stichting
mathematisch
centrum

MC

NUMERIEKE WISKUNDE

NN 7/76 MAART

P.H.M. WOLKENFELT

TAYLOR-RUNGE-KUTTA METHODEN

2e boerhaavestraat 49 amsterdam

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.

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

Taylor-Runge-Kutta methoden

door

P.H.M. Wolkenfelt

VOORWOORD

Dit verslag beschrijft de afleiding van Taylor-Runge-Kutta formules voor het oplossen van beginwaardeproblemen. De ALGOL-60 programma's zijn getest op de CONTROL DATA CYBER 73-28 van de Stichting Academisch Rekencentrum Amsterdam (SARA).

Het onderzoek werd aan de Universiteit van Amsterdam verricht onder leiding van Prof. P.J. van der Houwen.

Aangezien de onderzochte methoden grote verwantschap hebben met op het MC gedane onderzoekingen op het gebied van integratieformules voor beginwaardeproblemen is dit verslag gepubliceerd in de Numerieke Notitie-serie van het MC. De schrijver is de Directie van het MC erkentelijk voor de hem geboden publicatie-mogelijkheid.

KEY WORDS & PHRASES: *Numerical analysis, Ordinary differential equations, Initial value problems, Taylor-Runge-Kutta methods, Runge-Kutta methods.*

INHOUD

1. INLEIDING

We beschouwen het beginwaardeprobleem

$$(1.1) \qquad \frac{dy}{dx} = F(y) \ , \qquad y(x_0) = y_0 \ .$$

Hierin is $x \in \mathbb{R}$ de onafhankelijk veranderlijke en $F \in \mathbb{R}^r \to \mathbb{R}^r$ een (vector) functie, die voldoende vaak continu differentieerbaar is naar y. De oplossing y is een (vector) functie $\in \mathbb{R} \to \mathbb{R}^r$. Hierbij geeft r het aantal componenten aan.

Voor het numeriek oplossen van (1.1) is een groot aantal methoden ontwikkeld. Hierbij nemen de eenstapsmethoden, formeel gedefinieerd door

$$(1.2) \qquad y_{n+1} = E_n(y_n) \ , \qquad n = 0,1,2,\ldots$$

een belangrijke plaats in. Als van de rechterlidfunctie F(y) de hogere afgeleiden bekend zijn, kunnen Taylor methoden gebruikt worden (zie bijv. GEAR [1971]). Zijn er slechts enkele opvolgende afgeleiden bekend, te weinig om een voldoend hoge orde van nauwkeurigheid te krijgen, dan kan men Runge-Kutta formules gebruiken. Deze zijn gebaseerd op herhaalde evaluatie van F(y). Hierin wordt echter de informatie van de hogere afgeleiden niet gebruikt. Taylor-Runge-Kutta methoden gebruiken ook deze informatie, omdat ze gebaseerd zijn op herhaalde evaluatie van zowel de rechterlidfunctie F(y) als de hogere afgeleiden. Bij het onderzoek van deze methoden zullen we ons beperken tot het geval, waarbij slechts F(y) en $G(y) = \frac{d^2y}{dx^2}$ bekend zijn. Dergelijke formules komen we reeds tegen bij HENRICI [1962] en CESCHINO & KUNTZMANN [1963]. In het bijzonder zullen we methoden beschouwen waarvan het stabiliteitsgedrag nog aangepast kan worden aan de te integreren differentiaalvergelijking (vgl. VAN DER HOUWEN [1971]).

In het volgende hoofdstuk behandelen we enkele theoretische aspecten en zullen formules afleiden voor een deelklasse, die uit de algemene klasse van Taylor-Runge-Kutta formules ontstaat door een beperkt geheugengebruik te eisen. In het derde hoofdstuk worden de hierbij behorende ALGOL-60 implementaties gedocumenteerd. De methoden worden in hoofdstuk 4 getest op een aantal beginwaardeproblemen en vergeleken met bestaande Runge-Kutta

methoden. In de appendix tenslotte zijn ten behoeve van de geïnteresseerde lezer enkele afleidingen opgenomen.

2. TAYLOR-RUNGE-KUTTA METHODEN

In de eerste drie paragrafen van dit hoofdstuk geven we een beknopt overzicht van de algemene theorie van Taylor-Runge-Kutta methoden. In de daarop volgende paragrafen zullen we een bepaalde deelklasse beschouwen. Hierbij komen ter sprake consistentie, stabiliteit, vergelijking met Runge-Kutta methoden en schattingen voor de locale discretiseringsfout. Tenslotte worden formules voor locale foutschattingen afgeleid, welke ten grondslag liggen aan het stapkeuzemechanisme van de procedure STABTARK2VS.

2.1. Algemene structuur van Taylor-Runge-Kutta schema's

In het geval van een m-punts Taylor-Runge-Kutta formule wordt de eenstapsoperator $E_n(y_n)$ uit (1.2) beschreven door het volgende schema:

$$
(2.1)\qquad \begin{cases} y_{n+1}^{(0)} = y_n \\ y_{n+1}^{(j)} = y_n + h_n \sum_{\ell=0}^{j-1} \lambda_{j,\ell} F\left(y_{n+1}^{(\ell)}\right) + h_n^2 \sum_{\ell=0}^{j-1} \mu_{j,\ell} G\left(y_{n+1}^{(\ell)}\right), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j = 1,\ldots,m \\ y_{n+1} = y_{n+1}^{(m)} \end{cases}
$$

Hierin is y_n de numerieke oplossing in het punt $x = x_n$ en h_n de lengte van de $(n+1)^e$ integratiestap. Verder geldt

$$
G(y) = \frac{d^2y}{dx^2} = \frac{dF(y)}{dx} \quad .
$$

Schema (2.1) kan worden gekarakteriseerd door de "parametermatrices" $(\lambda_{j,\ell})$ en $(\mu_{j,\ell})$:

$$(2.2)\quad (\lambda_{j,\ell}) = \begin{pmatrix} 0 & \cdots & 0 \\ \lambda_{1,0} & 0 & \vdots \\ \lambda_{2,0} & \lambda_{2,1} & \ddots \\ \vdots & & 0 \\ \lambda_{m,0} & \cdots & \lambda_{m,m-1} \end{pmatrix} \text{ en } (\mu_{j,\ell}) = \begin{pmatrix} 0 & \cdots & 0 \\ \mu_{1,0} & 0 & \vdots \\ \mu_{2,0} & \mu_{2,1} & \ddots \\ \vdots & & 0 \\ \mu_{m,0} & \cdots & \mu_{m,m-1} \end{pmatrix}$$

2.2 Consistentievoorwaarden

Zij $y(x_n)$ de analytische oplossing van (1.1) in het punt $x = x_n$. We noemen schema (2.1) p^e orde consistent, indien

$$(2.3)\qquad y(x_{n+1}) - E_n(y(x_n)) = O(h_n^{p+1}) \ , \qquad h_n \to 0 \ .$$

Het linkerlid van (2.3) heet de locale afbreekfout. Ontwikkelen we $E_n(y(x_n))$ in een Taylorreeks in $x = x_n$ en identificeren we de eerste $(p+1)$ termen van deze reeks met de Taylorreeks van $y(x_{n+1})$ in $x = x_n$, dan krijgen we de voorwaarden voor p^e orde consistentie (zie Appendix).

STELLING 2.1. *Het schema* (2.1) *is consistent van de orde* 1, *als*

$$(2.4)\qquad \sum_{j=0}^{m-1} \lambda_{m,j} = 1 \ ,$$

en van de orde 2 *als bovendien*

$$(2.5)\qquad \sum_{j=1}^{m-1} \lambda_{m,j} \sum_{k=0}^{j-1} \lambda_{j,k} + \sum_{j=0}^{m-1} \mu_{m,j} = 1/2,$$

en van de orde 3 *als bovendien*

$$(2.6)\qquad \sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=1}^{j-1} \lambda_{j,k} \sum_{\ell=0}^{k-1} \lambda_{k,\ell} + \sum_{j=1}^{m-1} \lambda_{m,j} \sum_{k=0}^{j-1} \mu_{j,k} + \sum_{j=1}^{m-1} \mu_{m,j} \sum_{k=0}^{j-1} \lambda_{j,k} = 1/6$$

$$(2.7)\qquad \sum_{j=1}^{m-1} \lambda_{m,j} \left(\sum_{k=0}^{j-1} \lambda_{j,k} \right)^2 + 2 \sum_{j=1}^{m-1} \mu_{m,j} \sum_{k=0}^{j-1} \lambda_{j,k} = 1/3 \ ,$$

en van de orde 4 als bovendien

$$(2.8)\qquad \begin{aligned} &\sum_{j=3}^{m-1} \lambda_{m,j} \sum_{k=2}^{j-1} \lambda_{j,k} \sum_{\ell=1}^{k-1} \lambda_{k,\ell} \sum_{i=0}^{\ell-1} \lambda_{\ell,i} + \sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=1}^{j-1} \lambda_{j,k} \sum_{\ell=0}^{k-1} \mu_{k,\ell} + \\ &+ \sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=1}^{j-1} \mu_{j,k} \sum_{\ell=0}^{k-1} \lambda_{k,\ell} + \sum_{j=2}^{m-1} \mu_{m,j} \sum_{k=1}^{j-1} \lambda_{j,k} \sum_{\ell=0}^{k-1} \lambda_{k,\ell} + \\ &+ \sum_{j=1}^{m-1} \mu_{m,j} \sum_{k=0}^{j-1} \mu_{j,k} = 1/24 \end{aligned}$$

$$(2.9)\qquad \begin{aligned} &\sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=1}^{j-1} \lambda_{j,k} \left(\sum_{\ell=0}^{k-1} \lambda_{k,\ell} \right)^2 + 2 \sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=1}^{j-1} \mu_{j,k} \sum_{\ell=0}^{k-1} \lambda_{k,\ell} + \\ &+ \sum_{j=1}^{m-1} \mu_{m,j} \left(\sum_{k=0}^{j-1} \lambda_{j,k} \right)^2 = 1/12 \end{aligned}$$

$$(2.10)\qquad \begin{aligned} &\sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=0}^{j-1} \lambda_{j,k} \sum_{\ell=1}^{j-1} \lambda_{j,\ell} \sum_{i=0}^{\ell-1} \lambda_{\ell,i} + \sum_{j=1}^{m-1} \lambda_{m,j} \sum_{k=0}^{j-1} \lambda_{j,k} \sum_{k=0}^{j-1} \mu_{j,k} + \\ &+ \sum_{j=1}^{m-1} \mu_{m,j} \sum_{k=0}^{j-1} \mu_{j,k} + \sum_{j=2}^{m-1} \mu_{m,j} \sum_{k=1}^{j-1} \lambda_{j,k} \sum_{\ell=0}^{k-1} \lambda_{k,\ell} + \\ &+ \sum_{j=1}^{m-1} \mu_{m,j} \left(\sum_{k=0}^{j-1} \lambda_{j,k} \right)^2 = 1/8 \end{aligned}$$

$$(2.11)\qquad \sum_{j=1}^{m-1} \lambda_{m,j} \left(\sum_{k=0}^{j-1} \lambda_{j,k} \right)^3 + 3 \sum_{j=1}^{m-1} \mu_{m,j} \left(\sum_{k=0}^{j-1} \lambda_{j,k} \right)^2 = 1/4$$

Is de differentiaalvergelijking lineair, dan is schema (2.1):
3^e orde consistent als voldaan is aan (2.4), (2.5) *en* (2.6);
4^e orde consistent als voldaan is aan (2.4), (2.6) *en* (2.8).

BEWIJS. Zie appendix.

Stellen we in (2.5 - 2.11) alle $\mu_{j,\ell}$ gelijk aan nul, dan krijgen we de consistentievoorwaarden tot en met orde 4 van de traditionele Runge-Kutta methoden.

2.3. Stabiliteit

Beschouw de volgende lineaire differentiaalvergelijking

$$\frac{dy}{dx} = D\,y \,, \tag{2.12}$$

waarin D een matrix is, onafhankelijk van y, met eigenwaarden δ_i, waarvoor geldt $Re(\delta_i) \leq 0, \forall_i$. Toepassing van schema (2.1) op (2.12) levert

$$y_{n+1} = P(h_n D) y_n \,. \tag{2.13}$$

Hierin wordt P(z) recursief gedefinieerd door

$$\begin{cases} P^{(0)}(z) = 1 \\ P^{(j)}(z) = 1 + z \sum_{\ell=0}^{j-1} \lambda_{j,\ell}\, R^{(\ell)}(z) + z^2 \sum_{\ell=0}^{j-1} \mu_{j,\ell}\, R^{(\ell)}(z) \,, \quad j=1,\ldots,m \\ P(z) = P^{(m)}(z) \,. \end{cases} \tag{2.14}$$

P(z) heet de stabiliteitsfunctie van (2.1) en beschrijft voor lineaire vergelijkingen exact de wijze waarop fouten, gemaakt tijdens het integratieproces, accumuleren (zie bijv. VAN DER HOUWEN [1970]). De graad van het polynoom P is maximaal 2m namelijk wanneer alle $\mu_{j,j-1}$ ongelijk nul zijn; indien slechts $\mu_{1,0} = 0$ is de graad 2m-1. We schrijven nu P(z) als

$$P(z) = 1 + \beta_1 z + \ldots + \beta_{2m-1} z^{2m-1} + \beta_{2m} z^{2m} \,,$$

waarin de coëfficiënten β_i op grond van (2.14) functies zijn van de Taylor-Runge-Kutta parameters $\lambda_{j,\ell}$ en $\mu_{j,\ell}$ uit (2.2). Omgekeerd kunnen we, uitgaande van een polynoom P(z), de parameters $\lambda_{j,\ell}$ en $\mu_{j,\ell}$ uitdrukken in de coëfficiënten β_i en aldus een Taylor-Runge-Kutta schema opstellen (adaptiviteit).
Wanneer voor alle $z_i \stackrel{p.d}{=} h_n\delta_i$ geldt

$$|P(z_i)| \leq 1 \ ,$$

dan noemen we dit schema stabiel. Het begrip stabiliteitsgebied, evenals de begrippen reële (β_{real}), imaginaire (β_{im}) en absolute (β_{abs}) stabiliteitsgrens worden o.a. gedefinieerd in VAN DER HOUWEN [1974]. Deze begrippen hebben betrekking op het stabiliteitspolynoom en zijn dus onafhankelijk van de differentiaalvergelijking.

Voor niet-lineaire vergelijkingen is deze lineaire stabiliteitstheorie toepasbaar op de locaal-gelineariseerde voorstelling van (1.1). Hierop gebaseerde beschouwingen gelden dus slechts locaal. De eigenwaarden δ_i van de Jacobiaan $J(y_n) = \left(\frac{\partial f_k}{\partial y_j} \mid y=y_n\right)_{k,j}$ zijn dan bepalend voor locale stabiliteit:
Zij $\sigma(J(y_n))$ de spectraalradius van $J(y_n)$, dan moet de integratiestap h_n voldoen aan de "stabiliteitsvoorwaarde"

$$h_n \leq \frac{\beta}{\sigma(J(y_n))} \tag{2.15}$$

om locale stabiliteit te garanderen; hierin is, afhankelijk van het eigenwaardenspectrum van $J(y_n)$, β gelijk aan β_{real}, β_{im} of β_{abs}.

