MC SYLLABUS 29.2

COLLOQUIUM NUMERIEKE PROGRAMMATUUR

DEEL 2

H.J.J. TE RIELE (RED.)

MATHEMATISCH CENTRUM AMSTERDAM 1977

AMS(MOS) subject classification scheme (1970): 65-01

ISBN 90 6196 144 0

INHOUD

VOORWOORD

Het tweede deel van het Colloquium Numerieke Programmatuur werd, evenals het eerste, georganiseerd door de afdeling Numerieke Wiskunde van het Mathematisch Centrum. De bijeenkomsten vonden eenmaal per maand plaats in de maanden oktober, november, december 1976 en januari, februari 1977. Het aantal deelnemers bedroeg gemiddeld 33.

Lag in het eerste deel van het colloquium de nadruk op het *gebruik* van numerieke programmatuur, in het tweede deel werd daarnaast, mede op verzoek van enkele deelnemers, aandacht besteed aan de aan de programmatuur ten grondslag liggende *algoritmen*. Een ander belangrijk verschil is, dat veel van de in het tweede deel van het colloquium besproken programmatuur nog in ontwikkeling is, en daardoor (nog) niet in de grote programmatheken ACCULIB, IMSL, NAG en NUMAL beschikbaar is. De behandelde onderwerpen waren van zeer uiteenlopende aard, t.w. speciale 2^e orde differentiaalvergelijkingen, partiële differentiaalvergelijkingen, lineaire programmering, minimaliseren onder niet-lineaire nevenvoorwaarden, integraalvergelijkingen, interpolatie, sommatie en approximatie.

Helaas bleek het niet mogelijk een uitgebreid verslag van de voordracht over lineaire programmering (hoofdstuk 4) in deze syllabus op te nemen. Wel is een samenvatting van deze voordracht opgenomen. Van alle andere gehouden voordrachten bevat deze syllabus wel een uitgebreid verslag.

Het is inmiddels gebleken dat de syllabus van het eerste deel van dit colloquium reeds dienst doet als documentatie bij een cursus over numerieke programmatuur op een van de universitaire rekencentra. Wij hopen dat ook deze syllabus de weg van zijn voorganger zal volgen.

Tenslotte bedanken wij diegenen die hebben bijgedragen aan de realisatie van het colloquium en van deze syllabus, met name de sprekers van buiten het MC voor hun bijdrage (Drs. S.J. Polak, Dr. J.L. de Jong en Drs. C.G. van der Laan), Mevr. R.W.T. Riechelmann-Huis en Mevr. C.J. Klein Velderman-Los voor het bekwame typewerk, en de heren van de afdeling reproductie voor het drukken en binden van dit boekje.

H.J.J. te Riele.

1. HOGERE ORDE RUNGE-KUTTAMETHODEN VOOR SPECIALE 2e ORDE DIFFERENTIAAL-VERGELIJKINGEN

door P.A. Beentjes en W.J. Gerritsen
(Mathematisch Centrum)

1.1. Inleiding

De algemene vorm van een m-punts Runge-Kuttamethode voor de numerieke oplossing van het (vektor) beginwaardenprobleem

$$y'' = f(x,y), \quad y_0 = y(x_0) \quad \text{en} \quad y'_0 = y'(x_0) \tag{1.1.1}$$

wordt gedefinieerd door het volgende schema:

$$\begin{aligned} k_i &= h_\ell f\left(x_\ell + M_i h_\ell, y_\ell + h_\ell\left(M_i y'_\ell + \sum_{j=0}^{i-1} K_{ij} k_j\right)\right), \quad i = 0(1)m-1, \\ y_{\ell+1} &= y_\ell + h_\ell\left(y'_\ell + \sum_{i=0}^{m-1} A_i k_i\right), \quad y'_{\ell+1} = y'_\ell + \sum_{i=0}^{m-1} a_i k_i, \quad \ell = 0,1,2,\ldots,L. \end{aligned} \tag{1.1.2}$$

Hierin zijn M_i, K_{ij}, A_i en a_i nog vrij te kiezen parameters.

De eerste hogere orde methoden die voor 2e orde differentiaalvergelijkingen zonder 1e afgeleide werden ontwikkeld, dateren uit 1925 en werden geconstrueerd door NYSTRÖM [1925]. Deze methoden worden dan ook nog wel Runge-Kutta-Nyström (RKN-) methoden genoemd. Op een enkele uitzondering na (ALBRECHT [1955]) is er daarna weinig onderzoek meer gedaan op het gebied van hogere orde RKN-methoden. Pas in het laatste decennium is het onderzoek weer in belangrijke mate toegenomen, mede dankzij actuele toepassingsgebieden in de ruimtevaart en de sterrenkunde. De langdurige stilstand in het onderzoek is voornamelijk gelegen in het feit dat 2e orde differentiaalvergelijkingen zonder veel moeite te schrijven zijn als een stelsel gewone 1e orde differentiaalvergelijkingen; voor dit laatste type

vergelijkingen is een heel arsenaal van redelijk goede methoden aanwezig (zie BUS [1976]). De hernieuwde belangstelling voor RKN-methoden heeft onder andere opgeleverd een 5e orde formule van ZONNEVELD [1964] en 6e-, 7e- en 8e orde formules van FEHLBERG [1972] en HAIRER [1976]. De formules van Zonneveld en Fehlberg zijn interessant omdat ze zonder al te veel moeite in iedere stap een benadering van de locale fout kunnen maken, waarmee de stap verantwoord kan worden gevarieerd. Naar dit aspect van RKN-methoden is ook op het Mathematisch Centrum de laatste tijd onderzoek verricht (zie BEENTJES en GERRITSEN [1976]). In de tweede en derde paragraaf van dit hoofdstuk zal aandacht worden geschonken aan de constructie en stabiliteit van hogere orde RKN-methoden. In de vierde paragraaf zullen enkele testresultaten van een drietal 7e orde RKN-formules worden besproken.

1.2. Constructie van hogere orde RKN-formules

Het ontwerpen van hogere orde RKN-formules is in zekere zin gemakkelijker dan het ontwerpen van Runge-Kuttamethoden voor 1e orde differentiaalvergelijkingen. Dit komt doordat het aantal consistentievergelijkingen (dit zijn de niet-lineaire vergelijkingen waaraan de parameters van (1.1.2) moeten voldoen om een bepaalde orde van exactheid van het schema te verkrijgen) voor een RKN-methode kleiner is dan het aantal consistentievergelijkingen voor een gewone Runge-Kuttaformule van dezelfde orde (zie tabel 1.2.1). De praktijk leert dat vooral hogere orde RKN-schema's minder functie-evaluaties vragen dan gewone Runge-Kuttaschema's (zie tabel 1.2.2). Dat geeft dus al een goede reden voor het gebruik van RKN-formules.

Tabel 1.2.1

Aantal consistentievergelijkingen

Orde formule	RKN-methode	Runge-Kuttamethode voor 1e orde differentiaalvergelijkingen
1	1	1
2	2	2
3	3	3
4	5	8
5	8	17
6	13	37
7	22	85
8	37	200

Tabel 1.2.2

Minimaal aantal functie-evaluaties per stap

Orde formule	RKN-methode	Runge-Kuttamethode voor 1e orde differentiaalvergelijkingen
1	1	1
2	2	2
3	2	3
4	3	4
5	4	6
6	5	7
7	7	9
8	8	11

Het oplossen van de consistentievergelijkingen (= het ontwerpen van een RKN-formule) wordt moeilijker naarmate de beoogde orde van de formule hoger is. Voor schema's met een orde p, $p \leq 6$ heeft Hairer een min of meer uniforme algoritme opgesteld voor het oplossen van de consistentievergelijkingen.

Meer moeite kost het om nog hogere orde methoden te ontwerpen; daarbij neemt men dikwijls aan dat er tussen de te vinden parameters $K_{j\ell}$, $j = 1(1)m-1$, $\ell = 0(1)m-2$, bepaalde lineaire betrekkingen gelden waarmee dan de oorspronkelijke consistentievergelijkingen worden vereenvoudigd. Sommige consistentievergelijkingen geven restricties voor de M_i, $i = 1(1)m-1$; het vinden van een expliciete vorm van deze restricties vergt, verhoudingsgewijs, het meeste werk bij het ontwerp van RKN-methoden. Het resultaat van alle vereenvoudigingen mondt in de regel uit in het oplossen van lineaire (Van der Monde-achtige) stelsels vergelijkingen, waarin nog enkele vrijheidsgraden voorkomen.

De constructie van RKN-formules wordt lastiger als men een efficiente formule wil hebben waarmee stapsgewijs een of andere foutschatting kan worden gemaakt (om bijvoorbeeld de stapgrootte te kunnen regelen). In principe kan dit met iedere willekeurige formule gebeuren, door de oplossing $y_{n+1}^{(2h)}$ (verkregen na één stap ter grootte $2h_n$) met de oplossing $y_{n+1}^{(h+h)}$ (verkregen na twee stappen ter grootte h_n) te vergelijken:

$$\rho_y = y_{n+1}^{(2h)} - y_{n+1}^{(h+h)} = \text{constante} * h_n^{p+1}. \qquad (1.2.1)$$

Dit is echter een duur proces, omdat er per stap ongeveer 50% meer werk gedaan moet worden.

