGT, *ALGOL 60 procedures voor begin- en randwaardeproblemen.*

* MCS 21 L.J.M. GEURTS, D. GRUNE, Z. MANNA, L.G.L.T. MEERTENS & W.P. DE ROEVER, *Colloquium Programmacorrectheid.*

* MCS 22 R. HELMERS, J. OOSTERHOFF, F.H. RUYMGAART & M.C.A. VAN ZUYLEN, *Werkweek Statistiek* 1973.

* MCS 23.1 J. GRASMAN, J.H.B. KEMPERMAN, J.W. DE ROEVER & G.M. WILLEMS, *Colloquium Onderwerpen uit de Biomathematica, deel 1.*

* MCS 23.2 J. GRASMAN, J.H.B. KEMPERMAN, J.W. DE ROEVER & G.M. WILLEMS, *Colloquium Onderwerpen uit de Biomathematica, deel 2.*

MCS 24.1 P.J. VAN DER HOUWEN, *Numerieke integratie van differentiaalvergelijkingen, deel 1: Eenstapsmethoden*, 1974.

* MCS 25 *Colloquium Structuur Programmeertalen.*

De met een * gemerkte uitgaven moeten nog verschijnen.