2.4. Consistentievoorwaarden en stabiliteitsfunctie van een deelklasse

We willen nu aan de volgende eisen voldoen:

a) een beperkt geheugengebruik;
b) een maximale graad van het stabiliteitspolynoom;
c) een beperkt aantal evaluaties van F(y) en G(y).

Dit is te bereiken door in de matrices (2.2) te stellen:

$$\lambda_{j,\ell} = 0 \quad \text{voor} \quad j = 2,\ldots,m; \quad \ell = 1,\ldots,j-1$$
$$\text{en } \mu_{j,\ell} = 0 \quad \text{voor} \quad j = 2,\ldots,m; \quad \ell = 0,\ldots,j-2.$$

In dit geval wordt (2.1)

$$(2.16)\qquad \begin{cases} y_{n+1}^{(0)} = y_n \\ y_{n+1}^{(j)} = y_n + h_n\lambda_{j,0}F(y_n) + h_n^2\mu_{j,j-1}G\left(y_{n+1}^{j-1}\right), \quad j=1,\ldots,m \\ y_{n+1} = y_{n+1}^{(m)} \end{cases}$$

Dat wil zeggen 1 evaluatie van F(y) en (indien $\mu_{j,j-1} \neq 0$, $j=1,\ldots,m$) m evaluaties van G(y) en een werkruimte van 3 array's. De matrices uit (2.2) krijgen nu de gedaante:

$$(2.17)\qquad (\lambda_{j,\ell}) = \begin{pmatrix} 0 & \cdots & 0 \\ \lambda_{1,0} & \ddots & \vdots \\ \vdots & \ddots & 0 \\ \lambda_{m,0} & 0 \cdots & 0 \end{pmatrix} \quad \text{en} \quad (\mu_{j,\ell}) = \begin{pmatrix} 0 & \cdots & 0 \\ \mu_{1,0} & \ddots & \vdots \\ 0 & \ddots & 0 \\ 0 \cdots & 0 & \mu_{m,m-1} \end{pmatrix}$$

De aldus ontstane deelklasse bestaat uit 2m parameters. De consistentie-voorwaarden (2.4 - 2.11) reduceren nu tot

$$(2.18)\qquad \lambda_{m,0} = 1 \quad \text{voor orde 1.}$$

$$(2.19)\qquad \lambda_{m,0} = 1 \;;\; \mu_{m,m-1} = 1/2 \quad \text{voor orde 2.}$$

$$(2.20)\qquad \lambda_{m,0} = 1 \;;\; \mu_{m,m-1} = 1/2 \;;\; \lambda_{m-1,0} = 1/3 \quad \text{voor orde 3.}$$

$$(2.21)\qquad \begin{cases} \lambda_{m,0} = 1 \;;\; \mu_{m,m-1} = 1/2 \;;\; \lambda_{m-1,0} = 1/3 \;;\; \mu_{m,m-1} = 1/12 \;; \\ \mu_{m,m-1}\,\lambda_{m-1,0}^2 = 1/12 \quad \text{voor orde 4.} \end{cases}$$

De laatste vergelijking in (2.21) is strijdig met de overige. Orde 4 blijkt binnen deze klasse niet mogelijk, tenzij de beschouwde differentiaalvergelijking lineair is. Voor een orde p, p = 1,2,3 zijn er wegens (2.18 -

2.20) nog (2m-p) parameters vrij te kiezen, waarvan we in de volgende paragraaf gebruik zullen maken.
Toepassing van (2.14) levert de volgende stabiliteitsfunctie behorende bij schema (2.16)

$$(2.22) \qquad P(z) = 1 + \sum_{i=1}^{m} \left[\lambda_{m-i+1,0} \prod_{j=m}^{m-i+2} \mu_{j,j-1} \, z^{2i-1} + \prod_{j=m}^{m-i+1} \mu_{j,j-1} \, z^{2i} \right]$$

2.5. Genererende matrices van Taylor-Runge-Kutta schema's van de orde p, p = 1, 2 en 3

Identificatie van (2.22) met een voorgeschreven stabiliteitspolynoom $\beta_0 + \beta_1 z + \ldots + \beta_{2m-1} z^{2m-1} + \beta_{2m} z^{2m}$, waarin $\beta_0 = 1$, levert de volgende waarden van de Taylor-Runge-Kutta parameters, uitgedrukt in de coëfficiënten β_j

$$(2.23) \qquad \begin{cases} \lambda_{m-i+1,0} = \beta_{2i-1}/\beta_{2i-2} \\ \mu_{m-i+1,m-i} = \beta_{2i}/\beta_{2i-2} \end{cases} \qquad i = 1,\ldots,m \ .$$

Met behulp van (2.18 - 2.20) en (2.23) is eenvoudig te verifiëren, dat een 1^e, 2^e respectievelijk 3^e orde consistent stabiliteitspolynoom, (zoals gedefinieerd in VAN DER HOUWEN [1974], blz. 69) een 1^e, 2^e respectievelijk 3^e orde consistente Taylor-Runge-Kutta methode genereert. De matrices (2.17) worden nu

$$(2.24) \qquad (\lambda_{j,\ell}) = \begin{bmatrix} 0 & & \\ \beta_{2m-1}/\beta_{2m-2} & \ddots & \\ \vdots & & \\ \beta_1/\beta_0 & & \end{bmatrix} \quad \text{en} \quad (\mu_{j,\ell}) = \begin{bmatrix} 0 & \cdots & 0 \\ \beta_{2m}/\beta_{2m-2} & \ddots & \vdots \\ 0 & \ddots & 0 \\ \vdots & \ddots & \\ 0 \cdots 0 & & \beta_2/\beta_0 \end{bmatrix}$$

Voorbeeld 2.1. Het 2^e orde consistente stabiliteitspolynoom

$$P(z) = 1 + z + \frac{1}{2}z^2 + \frac{3}{16}z^3 + \frac{1}{32}z^4 + \frac{1}{128}z^5$$

met maximale imaginaire stabiliteitsgrens $\beta_{im} = 4$ (zie VAN DER HOUWEN [1971]) genereert de volgende matrices

$$(\lambda_{j,\ell}) = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 \\ \frac{3}{8} & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad \text{en} \quad (\mu_{j,\ell}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{16} & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}$$

De eenstapsoperator $E_n(y_n)$ uit (1.2) heeft in dit geval de gedaante

$$E_n(y_n) \equiv y_n + h_n F(y_n) + \tfrac{1}{2}h_n^2 G(y_n + \tfrac{3}{8}h_n F(y_n) + \tfrac{1}{16}h_n^2 G(y_n + \tfrac{1}{4}h_n F(y_n)))$$

De methode is 2^e orde consistent (vgl. (2.19)).

2.6. Kosten van Taylor-Runge-Kutta formules vergeleken met Runge-Kutta formules

In §2.4 hebben we gezien, dat we met 1 evaluatie van F(y) en m evaluaties van G(y) beschikken over een stabiliteitspolynoom van de graad 2m; in het geval $\mu_{1,0} = 0$ krijgen we met 1 evaluatie van F(y) en (m-1) evaluaties van G(y) een graad (2m-1). In onderstaande tabel 2.1 geven we een vergelijking tussen Taylor-Runge-Kutta formules en Runge-Kutta formules wat betreft het aantal evaluaties van F(y) en G(y) bij een vaste graad van het stabiliteitspolynoom. Het blijkt hierbij zinvol onderscheid te maken tussen even en oneven graad. Dit zal voortaan aangeduid worden met $\mu_{1,0} \neq 0$ (even graad) en $\mu_{1,0} = 0$ (oneven graad). Met e(F) resp. e(G) bedoelen we het aantal evaluaties van F(y) resp. G(y) per integratiestap.

GRAAD	METHODE	e(F)	e(G)
2m	R.K.	2m	0
	T.R.K.	1	m
2m-1	R.K.	2m-1	0
	T.R.K.	1	m-1

Tabel 2.1.
Vergelijking tussen Taylor-Runge-Kutta en Runge-Kutta formules

Stellen nu c(F) resp. c(G) de kosten voor het berekenen van F(y) resp. G(y) voor, dan is een Taylor-Runge-Kutta formule voordeliger dan een Runge-Kutta formule als:

$$\text{(2.25)} \qquad \begin{aligned} &\text{a)}\ c(G) < \frac{2m-1}{m}\, c(F) \quad \text{en} \quad \mu_{1,0} \neq 0 \\ &\text{b)}\ c(G) < 2\ c(F) \quad \text{en} \quad \mu_{1,0} = 0. \end{aligned}$$

Een van deze voordelen is, dat men met dezelfde hoeveelheid rekentijd per integratiestap over grotere stabiliteitsgebieden kan beschikken. We merken nog op, dat met 1 evaluatie van F(y) en m evaluaties van G(y) graad (2m+1) bereikt kan worden. Dat er inderdaad belangrijke klassen van differentiaalvergelijkingen bestaan, waarvoor (2.25) geldt, zal blijken bij de testvoorbeelden.

2.7. Schattingen van de locale discretiseringsfout

Zij $z(x_n,y_n;x)$ de locaal-analytische oplossing van (1.1) door het punt (x_n,y_n). We definieren nu de locale discretiseringsfout ρ_n

$$\text{(2.26)} \qquad \rho_n \overset{p.d.}{=} z(x_n,y_n;x_{n+1}) - y_{n+1}.$$

Omdat $z(x_n,y_n;x)$ in het algemeen niet bekend is, benaderen we ρ_n door met een referentieformule een referentieoplossing $\tilde{y}_{n+1}$ te bepalen, die nauwkeuriger is dan de numerieke oplossing, zodat geldt

$$\rho_n \simeq \tilde{\rho}_n \overset{p.d.}{=} \tilde{y}_{n+1} - y_{n+1}.$$

Analoog aan VAN DER HOUWEN [1974] (blz. 147-148) beschouwen we referentieformules, die voortgebracht worden door de volgende matrices

$$(2.27) \qquad (\tilde{\lambda}_{j,\ell}) = \left[\begin{array}{ccccc} 0 & - & - & - & 0 & 0 \\ \lambda_{1,0} & 0 & & & & \\ & & \ddots & & 0 & \\ \lambda_{m,0} & 0 & - & - & 0 & 0 \\ \hline \tilde{\lambda}_{m,0} & 0 & - & - & \tilde{\lambda}_{m,m} & \end{array}\right] \quad \text{en} \quad (\tilde{\mu}_{j,\ell}) = \left[\begin{array}{ccccc} 0 & - & - & - & 0 & 0 \\ \mu_{1,0} & 0 & & & & \\ 0 & \ddots & \ddots & & 0 & \\ 0 & - & - & - & \mu_{m,m-1} & 0 \\ \hline \tilde{\mu}_{m,0} & - & - & - & \tilde{\mu}_{m,m-1} & \tilde{\mu}_{m,m} \end{array}\right]$$

waarin de matrices bestaande uit de eerste (m+1) rijen de gebruikte Taylor-Runge-Kutta formule voorstelt. De formule voor de referentieoplossing luidt

$$(2.28) \qquad \begin{aligned} \tilde{y}_{n+1} = y_n &+ h_n \tilde{\lambda}_{m,0} F(y_{n+1}^{(0)}) + h_n \tilde{\lambda}_{m,m} F(y_{n+1}^{(m)}) + \\ &+ h_n^2 \sum_{j=0}^{m} \tilde{\mu}_{m,j} G(y_{n+1}^{(j)}). \end{aligned}$$

Indien genomen stappen niet worden verworpen, geldt dat deze formule geen extra evaluatie van F en G vereist; immers $F(y_{n+1}^{(m)}) = F(y_{n+1}) = F(y_{n+2}^{(0)})$ resp. $G(y_{n+1}) = G(y_{n+2}^{(0)})$, tenzij de graad van het stabiliteitspolynoom oneven is. Daarom stellen we $\tilde{\mu}_{m,0} = \tilde{\mu}_{m,m} = 0$ als $\mu_{1,0} = 0$.
De referentieformule heeft m + 3 resp. m + 1 vrij te kiezen parameters als $\mu_{1,0} \neq 0$ resp. $\mu_{1,0} = 0$. Met behulp van (2.4-2.11) kunnen de consistentievoorwaarden bij schema (2.27) opgesteld worden. Deze vergelijkingen blijken lineair in de parameters $\tilde{\lambda}_{m,0}$, $\tilde{\lambda}_{m,m}$ en $\tilde{\mu}_{m,j}$, j = 0,...,m. Zij nu p de orde van de gebruikte Taylor-Runge-Kutta formule (2.16) en $\tilde{p}$ de orde van de referentieformule en $c(\tilde{p})$ het aantal consistentievoorwaarden voor een $\tilde{p}^{e}$ orde referentieformule, dan kan $c(\tilde{p})$ de volgende waarden aannemen:

$\tilde{p}$	1	2	3	4	
$c(\tilde{p})$	1	2	4	6	als p = 1
$c(\tilde{p})$	1	2	3	6	als p = 2,3

Hieruit volgt, dat een $\tilde{p}^e$ orde referentieformule geconstrueerd kan worden als $m + 3 \geq c(\tilde{p})$ en $\mu_{1,0} \neq 0$ of als $m + 1 \geq c(\tilde{p})$ en $\mu_{1,0} = 0$.

2.8. Formules voor locale foutschattingen voor het geval p = 2

Zij $P(z) = 1 + \beta_1 z + \ldots + \beta_{2m} z^{2m}$ het stabiliteitspolynoom van de gebruikte Taylor-Runge-Kutta formule. Toepassing van de theorie uit §2.7 levert voor het geval p = 2 de volgende formules voor een schatting van de locale discretiseringsfout. We onderscheiden hierbij de gevallen $\mu_{1,0} \neq 0$ en $\mu_{1,0} = 0$.

A. Het geval $\mu_{1,0} \neq 0$

$$(2.29) \qquad \begin{cases} \tilde{\lambda}_{m,0} = 1 - \tilde{\lambda}_{m,m};\ \tilde{\mu}_{m,0} = \frac{1}{3} - \frac{1}{2}\tilde{\lambda}_{m,m};\ \tilde{\mu}_{m,m} = \frac{1}{6} - \frac{1}{2}\tilde{\lambda}_{m,m}; \\ \tilde{\mu}_{m,j} = 0 \quad \text{voor} \quad j = 1,\ldots,m-1 \end{cases}$$

waarbij

$$(2.30a) \qquad \tilde{\lambda}_{m,m} = \frac{1}{6 - 24\beta_3} \quad \text{als} \quad m \geq 2 \text{ en } \left|\beta_3 - \tfrac{1}{4}\right| \geq \frac{1}{24}$$

of

$$(2.30b) \qquad \tilde{\lambda}_{m,m} = \frac{1}{2} \quad \text{anders.}$$

De referentieformule heeft consistentieorde 3 en in geval (2.30a) zelfs orde 4 voor lineaire differentiaalvergelijkingen.