Er zijn echter goedkopere foutschattingen van het volgende type

$$\rho_y = h^2 \sum_{i=0}^{n-1} B_i k_i, \qquad \text{(Zonneveld, Fehlberg)}, \tag{1.2.2}$$

$$\rho_{y'} = h \sum_{i=0}^{n-1} b_i k_i, \qquad \text{(Zonneveld)}, \tag{1.2.3}$$

waarin B_i en b_i, $i = 0(1)n-1$, geschikt gekozen parameters zijn, waarvoor extra consistentievergelijkingen moeten worden opgelost. Bij Zonneveld zijn de bovenstaande schattingen $0(h^p)$, bij Fehlberg $0(h^{p+1})$. Verder geldt voor de index n uit (1.2.2) en (1.2.3)

$$n = m+1, \quad \text{(Fehlberg; zie (1.1.2) voor definitie van m)}, \tag{1.2.4}$$

$$n \geq m+1, \quad \text{(Zonneveld)};$$

met andere woorden: er worden extra functie-evaluaties gemaakt om de foutschatting te maken. Beide genoemde auteurs gebruiken hiervoor echter ook steeds de eerste functie-evaluatie van de volgende stap, $f(x_{n+1}, y_{n+1})$. Als de laatste stap geaccepteerd wordt, zal deze evaluatie ook voor de volgende stap nodig zijn; hij kost dan in feite niets.

De op het Mathematisch Centrum ontwikkelde formules hebben ook foutschattingen van het type (1.2.2) en (1.2.3). De schattingen zijn $0(h^p)$, maar kunnen worden berekend zonder extra functie-evaluaties, dus $n = m$.

1.3. Stabiliteit

In deze paragraaf zal aandacht besteed worden aan de stabiliteit van het schema (1.1.2). In VAN DER HOUWEN [1975] wordt, voor een m-punts Runge-Kuttaformule, de volgende amplificatie-matrix $R^{(m)}(z)$ afgeleid:

$$R^{(m)}(z) = \begin{pmatrix} 1 + \sum_{\ell=0}^{m-1} A_{\ell} z R_{11}^{(\ell)}(z) & 1 + \sum_{\ell=0}^{m-1} A_{\ell} z R_{12}^{(\ell)}(z) \\ \sum_{\ell=0}^{m-1} a_{\ell} z R_{11}^{(\ell)}(z) & 1 + \sum_{\ell=0}^{m-1} a_{\ell} z R_{12}^{(\ell)}(z) \end{pmatrix},$$

waarin

$$R_{11}^{(j)} = 1 + \sum_{\ell=0}^{j-1} K_{j\ell}\, z\, R_{11}^{(\ell)}(z),\quad R_{12}^{(j)} = M_j + \sum_{\ell=0}^{j-1} K_{j\ell}\, z\, R_{12}^{(\ell)}(z),$$

$$R_{11}^{(0)} = 1,\quad R_{12}^{(0)} = 0,\quad z = h^2\delta,$$

$\delta \in \Delta$, Δ is het eigenwaardenspectrum van de Jacobiaan van het rechterlid van (1.1.1), h is de stapgrootte. Als we de eigenwaarden van $R^{(m)}(z)$ α noemen, dan heet een Runge-Kuttaschema stabiel als geldt dat $|\alpha_{1,2}| < 1$ en zwak stabiel wanneer $|\alpha_{1,2}| \leq 1$. Uit een onderzoek naar de analytische voortplanting van verstoringen voor vergelijkingen van de vorm

$$y'' = Jy$$

blijkt, dat de analytische amplificatie-matrix van de volgende vorm is:

$$A = \begin{pmatrix} \cosh(Dh) & (Dh)^{-1}\sinh(Dh) \\ Dh\,\sinh(Dh) & \cosh(Dh) \end{pmatrix},$$

waarin $D^2 = J$.
Voor de eigenwaarden α van A geldt:

$$\alpha^2 - 2\alpha\cosh\sqrt{z} + 1 = 0,\quad z = h^2\delta,\ \delta \in \Delta.$$

Hieruit volgt meteen dat $\alpha_{\pm} = \exp(\pm\sqrt{z})$. Dit betekent dat er zwakke stabiliteit is op de negatieve z - as en geen stabiliteit voor andere waarden van z. Juist om deze reden is het realistisch om die schema's te verkrijgen die een groot negatief *stabiliteitsinterval* - dat is het interval $[-\beta,0]$ bestaande uit z-waarden, waarvoor geldt dat $|\alpha_{1,2}| \leq 1$ - hebben. De linkergrens $-\beta$ wordt de *stabiliteitsgrens* van de methode genoemd. Het zal duidelijk zijn dat een Runge-Kuttaformule, met stabiliteitsgrens $-\beta$, stabiel is voor $|h| \leq |h_{max}| = \sqrt{\dfrac{\beta}{|\delta|_{max}}}$, waarin $|\delta|_{max}$ de maximale absolute eigenwaarde van de Jacobiaan van het rechterlid van (1.1.1) voorstelt. Voor m-punts Runge-Kuttaformules kunnen we in het algemeen zeggen, dat voor de eigenwaarden α van $R^{(m)}(z)$ geldt:

$$\alpha^2 - S\cdot\alpha + P = 0 \qquad (1.3.1)$$

met S: het spoor van $R^{(m)}(z)$
en P: de determinant van $R^{(m)}(z)$.

Wanneer we nu het Hurwitz-criterium toepassen op (1.3.1), dan vinden we dat $|\alpha_{1,2}| \leq 1$, als geldt:

(1.3.2) $\quad p_1(z) = P - 1 \leq 0$

(1.3.3) $\quad p_2(z) = S - P - 1 \leq 0$

(1.3.4) $\quad p_3(z) = -S - P - 1 \leq 0.$

Voor ieder polynoom $p_i(z)$, $i = 1,2,3$ geldt dat er een maximaal interval $[-\beta_i,0]$ is, waarvoor (1.3.2), (1.3.3) en (1.3.4) gelden. De stabiliteitsgrens $-\beta$ nu, wordt gevonden uit

$$[-\beta,0] = \bigcap_{i=1,2,3} [-\beta_i,0].$$

Voor een gegeven RKN-schema worden de grenzen $-\beta_i$, $i = 1,2,3$ gemakkelijk bepaald door de nulpunten van p_i, $i = 1,2,3$ te berekenen.

Voor hogere orde RKN-methoden is het uiterst moeilijk om de stabiliteitseigenschappen van schema's gunstig te beïnvloeden (althans analytisch) door middel van de weinige vrijheidsgraden die er nog in de schema's over zijn na het oplossen van de consistentievergelijkingen. In dit feit ligt waarschijnlijk de oorzaak dat de ontwerpers van hogere orde RKN-methoden tot nu toe geen aandacht aan de stabiliteit van hun schema's hebben geschonken. Verschillende van de nu bestaande schema's zijn dan ook in het geheel niet, of slechts in geringe mate, stabiel.

Voor de onlangs op het Mathematisch Centrum ontworpen hogere orde methoden is wél onderzoek gedaan naar de stabiliteit. Het is onder meer gelukt om analytisch een efficiente (3-punts) 4e orde formule met optimale stabiliteitsgrens te construeren, terwijl ook 5e-, 6e- en 8e orde formules zijn ontworpen, waarvan de stabiliteitsgrenzen aanzienlijk groter zijn dan van bestaande schema's van overeenkomstige orde.

1.4. Numerieke experimenten

We hebben drie 7e orde formules gekozen om de efficientie en nauw-

keurigheid van RKN-methoden te illustreren, te weten

A. de 9-puntsformule van FEHLBERG [1972];

B. de 7-puntsformule van HAIRER [1976];

C. de 7-puntsformule van BEENTJES & GERRITSEN [1976].

Deze drie schema's hebben reële stabiliteitsgrenzen van, respectievelijk, -9.9, -5.8 en -9.8. Voor alle methoden werd bij de experimenten de volgende stapkeuzestrategie gevolgd: zij $\rho_n(\rho_n')$ de geschatte fout in $y_n(y_n')$, dan worden de resultaten (x_n, y_n, y_n') verworpen als geldt

(1.4.1) $\quad |\rho_n| \geq \eta_n = \text{reltol}\ \|y_n\| + \text{abstol}$ (idem voor ρ_n'),

met reltol en abstol een relatieve - en absolute stuurprecisie. De nieuwe integratiestap h, wordt hierna als volgt berekend

$$h = h_n \sqrt[q]{\left|\frac{\eta_n}{\rho_n}\right|} * .95,$$

waarin q gelijk is aan de orde van de geschatte fout. Bij de formule van Hairer lieten we de integratiestap steeds uit twee stappen ter grootte h_n, gevolgd door één stap ter grootte $2*h_n$ (weer vanuit x_n) bestaan, om twee resultaten op $x_n + 2h_n$ te krijgen. Hiermee werd dan $\rho_n(\rho_n')$ berekend. Indien (1.4.1) daartoe aanleiding gaf, werden echter óók twee stappen h_n verworpen.