B. Het geval $\mu_{1,0} = 0$

$$(2.31) \qquad \begin{cases} \tilde{\lambda}_{m,0} = \frac{1}{2} + \tilde{\mu}_{m,m-1};\ \tilde{\lambda}_{m,m} = \frac{1}{2} - \tilde{\mu}_{m,m-1};\ \tilde{\mu}_{m,m} = 0; \\ \tilde{\mu}_{m,j} = 0 \quad \text{voor} \quad j = 0,\ldots,m-2. \end{cases}$$

waarbij

(2.32a) $\tilde{\mu}_{m,m-1} = \frac{1}{6 - 24\beta_3}$ als $|\beta_3 - \frac{1}{4}| \geq \frac{1}{24}$ en $|\beta_3 - \frac{1}{6}| \geq \frac{1}{96}$

of

(2.32b) $\tilde{\mu}_{m,m-1} = 0$ anders.

De referentieformule is in geval (2.32a) 3^e orde en in geval (2.32b) 2^e orde consistent.
Voor een gedetailleerde afleiding van bovenstaande formules wordt de lezer verwezen naar de appendix.

Voorbeeld 2.2. Beschouw de formule uit voorbeeld 2.1. Wegens oneven graad van het polynoom geldt $\mu_{1,0} = 0$ zodat we (2.31) en (2.32a) moeten toepassen. De genererende matrices voor de referentieformule worden nu

$$(\tilde{\lambda}_{j,\ell}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 0 \\ 3/8 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 7/6 & 0 & 0 & -1/6 \end{pmatrix} \text{ en } (\tilde{\mu}_{j,\ell}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1/16 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ 0 & 0 & 4/6 & 0 \end{pmatrix}$$

Derhalve luidt de formule voor $\tilde{y}_{n+1}$

$$\tilde{y}_{n+1} = y_n + \frac{7}{6} h_n F(y_n) - \frac{1}{6} h_n F(y_{n+1}) + \frac{4}{6} h_n^2 G(y_{n+1}^{(2)})$$

zodat voor $\tilde{\rho}_n$ geldt

$$\tilde{\rho}_n = \frac{1}{6} h_n F(y_n) - \frac{1}{6} h_n F(y_{n+1}) +$$

$$\frac{1}{6} h_n^2 G(y_n + \frac{3}{8} h_n F(y_n) + \frac{1}{16} h_n^2 G(y_n + \frac{1}{4} h_n F(y_n))).$$

3. DE ALGOL-60 PROCEDURES

In dit hoofdstuk worden de procedures STABTARK (stabilized Taylor-Runge-Kutta) en STABTARK2VS (stabilized Taylor-Runge-Kutta, second order, variable step) besproken. Voor beide procedures geldt, dat de eerste 8 parameters betrekking hebben op het op te lossen beginwaardeprobleem. Met behulp van de daaropvolgende parameters definieert de gebruiker enerzijds het gewenste stabiliteitsgedrag van de methode; anderzijds kan hij beschikken over voor hem nuttige (uitvoer)gegevens

3.1. De procedure STABTARK

Deze procedure beschrijft, afhankelijk van het opgegeven stabiliteitspolynoom, een gestabiliseerde 1^e, 2^e of 3^e orde Taylor-Runge-Kutta formule. STABTARK integreert met constante staplengte.

3.1.1. Heading en parameters van STABTARK

De heading van deze procedure is:

```
procedure STABTARK (x, xe, y, low, up, firstder, secondder, specrad,
                    degree, beta, stabbound, step, n, out);
integer   low, up, degree, n;
real      x, xe, specrad, stabbound, step;
array     y, beta;
procedure firstder, secondder, out;
```

Betekenis van de parameters:

x : <variable>;
de onafhankelijke variable;
bij aanroep van STABTARK moet x de beginwaarde x_0 van de onafhankelijke variabele hebben; na uitvoering van de procedure heeft x de waarde xe;

xe : <arithmetic expression>;
de eindwaarde van x;

y : <array identifier>;
het array y[low : up] bevat gedurende het integratieproces de numerieke oplossing in het punt x; bij aanroep van STABTARK moet y de startwaarde $y(x_0)$ hebben;

low,up : <arithmetic expression>;
indices van de eerste en de laatste vergelijking van het stelsel differentiaalvergelijkingen;

firstder,
secondder: <procedure identifier>;
firstder en secondder moeten alsvolgt door de gebruiker gedeclareerd worden:
procedure firstder(a); array a; <body 1>;
procedure secondder(a); array a; <body 2>;
firstder en secondder definiëren de evaluatie van de rechterlidfunctie F resp. de eerste afgeleide G naar x van de rechterlidfunctie met als afhankelijke variabelen a[low : up]; na uitvoering van deze procedures moet het array a de componenten van F(a(x)) resp. G(a(x)) bevatten;
omdat de differentiaalvergelijking in autonome vorm moet zijn, mag de variabele x niet gebruikt worden in firstder en secondder;

specrad : <arithmetic expression>;
de spectraalradius van die eigenwaarden van de Jacobiaan van F(y), die niet in het positieve halfvlak liggen; specrad moet door de gebruiker worden meegegeven en kan een functie zijn van y[low : up];

degree : <arithmetic expression>;
de graad van het op te geven stabiliteitspolynoom;

beta : <array identifier>;
in het array beta[1 : degree] moeten de coëfficiënten β_j van het stabiliteitspolynoom gegeven worden;

stabbound: <arithmetic expression>;
de gebruiker geeft hierin mee de stabiliteitsgrens behorende bij het opgegeven stabiliteitspolynoom;

step : <variable>;
de gebruiker moet hierin de staplengte geven, die hij op grond van nauwkeurigheid wenst; indien deze staplengte geen stabiliteit

waarborgt, wordt door STABTARK de staplengte verkleind tot stab-
bound/specrad(zie §2.3);

n : <variable>;
n telt het aantal integratiestappen en wordt door STABTARK zelf geïnitialiseerd op nul;
n kan worden gebruikt als Jensenvariabele in de expressie voor xe, step en specrad en in de procedure out;

out : <procedure identifier>;
out moet als volgt door de gebruiker gedeclareerd worden:
procedure out; <body>;
na iedere integratiestap wordt de procedure out aangeroepen;
via deze procedure kan men enerzijds de waarden opvragen van x, y[low:up], specrad, step en n, anderzijds kan men verschillende parameters wijzigen zoals xe, specrad en step zonder dat STABTARK opnieuw gestart behoeft te worden;
voorbeeld:
```
if xe < 10 then xe := xe + 2;
   specrad := 2 * y[1];
```

3.1.2. De body van STABTARK

```
"PROCEDURE" STABTARK(X, XE, Y, LOW, UP, FIRSTDER, SECONDDER, SPECRAD,
                                  DEGREE, BETA, STABBOUND, STEP, N, OUT);
    "INTEGER" LOW, UP, DEGREE, N;
    "REAL" X, XE, SPECRAD, STABBOUND, STEP;
    "ARRAY" Y, BETA;
    "PROCEDURE" FIRSTDER, SECONDDER, OUT;

  "BEGIN" "INTEGER" M; "REAL" MACHTOL; "BOOLEAN" EVEN, LAST;
    "REAL" "ARRAY" LAMBDA, MU[1:(DEGREE+1)//2], F, YN[LOW:UP];

    "PROCEDURE"  TAYLOR RUNGE KUTTA PARAMETERS;
      "BEGIN" "INTEGER" S, J;
        EVEN:= DEGREE//2*2 = DEGREE; M:= (DEGREE+1)//2;
        "IF" M = 1 "THEN"
        "BEGIN" LAMBDA[1]:= BETA[1];
                MU[1]:= "IF" EVEN "THEN" BETA[2] "ELSE" 0
        "END"   "ELSE"
        "BEGIN" LAMBDA[M]:= BETA[1]; MU[M]:= BETA[2]; S:= 2;
          "FOR" J:= M-1 "STEP" -1 "UNTIL" 2 "DO"
          "BEGIN" LAMBDA[J]:= BETA[S+1]/BETA[S];
                  MU[J]:= BETA[S+2]/BETA[S]; S:= S+2
          "END";
          LAMBDA[1]:= BETA[S+1]/BETA[S];
          MU[1]:= "IF" EVEN "THEN" BETA[S+2]/BETA[S] "ELSE" 0
        "END"
      "END" PARAMETERS;

    "PROCEDURE"  CHECK STEP;
      "BEGIN"
        "IF" STEP*SPECRAD>STABBOUND "THEN" STEP:= STABBOUND/SPECRAD;
        "IF" X+STEP>XE*(1-N*MACHTOL) "THEN"
        "BEGIN" STEP:= XE-X; LAST:= "TRUE"  "END";
        "IF" STEP<X*MACHTOL "THEN"  "GOTO" END OF STABTARK
      "END"  CHECK;

    "PROCEDURE"  DIFFERENCE SCHEME;
      "BEGIN" "INTEGER" I, J; "REAL" LT, MT;
        "FOR" I:= LOW "STEP" 1 "UNTIL" UP "DO" YN[I]:= F[I]:= Y[I];
        FIRSTDER(F); "IF" EVEN "THEN" SECONDDER(Y);
        LT:= LAMBDA[1]*STEP; MT:= MU[1]*STEP*STEP;
        "FOR" I:= LOW "STEP" 1 "UNTIL" UP "DO"
            Y[I]:= YN[I]+(LT*F[I]+MT*Y[I]);
        "FOR" J:= 2 "STEP" 1 "UNTIL" M "DO"
        "BEGIN"  SECONDDER(Y);
          LT:= LAMBDA[J]*STEP; MT:= MU[J]*STEP*STEP;
          "FOR" I:= LOW "STEP" 1 "UNTIL" UP "DO"
              Y[I]:= YN[I]+(LT*F[I]+MT*Y[I])
        "END";
        "IF" LAST "THEN" "BEGIN" X:= XE; LAST:= "FALSE" "END"
        "ELSE" X:= X+STEP
      "END"  DIFF. SCHEME;

    MACHTOL:= 2**(-48); N:= 0; LAST:= "FALSE";
    TAYLOR RUNGE KUTTA PARAMETERS;
    NEXT STEP :
        CHECK STEP; DIFFERENCE SCHEME; N:= N+1; OUT;
        "IF" X<XE "THEN"  "GOTO"  NEXT STEP;
    END OF STABTARK :
  "END"  STABILIZED TAYLOR-RUNGE-KUTTA METHOD;
```

3.1.3. De deelprocedures van STABTARK

1) De procedure taylor runge kutta parameters.

Uit de coëfficiënten beta[1 : degree] worden met behulp van (2.23) de waarden van $\lambda_{j,0}$ en $\mu_{j,j-1}$, j = 1, ... , m uit (2.17) berekend.

2) De procedure check step.

In deze procedure wordt de staplengte onderworpen aan de stabiliteitsvoorwaarde (2.15) en zonodig verkleind tot de maximaal stabiele staplengte. Indien de staplengte te klein is (zie §3.2.3), worden verdere berekeningen gestaakt.

3) De procedure difference scheme.

Deze procedure berekent de numerieke oplossing y_{n+1} volgens schema (2.16) en levert deze af in y[low : up].

3.2. De procedure STABTARK2VS

Deze procedure beschrijft een gestabiliseerde 2^e orde Taylor-Runge-Kutta formule. STABTARK2VS bepaalt zelf de grootte van de integratiestappen; om geheugenruimte te besparen worden genomen stappen niet verworpen.

3.2.1. Heading en parameters van STABTARK2VS

De heading van deze procedure luidt:

```
procedure STABTARK2VS (x, xe, y, low, up, firstder, secondder, specrad,
                       data, out);
integer   low, up;
real      x, xe, specrad;
array     y, data;
procedure firstder, secondder, out;
```

Betekenis van de parameters:

x, xe, y, low, up, firstder, secondder, specrad: zie §3.1.1.

data: <array identifier>;
in het array data[1 : 10 + data [1]] moet de gebruiker meegeven:

data [1] : de graad van het gewenste stabiliteitspolynoom;
data [2] : de stabiliteitsgrens;
data [3] : de minimale staplengte;
data [4] : de absolute tolerantie voor de locale fout;
data [5] : de relatieve tolerantie voor de locale fout;
De procedure berekent na elke stap altijd een schatting van de locale fout. Op grond hiervan en de opgegeven tolerantie wordt een nieuwe staplengte berekend, tenzij data [4] < 0 en data [5] < 0. In dit geval integreert STABTARK2VS met een constante staplengte gelijk aan data [3];
data [11 : 10 + data [1]] :
de coëfficiënten β_j van het stabiliteitspolynoom; dit polynoom moet 2^e orde consistent zijn;
Na iedere stap worden de volgende gegevens afgeleverd:
data [6] : het aantal stappen;
data [7] : schatting van de gemaakte locale fout;
data [8] : tolerantie voor de locale fout;
data [9] : lengte van de integratiestap;
data [10]: foutmeldingen;
data [10] =
0 : geen moeilijkheden;
1 : stabiliteitspolynoom is niet 2^e orde consistent;
2 : de minimale staplengte is groter dan de maximaal stabiele staplengte;
3 : de staplengte is te klein (zie §3.2.3);

out : <procedure identifier>;
procedure out; <body>;
via deze procedure kan men enerzijds de waarden opvragen van x, y[low:up], specrad en data [6:10], anderzijds de waarden van verschillende parameters veranderen, zoals xe, specrad en data[3 : 5] zonder dat STABTARK2VS opnieuw gestart behoeft te worden.
Bijvoorbeeld : if data [7] > data [8] then tel := tel + 1;
unit := data [7]/data [9];

```
if x = xe ∧ xe = 5 then
begin data [4]:= data [5]:= data [4]/10;
      xe := 10
end;
```

3.2.2. De body van STABTARK2VS

```
"PROCEDURE" STABTARK2VS(X, XE, Y, LOW, UP, FIRSTDER, SECONDDER, SPECRAD,
                                                              DATA, OUT);
    "INTEGER" LOW, UP;
    "REAL" X, XE, SPECRAD;
    "REAL" "ARRAY" Y, DATA;
    "PROCEDURE" FIRSTDER, SECONDDER, OUT;

  "BEGIN" "INTEGER" DEGREE, M; "BOOLEAN" EVEN;
    DEGREE:= DATA[1]; EVEN:= DEGREE//2*2 = DEGREE; M:= (DEGREE+1)//2;

  "BEGIN"  "REAL" MACHTOL, H; "BOOLEAN" START, STEP10, LAST;
    "REAL" "ARRAY" LAMBDA, MU[1:M], LAMBT, MUT[0:M], F, G, RHO[LOW:UP];

    "PROCEDURE"  INIVEC(L, U, A, X); "CODE" 31010;
    "PROCEDURE"  DUPVEC(L, U, SHIFT, A, B); "CODE" 31030;
    "PROCEDURE"  ELMVEC(L, U, SHIFT, A, B, X); "CODE" 34020;
    "REAL" "PROCEDURE"  VECVEC(L, U, SHIFT, A, B); "CODE" 34010;

    "PROCEDURE"  ERROR(I); "VALUE" I; "INTEGER" I;
      "BEGIN" DATA[10]:= I; "GOTO" ENDOFSTABTARK2VS "END";

    "REAL" "PROCEDURE"  NORM(VECTOR); "ARRAY" VECTOR;
         NORM:= SQRT(VECVEC(LOW, UP, 0, VECTOR, VECTOR));

    "PROCEDURE"  SUM(LT, MT); "VALUE" LT, MT; "REAL" LT, MT;
      "BEGIN" "INTEGER" I;
        "FOR" I:= LOW "STEP" 1 "UNTIL" UP "DO"
             G[I]:= Y[I]+(LT*F[I]+MT*G[I])
      "END" SUM;

    "PROCEDURE"  INITIALIZATION;
      "BEGIN"  DATA[10]:= 0; MACHTOL:= 2**(-48);
        EVALUATE PARAMETERS;
        DUPVEC(LOW, UP, 0, F, Y); DUPVEC(LOW, UP, 0, G, Y);
        FIRSTDER(F); "IF" EVEN "THEN" SECONDDER(G);
        DATA[6]:= 0; START:= "TRUE"; STEP10:= LAST:= "FALSE"
      "END" INIT;
```

```
"PROCEDURE"  EVALUATE PARAMETERS;
  "BEGIN" "INTEGER" S, J; "REAL" R;
     "IF" M = 1 "THEN"
     "BEGIN" "IF" DATA[11] ^= 1 "THEN"  ERROR(1);
        LAMBDA[1]:= DATA[11];
        MU[1]:= "IF" EVEN "THEN" DATA[12] "ELSE" 0;
        "IF" MU[1] ^= 0.5 "THEN" ERROR(1);
     "END"  "ELSE"
     "BEGIN"
        "IF" DATA[11] ^= 1 "OR" DATA[12] ^= 0.5 "OR"
             ABS(6*DATA[13]-1) < 10000*MACHTOL "THEN" ERROR(1);
        LAMBDA[M]:= DATA[11]; MU[M]:= DATA[12]; S:= 12;
        "FOR" J:= M-1 "STEP" -1 "UNTIL" 2 "DO"
        "BEGIN"  R:= DATA[S];
           LAMBDA[J]:= DATA[S+1]/R; MU[J]:= DATA[S+2]/R; S:= S+2
        "END"; R:= DATA[S];
        LAMBDA[1]:= DATA[S+1]/R;
        MU[1]:= "IF" EVEN "THEN" DATA[S+2]/R "ELSE" 0
     "END"
      AND NOW EVALUATE PARAMETERS FOR REFERENCE-SOLUTION;
     INIVEC(0, M , LAMBT, 0); INIVEC(0, M, MUT, 0);
     "IF" EVEN "THEN"
     "BEGIN" "IF" M >= 2 "THEN"
        "BEGIN" "IF" ABS(24*DATA[13]-6) < 1 "THEN"
           "BEGIN" LAMBT[0]:= LAMBT[M]:= 0.5;
              MUT[0]:= 1/12; MUT[M]:= -1/12
           "END"  "ELSE"
           "BEGIN" R:= LAMBT[M]:= 1/(6-24*DATA[13]);
              LAMBT[0]:= 1-R; MUT[0]:= (2-3*R)/6;
              MUT[M]:= (1-3*R)/6
           "END"
        "END"  "ELSE"
         "BEGIN" LAMBT[0]:= LAMBT[M]:= 0.5;
            MUT[0]:= 1/12; MUT[M]:= -1/12
         "END"
     "END"  "ELSE"
     "IF" ABS(24*DATA[13]-6)<1 "OR" ABS(96*DATA[13]-16)<1 "THEN"
     "BEGIN" LAMBT[0]:= LAMBT[M]:= 0.5 "END"  "ELSE"
     "BEGIN" R:= MUT[M-1]:= 1/(6-24*DATA[13]);
        LAMBT[0]:= 0.5+R; LAMBT[M]:= 0.5-R
     "END";
     LAMBT[0]:= LAMBT[0]-LAMBDA[M]; MUT[M-1]:= MUT[M-1]-MU[M]
"END" PARAMETERS;

"PROCEDURE"  LOCAL ERROR CONSTRUCTION(I); "VALUE" I; "INTEGER" I;
  "BEGIN"
     "IF" LAMBT[I]^=0 "THEN" ELMVEC(LOW, UP, 0, RHO, F, LAMBT[I]);
     "IF" MUT[I] ^= 0 "THEN" ELMVEC(LOW, UP, 0, RHO, G, MUT[I]*H);
     "IF" I=M "THEN" DATA[7]:= NORM(RHO)*H
  "END"  LOC. ERROR CONSTR.;
```

```
"PROCEDURE"  STEPSIZE;
  "BEGIN" "REAL" TOL, RO, HOLD, HACC, HSTAB;
     TOL:= DATA[8]:= DATA[4]+DATA[5]*NORM(Y);
     "IF" TOL > 0 "THEN"
     "BEGIN" "IF" START "THEN"
        "BEGIN" HACC:= DATA[3]; START:= "FALSE";
           STEP10:= "TRUE"
        "END"
        "ELSE"
        "BEGIN" RO:= DATA[7]; HOLD:= DATA[9];
           "IF" STEP10 "THEN"
           "BEGIN" "IF" TOL > 1000*RO "THEN" HACC:= 10*HOLD "ELSE"
              "BEGIN" HACC:= (TOL/(0.75*(TOL+RO))+0.33333333)*HOLD;
                 STEP10:= "FALSE"
              "END"
           "END"
           "ELSE" HACC:= (TOL/(0.75*(TOL+RO))+0.33333333)*HOLD
        "END"
     "END"  "ELSE" HACC:= DATA[3];
     "IF" HACC < DATA[3] "THEN" HACC:= DATA[3];
     HSTAB:= DATA[2]/SPECRAD; "IF" HSTAB < DATA[3] "THEN" ERROR(3);
     H:= "IF" HACC >HSTAB "THEN" HSTAB "ELSE" HACC;
     "IF" H < X*MACHTOL "THEN" ERROR(4);
     "IF" X+H >= XE "THEN"
     "BEGIN" H:= XE-X; LAST:= "TRUE"; STEP10:= "TRUE" "END";
     DATA[9]:= H
  "END" STEPSIZE;


  "PROCEDURE"  DIFFERENCE SCHEME;
    "BEGIN" "INTEGER" J;
       INIVEC(LOW, UP, RHO, 0); LOCAL ERROR CONSTRUCTION(0);
       SUM(LAMBDA[1]*H, MU[1]*H*H);
       "FOR" J:= 2 "STEP" 1 "UNTIL" M "DO"
       "BEGIN"  SECONDDER(G);
          "IF" J = M "THEN" LOCAL ERROR CONSTRUCTION(M-1);
          SUM(LAMBDA[J]*H, MU[J]*H*H)
       "END";
       DUPVEC(LOW, UP, 0, Y, G); DUPVEC(LOW, UP, 0, F, G);
       FIRSTDER(F); "IF" EVEN "THEN" SECONDDER(G);
       LOCAL ERROR CONSTRUCTION(M);
       "IF" LAST "THEN" "BEGIN" X:= XE; LAST:= "FALSE" "END"
       "ELSE" X:= X+H
    "END"  DIFF. SCHEME;


  INITIALIZATION;

  NEXT STEP :
      STEPSIZE; DIFFERENCE SCHEME; DATA[6]:= DATA[6]+1; OUT;
      "IF" X < XE "THEN"  "GOTO"  NEXT STEP;

  END OF STABTARK2VS;
"END"
"END" STABILIZED TAYLOR-RUNGE-KUTTA, SECOND ORDER, VARIABLE STEPSIZE;
```

3.2.3. De deelprocedures van STABTARK2VS

1) De procedures inivec, dupvec, elmvec en vecvec.

Deze elementaire vectorprocedures zijn genomen uit bibliotheek NUMAL (ref. NUMAL, a library of numerical procedures in ALGOL-60, section 1.1., Mathematisch Centrum, Amsterdam).

2) De procedure error.

Verzorgt de foutmeldingen. Na een aanroep van "error" worden verdere berekeningen gestaakt;

3) De procedure norm.

Berekent de Euclidische norm van een vector;

4) De procedure sum.

Berekent $y_{n+1}^{(j)} = y_n + h_n\lambda_{j,0}F(y_n) + h_n^2\mu_{j,j-1}G\left(y_{n+1}^{(j-1)}\right)$;

5) De procedure initialization.

Initialiseert een aantal variabelen, waaronder data [6], en roept de procedure "evaluate parameters" aan;

6) De procedure evaluate parameters.

Uit de coëfficiënten data [11 : 10 + data [1]] worden met behulp van (2.23) de waarden van $\lambda_{j,0}$ en $\mu_{j,j-1}$, $j = 1, \ldots m$ uit (2.17) berekend; tevens worden de waarden van $\tilde{\lambda}_{m,0}$, $\tilde{\lambda}_{m,m}$ en $\tilde{\mu}_{m,j}$, $j = 0, \ldots, m$ uit (2.28) bepaald door middel van de formules uit §2.8;

7) De procedure local error construction.

Hierin wordt de schatting van de locale fout $\tilde{\rho}_n$ berekend; daarna wordt $\|\rho_n\|_2$ bepaald.

8) De procedure stepsize.

Deze procedure bepaalt de nieuwe integratiestap h_n, uitgaande van de berekende locale fout $\|\tilde{\rho}_{n-1}\|_2$ in $x = x_n$, de tolerantie tol_n = data [4] + data [5] * $\|y_n\|_2$ en de laatst gebruikte staplengte h_{n-1}, volgens de volgende strategie:

a) eerste stap: h_0 = data [3] = de minimale staplengte;

b) tussenfase : de tussenfase gaat over in de eindfase als
$\|\tilde{\rho}_n\|_2 > \frac{1}{1000}\,tol_n$;
in de tussenfase geldt:

$h_n = 10 * h_{n-1}$ (vgl. BEENTJES [1972]);

c) eindfase : $h_n = h_{n-1} * \left(\frac{\frac{4}{3}\,tol_n}{tol_n + \|\tilde{\rho}_{n-1}\|_2} + \frac{1}{3}\right)$

(vgl. VAN KAMPEN [1974]);

De aldus berekende nieuwe stap wordt steeds onderworpen aan de stabiliteitsvoorwaarde (2.15) $h_n \leq$ data [2]/specrad.
Indien $h_n < \varepsilon * x_n$, waarbij ε de machine-precisie voorstelt ($\varepsilon \simeq 10^{-14}$ voor de CD CYBER 73-28), geldt $x_n = x_{n+1}$ en worden verdere berekeningen gestaakt. Na het bereiken van het eindpunt xe, komt het proces in de tussenfase.

9) *de procedure difference scheme*.

Berekent de numerieke oplossing y_{n+1} volgens schema (2.16) en levert deze af in y[low:up]; tijdens de berekening van y_{n+1} wordt een schatting van de locale fout opgebouwd door aanroepen van de procedure "local error construction".

4. TESTRESULTATEN

In dit hoofdstuk bespreken we vier problemen, waarmee STABTARK en STABTARK2VS zijn getest.

1) een lineair stelsel differentiaalvergelijkingen; dit werd gekozen om de mogelijkheid van exponentiële aanpassing aan te tonen;

2) een niet-lineaire vergelijking; hierbij blijkt het voordeel van een stapkeuze;

3) een niet-lineair stelsel differentiaalvergelijkingen; dit kozen we om te laten zien, dat sommige problemen door een transformatie efficiënter zijn op te lossen;

4) een parabolische partiële differentiaalvergeljking; hierbij wordt aangetoond:
   a. het voordeel van een geschikte keuze van het stabiliteitspolynoom;
   b. het nadeel van integratie met een variabele staplengte.

Bij de problemen 1, 3 en 4 zullen Taylor-Runge-Kutta schema's en Runge-Kutta schema's, gegenereerd door hetzelfde stabiliteitspolynoom, worden vergeleken.

4.1. Een lineair stelsel differentiaalvergelijkingen

FOWLER en WARTEN geven het gekoppelde stelsel stijve differentiaalvergelijkingen

$$(4.1) \qquad y' = Ay + B \quad , \; y(0) = \begin{pmatrix} -0.1 \\ 0.1 \end{pmatrix}$$

waarin

$$A = \begin{pmatrix} -500.5 & 499.5 \\ 499.5 & -500.5 \end{pmatrix} \qquad \text{en} \qquad B = \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

De analytische oplossing van (4.1) luidt:

$$y(x) = 2(1-e^{-x})\begin{pmatrix} 1 \\ 1 \end{pmatrix} + e^{-1000x}\begin{pmatrix} -0.1 \\ 0.1 \end{pmatrix} .$$

De Jacobiaan van het stelsel heeft de eigenwaarden -1000 en -1, zodat een exponentieel aangepaste methode het meest geschikt lijkt. Het probleem (4.1) is opgelost door integratie met constante staplengte h over het interval [0,1]. Hiervoor gebruikten we de procedure STABTARK en gaven de volgende stabiliteitspolynomen mee:

$$P_2(z) = 1 + z + \tfrac{1}{2} z^2 + \beta_3 z^3$$

$$P_3(z) = 1 + z + \tfrac{1}{2} z^2 + \frac{1}{6} z^3 + \beta_4 z^4$$

$$P_4(z) = 1 + z + \tfrac{1}{2} z^2 + \frac{1}{6} z^3 + \frac{1}{24} z^4 + \beta_5 z^5,$$

met β_i zodanig, dat $P_{i+1}(z_1) = e^{z_1}$, $i = 1,2,3$ met $z_1 = -1000*h$. En tevens

$$P_1(z) = 1 + z + \beta_2 z^2 + \beta_3 z^3,$$

met β_2 en β_3 zodanig, dat $P_1(z_1) = e^{z_1}$ en $P_1(z_2) = e^{z_2}$, waarbij $z_1 = -1000*h$ en $z_2 = -h$.