Voorbeeld 1.4.1. Baanvergelijking met excentriciteit $\varepsilon = .3$ (HULL [1972]).

$$y_1'' = -y_1/(y_1^2+y_2^2)^{3/2}, \quad y_1(0) = 1-\varepsilon, \ y_1'(0) = 0,$$

$$y_2'' = -y_2/(y_1^2+y_2^2)^{3/2}, \quad y_2(0) = 0, \quad y_2'(0) = \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\ .$$

Eindpunt van integratie: $t_e = 6\pi$. In tabel 1.4.1 zijn de testresultaten van de drie methoden voor dit voorbeeld gegeven.
In beide voorbeelden werd een beginstap van .01 genomen.

Tabel 1.4.1

Testresultaten voor voorbeeld 1.4.1

reltol(=abstol)	aantal functie-evaluaties ( ): aantal verworpen stappen			abs. precisie in t_e voor y_1		
	methode A	methode B	methode C	methode A	methode B	methode C
1e-6	279(7)	540(8)	317(12)	-3.6e-5	3.0e-6	2.5e-6
3e-7	315(8)	600(9)	376(16)	-1.0e-5	6.9e-7	5.8e-7
1e-7	369(10)	660(9)	436(19)	-3.2e-6	4.5e-7	1.5e-7
1e-8	486(14)	840(12)	572(23)	-2.6e-7	7.0e-8	7.7e-9
1e-9	648(20)	1080(15)	768(30)	-1.8e-8	1.2e-8	1.9e-10

Voorbeeld 1.4.2. FEHLBERG [1972]

$$\begin{cases} y_1'' = -4t^2 y_1 - 2y_2/(y_1^2+y_2^2)^{\frac{1}{2}}, \; y_1(t_i) = 0, \; y'(t_i) = -\sqrt{2\pi}, \; t_i = \sqrt{\frac{\pi}{2}}, \\ y_2'' = -4t^2 y_2 + 2y_1/(y_1^2+y_2^2)^{\frac{1}{2}}, \; y_2(t_i) = 1, \; y_2'(t_i) = 0. \end{cases}$$

Eindpunt van integratie: $t_e = 10$. In tabel 1.4.2 zijn de testresultaten van dit tweede voorbeeld gegeven.

Tabel 1.4.2

Testresultaten voor voorbeeld 1.4.2

reltol(=abstol)	aantal functie-evaluaties ( ): aantal verworpen stappen			abs. precisie in t_e voor y_1		
	methode A	methode B	methode C	methode A	methode B	methode C
3e-6	675(18)	1160(11)	1012(38)	2.5e-5	7.9e-5	9.5e-6
3e-7	1053(32)	1380(9)	1471(62)	7.9e-6	2.2e-5	1.1e-6
3e-8	1323(34)	1720(8)	1880(59)	1.5e-6	4.2e-6	1.4e-7
3e-9	1530(21)	2300(11)	2528(69)	2.8e-7	7.0e-7	1.6e-8
3e-10	1890(13)	3580(41)	3194(47)	4.3e-8	1.1e-7	1.8e-9
3e-11	2439(10)		4055(5)	6.4e-9		2.0e-10

Zet men de resultaten van de twee voorbeelden uit in een nauwkeurigheid-bewerkelijkheid-grafiek, dan blijkt dat methode B voor beide voorbeelden het minst efficient is. Verder is te zien dat methode A de meest efficiënte methode is met betrekking tot voorbeeld 2; bij voorbeeld 1 levert methode C de beste resultaten af.

LITERATUUR

ALBRECHT, J. [1955], *Beiträge zum Runge-Kutta-Verfahren*, Z. Angew. Math. Mech. 35, 100-110.

BEENTJES, P.A. & GERRITSEN, W.J. [1976], *Higher order Runge-Kutta methods for the numerical solution of second order differential equations without first derivatives*, Report NW 34/76, Mathematisch Centrum, Amsterdam.

BUS, J.C.P. (red.) [1976], *Colloquium Numerieke Programmatuur*, deel 1, MC Syllabus 29.1, Mathematisch Centrum, Amsterdam.

FEHLBERG, E. [1972], *Classical Eighth- and Lower-Order Runge-Kutta-Nyström Formulas with Stepsize Control for Special second-Order Differential Equations*, NASA Technical Report R-381.

HAIRER, E. [1976], *Méthodes de Nyström pour l'équation différentielle* y" = f(x,y), Université de Genève, Section de mathématiques.

HOUWEN, P.J. VAN DER [1975], *Stabilized Runge-Kutta methods for second-order differential equations without first derivatives*, Report NW 26/75, Mathematisch Centrum, Amsterdam.

HULL, T.E. [1972], *Comparing numerical methods for ordinary differential equations*, SIAM J. Numer. Anal. 9, 603-637.

NYSTRÖM, E.J. [1925], *Über die numerische Integration von Differentialgleichungen*, Acta Soc. Sci. Fenn. 50, 1-55.

ZONNEVELD, J.A. [1964], *Automatic Numerical Integration*, MC Tract 8, Mathematisch Centrum, Amsterdam.

2. TEDDY2, EEN PROGRAMMAPAKKET VOOR 2-DIMENSIONALE PARABOLISCHE PROBLEMEN IN SAMENGESTELDE GEBIEDEN

door S.J. Polak
(Philips, Eindhoven)

2.1. Inleiding

Toegepaste natuurkundigen willen gewoonlijk hun problemen niet oplossen *met* een computer, ze willen die problemen *door* een computer op laten lossen.

Er zijn een klein aantal programma's en programmapakketten (CARDENAS c.s. [1968], CARVER [1973], CSENDES [1975], GARY c.s. [1972], HELGASON c.s. [1971], NILSEN [1974], SINCOVEC c.s. [1975]) gemaakt met dit essentiële onderscheid als uitgangspunt. Van deze programma's en programmapakketten zijn de meeste bescheiden van opzet in verhouding tot de praktische problemen. Andere zijn ambitieuzer maar weinig robuust. Realistische problemen zijn in het algemeen ruimtelijk driedimensionaal. Alleen bij voldoende symmetrie is vereenvoudigen tot twee- of ééndimensionale problemen mogelijk. De meeste fysische opstellingen bestaan uit meerdere onderdelen.

In PDEL, het programmapakket dat op een probleemklasse gericht is die de probleemklasse van TEDDY2 gedeeltelijk overlapt, zijn alleen ADI-methoden beschikbaar. In het vervolg zullen we zien dat deze algoritmen alleen voor rechthoekige polygonen met voldoende resultaat onderzocht zijn. De eventueel niet rechtlijnige randen worden daar dan ook trapsgewijs benaderd. Verder mogen daar gebieden niet samengesteld zijn uit verschillende delen met continuïteitsvoorwaarden op de grenzen. In het vervolg zullen we zien dat TEDDY2 voor de verschillende hier gesignaleerde tekortkomingen goede oplossingen biedt. Anderzijds moet worden opgemerkt dat PDEL faciliteiten biedt voor hyperbolische problemen die in TEDDY2 niet aanwezig zijn. Ook 3-dimensionale problemen kunnen met TEDDY2 niet opgelost worden, terwijl in PDEL daar wel faciliteiten voor bestaan.

Met het programmapakket TEDDY2 kunnen we voor problemen met parabolische partiële differentiaalvergelijkingen in samengestelde gebieden benaderde oplossingen berekenen met de robuuste ADI- en LOD- methoden.

Het pakket kan gebruikt worden m.b.v. de probleemgeorienteerde taal PARDEL (partial differential equations language). Met deze taal kunnen problemen uit een klasse beschreven worden, die veel groter is dan de klasse waarvoor convergentiebewijzen bekend zijn voor de ADI- en LOD-methoden. Evenwel wijst de praktijk uit (vergelijkingen met meetresultaten b.v.) dat inderdaad deze algoritmen voor een grote klasse problemen nuttig gebruikt kunnen worden.

De hoofdzaken voor de gebruiker m.b.t. TEDDY2 en PARDEL zijn:
- het kost maar een paar uur om PARDEL te leren gebruiken,
- invoer voor TEDDY2 is, met alle voorbereiding (afgezien van het meten van materiaaleigenschappen, enz.) in een paar uur klaar, terwijl het schrijven van een speciaal programma voor hetzelfde probleem weken of zelfs maanden kan duren,
- de rekentijd en het geheugengebruik zijn gunstig in vergelijking met hetgeen door ons in speciale programma's voor vergelijkbare problemen werd geconstateerd.

In dit hoofdstuk beschrijven we eerst de klasse van parabolische begin-randwaardenproblemen in samengestelde gebieden (PBRP-en) in 2.2, vervolgens wordt het programmapakket TEDDY2 besproken met de taal PARDEL en in 2.7 e.v. bespreken we de ADI- en LOD-methoden en geven we konvergentiebewijzen voor een eenvoudig probleem.