De door deze polynomen bepaalde Taylor-Runge-Kutta methoden zijn 2^e, 3^e, 4^e respectievelijk 1^e orde consistent. Met het schema, gegenereerd door $P_1(z)$ verwachten we exacte integratie van (4.1). In tabel 4.1 is een overzicht gegeven van het aantal goede cijfers van de numerieke oplossing (= $-^{10}\log|\text{relatieve fout}|$). De resultaten verkregen met STABTARK en het polynoom $P_1(z)$ zijn slechter dan was verwacht. Dit wordt verklaard door de onnauwkeurige berekening van β_1 en β_2. Bovendien is (4.1) opgelost met een exponentieel aangepaste Runge-Kutta methode. Hiervoor werd de procedure EFRK van DEKKER [1974] gebruikt. Als stabiliteitspolynomen werden $P_2(z)$, $P_3(z)$ en $P_4(z)$ gekozen. De hiermee verkregen resultaten zijn gelijk aan die verkregen met STABTARK. Voor de tweede afgeleide geldt $y'' = A^2y + AB$, zodat voor dit probleem $c(G) = c(F)$ (zie §2.6). Tellen we nu een evaluatie van $G(y)$ als een evaluatie van $F(y)$, dan kunnen we een vergelijking maken tussen STABTARK en EFRK wat betreft het aantal evaluaties van $F(y)$ per integratiestap (zie onderstaande tabel)

	$P_1(z)$	$P_2(z)$	$P_3(z)$	$P_4(z)$
STABTARK	2	2	3	3
EFRK	x	3	4	5

Hieruit blijkt, dat STABTARK voor dit soort problemen efficiënter is dan EFRK en dat stabiliteitspolynomen van oneven graad de voorkeur genieten. Om aan te tonen, dat een exponentieel aangepaste methode in dit geval vele malen sneller is dan een gewone Runge-Kutta methode, hebben we (4.1) ook opgelost met de procedure RKE van BEENTJES [1974a]. Dit is een 5e orde Runge-Kutta methode, die 6 evaluaties van $F(y)$ per stap gebruikt; RKE bepaalt zelf de stapgrootte en verwerpt deze zonodig.
De resultaten staan in tabel 4.2.

Tabel 4.1

Aantal goede cijfers voor probleem (4.1) opgelost met STABTARK

POLYNOOM	AANTAL STAPPEN							
	1	2	4	8	16	32	64	128
P_2	0.7	1.4	2.1	2.8	3.4	4.1	4.7	5.4
P_3	1.3	2.3	3.3	4.3	5.2	6.2	7.1	8.1
P_4	2.0	3.3	4.6	5.9	7.1	8.4	9.6	11.0
P_1	10.8	10.0	10.2	10.8	11.7	12.1	13.0	12.4

Tabel 4.2

Resultaten met de procedure RKE voor probleem (4.1)

aantal geaccepteerde stappen	aantal verworpen stappen	aantal evaluaties van F(y)	aantal goede cijfers
340	175	3090	6.5
346	159	3030	10.2
561	142	4218	13.6

4.2. Een eenvoudige niet-lineaire vergelijking

Beschouw het beginwaardeprobleem

$$y' = a^2 - y^2 \quad , \quad y(0) = 0, \quad a \in \mathbb{R} \tag{4.2}$$

met als analytische oplossing

$$y(x) = a - 2a/(\exp(2ax) + 1).$$

Het integratieinterval is [0,10]. Op dit interval is de spectraalradius maximaal 2a. Gezien het gedrag van de analytische oplossing, zal de staplengte aanvankelijk bepaald worden door de gewenste nauwkeurigheid. Is de oplossing eenmaal in de asymptotische fase, dan zal de staplengte slechts bepaald worden door stabiliteit. Integratie met een variabele staplengte lijkt in dit geval het verstandigst.

Daartoe hebben we probleem (4.2) voor a = 2,5 en 10 opgelost met de procedure STABTARK2VS. De beschouwde stabiliteitspolynomen waren (vgl. DEKKER e.a. [1972], blz. 65):

$$P_4(z) = 1 + z + \tfrac{1}{2}z^2 + 78084485_{10}\text{-}9z^3 + 36084541_{10}\text{-}10z^4,$$

$$\beta_{real} = 12.05$$

en

$$P_5(z) = 1 + z + \tfrac{1}{2}z^2 + 84608499_{10}\text{-}9z^3 + 55271248_{10}\text{-}10z^4 +$$

$$+ 12219644_{10}\text{-}11z^5, \qquad \beta_{real} = 19.45$$

Beide polynomen zijn 2^e orde consistent en gebruiken elk 1 evaluatie van F(y) en 2 evaluaties van G(y) per integratiestap. Hierbij geldt $G(y) = 2y(y^2-a^2)$, zodat c(G) ≃ 2 * c(F). De waarden van enkele besturingsparameters van STABTARK2VS waren:
specrad = 2 * y_n; data [3] = $_{10}$-4; data [4] = data [5] = $_{10}$-3, $_{10}$-5, $_{10}$-7.

Gemeten is de maximale globale fout $\|e\|_\infty \doteq \max_n |y(x_n) - y_n|$.
Deze fout bleek in alle gevallen in het interval [0,1] op te treden. Voorts is probleem (4.2) opgelost door integratie met verschillende vaste staplengtes met behulp van de procedure STABTARK met de voorgeschreven polynomen $P_4(z)$ en $P_5(z)$. Door middel van interpolatie werd het aantal stappen bepaald om een fout $\|e\|_\infty$ te krijgen, zodat vergelijking met STABTARK2VS mogelijk is. In tabel 4.3 wordt een overzicht gegeven van de verkregen resultaten. Hierbij

gelden de volgende definities:

n(i,j) $\stackrel{p.d.}{=}$ het aantal stappen nodig om (4.2) te integreren over het interval [0,j] met STABTARK2VS met stabiliteitspolynoom $P_i(z)$, j = 1, 2, 10; i = 4, 5.

m(i) $\stackrel{p.d.}{=}$ het aantal stappen nodig om (4.2) te integreren over het interval [0,10] met STABTARK met stabiliteitspolynoom $P_i(z)$, i = 4, 5.

Wegens het grotere stabiliteitsgebied behorende bij polynoom P_5 is in tabel 4.3 het aantal stappen behorende bij P_5 kleiner dan bij P_4. Voorts blijkt, dat integratie met een variabele staplengte voor dit soort problemen te verkiezen is boven integratie met constante staplengte. Wat de totale kosten betreft, is integratie met $P_5(z)$ efficiënter dan integratie met $P_4(z)$.

Tabel 4.3

De resultaten met STABTARK en STABTARK2VS voor a = 2, 5 en 10

waarde van a	poly-noom	STABTARK2VS			STABTARK	
		n(i,1)	n(i,2)	n(i,10)	m(i)	$\|e\|_\infty$
a = 2	P_4	14	20	27	101	$4.6_{10}{}^{-3}$
	P_5	14	21	28	97	$4.6_{10}{}^{-3}$
	P_4	38	53	65	498	$1.8_{10}{}^{-4}$
	P_5	37	52	65	475	$1.8_{10}{}^{-4}$
	P_4	163	226	256	2288	$8.2_{10}{}^{-6}$
	P_5	159	220	250	2260	$8.0_{10}{}^{-6}$
a = 5	P_4	18	21	29	276	$9.7_{10}{}^{-3}$
	P_5	18	21	27	262	$9.9_{10}{}^{-3}$
	P_4	62	68	78	1332	$3.9_{10}{}^{-4}$
	P_5	61	67	74	1305	$3.8_{10}{}^{-4}$
	P_4	265	281	292	6094	$1.8_{10}{}^{-5}$
	P_5	259	275	283	6083	$1.7_{10}{}^{-5}$
a = 10	P_4	23	27	41	574	$1.8_{10}{}^{-2}$
	P_5	23	26	35	539	$1.9_{10}{}^{-2}$
	P_4	69	73	89	2738	$7.3_{10}{}^{-4}$
	P_5	68	72	82	2620	$7.3_{10}{}^{-4}$
	P_4	294	300	314	12875	$3.3_{10}{}^{-5}$
	P_5	287	292	304	12515	$3.2_{10}{}^{-5}$

4.3. Een niet-lineair stelsel differentiaalvergelijkingen

DAVIS [1962] geeft het volgende stelsel differentiaalvergelijkingen

$$(4.3a)\qquad \begin{cases} \dfrac{dy_1}{dx} = 2y_1(1-y_2) , & y_1(0) = 1 \\ \dfrac{dy_2}{dx} = y_2(y_1-1) , & y_2(0) = 3 . \end{cases}$$

Dit stelsel is een wiskundig model voor de groei van twee in conflict levende populaties. Een analytische oplossing is ons niet bekend. Eliminatie van de "tijdvariabele" x levert de vergelijking van de (gesloten) oplossingskrommen in het (y_1-y_2) vlak:

$$y_1 + 2y_2 - \ln 2y_1y_2^2 = C,$$

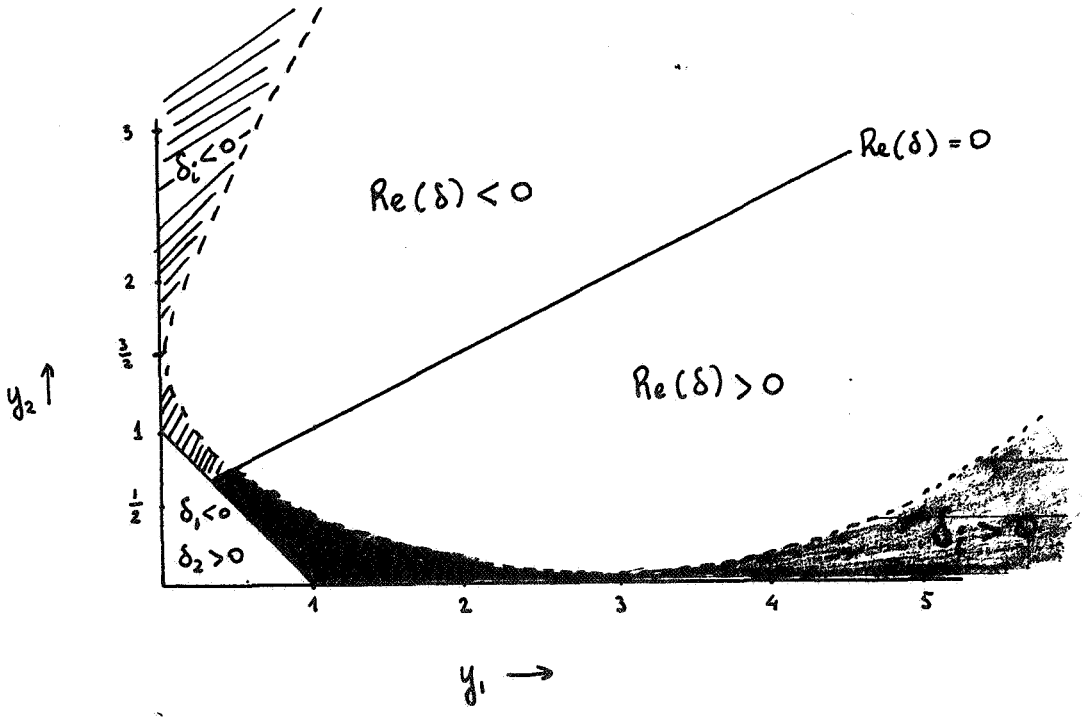
waarbij C bepaald wordt door de beginvoorwaarde. In figuur 4.1 zijn enkele krommen getekend behorende bij de beginvoorwaarden (1,3); (1,2.5); (1,2) en (1,1.5). De oplossing van (4.3a) is periodiek. De Jacobiaan van het rechterlid van (4.3a) luidt

$$J_1 = \begin{bmatrix} 2(1-y_2) & -2y_1 \\ y_2 & y_1-1 \end{bmatrix}$$

De eigenwaarden $\delta_{1,2}$ van J_1 zijn

$$\delta_{1,2} = \frac{1}{2}\left[(1-2y_2+y_1) \pm \sqrt{(y_1-2y_2)^2 - 6(y_1+2y_2) + 9}\right].$$

De nu volgende figuur geeft een idee met wat voor soort (negatief, positief of complex) eigenwaarden we te maken zullen krijgen.



Hierin is te zien, dat we over een groot traject te maken hebben met eigenwaarden waarvoor Re(δ) > 0, zodat integratie met een lage orde methode zonder stapverwerping gevaarlijk lijkt. Voor de kosten van y'' geldt c(G) ≃ 3.5 * c(F) zodat in dit geval een Taylor-Runge-Kutta methode altijd bewerkelijker is dan een Runge-Kutta methode (zie (2.25)).

Door de transformatie $v_1(x) = \ln y_1(x)$ en $v_2(x) = \ln y_2(x)$ gaat (4.3a) over in

$$(4.3b)\qquad \begin{cases} \dfrac{dv_1}{dx} = 2(1-\exp(v_2)) \ , & v_1(0) = 0 \\[2ex] \dfrac{dv_2}{dx} = \exp(v_1) - 1 \ , & v_2(0) = \ln 3. \end{cases}$$

De Jacobiaan van het rechterlid van (4.3b) is

$$J_2 = \begin{pmatrix} 0 & -2\exp(v_2) \\ \exp(v_1) & 0 \end{pmatrix}$$

met als eigenwaarden $\delta'_{1,2} = \pm i\sqrt{2\exp(v_1+v_2)}$. De eigenwaarden zijn nu zuiver imaginair. Voor probleem (4.3b) hebben we F(y) en G(y) als volgt gedefinieerd door de proceduredeclaraties

```
procedure   firstder (a);   array a;
begin   real a1;   a1 := exp(a[1]) - 1;
        a[1] := 2 * (1-exp(a[2])); a[2] := a1; f := f + 1
end;

procedure   secondder (a);   array a;
begin   real a1, a2, a3;   a1 := exp(a[1]); a2 := exp(a[2]);
                           a3 := a1 * a2;
            a[1] := 2 * (a2-a3); a[2] := 2 * (a1-a3); g := g + 1
end;
```

Hierin telt f resp. g het aantal evaluaties van F(y) resp. G(y). Voor de voor de kosten van y" geldt blijkbaar c(G) $\simeq$ c(F). Nu lijkt een Taylor-Runge-Kutta methode aantrekkelijker.

Het beschouwde integratieinterval was [0,20]. We zullen de berekende oplossing in x = 20 vergelijken met de referentieoplossing

$$y_1(20) = 0.6761\ 8760\ 0858$$
$$y_2(20) = 0.1860\ 8160\ 9964\ ,$$

die we met een hoge orde Runge-Kutta formule en een kleine staplengte hebben berekend. Bij alle beschouwde oplossingsmethoden bleek, dat de berekende component $y_2(20)$ 1 tot 3 cijfers nauwkeuriger was dan $y_1(20)$ zodat in alle tabellen van deze paragraaf het aantal goede cijfers in $y_1(20)$ staat vermeld.