2.2. Parabolische problemen in samengestelde gebieden

Tweedimensionale parabolische problemen in samengestelde gebieden worden gekarakteriseerd door

- een aantal gebieden in $\mathbb{R}_2$,
- per gebied een parabolische partiële differentiaalvergelijking,
- rand- en grensvoorwaarden.

De parabolische partiële differentiaalvergelijkingen hebben de vorm

$$\frac{\partial u}{\partial t} = a_1 \frac{\partial^2 u}{\partial x^2} + b_1 \frac{\partial^2 u}{\partial y^2} + a_2 \frac{\partial u}{\partial x} + b_2 \frac{\partial u}{\partial y} + cu + d. \qquad (2.2.1)$$

Een gebied is begrensd door (buiten-)randen en grenzen met andere gebieden. Randvoorwaarden hebben de vorm

$$(2.2.2) \qquad \frac{\partial u}{\partial n} = gu + f,$$

waarbij $\frac{\partial u}{\partial n}$ de normale afgeleide is.

De grensvoorwaarden hebben de vorm

$$(2.2.3) \qquad s_1 \frac{\partial u}{\partial n}1 + p_1 u_1 = s_2 \frac{\partial u}{\partial n}2 + p_2 u_2 + q;$$

hierbij slaan de indexen 1 en 2 op de twee gebieden die aan elkaar grenzen.

De coefficienten kunnen allemaal functies zijn van de ruimte- en tijdvariabelen en ook van de oplossing zelf. We veronderstellen verder $a_1 > 0$ en $b_1 > 0$. Op de grenzen moeten altijd twee voorwaarden gegeven zijn.

Opmerking 1. We houden ons hier niet bezig met existentie en eenduidigheid van oplossingen.

Opmerking 2. De randvoorwaarden hebben meestal een van de volgende vormen

- $u = f(x,y,t)$
- $\lambda \frac{\partial u}{\partial n} = c(u-u_0)$
- $\lambda \frac{\partial u}{\partial n} = c(u^4-u_0^4)$.

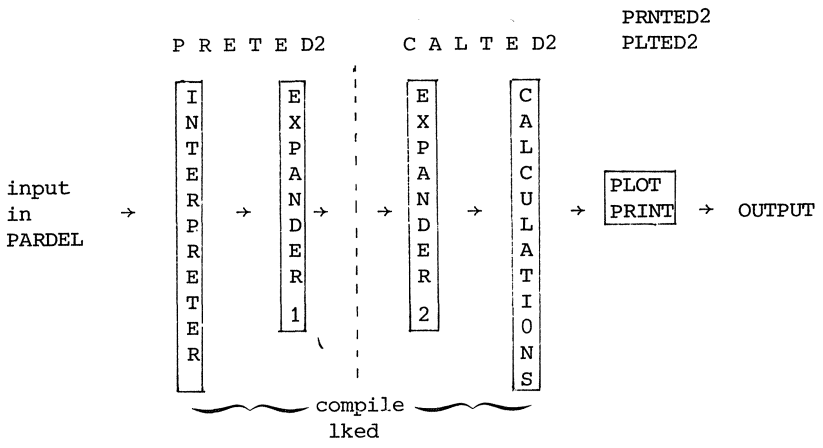
Opmerking 3. Op grenzen zijn de meest gebruikte voorwaarden

- $\lambda_1 \frac{\partial u}{\partial n}1 = \lambda_2 \frac{\partial u}{\partial n}2$
- $u_1 = u_2$
- $\lambda_1 \frac{\partial u}{\partial n}1 = \alpha(u_2-u_1)$.

Opmerking 4. In de praktijk geldt vaak dat in de operator $\frac{\partial}{\partial x}(a_1 \frac{\partial}{\partial x} u)$ voorkomt. Het is een tekortkoming van de huidige probleemklasse dat dit niet eenvoudig toegelaten is. In de praktijk vangen we dit op door a_2 van $\frac{\partial u}{\partial x}$ af te laten hangen. In de nabije toekomst wordt dit evenwel veranderd.

2.3. De structuur van TEDDY2.

De structuur van TEDDY2 is in fig. 2.3.1 te vinden.



Figuur 2.3.1

Van links naar rechts vinden we het volgende.

- PARDEL is de probleemgeorienteerde taal waarin de invoer voor TEDDY2 gegegeven moet worden.

Het hele programma is geschreven in FORTRAN en bestaat uit de modulen PRETED2, CALTED2, PRNTED2 en PLTED2.

- PRETED2 bestaat uit twee delen, het eerste deel, de interpreter, transformeert de invoer in tabellen. Deze tabellen bevatten een geordende versie van de invoer. Dan worden uit deze gegevens tabellen geconstrueerd, in expander 1, waarin de gegevens zo goed mogelijk beschikbaar zijn voor het uitvoeren van de algoritmen.

Omdat PRETED2 in Fortran geschreven is hebben we de gebruikelijke dimensioneringsproblemen. In PRETED2 hebben we daarom een stel arrays met vaste dimensie. Wanneer zo'n array vol is wordt disk-ruimte gebruikt.

De arrayruimte in PRETED2 is nog erg klein en varieert niet sterk per probleem. In PRETED2 worden verschillende programma's gegenereerd voor de berekeningen. Zo wordt het dimensieprobleem voor CALTED2 opgelost door de juiste dimensiestatements te genereren in PRETED2. Bij de beschrijving van PARDEL zal duidelijk worden dat de uitbreidingsfase, afhankelijk van de

invoer, moet worden voortgezet in CALTED2.

De gegenereerde Fortranprogramma's moeten worden gecompileerd en met de bestaande CALTED2-modulen tot één uitvoerbaar programma worden samengevoegd. Na de "compile-link"-stappen komt dan, nog steeds van links naar rechts

- CALTED2. Dit module bestaat uit twee delen, een expander en een algoritme gedeelte.

Verder zijn uitgebreide plot- en printfaciliteiten beschikbaar.

Opmerking. Het genereren van programmatuur is een heel goede manier om het extra werk dat de algemeenheid van een pakket met zich meebrengt tot een minimum te beperken.

2.4. De probleemgeorienteerde taal PARDEL

Vier soorten informatie zijn nodig voor de analyse van een probleem met TEDDY2.
Deze zijn

- probleembeschrijving,
- algoritme-informatie,
- outputwensen en
- systeemcommando's.

De eerste drie soorten moeten voor TEDDY2 in PARDEL gegeven worden.

De probleem beschrijving moet per gebied worden gegeven. De invoer bevat daarom één of meer

- regionblokken.

De algoritme-informatie wordt gegeven in een

- algoritmeblok

en de output-wensen in het

- informatieblok

voor PRETED2 en direkt, niet in een blok, voor PRNTED2 en PLTED2.

De systeemcommando's moeten gegeven worden in een taal (b.v. JCL voor IBM), afhankelijk van de rekenmachine.

Blokken bestaan uit statements. Blokken en statements in blokken kunnen in willekeurige volgorde gegeven worden. Men moet zich hierbij realiseren dat de statements geen opdrachten zijn die in volgorde worden uitgevoerd, maar een beschrijvende inhoud hebben.

Blokken worden voorafgegaan door blok identifier statements en input voor PARDEL wordt beëindigd met een enddata statement.

De schematische representatie in paragraaf 2.5 laat de invoerstructuur zien. In BARNEVELD BINKHUYSEN [1975] en POLAK [1975] wordt een gedetailleerde beschrijving met voorbeelden gegeven.

Waar het woord

- finter

wordt gebruikt mag een integer, real of wel een *gebruikersgekozen functienaam* gegeven of weggelaten worden. *Deze constructie* bepaalt grotendeels de flexibiliteit van TEDDY2. Fortranfuncties, corresponderend met de namen in de PARDEL-invoer, moeten onmiddellijk op het enddata statement volgen.

Deze functies mogen afhangen van

- één parameter die T aanduidt,
- twee parameters die X en Y aanduiden in die volgorde,
- drie parameters die X,Y en T aanduiden in die volgorde.

In functies die van X,Y en T afhangen kunnen de volgende commons gebruikt worden:

```
COMMON/UU/U
```

en

```
COMMON/UXY/U,UXL,UXR,UYL,UYR,DXL,DXR,DYL,DYR
```

In het eerste common is de laatst berekende U-waarde beschikbaar in het punt X,Y, in het tweede zijn alle gegevens van de vijfpuntsster (ISAACSON c.s. [1966]) om X,Y heen beschikbaar met de laatst berekende temperatuurwaarden.

2.5. Schematische representatie van PARDEL

Invoer PRETED2

(**) regionblokken voorafgegaan door een blokidentifier
- (*) boundary statement
- (**) boundary condition statement
- (*) equation statement
- (*) initial value statement

(*) algoritmeblok voorafgegaan door een blokidentifier.
  (*) x mesh statement
  (*) y mesh statement
  (*) tau statement
  (*) t begin statement
  (*) tend statement
  (*) method statement
  (*) accuracy statement

(*) informationblok
  (*) store statement

De invoer voor OUTTED2 is niet in blokken georganiseerd maar omvat
  (**) print statements
  (**) plot statements

* betekent één en slechts één
** betekent één of meer.

2.6. Statements in PARDEL

Invoer PRETED2

- region block.

region block identifier statement:

```
REGION;
```

equation statement:

```
DUDT = finter1 * D2UDX2 ± finter2 * D2UDY2
     ± finter3 * DUDX ± finter4 * DUDY ±
       finter5 * U ± finter6;
```

boundary statement:

```
(x1,y1) STRAIGHT (x2,y2) STRAIGHT ... etc. ...
  CIRCLE (a,b,flag) (xi,yi) .....;
```

boundary condition statement:

```
BC BETWEEN (x1,y1) AND (x2,y2) IS
 0 = finter1 * DUDN1 ± finter2 * DUDN2 ±
     finter3 * U1 ± finter4 * U2 ± finter5;
```

initial value statement:

U0 = finter;

or U0 = FILE;

- algorithm block.

algorithm block identifier statement:

ALGORITHM;

x mesh statement:

X (integer) = number, , , , ;

y mesh statement:

Y (integer) = number, , , , ;

accuracy statement:

ACCURACY = [DOUBLE PRECISION;
SINGLE PRECISION;]

tau statement:

TAU = finter;

t begin statement:

TBEGIN = number;

t end statement:

TEND = number;

method statement:

METHOD = [finter;
ADI ;
LOD ;]

information block.

store statement:

STORE AT

[TIMES = ...,...,...
T (integer) = ..., T (integer) ...,...
STEPNO = integer, integer, ...]

- last PRETED2 statement:

ENDDATA;

- print statement:

```
[ FULL      ]
[ RECTANGLE ]
[ LINE      ]   PRINT (integer)*
[ BOUNDARY  ]

[ DIAGONAL (...,...) AND (...,...)                  ]
[ X = ..., X = ..., ..., Y = ..., Y = ...,         ]
[ BETWEEN (...,...) AND (...,...)                   ]

[ TIMES = ...,...,...                               ]
[ T (integer) = .., T (integer) = ... ... ...       ]
[ STEPNO = integer, integer ..                      ]

[ REAL*N    ]
[ INTEGER*N ]
[ EXP*N     ]
```

- plot statement:

```
[ FULL      ]
[ RECTANGLE ]
[ LINE      ]   PLOT (format identifier)*
[ BOUNDARY  ]

[ DIAGONAL (..,..) AND (..,..)                      ]
[ X = .., X = .. .. , Y = ..., Y = ...,             ]
[ BETWEEN (..,..) AND (..,..)                       ]

[ TIMES = ..,..                                     ]
[ T (integer) = .., T (integer) = ...               ]
[ STEPNO = integer, integer ....                    ]

[              { U = ...,...                      } ]
[ ISOTHERM AT  {                                  } ]  ;
[              { U (integer) = ..., U (integer) = ..} ]
```

De integer in het print statement duidt het aantal beschikbare posities op de printer aan.

De "format identifier" kan zijn

A1,A2,A3,A4,A5,A6 of een paar getallen.

2.7. ADI- en LOD-methoden voor parabolische partiële differentiaalvergelijkingen.

We zullen in deze paragraaf de ADI-(alternating directions implicit) en LOD-(locally one dimensional) methoden bespreken aan de hand van een eenvoudig voorbeeld.

Als voorbeeld nemen we

(2.7.1) $U_t = U_{xx} + U_{yy}$ op een gebied $\Omega \subset \mathbb{R}_2$,

met rand $\delta\Omega$, $0 \leq t \leq T$, en

(2.7.2) $U = f(t,x,y)$ op $\delta\Omega$,

(2.7.3) $U(t=0) = g(x,y)$ op $\Omega \cup \delta\Omega$.

De rand $\delta\Omega$ en de functies f en g zijn zó, dat de oplossing U bestaat en éénduidig is. Verder veronderstellen we deze oplossing voldoende vaak differentieerbaar voor alle gebruikte afschattingen.

Op $(\Omega \cup \delta\Omega) \times [0,T]$ is een maas gedefinieerd door maaslijnen evenwijdig aan de x- en y-assen voor coördinaten x_i, $i = 1,\ldots,N$ en y_j, $j = 1,\ldots,M$ en tijdstappen $\tau_k > 0$ met $\sum_{k=1}^{\ell} \tau_k = t_\ell$, $t_L = T$, $t_0 = 0$.

Deze maas wordt aangevuld met alle snijpunten van maaslijnen met $\delta\Omega$. We noemen de snijpunten van maaslijnen in Ω reguliere maaspunten en snijpunten van maaslijnen met $\delta\Omega$ irreguliere maaspunten.

De volgende differentieoperatoren worden gebruikt

$$\Delta_\xi u(x_i,y_j,t) = (u(x_{i+1},y_j,t) - u(x_i,y_j,t))/(x_{i+1}-x_i)$$
$$\Delta_\eta u(x_i,y_j,t) = (u(x_i,y_{j+1},t) - u(x_i,y_j,t))/(y_{j+1}-y_j)$$
$$\nabla_\xi u(x_i,y_j,t) = (u(x_i,y_j,t) - u(x_{i-1},y_j,t))/(x_i-x_{i-1})$$
$$\nabla_\eta u(x_i,y_j,t) = (u(x_i,y_j,t) - u(x_i,y_{j-1},t))/(y_j-y_{j-1})$$
$$\nabla_\tau u(x_i,y_j,t_\ell) = (u(x_i,y_j,t_\ell) - u(x_i,y_j,t_\ell-\tau_\ell))/\tau_\ell$$
$$D^2_\xi = \Delta_\xi\nabla_\xi, \quad D^2_\eta = \Delta_\eta\nabla_\eta.$$

Waar misverstanden onwaarschijnlijk zijn worden argumenten x_i,y_j,t_ℓ weggelaten. Verder gebruiken we $h_x = \max_i|x_i-x_{i-1}|$, $h_y = \max_j|y_j-y_{j-1}|$ en

$h = \max(h_x, h_y)$, $\tau = \max_\ell \tau_\ell$.

In de maas heten twee punten (x_i, y_j) en (x_k, y_ℓ) buren als ófwel $|i-k| = 1$ en $j = \ell$ ófwel $i = k$ en $|j-\ell| = 1$. Voor twee functies z_1 en z_2 die gedefinieerd zijn op de reguliere maaspunten is het volgende inproduct gedefinieerd

$$(z_1, z_2) = (\sum z_1(x_i, y_j) z_2(x_i, y_j))/Q$$

waarbij over alle reguliere maaspunten gesommeerd wordt en Q het aantal reguliere maaspunten is. Verder is $\|z\| = (z,z)^{\frac{1}{2}}$.

Voor het vervolg veronderstellen we de begrippen consistentie, stabiliteit en convergentie bekend (zie b.v. ISAACSON c.s. [1968]). Eenstaps expliciete methoden hebben het nadeel dat moet gelden $\tau \leq ch^2$. Bij het gebruiken van impliciete methoden geldt zo'n beperking niet, maar nu moet bij iedere tijdstap een groot stelsel vergelijkingen opgelost worden. Weliswaar is de oplossing van het vorige tijdstip altijd voorhanden als benadering voor de oplossing op het volgende tijdstip, toch is het vaak te veel werk om zo'n stelsel op te lossen. Voor $N = M$ is met Gaussdecompositie $O(N^4)$ operaties nodig voor het oplossen van zo'n stelsel. We houden dit, hoewel i.h.a. minder nodig is, als referentie aan omdat we bij de ADI- en LOD-methoden zeker weten dat het aantal operaties $O(N^2)$ is. We zullen dit laatste in het vervolg nog bespreken.

Voor de analyse van de ADI-methode voor (2.7.1-3) veronderstellen we Ω rechthoekig met zijden evenwijdig aan de x- en y-assen. Voor de maas veronderstellen we $x_{i+1}-x_i = x_{j+1}-x_j$ voor alle i,j en $y_{i+1}-y_i = y_{j+1}-y_j$ voor alle i,j en $\tau_\ell = \tau$ voor alle ℓ. Het verhoogt het inzicht de vervanging van (2.7.1) met de ADI-methode op twee manieren te bekijken.

Voor de eerste afleiding splitsen we een tijdstap τ_ℓ in twee helften en benaderen (2.7.1) door

$$\nabla_{\frac{1}{2}\tau} u(t+\tfrac{1}{2}\tau) = D_\xi^2 u(t+\tfrac{1}{2}\tau) + D_\eta^2 u(t),$$

ofwel

$$(I-\tfrac{1}{2}\tau D_\xi^2) u(t+\tfrac{1}{2}\tau) = (I+\tfrac{1}{2}\tau D_\eta^2) u(t) \qquad (2.7.4)$$

op de eerste helft en door

$$\nabla_{\frac{1}{2}\tau} u(t+\tau) = D_\xi^2 u(t+\tfrac{1}{2}\tau) + D_\eta^2 u(t+\tau),$$

ofwel

$$(2.7.5) \qquad (I-\tfrac{1}{2}\tau D_\eta^2)u(t+\tau) = (I+\tfrac{1}{2}\tau D_\xi^2)u(t+\tfrac{1}{2}\tau)$$

op de tweede helft. De formules (2.7.4) en (2.7.5) zijn beide consistent met (2.7.1) met een fout $O(\tau^2+\tau h^2)$.