Met de procedure STABTARK losten we de problemen (4.3a) en (4.3b) op met als stabiliteitspolynoom

$$P_4(z) = 1 + z + \tfrac{1}{2}z^2 + \tfrac{1}{6}z^3 + \tfrac{1}{24}z^4\ ; \qquad \beta_{abs} = 2.63.$$

De door $P_4(z)$ gegenereerde methode is 3^e orde consistent. We gebruikten hierbij de staplengtes: $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, $\frac{1}{64}$ en $\frac{1}{128}$. In tabel 4.4 staan bij verschillende staplengtes het aantal goede cijfers in $y_1(20)$ en de gemeten

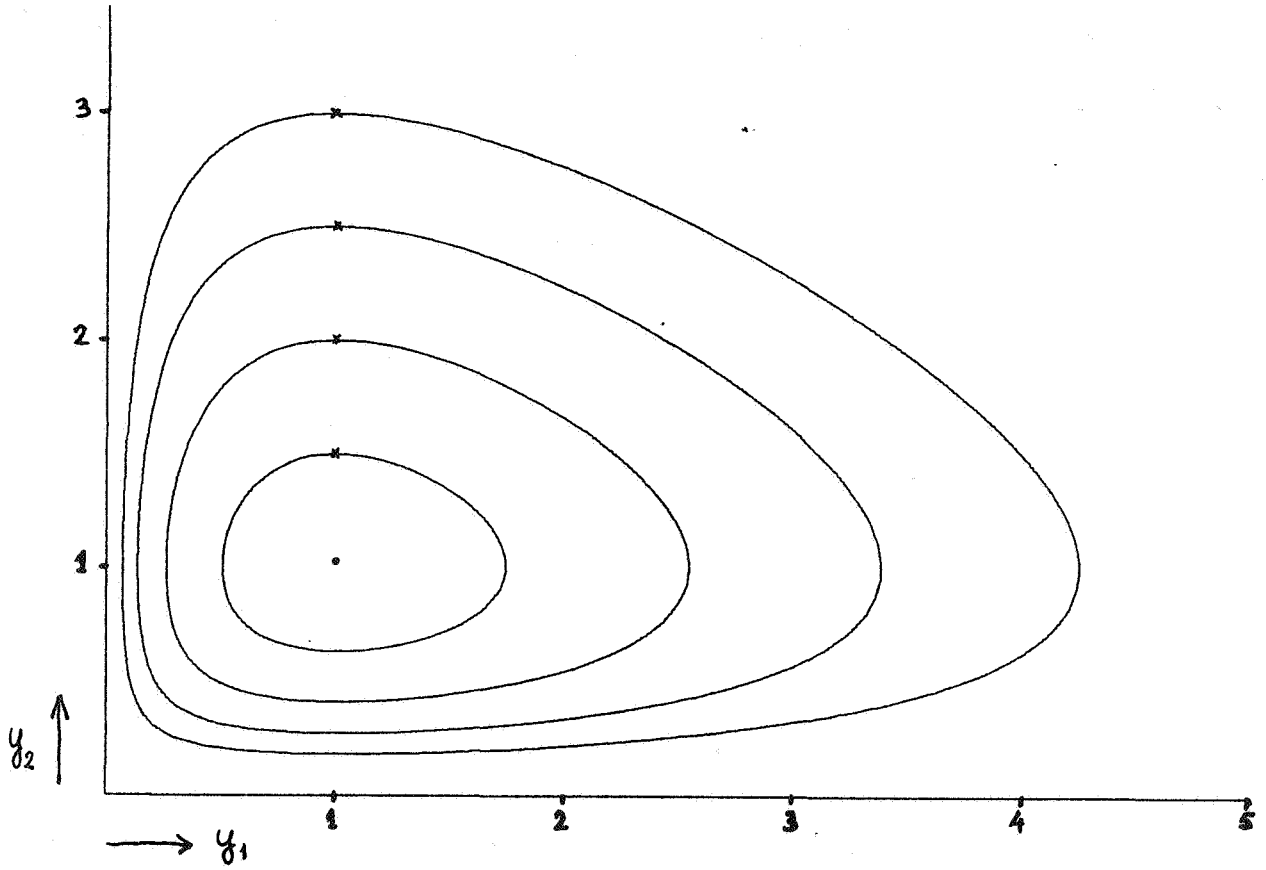


Fig. 4.1

Enkele oplossingskrommen van probleem (4.3a)

Tabel 4.4

Resultaten met STABTARK voor de problemen (4.3a) en (4.3b)

totale aantal stappen	PROBLEEM (4.3a)		PROBLEEM (4.3b)	
	aantal goede cijfers	gemeten rekentijd	aantal goede cijfers	gemeten rekentijd
80	1.2	0.3	0.7	0.4
160	2.5	0.6	1.5	0.8
320	3.1	1.3	2.4	1.6
640	3.9	2.6	3.3	3.2
1280	4.8	5.2	4.2	6.4
2560	5.7	10.2	5.1	13.0

rekentijd in seconden. Hieruit blijkt, dat probleem (4.3b) een minder lineair karakter heeft dan (4.3a), hetgeen ook aan de corresponderende Jacobianen J_1 en J_2 te zien is. Vervolgens hebben we STABTARK2VS vergeleken met een gestabiliseerde 2^e orde Runge-Kutta methode. Hiervoor werd de procedure ARK van BEENTJES [1974b] genomen. De beschouwde 2^e orde schema's werden gegenereerd door de volgende polynomen

1) $P_2(z) = 1 + z + \frac{1}{2}z^2$, $\beta_{real} = 2$; voor het probleem (4.3a);
2) $P_3(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{4}z^3$, $\beta_{im} = 2$; voor de problemen (4.3a) en (4.3b);
3) $P_5(z) = 1 + z + \frac{1}{2}z^2 + \frac{3}{16}z^3 + \frac{1}{32}z^4 + \frac{1}{128}z^5$, $\beta_{im} = 4$ voor het probleem (4.3b);

De besturingsparameters data [3 : 5] hadden de waarden: data [3] = 10^{-4}; data [4] = data [5] = 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}. We willen nog benadrukken dat deze toleranties betrekking hebben op de locale fout en als zodanig de globale fout kunnen besturen. In de tabellen 4.5 t/m 4.8 staan bij verschillende toleranties het aantal stappen, het aantal goede cijfers in $y_1(20)$ en de gemeten rekentijd vermeld. Hieruit blijkt, dat het getransformeerde probleem (4.3b) met minder stappen nauwkeuriger wordt opgelost dan (4.3a). Dit komt doordat de eigenwaarden van (4.3b) zuiver imaginair zijn. Opgemerkt wordt nog, dat beide procedures geen stappen verwerpen. De schatting van de locale fout is bij ARK in het geval van polynoom $P_3(z)$ gebaseerd op een 3^e orde en bij STABTARK2VS op een 2^e orde referentieformule. Hiermee worden in de tabellen 4.6 en 4.7 de verschillen tussen beide procedures verklaard. Het stapkeuzemechanisme van ARK is gebaseerd op de laatste 3 gebruikte stappen, terwijl STABTARK2VS slechts de laatsgenomen stap gebruikt; dit verklaart de verschillen in tabel 4.5.

Tabel 4.5

Probleem (4.3a) opgelost met schema's, gegenereerd door het polynoom $P_2(z)$

opgegeven tolerantie	STABTARK2VS			ARK		
	aantal stappen	aantal goede cijfers	rekentijd	aantal stappen	aantal goede cijfers	rekentijd
10^{-3}	148	1.1	1.7	165	1.5	1.7
10^{-4}	312	1.6	3.7	378	1.8	3.9
10^{-5}	667	2.0	7.9	852	2.8	8.9
10^{-6}	1435	2.6	16.9	1860	3.1	19.4

Tabel 4.6

Probleem (4.3a) opgelost met schema's, gegenereerd door het polynoom $P_3(z)$

opgegeven tolerantie	STABTARK2VS			ARK		
	aantal stappen	aantal goede cijfers	rekentijd	aantal stappen	aantal goede cijfers	rekentijd
10^{-3}	101	0.2	1.2	97	0.1	1.3
10^{-4}	173	0.7	2.2	204	1.4	2.7
10^{-5}	306	1.4	3.7	444	2.0	5.6
10^{-6}	540	2.1	6.7	957	2.6	12.0

Tabel 4.7

Probleem (4.3b) opgelost met schema's, gegenereerd door het polynoom $P_3(z)$

opgegeven tolerantie	STABTARK2VS			ARK		
	aantal stappen	aantal goede cijfers	rekentijd	aantal stappen	aantal goede cijfers	rekentijd
10^{-3}	72	0.7	0.9	88	1.5	1.3
10^{-4}	123	1.1	1.6	188	2.6	2.7
10^{-5}	216	1.3	2.8	406	2.7	5.7
10^{-6}	379	1.7	4.9	870	3.6	12.2

Tabel 4.8

Probleem (4.3b) opgelost met schema's, gegenereerd door het polynoom $P_5(z)$

opgegeven tolerantie	STABTARK2VS			ARK		
	aantal stappen	aantal goede cijfers	rekentijd	aantal stappen	aantal goede cijfers	rekentijd
10^{-3}	66	1.6	1.0	76	0.5	1.5
10^{-4}	136	1.8	2.0	165	1.6	3.0
10^{-5}	282	2.2	4.1	352	2.3	6.4
10^{-6}	601	2.8	9.0	756	2.9	13.6

4.4. De parabolische partiële differentiaalvergelijking $U_t = U_{xx}$

Beschouw het beginrandwaardeprobleem

$$(4.4)\qquad \begin{cases} U_t = U_{xx}\ , & 0 \le x \le \pi\ , \quad t \ge 0 \\ U(0,t) = U(\pi,t) = 0 \\ U(x,0) = \sin(x) \end{cases}$$

met als analytische oplossing

$$U(x,t) = \sin(x)\ \exp(-t).$$

Door partiële discretisatie naar de plaatsvariabele (vgl. VAN DER HOUWEN [1974], blz. 8-13), wordt (4.4) herleid tot een (groot) lineair stelsel gewone differentiaalvergelijkingen van de vorm

$$(4.5)\qquad \frac{dU}{dt} = F(U) = DU,$$

hierin is D de tridiagonale matrix

$$D = \frac{1}{(\Delta x)^2}\begin{pmatrix} -2 & 1 & & & \bigcirc \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ \bigcirc & & & 1 & -2 \end{pmatrix}$$

met Δx de gekozen maaswijdte. De vector $U(t)$ bestaat uit N componenten $U_j(t) = U(j\Delta x,t)$, waarbij N volgt uit de relatie $(N+1)\Delta x = \pi$.
De eigenwaarden van de Jacobiaan D van F zijn

$$\delta_j = -\frac{4}{(\Delta x)^2}\sin^2\left(\frac{j\Delta x}{2}\right), \qquad j = 1,\ldots,N \quad .$$

Voor de spectraalradius $\sigma(D)$ geldt $\sigma(D) \simeq \frac{4}{(\Delta x)^2} = \frac{4(N+1)^2}{\pi^2}$. De eigenwaarden δ_j liggen dus verspreid over het interval $[-\frac{4}{(\Delta x)^2},0]$. Op grond hiervan zullen we voor onze Taylor-Runge-Kutta schema's polynomen met een maximale reële stabiliteitsgrens gebruiken. De stabiliteitsconditie (2.15) heeft tot gevolg dat de maximaal stabiele staplengte $\frac{1}{4}\beta_{real}(\Delta x)^2$ voldoende klein is om aan nauwkeurigheidseisen te voldoen.
Voor u'' geldt $u'' = G(u) = D^2U$ met

$$D^2 = \frac{1}{(\Delta x)^4}\begin{pmatrix} 5 & -4 & 1 & & & \bigcirc \\ -4 & 6 & -4 & 1 & & \\ 1 & -4 & 6 & -4 & 1 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ \bigcirc & & \ddots & \ddots & \ddots & \ddots \\ & & & 1 & -4 & 6 & -4 \\ & & & & 1 & -4 & 5 \end{pmatrix}$$

Integreren we (4.4) met het algemene polynoom $P(z) = 1 + z + \beta_2 z^2 + \beta_3 z^2 + \ldots$ met stabiliteitsgrens β_{real}, dan hebben we in elke stap te maken met de volgende twee fouten ρ_n en ε_n:

1) $\rho_n \underline{\underline{\text{p.d.}}}$ *de locale discretiseringsfout;*
hiervoor geldt bij benadering

$$\rho_n \simeq (\tfrac{1}{2}-\beta_2)h_n^2 U_{tt} + (\tfrac{1}{6}-\beta_3)h_n^3 U_{ttt};$$

2) $\varepsilon_n \underline{\underline{\text{p.d.}}}$ *de fout door de benadering van* U_t *en* U_{tt} *door* $F(U)$ *en* $G(U)$;
deze fout is gevolg van de partiële discretisatie; hiervoor geldt bij benadering

$$\varepsilon_n \simeq -\frac{1}{12}(\Delta x)^2 h_n U_{tt} - \frac{1}{6}(\Delta x)^2 h_n^2 U_{ttt};$$

waarbij $h_n = \frac{1}{4}\beta_{real}(\Delta x)^2$.
We onderscheiden nu 2 gevallen

Geval 1. Het polynoom is 1^e orde consistent. In dit geval is $\beta_2 \neq \frac{1}{2}$ en is voor hoge graad van $P(z)$ ongeveer 1/6 zodat

$$\rho_n \simeq \frac{1}{48}\beta_{real}^2 (\Delta x)^4 U_{tt} \quad \text{en} \quad \varepsilon_n \simeq -\frac{1}{48}\beta_{real}(\Delta x)^4 U_{tt}.$$

ρ_n is weliswaar in orde gelijk aan ε_n, maar is een factor β_{real} groter.
Gevolg: Voor grote waarden van β_{real} wordt de totale locale fout in grote mate bepaald door de locale discretiseringsfout en dus door de orde van de gebruikte formule.

Geval 2. Het polynoom is 2^e orde consistent. In dit geval is $\beta_2 = \frac{1}{2}$ en geldt $\beta_3 \simeq 1/10$ voor hoge graad van $P(z)$ zodat

$$\rho_n \simeq \frac{1}{960}\beta_{real}^3 (\Delta x)^6 U_{ttt} \quad \text{en} \quad \varepsilon_n \simeq -\frac{1}{48}\beta_{real}(\Delta x)^4 U_{tt}$$

Gevolg: De totale locale fout wordt hoofdzakelijk bepaald door de discretisatiefout ε_n en dus door de keuze van Δx.

We hebben probleem (4.4) opgelost met gestabiliseerde Taylor-Runge-Kutta schema's (procedure STABTARK) en met gestabiliseerde Runge-Kutta schema's (procedure ARK (zie §4.3)). Hierbij zijn de volgende polynomen voorgeschreven

$T_3(1+z/9)$, $\beta_{real} = 18$;

$T_4(1+z/16)$, $\beta_{real} = 32$;

$T_5(1+z/25)$, $\beta_{real} = 50$;

met T_i het i^e graads Chebyshev-polynoom;
en

$P_4(z)$ uit §4.2, $\beta_{real} = 12.05$;

$P_5(z)$ uit §4.2, $\beta_{real} = 19.45$;

De eerste 3 polynomen zijn 1^e orde, de laatste 2 2^e orde consistent. Het integratie-interval was [0,2]; de staplengte bedroeg $\frac{1}{4}\beta_{real}(\Delta x)^2$; Δx had de waarden $\pi/20$, $\pi/50$ en $\pi/100$. De procedures "firstder" en "secondder" zijn zo geprogrammeerd, dat voor het aantal bewerkingen geldt (als functie van het aantal vergelijkingen)

	firstder	secondder
aantal optellingen	3n	5n-6
aantal vermenigvuldigingen	2n	3n
aantal assignments	4n+2	6n+3

zodat geldt $c(G) \simeq \frac{3}{2} c(F)$.