Per vergelijking komen drie onbekenden voor. Per halfstap vinden we zo een aantal ontkoppelde stelsels met tridiagonale matrices. Ieder stelsel correspondeert met de onbekenden op één maaslijn. De vergelijkingen (2.7.4) corresponderen zo met x-lijnen, (2.7.5) met y-lijnen. Voor het oplossen van een tridiagonaal stelsel met n onbekenden is 5n-4 operaties nodig (b.v. ISAACSON c.s. [1968]) dus bij N × M maaslijnen zijn 10MN + O(M+N) operaties per tijdstap nodig. De vergelijkingen (2.7.4) en (2.7.5) zijn wel consistent maar niet onvoorwaardelijk stabiel (als $h_x \neq h_y$). Samen zijn ze wel onvoorwaardelijk stabiel. We bespreken dit wat precieser n.a.v. de tweede afleiding.

Voor een oplossing van (2.7.1) geldt

$$(2.7.6) \qquad \exp\left(\tfrac{1}{2}\tau \frac{\partial^2}{\partial x^2}\right) \exp\left(\tfrac{1}{2}\tau \frac{\partial^2}{\partial y^2}\right) U(t) = \exp\left(-\tfrac{1}{2}\tau \frac{\partial^2}{\partial x^2}\right) \exp\left(-\tfrac{1}{2}\tau \frac{\partial^2}{\partial y^2}\right) U(t+\tau).$$

Hierbij is de commutativiteit van $\frac{\partial^2}{\partial x^2}$ en $\frac{\partial^2}{\partial y^2}$ voor U gebruikt en het feit dat linker en rechterlid gelijk zijn aan $U(t+\tfrac{1}{2}\tau_\ell)$. Als benadering nemen we

$$(2.7.7) \qquad (I-\tfrac{1}{2}\tau D_\xi^2)(I-\tfrac{1}{2}\tau D_\eta^2)u(t+\tau) = (I+\tfrac{1}{2}\tau D_\xi^2)(I+\tfrac{1}{2}\tau D_\eta^2)u(t).$$

Deze vergelijking is in reguliere maaspunten met vier reguliere buren equivalent met (2.7.4-5). Bij deze equivalentie komen we op een probleem dat in de westerse literatuur in een aantal publikaties over het hoofd gezien is. Dit probleem werd voor het eerst gesignaleerd in D'JAKONOV [1962] en wordt verder b.v. besproken in MITCHELL [1969], YANENKO [1971] en FAIRWEATHER c.s. [1967]. We kunnen de overeenkomstige leden van (2.7.4) en (2.7.5) aftrekken en vinden dan

$$\nabla_{\frac{1}{2}\tau} u(t+\tfrac{1}{2}\tau) - \nabla_{\frac{1}{2}\tau} u(t+\tau) = D_\eta^2 (u(t)-u(t+\tau)),$$

zodat

(2.7.8) $\quad 2u(t+\frac{1}{2}\tau) = \frac{1}{2}\tau D_\eta^2(u(t)-u(t+\tau))+u(t)+u(t+\tau).$

Invullen in b.v. (2.7.4) levert (2.7.7). Evenwel hebben we bij dit invullen *drie* waarden van $u(t+\frac{1}{2}\tau_\ell)$ nodig in $D_\xi^2 u(t+\frac{1}{2}\tau_\ell)$. Een van deze drie kan op $\delta\Omega$ liggen en voldoet niet noodzakelijkerwijs aan (2.7.8). Voor rechthoekige Ω is dit eenvoudig te verhelpen door i.p.v. $u(t+\frac{1}{2}\tau_\ell) = f(t+\frac{1}{2}\tau_\ell)$

(2.7.9) $\quad 2u(t+\frac{1}{2}\tau) = \frac{1}{2}\tau D_\eta^2(f(t)-f(t+\tau))+f(t)+f(t+\tau)$

te nemen.

Voor rechthoekige polygonen is er ook nog een manier (zie b.v. MITCHELL [1969]) om de equivalentie te handhaven. Voor andere gebieden is mij hiervoor geen methode bekend.

Bewijzen voor convergentie van oplossingen met de ADI-methode berekend maken altijd gebruik van de equivalentie tussen (2.7.4-5) en (2.7.7). We geven hier nu zo'n bewijs.

We veronderstellen rand- en beginwaarden exact gerepresenteerd. De foutfunctie $e(x_i,y_j,t_\ell)$ is gedefinieerd door

$$e = u - U.$$

Deze functie is bepaald door

(2.7.10) $\quad (I-\frac{1}{2}\tau D_\xi^2)(I-\frac{1}{2}\tau D_\eta^2)e(t+\tau) = (I+\frac{1}{2}\tau D_\xi^2)(I+\frac{1}{2}\tau D_\eta^2)e(t) + \psi$

met $\psi = O(\tau^3+\tau h^2)$ aangezien ψ juist de consistentiefout is.

(2.7.11) $\quad e = 0 \qquad$ op $\delta\Omega$,

(2.7.12) $\quad e(t=0) = 0 \quad$ op $\Omega \cup \delta\Omega$.

We definieren $\bar{e}(t)$ als de vector van $e(t)$ waarden in de reguliere maaspunten. Stel $\frac{\tau}{h^2} = c_x$. De matrix die bij $\frac{1}{2}\tau D_\xi^2$ hoort heeft de vorm $B_x = \frac{1}{2}c_x E_M$ met^x

$$E_M = \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}_{M\times M},$$

voor een zekere ordening van de onbekenden per lijn. Voor $\frac{\tau}{h_y^2} = c_y$ vinden we zo $B_y = \frac{1}{2}c_y E_N$.

De eigenwaarden van E_N worden gegeven door $-2 + 2\cos\frac{s\pi}{N+1}$. Met deze gegevens is eenvoudig in te zien dat voor de matrix $A = (I-B_x)^{-1}(I-B_y)^{-1}(I+B_x)(I+B_y)$ geldt $\|A\| \leq 1$. Uit (2.7.10) volgt

$$\bar{e}(t+\tau) = A\bar{e}(t) + \bar{\psi}, \tag{2.7.13}$$

met $\bar{\psi}$ de vector behorende bij $(I-B_x)^{-1}(I-B_y)^{-1}\psi$ op de maaspunten. Ook $\bar{\psi}$ heeft dus coëfficiënten $O(\tau^3+\tau h^2)$. Dan geldt

$$\|\bar{e}(t+\tau)\|^2 \leq \|A\|\ \|\bar{e}(t)\| + \|\bar{\psi}\|.$$

Dit herhaald toepassend vinden we dan

$$\|\bar{e}(t+\tau)\| = O(\tau^2+h^2).$$

We bekijken nu de LOD-methode. Voor een functie U die voldoet aan (2.7.1) geldt

$$\exp\left(-\tau_k \frac{\partial^2}{\partial x^2}\right)\exp\left(-\tau_k \frac{\partial^2}{\partial y^2}\right)U(t+\tau_k) = U(t) \tag{2.7.14}$$

en als benadering gebruiken we

$$(I-\tau_k D_\xi^2)(I-\tau_k D_\eta^2)u(t+\tau_k) = u(t). \tag{2.7.15}$$

Op punten met reguliere buren is dit equivalent met

(2.7.16) $(I-\tau_k D_\xi^2)u^* = u(t)$,

(2.7.17) $(I-\tau_k D_\eta^2)u(t+\tau_k) = u^*$.

Weer krijgen we tridiagonale stelsels per lijn. Wanneer we de equivalentie tussen (2.7.16), (2.7.17) en (2.7.15) moeten gebruiken voor convergentiebewijzen hebben we dezelfde moeilijkheden als bij ADI. Voor rechthoekige gebieden kunnen we weer

$$(I-\tau_k \delta_\eta^2)f(t+\tau_k) = u^*$$

op randen gebruiken.

In SAMARSKII [1962] wordt een convergentiebewijs gegeven waarbij deze equivalentie niet gebruikt wordt. Dit bewijs wordt voor heel algemene problemen gegeven en is in een groot aantal publikaties nog uitgebreid.

We geven hier het bewijs voor (2.7.1) - (2.7.3) (nu met willekeurige Ω en maas).

We gebruiken in dit bewijs de volgende gelijkheden, analoog aan $2u\frac{du}{dt} = \frac{d(u^2)}{dt}$

$$2u\nabla_{\tau_k} u = \nabla_{\tau_k}(u^2) + \tau_k(\nabla u)^2$$

en analoog aan $2u\frac{d^2u}{dx^2} + 2\left(\frac{du}{dx}\right)^2 = \frac{d^2(u^2)}{dx^2}$ gebruiken we

$$2uD_\xi^2 u = D_\xi^2(u^2) - 2\sigma[(x_i-x_{i-1})(\Delta_\xi u)^2 + (x_{i+1}-x_i)(\nabla_\xi u)^2]$$

met $\sigma = (x_{i+1}-x_{i-1})^{-1}$.

We stellen weer een differentieschema op waaraan de fout $e = u-U$ moet voldoen waarbij $e^* = u^* - U(t+\frac{1}{2}\tau)$. Op $\delta\Omega$ stellen we weer $u(t+\tau_k) = f(t+\tau_k)$ en $u^* = f(t+\frac{1}{2}\tau_k)$. De beginvoorwaarden zijn exact gerepresenteerd zodat e voldoet aan het volgende differentieprobleem.

(2.7.18) $(I-\tau_k D_\xi)^2 e^* = e(t) + \psi^*$,

(2.7.19) $(I-\tau_k D_\eta^2) e(t+\tau_k) = e^* + \psi(t+\tau_k)$,

met

$$\psi^* = (I-\tau_k D_\xi^2) U(t+\tfrac{1}{2}\tau_k) - U(t),$$

$$\psi(t+\tau_k) = (I-\tau_k D_\eta^2) U(t+\tau_k) - U(t+\tfrac{1}{2}\tau_k) ,$$

(2.7.20) $e^* = 0$ op $\delta\Omega$ en

(2.7.21) $e^*(t=0) = 0$ op Ω.

We definiëren nu $\psi_1^* = \psi^* - \psi_2^*$ met

$$\psi_2^* = \left[\tau_k \frac{\partial^2 U}{\partial x^2} (t+\tau) - \tfrac{1}{2} \frac{\partial U}{\partial t} (t+\tau) \right] ,$$

dan geldt

$$\psi_1^* = \tau_k \left\{ \frac{U(t+\frac{1}{2}\tau_k) - U(t)}{\tau_k} - \tfrac{1}{2} \frac{\partial U}{\partial t} (t+\tau_k) \right.$$

$$\left. + \frac{\partial^2 U}{\partial x^2} (t+\tau_k) - D_\xi^2 U(t+\tfrac{1}{2}\tau_k) \right\} = O(\tau^2+\tau h^2).$$

Analoog definiëren we $\psi_1(t+\tau) = \psi(t+\tau) - \psi_2(t+\tau)$ met $\psi_2(t+\tau_k) =$

$= \tau_k \left(\frac{\partial^2 U}{\partial x^2}(t+\tau_k) - \tfrac{1}{2} \frac{\partial U}{\partial t} (t+\tau_k) \right)$ zodat $\psi_1(t+\tau_k) = O(\tau^2+\tau h^2)$.

We definiëren nu $e_1 = e - e_2$ waarbij e_2 voldoet aan

$$\nabla_{\frac{1}{2}\tau_k} e_2^* = \psi_2^* ,$$

$$\nabla_{\frac{1}{2}\tau_k} e_2(t+\tau) = \psi_2(t+\tau_k)$$

en $e_2(t=0) = 0$. Omdat $\psi_2^* + \psi(t+\tau_k) = 0$ geldt dan $e_2^* = 0(\tau)$ en $e_2(t+\tau) = 0$. De functie e_1^* voldoet nu aan

$$2e_1^* \nabla_{\frac{1}{2}\tau_k} e_1^* = 2e_1^* D_\xi^2 e_1^* + 2e_1(\psi_1^* + D_\xi^2 e_2^*),$$

of (stel $\bar{\psi}_1^* = \psi_1^* + D_\xi^2 e_2^*$)

$$2\sigma\{(x_i - x_{i-1})(\Delta_\xi e_1^*)^2 + (x_{i+1} - x_i)(\nabla_\xi e_1^*)^2\} + \nabla_{\frac{1}{2}\tau_k}(e_1^*)^2 +$$

$$+ \tfrac{1}{2}\tau_k (\nabla_{\frac{1}{2}\tau_k} e_1^*)^2 - D_\xi^2 (e_1^*)^2 = 2e_1^* \bar{\psi}_1^* \leq \frac{1}{c_1}(e_1^*)^2 + c_2(\bar{\psi}_1^*)^2.$$

e^{*2} kan op de rand geen maximum hebben, tenzij $e^{*2} = 0$ op $\Omega \cup \delta\Omega$. Stel nu e^{*2} heeft een maximum in een regulier maaspunt, dan geldt daar $D_\xi^2(e_1^{*2}) < 0$. Samen met $(\Delta_\xi e_1^*)^2 \geq 0$ enz. kunnen we concluderen dat

$$\max_{(x_i, y_j)} e_1^{*2} \leq c^* \bar{\psi}_1^{*2} + c^* e(t)^2, \quad c^* > 0.$$

Voor $e(t)^2$ geldt een analoge afschatting, dus, aangezien $e_2(t=0) = 0$, geldt

$$\max_{i,j}(e_1(t_k))^2 \leq c^* \sum_{j=1}^{k} 0(\tau^2 + \tau h^2) = 0(\tau + h^2).$$

2.8. Tabellen met gegevens in TEDDY2

Ten eerste moet opgemerkt worden dat de gebruiker van TEDDY2 niets te maken heeft met deze tabellen en er dus ook niets over hoeft te weten. We geven hier alleen een schets van deze tabellen met enkele voorbeelden om een indruk te geven van de werkwijzen.

Er zijn twee klassen tabellen, één voor de berekeningen met (2.7.4) en één voor de berekeningen met (2.7.5). Voor berekeningen met (2.7.4) volgen de tabellen de x-maaslijnen. We geven een aantal van deze tabellen met een korte inhoudsaanduiding.

x	-	tabellen met x_i waarden
DYREG	-	tabel met $y_i - y_{i-1}$ waarden
IXPCE	-	per x-maaslijn het aantal verschillende gebieden
YB	-	y-waarden van de snijpunten van x maaslijnen met grenzen en randen

COSY	-	cosinuswaarden op randen en grenzen m.b.t. de uitwendige normaalrichting
DYBP	-	"irreguliere" maaswijdten m.b.t. randen
DYBPI	-	"irreguliere" maaswijdten m.b.t. de grenzen
IYIST	-	y-maaslijnen nummer van het eerste reguliere maaspunt per x-maaslijn
		etc.

Voor een zekere x-maaslijn moeten we eerst het kleinste randpunt, dan het eerste reguliere maaspunt en vervolgens de maaswijdte tot het volgende rand- of grenspunt weten, enz.

Voor iedere vergelijking hebben we berekende oplossingen van het vorige tijdstip nodig. Voor de ADI-methode zijn dit er drie. Daarbij hebben we ook gegevens van de maas nodig. Deze gegevens zijn dan nodig, voor (2.7.4) uit de organisatie van (2.7.5) en v.v. Daarom zijn er twee afbeeldingstabellen die de beide organisaties met elkaar verbinden.

De ADI-methoden zijn in de loop der jaren uitgebreid tot problemen met parabolische partiële differentiaalvergelijkingen van aanzienlijke verscheidenheid. Evenwel uitsluitend die bewijzen zijn correct die voor rechthoekige gebieden gelden.

De LOD-methoden zijn onderzocht voor een veel grotere klasse problemen. Ook convergentiebewijzen zijn voor een grotere klasse problemen beschikbaar.

Voor samengestelde gebieden is mij maar één publikatie bekend nl. FRJAZINOV [1975]. Verder is door mij een klasse problemen onderzocht in POLAK [1974] waarbij een niet-realistische grensvoorwaarde werd aangenomen.

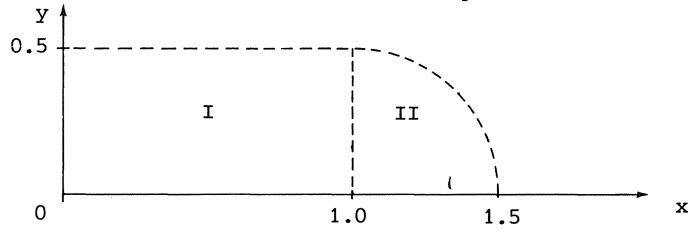
We kunnen nu m.b.t. de problemen die via PARDEL aan TEDDY2 aangeleverd kunnen worden het volgende constateren.
Deze problemen vormen een verzameling waarvoor slechts voor een deel het gebruik van de ADI- of LOD-methode verantwoord kan worden i.v.m. convergentie. We noemen deze deelverzameling de kern, de rest de marge. We moeten evenwel constateren dat problemen die fysici willen analyseren met TEDDY2 vrijwel allemaal uit de marge zijn.