Stellen nu f en g het totale aantal evaluaties van F(u) en G(u) voor, dan definiëren we de *bewerkelijkheid* B

$$B \underline{\underline{p.d.}} \frac{f+\alpha*g}{100\Delta x} \quad ,$$

hierin heeft α de waarde $\frac{3}{2}$ resp. 0 bij integratie met STABTARK resp. ARK.

De figuren 4.2 en 4.3 tonen de precisie-bewerkelijkheidsgrafieken voor verschillende polynomen. De hierbij behorende getalwaarden en de gemeten rekentijden staan vermeld in tabel 4.9.

Hieruit blijkt, dat voor dezelfde nauwkeurigheid 2^e orde schema's inderdaad minder bewerkelijk zijn dan 1^e orde schema's. Tevens geldt, dat, gegeven een stabiliteitspolynoom van oneven graad, het gestabiliseerde Taylor-Runge-Kutta schema efficiënter is dan het corresponderende Runge-Kutta schema. Om aan te tonen, dat

Fig. 4.2

Precisie-bewerkelijkheidsgrafieken van verschillende polynomen voor probleem (4.4) opgelost met STABTARK

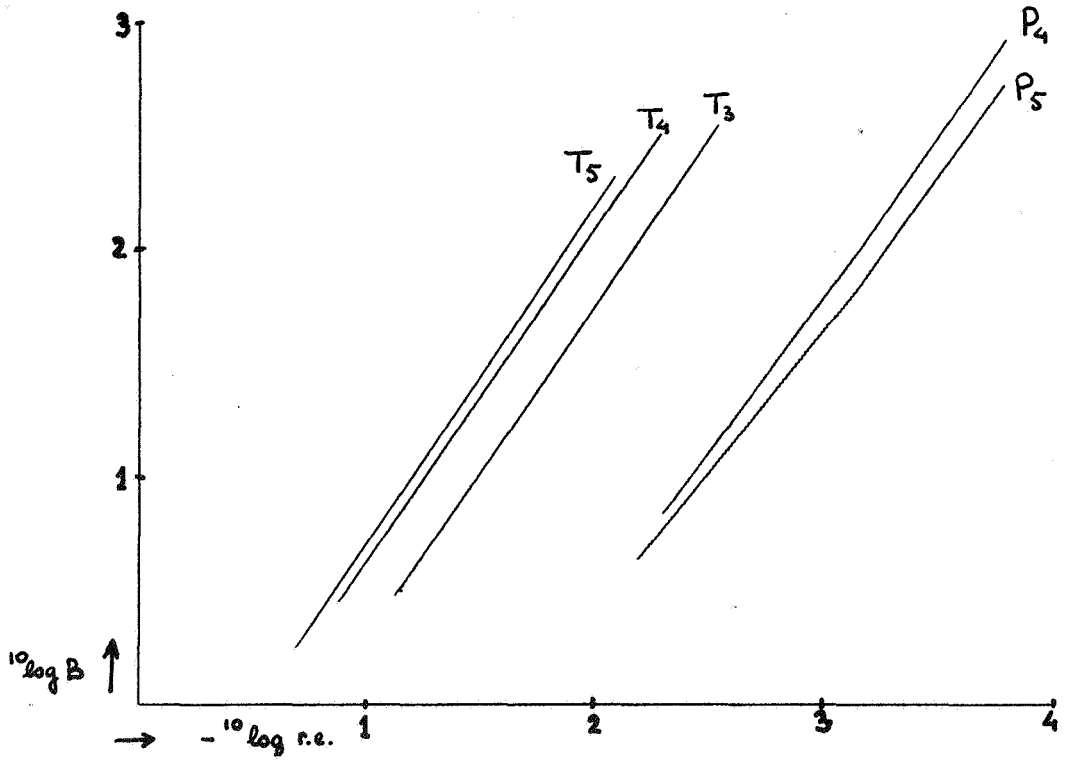
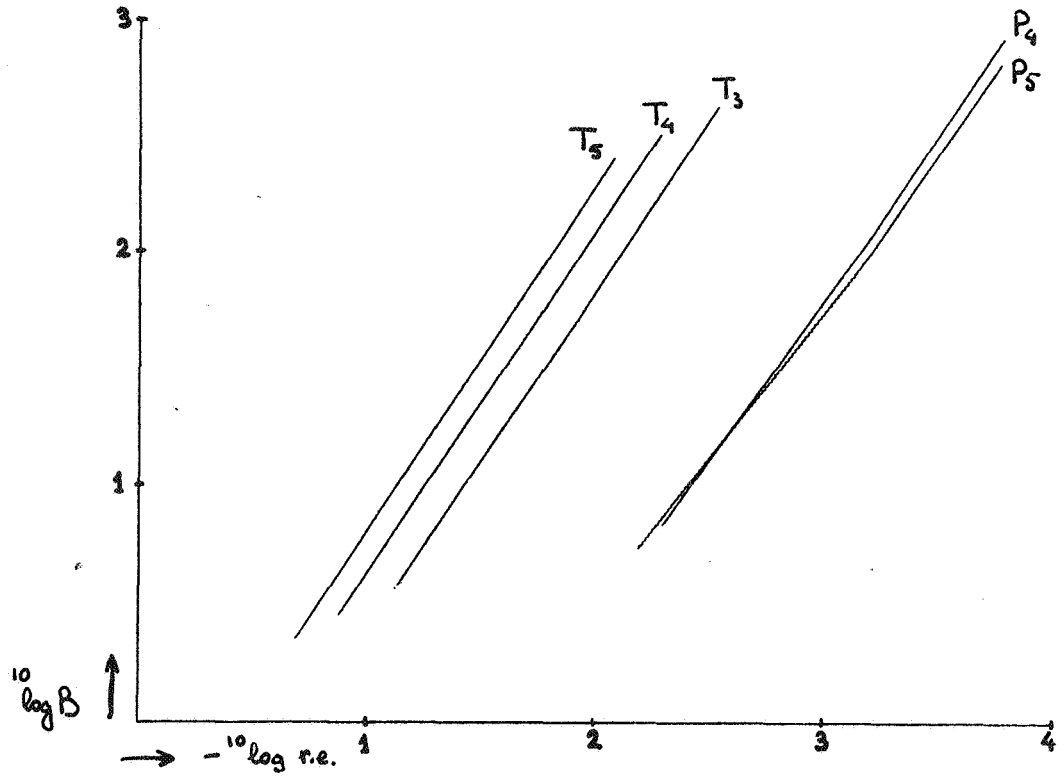


Fig. 4.3

Precisie-bewerkelijkheidsgrafieken van verschillende polynomen voor probleem (4.4) opgelost met ARK



Tabel 4.9

Resultaten met STABTARK en ARK voor verschillende polynomen

poly-noom	maas-wijdte Δx	relatieve fout	STABTARK			ARK		
			B	$^{10}\log B$	tijd	B	$^{10}\log B$	tijd
T_3	$\frac{\pi}{20}$	$7.6_{10}{}^{-2}$	3.0	0.5	0.3	3.7	0.6	0.5
	$\frac{\pi}{50}$	$1.2_{10}{}^{-2}$	45.0	1.7	4.1	54.1	1.7	5.5
	$\frac{\pi}{100}$	$3.0_{10}{}^{-3}$	358.9	2.6	32.5	431.0	2.6	40.1
T_4	$\frac{\pi}{20}$	$1.4_{10}{}^{-1}$	2.8	0.5	0.2	2.9	0.5	0.4
	$\frac{\pi}{50}$	$2.1_{10}{}^{-2}$	40.1	1.6	2.8	40.9	1.6	4.0
	$\frac{\pi}{100}$	$5.3_{10}{}^{-3}$	323.4	2.5	22.3	323.7	2.5	29.5
T_5	$\frac{\pi}{20}$	$2.1_{10}{}^{-1}$	1.8	0.3	0.1	2.3	0.4	0.3
	$\frac{\pi}{50}$	$3.3_{10}{}^{-2}$	26.1	1.4	2.2	32.8	1.5	3.2
	$\frac{\pi}{100}$	$8.2_{10}{}^{-3}$	207.5	2.3	17.0	260.0	2.4	23.0
P_4	$\frac{\pi}{20}$	$5.1_{10}{}^{-3}$	6.9	0.8	0.5	6.9	0.8	0.8
	$\frac{\pi}{50}$	$6.8_{10}{}^{-4}$	107.6	2.0	7.7	107.8	2.0	10.5
	$\frac{\pi}{100}$	$1.7_{10}{}^{-4}$	856.9	2.9	59.4	857.2	2.9	78.2
P_5	$\frac{\pi}{20}$	$6.6_{10}{}^{-3}$	4.3	0.6	0.4	5.5	0.7	0.7
	$\frac{\pi}{50}$	$7.2_{10}{}^{-4}$	66.8	1.8	5.7	83.7	1.9	7.9
	$\frac{\pi}{100}$	$1.7_{10}{}^{-4}$	530.9	2.7	43.6	664.0	2.8	58.4

Tabel 4.10

Resultaten met STABTARK2VS voor $\Delta x = \pi/20, \pi/50$

poly-noom	$\Delta x = \frac{\pi}{20}$; tol = 10^{-7}			$\Delta x = \frac{\pi}{50}$; tol = 10^{-8}		
	relatieve fout	B	tijd	relatieve fout	B	tijd
P_4	$4.2_{10}{}^{-3}$	40.4	5.3	$6.6_{10}{}^{-4}$	229.3	24.0
P_5	$4.2_{10}{}^{-3}$	39.2	5.5	$6.6_{10}{}^{-4}$	222.9	24.6

bij dit soort problemen integratie met een variabele staplengte de bewerkelijkheid doet toenemen zonder een grotere nauwkeurigheid te krijgen, is (4.4) opgelost met STABTARK2VS en voorgeschreven polynomen $P_4(z)$ en $P_5(z)$. De waarden van Δx waren $\pi/20$ en $\pi/50$; de besturingsparameters data [3 : 5] hadden de waarden 10^{-4}, 10^{-7}, 10^{-7} bij $\Delta x = \pi/20$ en 10^{-4}, 10^{-8}, 10^{-8} bij $\Delta x = \pi/50$. De resultaten staan in tabel 4.10.

Kijken we in tabel 4.9 bij P_5, dan zien we, dat door verkleinen van Δx met een factor c de fout met een factor c^2 wordt verkleind en de bewerkelijkheid met een factor $\frac{1}{c^3}$ wordt vergroot. Op grond hiervan kunnen we voor P_5 bij $\Delta x = 0.96 * \frac{\pi}{50}$ een fout van $6.6_{10}{}^{-4}$ verwachten bij een bewerkelijkheid van ongeveer 80. Dit in vergelijking met een bewerkelijkheid van 223 in tabel 4.10 voor dezelfde precisie.

5. CONCLUSIE

In dit verslag zijn gestabiliseerde Taylor-Runge-Kutta methoden beschouwd. De theorie van deze methoden blijkt grote overeenkomsten te hebben met gestabiliseerde Runge-Kutta methoden. Wat nauwkeurigheid betreft zijn de door ons beschouwde methoden gelijk aan Runge-Kutta methoden. Is de 2^e afgeleide niet te bewerkelijk, dan lossen Taylor-Runge-Kutta methoden het betreffende probleem efficiënter op dan de corresponderende Runge-Kutta methode. Dit betekent vooral voor grote problemen een aanzienlijke winst, hetgeen in testvoorbeeld 4.4 duidelijk naar voren komt. Staat men een vast aantal evaluaties van de 2^e afgeleide toe, dan wordt aangeraden een Taylor-Runge-Kutta methode te gebruiken, die wordt gegenereerd door een stabiliteitspolynoom van oneven graad.

Uit de testvoorbeelden is gebleken, dat de methoden goede resultaten geven vooral voor het geval de eigenwaarden van de Jacobiaan in het negatieve halfvlak liggen.

6. APPENDIX

Het bewijs van stelling 2.1

Indien we $E_n(y(x_n))$ met behulp van een Taylor reeks in machten van h_n willen ontwikkelen, kunnen de coëfficiënten niet uitgedrukt worden in de afgeleiden van y, maar moeten we deze coëfficiënten uitdrukken in de functies F(y) en G(y).

In het onderstaande maken we gebruik van de sommatieconventie van Einstein en moet voor F en G consequent F(y) en G(y) gelezen worden. Voor verdere details wat betreft de gebruikte notatie in de afleiding verwijzen we naar VAN KAMPEN [1974], blz. 4 - 5.

De Taylor ontwikkeling voor $E_n(y(x_n))$ luidt

$$
\begin{aligned}
E_n(y) = y &+ h_n \sum_{j=0}^{m-1} \lambda_{m,j} F + \\
&+ h_n^2 \left[\sum_{j=1}^{m-1} \lambda_{m,j} \sum_{k=0}^{j-1} \lambda_{j,k} F_r F^r + \sum_{j=0}^{m-1} \mu_{m,j} G \right] + \\
&+ h_n^3 \left[\sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=1}^{j-1} \lambda_{j,k} \sum_{\ell=0}^{k-1} \lambda_{k,\ell} F_r F_s^r F^s + \right. \\
&\quad + \sum_{j=1}^{m-1} \lambda_{m,j} \sum_{k=0}^{j-1} \mu_{j,k} F_r G^r + \\
&\quad + \tfrac{1}{2} \sum_{j=1}^{m-1} \lambda_{m,j} \left(\sum_{k=0}^{j-1} \lambda_{j,k} \right)^2 F_{rs} F^r F^s + \\
&\quad \left. + \sum_{j=1}^{m-1} \mu_{m,j} \sum_{k=0}^{j-1} \lambda_{j,k} G_r F^r \right] + \\
&+ h_n^4 \left[\sum_{j=3}^{m-1} \lambda_{m,j} \sum_{k=2}^{j-1} \lambda_{j,k} \sum_{\ell=1}^{k-1} \lambda_{k,\ell} \sum_{i=0}^{\ell-1} \lambda_{\ell,i} F_r F_s^r F_t^s F^t + \right. \\
&\quad \left. + \sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=1}^{j-1} \lambda_{j,k} \sum_{\ell=0}^{k-1} \mu_{k,\ell} F_r F_s^r G^s + \right.
\end{aligned}
$$

$$+ \frac{1}{2}\sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=1}^{j-1} \lambda_{j,k} \left(\sum_{\ell=0}^{k-1} \lambda_{k,\ell}\right)^2 F_r F^r_{st} F^s F^t +$$

$$+ \sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=1}^{j-1} \mu_{j,k} \sum_{\ell=0}^{k-1} \lambda_{k,\ell} F_r G^r_s F^s +$$

$$+ \sum_{j=2}^{m-1} \lambda_{m,j} \sum_{k=0}^{j-1} \lambda_{j,k} \sum_{\ell=1}^{j-1} \lambda_{j,\ell} \sum_{i=0}^{\ell-1} \lambda_{\ell,i} F_{rs} F^r_t F^t F^s +$$

$$+ \sum_{j=1}^{m-1} \lambda_{m,j} \sum_{k=0}^{j-1} \lambda_{j,k} \sum_{k=0}^{j-1} \mu_{j,k} F_{rs} F^r G^s +$$

$$+ \frac{1}{6}\sum_{j=1}^{m-1} \lambda_{m,j} \left(\sum_{k=0}^{j-1} \lambda_{j,k}\right)^3 F_{rst} F^r F^s F^t +$$

$$+ \sum_{j=2}^{m-1} \mu_{m,j} \sum_{k=1}^{j-1} \lambda_{j,k} \sum_{\ell=0}^{k-1} \lambda_{k,\ell} G_r F^r_s F^s +$$

$$+ \sum_{j=1}^{m-1} \mu_{m,j} \sum_{k=0}^{j-1} \mu_{j,k} G_r G^r +$$

$$+ \frac{1}{2}\sum_{j=1}^{m-1} \mu_{m,j} \left(\sum_{k=0}^{j-1} \lambda_{j,k}\right)^2 G_{rs} F^r F^s \Bigg] +$$

$$+ O\left(h_n^5\right) \quad , \quad h_n \to 0.$$

Deze gehele uitdrukking geëvalueerd in het punt $y = y(x_n)$.

De Taylor ontwikkeling voor $y(x_{n+1}) = y(x_n+h_n)$ luidt

$$y(x_{n+1}) = y + h_n F + \frac{1}{2}h_n^2 F_r F^r + \frac{1}{6}h_n^3\left[F_r F^r_s F^s + F_{rs} F^r F^s\right] +$$

$$+ \frac{1}{24} h_n^4\left[F_r F^r_s F^s_t F^t + F_r F^r_{st} F^s F^t + 3F_{rs} F^r_t F^t F^s + F_{rst} F^r F^s F^t\right] +$$

$$+ O\left(h_n^5\right) \quad , \quad h_n \to 0.$$

Het rechterlid geëvalueerd in $y = y(x_n)$.

In de ontwikkeling van $E_n(y(x_n))$ komen nog partiële afgeleiden van G voor. Deze zijn echter uit te drukken in afgeleide van F door middel van de relatie $G = F_r F^r$, immers $G(y) = J(y)F(y)$. We geven hiervan een voorbeeld:

$$G_{rs}F^rF^s = \left(G_r\right)_s F^rF^s = \left(\left(F_tF^t\right)_r\right)_s F^rF^s =$$

$$\left(F_{rt}F^t + F_tF^t_r\right)_s F^rF^s = \left(F_{rst}F^t + F_{rt}F^t_s + F_{st}F^t_r + F_tF^t_{rs}\right)F^rF^s =$$

$$F_{rst}F^rF^sF^t + 2F_{rs}F^r_tF^tF^s + F_rF^r_{st}F^sF^t \ .$$

In de reeksen voor $E_n(y(x_n))$ en $y(x_{n+1})$ zijn dan de coëfficiënten van h_n uitgedrukt in F; identificatie van beide reeksen geeft de consistentie voorwaarden (2.4 t/m 2.11).

Is de beschouwde differentiaalvergelijking lineair, dan komen in beide reeksen slechts de partiële afgeleiden F_rF^r, $F_rF^r_sF^s$ en $F_rF^r_sF^s_tF^t$ voor. Deze afgeleiden zijn gelijk aan JF, J^2F respectievelijk J^3F. Hieruit volgt, dat bij lineaire vergelijkingen aan p voorwaarden voldaan moet zijn voor p^e orde consistentie. Hiermee zijn beide gedeelten van stelling 2.1 bewezen. □

Afleiding van de formules uit §2.8

Het geval p = 2 impliceert voor het opgegeven stabiliteitspolynoom $\beta_1 = 1$, $\beta_2 = \frac{1}{2}$, $\beta_3 \neq \frac{1}{6}$. Hieruit volgt $\lambda_{m,0} = 1$ en $\mu_{m,m-1} = \frac{1}{2}$. Indien de graad van het opgegeven polynoom even is, geldt $\mu_{1,0} \neq 0$. We stellen nu $\tilde{\mu}_{m,j} = 0$ voor $j = 1, \ldots, m-1$ om het rekenwerk te vergemakkelijken. De voorwaarden voor een 3^e orde referentieformule luiden nu

$$(6.1) \qquad \begin{cases} \tilde{\lambda}_{m,0} + \tilde{\lambda}_{m,m} = 1 \\ \tilde{\lambda}_{m,m} + \tilde{\mu}_{m,0} + \tilde{\mu}_{m,m} = \frac{1}{2} \\ \tilde{\lambda}_{m,m} + 2\tilde{\mu}_{m,m} = \frac{1}{3} \end{cases}$$

met als oplossing (vgl. (2.29))

$$\tilde{\lambda}_{m,0} = 1 - \tilde{\lambda}_{m,m} \; ; \quad \tilde{\mu}_{m,0} = \frac{1}{3} - \frac{1}{2}\tilde{\lambda}_{m,m} \; ; \quad \tilde{\mu}_{m,m} = \frac{1}{6} - \frac{1}{2}\tilde{\lambda}_{m,m} \; ;$$

De vrijheid in $\tilde{\lambda}_{m,m}$ kunnen we benutten door te eisen, dat de referentieformule 4^e orde consistent is voor lineaire differentiaalvergelijkingen. Voor "bijna" lineaire vergelijkingen (dat wil zeggen vergelijkingen, waarvan de rechterlidfunctie slechts langzaam varieert met y) zal dan de referentieformule "bijna" 4^e orde consistent zijn. Wegens (2.8) geeft dit nog de extra voorwaarde (mits $m \geq 2$)

$$(6.2) \qquad \frac{1}{2}\tilde{\lambda}_{m,m}\,\lambda_{m-1,0} + \frac{1}{2}\tilde{\mu}_{m,m} = \frac{1}{24}$$

waarbij $\lambda_{m-1,0} = 2\beta_3$, zodat

$$\tilde{\lambda}_{m,m} = \frac{1}{6-24\beta_3} \qquad \text{voor } \beta_3 \neq \frac{1}{4} \; .$$

In §4.3 hebben we gezien, dat het geval $\beta_3 = 1/4$ inderdaad kan optreden. Indien β_3 dichtij 1/4 ligt, blijkt het stelsel (6.1) + (6.2) slecht geconditioneerd. De waarden van de 4 parameters worden dan groot negatief of positief, hetgeen we met het oog op nauwkeurigheid en stabiliteit willen vermijden. We onderzoeken daartoe het geval $|\beta_3 - \frac{1}{4}| \geq \varepsilon$. Onder de aanname, dat $0 \leq \beta_3 \leq \beta_2 = 1/2$ geldt

$$-\frac{1}{24\varepsilon} \leq \tilde{\lambda}_{m,m} \leq -\frac{1}{6} \quad \text{als} \quad \frac{1}{4} + \varepsilon \leq \beta_3 \leq \frac{1}{2}$$

of

$$\frac{1}{6} \leq \tilde{\lambda}_{m,m} \leq \frac{1}{24\varepsilon} \quad \text{als} \quad 0 \leq \beta_3 \leq \frac{1}{4} - \varepsilon \; .$$

Eisen we $|\tilde{\lambda}_{m,m}| \leq 1$ dan impliceert dit $\varepsilon \geq \frac{1}{24}$. We lossen nu stelsel (6.1) + (6.2) op als $|\beta_3 - \frac{1}{4}| \geq \frac{1}{24}$. De parameters kunnen dan de volgende waarden aannemen

$$1) \qquad -1 \leq \tilde{\lambda}_{m,m} \leq -\frac{1}{6} \; ; \quad \frac{1}{4} \leq \tilde{\mu}_{m,m} \leq \frac{2}{3} \; ; \quad \frac{7}{6} \leq \tilde{\lambda}_{m,0} \leq 2 \; ; \quad \frac{5}{12} \leq \tilde{\mu}_{m,0} \leq \frac{5}{6} \; ;$$

of

$$2)\qquad \frac{1}{6} \le \tilde{\lambda}_{m,m} \le 1 \; ; \quad -\frac{1}{3} \le \tilde{\mu}_{m,m} \le \frac{1}{12} \; ; \quad 0 \le \tilde{\lambda}_{m,0} \le \frac{5}{6} \; ; \quad -\frac{1}{6} \le \tilde{\mu}_{m,0} \le \frac{1}{4} .$$

Voor het geval m = 1 of $|\beta_3 - \frac{1}{4}| < \frac{1}{24}$ hebben we $\tilde{\lambda}_{m,0} = \tilde{\lambda}_{m,m} = 1/2$ gekozen, hetgeen een 3^e orde referentieformule oplevert. Bovenstaande levert de formules (2.29), (2.30a) en (2.30b).

Indien de graad van het opgegeven polynoom oneven is, geldt $\mu_{1,0} = 0$. We stellen nu $\tilde{\mu}_{m,j} = 0$ voor $j = 0,\ldots,m-2$ en $\tilde{\mu}_{m,m} = 0$. De voorwaarden voor een 3^e orde consistente referentieformule zijn

$$(6.3)\qquad \begin{cases} \tilde{\lambda}_{m,0} + \tilde{\lambda}_{m,m} = 1 \\ \tilde{\lambda}_{m,m} + \tilde{\mu}_{m,m-1} = \frac{1}{2} \\ \tilde{\lambda}_{m,m} + 2\tilde{\mu}_{m,m-1}\ \lambda_{m-1,0} = \frac{1}{3} \end{cases}$$

met als oplossing (vgl. (2.31))

$$\begin{cases} \tilde{\lambda}_{m,0} = \frac{1}{2} + \tilde{\mu}_{m,m-1} \; ; \quad \tilde{\lambda}_{m,m} = \frac{1}{2} - \tilde{\mu}_{m,m-1} \; ; \\ \tilde{\mu}_{m,m-1} = \frac{1}{6-24\beta_3} \quad \text{voor } \beta_3 \neq \frac{1}{4} \end{cases}$$

Voor β_3 in een omgeving van 1/4 is stelsel (6.3) weer slecht geconditioneerd; voor β_3 in een omgeving van 1/6 geldt $\tilde{\lambda}_{m,m} \simeq 0$, zodat de referentieformule veel gaat lijken op de gebruikte Taylor-Runge-Kutta formule. We onderzoeken daarom het geval $|\beta_3 - \frac{1}{4}| \ge \varepsilon$ en $|\beta_3 - \frac{1}{6}| \ge \eta$. Onder de aanname dat $0 \le \beta_3 \le \frac{1}{2}$ geldt

$$-\frac{1}{24\varepsilon} \le \tilde{\mu}_{m,m-1} \le -\frac{1}{6} \quad \text{als } \frac{1}{4} + \varepsilon \le \beta_3 \le \frac{1}{2} \; ;$$

$$\frac{1}{2-24\eta} \le \tilde{\mu}_{m,m-1} \le \frac{1}{24\varepsilon} \quad \text{als } \frac{1}{6} + \eta \le \beta_3 \le \frac{1}{4} - \varepsilon \; ;$$

$$\frac{1}{6} \le \tilde{\mu}_{m,m-1} \le \frac{1}{2+24\eta} \quad \text{als } 0 \le \beta_3 \le \frac{1}{6} - \eta \; .$$

We nemen $\varepsilon = \frac{1}{24}$ en $\eta = \frac{1}{96}$. Als nu $|\beta_3 - \frac{1}{4}| \geq \frac{1}{24}$ en $|\beta_3 - \frac{1}{6}| \geq \frac{1}{96}$ dan lossen we stelsel (6.3) op. De parameters kunnen dan de volgende waarden aannemen

1) $-1 \leq \tilde{\mu}_{m,m-1} \leq -\frac{1}{6}$; $-\frac{1}{2} \leq \tilde{\lambda}_{m,0} \leq \frac{1}{3}$; $\frac{2}{3} \leq \tilde{\lambda}_{m,m} \leq \frac{3}{2}$;

of

2) $\frac{4}{7} \leq \tilde{\mu}_{m,m-1} \leq 1$; $\frac{15}{14} \leq \tilde{\lambda}_{m,0} \leq \frac{3}{2}$; $-\frac{1}{2} \leq \tilde{\lambda}_{m,m} \leq -\frac{1}{14}$;

of

3) $\frac{1}{6} \leq \tilde{\mu}_{m,m-1} \leq \frac{4}{9}$; $\frac{2}{3} \leq \tilde{\lambda}_{m,0} \leq \frac{17}{18}$; $\frac{1}{18} \leq \tilde{\lambda}_{m,m} \leq \frac{1}{3}$.

Indien $|\beta_3 - \frac{1}{4}| < \frac{1}{24}$ of $|\beta_3 - \frac{1}{6}| < \frac{1}{96}$, dan nemen we een 2^e orde referentieformule door $\tilde{\mu}_{m,m-1} = 0$ te keizen. Bovenstaande levert de formules (2.31), (2.32a) en (2.32b).

REFERENTIES

BEENTJES, P.A. [1972]: *Een ALGOL-60 versie van gestabiliseerde Runge-Kutta methoden*, NR 23/72, Mathematisch Centrum, Amsterdam.

BEENTJES, P.A. [1974a]: *NUMAL, a library of numerical procedures in ALGOL-60*, section 5.2.1.1.1.1.B, Mathematisch Centrum, Amsterdam.

BEENTJES, P.A. [1974b]: *NUMAL, a library of numerical procedures in ALGOL-60*, section 5.2.1.1.1.1.H, Mathematisch Centrum, Amsterdam.

CESCHINO, F. & J. KUNTZMANN [1963]: *Méthodes numériques - Problèmes différentiels de conditions initiales*, Dunod, Paris.

DAVIS, H.T. [1962]: *Introduction to nonlinear differential and integral equations*, Dover, New York.

DEKKER, K. [1974]: *NUMAL, a library of numerical procedures in ALGOL-60*, section 5.2.1.1.1.1.I, Mathematisch Centrum, Amsterdam.

DEKKER, T.J., P.W. HEMKER & P.J. VAN DER HOUWEN [1972]: *Colloquium stijve differentiaalvergelijkingen*, deel 1, MCS 15.1, Mathematisch Centrum, 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.

HOUWEN, P.J. VAN DER [1970]: *One step methods for linear initial value problems I. Polynomial methods*, TW 119/70, Mathematisch Centrum, Amsterdam.

HOUWEN, P.J. VAN DER [1971]: *Stabilized Runge-Kutta methods with limited storage requirements*, TW 124/71, Mathematisch Centrum, Amsterdam.

HOUWEN, P.J. VAN DER [1974]: *Numerieke integratie van differentiaal-vergelijkingen*, Deel I: Eenstapsmethoden, MCS 24.1, Mathematisch Centrum, Amsterdam.

KAMPEN, S.P.N. VAN [1974]: *Een 4^e orde gegeneraliseerde Runge-Kutta methode*, NN 2/74, Mathematisch Centrum, Amsterdam.