Figuur 2.9.1

```
REGION;
  (0,0) STRAIGHT (0,0.5) STRAIGHT (1,0.5) STRAIGHT (1,0) STRAIGHT;
  U0=FUN1;
  DUDT=0.05*D2UDX2+0.05*D2UDY2;
  BC BETWEEN (0,0) AND (0,0.5) IS 0=DUDN1-FUN2;
  BC BETWEEN (0,0.5) AND (1,0.5) IS 0=DUDN1-FUN3;
  BC BETWEEN (1,0.5) AND (1,0) IS 0=0.2*DUDN1-0.5*DUDN2;
  BC BETWEEN (1,0.5) AND (1,0) IS 0=0.2*DUDN1-FUN6*U2+FUN6*U1;
  BC BETWEEN (1,0) AND (0,0) IS 0=DUDN1;
REGION;
  (1,0) STRAIGHT (1,0.5) CIRCLE (1.0,0)(1.5,0) STRAIGHT;
  U0=FUN4;
  BC BETWEEN (1,0.5) AND (1.5,0) IS 0=U1-FUN5;
  BC BETWEEN (1.5,0) AND (1,0) IS 0=DUDN1;
  DUDT=0.05*D2UDX2+0.05*D2UDY2;
ALGORITHM;
  METHOD=ADI;
  TBEGIN=0;
  TAU=0.025;
  TEND=3;
  X(1)=0,X(31)=1.5;
  Y(1)=0,Y(11)=0.5;
  ACCURACY=DOUBLE PRECISION;
INFORMATION;
  STORE AT TIMES=0.2, 0.6, 1.0, 2.2, 3;
ENDDATA;
```

2.9. Voorbeeld.

Dit voorbeeld bestaat uit twee gebieden waarvan één niet-rechthoekig.



De probleembeschrijving en algoritme-informatie worden in fig. 2.9.1 gegegeven. De bijbehorende functies zijn voor de berekeningen in Fortran gegeven. Hier geven we alleen de essentiële expressies.
De oplossing van het probleem is

$$(2.9.1) \qquad U_i = \exp(-kt)\{B_i \sin((y+0.5)\pi)+C_i \cos((x-0.5)\pi)\}+E_i$$

met i = 1,2, resp. voor gebied I en II.
De constanten hebben de volgende waarden

$$B_1=B_2=8,\ C_1=10,\ C_2=4,\ E_1=20,\ E_2=15$$

en in het volgende

$$\lambda_1=0.2,\ \lambda_2=0.5,\ \rho_1c_1=4,\ \rho_2c_2=10,\ \lambda_1/\rho_1c_1=\lambda_2/\rho_2c_2=0.05,$$

$$k=\pi^2\lambda_i/\rho_i c_i$$

$$\text{FUN1} = U_1(t=0),\ \text{FUN4} = U_2(t=0)$$
$$\text{FUN2} = -C_1\pi \exp(-kt),\ \text{FUN3} = -B_1\pi \exp(-kt)$$
$$\text{FUN6} = C_1\lambda_1\pi \exp(-kt)/(E_1-E_2)$$
$$\text{FUN5} = U_2.$$

De berekende oplossing op een maas met N reguliere punten is u. We beschouwen

$$\|U-u\|_1 = \Sigma\frac{|U-u|}{N},\ \|U-u\|_2 = \left(\Sigma\frac{(U-u)^2}{N}\right)^{\frac{1}{2}},\ \|U-u\|_\infty = \max|U-u|\ ,$$

waarbij de sommatie en het maximum over de reguliere maaspunten wordt genomen. In de volgende tabellen wordt het resultaat van drie berekeningen vergeleken met de analytische oplossing.

run	Δx, Δy	Δt	reguliere maaspunten
I	0.05	0.025	277
II	0.025	0.00625	1112
III	0.0125	0.0015625	4454

De berekeningen werden gedaan op een IBM 370/168 met de volgende tijden en geheugengebruiken.

RUN	PRETED (GO.)TIME	CALTED (GO.)TIME	PRETED CORE	CALTED CORE
I	.98*	9.75	150K	130K
II	1.33	99.98	150K	202K
III	1.27	1314.5	150K	466K

In de volgende tabellen vinden we U-u in de verschillende normen op verschillende tijdstippen t met de berekeningen I, II en III.

	run \ t	0.2	0.6	1.0	2.2	3.0
$\|\cdot\|_1$	I	928E-6	404E-5	523E-5	585E-5	678E-5
	II	218E-6	105E-5	136E-5	152E-5	175E-5
	III	535E-7	266E-6	346E-6	385E-6	444E-6
$\|\cdot\|_2$	I	110E-5	490E-5	611E-5	670E-5	769E-5
	II	260E-6	126E-5	158E-5	173E-5	198E-5
	III	640E-7	320E-6	401E-6	440E-6	503E-6
$\|\cdot\|_\infty$	I	346E-5	984E-5	112E-4	115E-4	116E-4
	II	912E-6	251E-5	287E-5	296E-5	299E-5
	III	243E-6	633E-6	724E-6	747E-6	757E-6

$\|U(t=0)\|_1 = 27.814$, $\|U(t=0)\|_2 = 28.634$, $\|U(t=0)\|_\infty = 37.996$.

LITERATUUR

BARNEVELD BINKHUYSEN, C.F.E., S.J. POLAK, J.G. Schrooten & A.J.H. WACHTERS [1975], *TEDDY2 examples* (preliminary issue), Philips-ISA-UDV-DSA-SCA/BB/SP/JS/AW/075/027/jf.

CARDENAS, A.F. & W.J. KARPLUS, [1968], PDEL - *A language for partial differential equations*, Communications of the ACM 13, 184-191.

CARVER, M.B., [1973], *A Fortran-oriented simulation system for the general solution of Partial differential equations*, Proceedings of the 1973 summer computer simulation conference, Montreal.

CSENDES, Z.J., [1975], DECL - *A computer language for the solution of arbitrary partial differential equations,* Advances in comp. methods for partial differential equations. AICA.

D'JAKONOV, E.G., [1962], *Method of nets for the solution of parabolic equations of order* 2m *with separable variables* (Russisch), Dokl. Akad. Nauk SSSR 142, 1236-1238.

FAIRWEATHER, G. & A.R. MITCHELL, [1967], *A new computational procedure for A.D.I. methods,* Siam J. Numer. Anal. 4, 163-170.

FRJAZINOV, I.V., [1975], *A certain class of schemes for equations of parabolic type* (Russisch), Z. Vychisl. Mat. i Mat. Fiz. 15, 113-125.

GARY, I. & R. HELGASON, [1972], *An extension of Fortran containing finite difference operators, Software - practice and experience* 2, 321-326.

HELGASON, R & I. GARY, [1975], PDELAN *users manual version I,* National center for atmospheric research, Boulder, Colorado.

ISAACSON, E. & H.B. KELLER, [1966], *Analysis of numerical methods,* John Wiley and Sons.

NILSEN, R.N. & W.J. KARPLUS, [1974], *Continuous-system simulation languages: a state of the art survey,* Annales de l'association internationale pour le calcul analogique 1.

MITCHELL, A.R., [1969], *Computational methods in partial differential equations,* John Wiley and Sons.

POLAK, S.J. & J. SCHROOTEN, [1975], *Preliminary TEDDY2 user manual,* Philips-ISA-UDV-DSA-SCA/SP/75/024/mw.

POLAK, S.J., [1974], *Convergence of an* L.O.D.*-method for a composite region problem,* Philips I.S.C.A.

SAMARSKII, A.A., [1962], *An efficient difference method for solving a multidimensional parabolic equation in an arbitrary domain* (Russisch), Z. Vychisl. mat. i Mat. Fiz. 2.

SINCOVEC, R.F. & N.K. MADSEN, [1975], *Software for nonlinear partial differential equations,* ACM. Transactions on Mathematical software 2, 232-260.

YANENKO, M.M., [1971], *The method of fractional steps,* Springer.

3. BEGIN-RANDWAARDEPROBLEMEN VOOR PARTIËLE DIFFERENTIAALVERGELIJKINGEN

3.1. Inleiding

In sectie 2.2 van het eerste deel van het colloquium Numerieke Programmatuur is reeds opgemerkt dat het numeriek oplossen van partiële differentiaalvergelijkingen een zeer gevarieerd en gecompliceerd onderwerp is. Zoals bekend mag worden verondersteld bestaan er zeer veel ad hoc methoden die slechts gebruikt kunnen worden voor speciale klassen van problemen. Zulke ad hoc methoden zijn in het algemeen zeer efficiënt wat betreft rekentijd, maar bijzonder inefficiënt wat betreft programmeertijd. Deze inefficiëntie doet zich vooral gelden indien men bestaande programmatuur aan wil passen bij een andere probleemklasse.

Een methode die leidt tot een meer uniforme aanpak bij het numeriek oplossen van beginwaardeproblemen voor partiële vergelijkingen is de methode der *semi-discretisatie,* welke ook de methode der lijnen genoemd wordt. Deze methode is in de boven vermelde sectie geïllustreerd aan de hand van een drietal concrete praktijk-problemen, en komt er op neer dat de partiële differentiaalvergelijking herleid wordt tot een stelsel gewone differentiaalvergelijkingen door de plaatsvariabelen in de partiële differentiaaloperator te discretiseren. Bij semi-discretisatie wordt de tijdvariabele dus continu gelaten. Dit betekent dat dezelfde semi-discretisatie net zo makkelijk toegepast kan worden bij hogere orde vergelijkingen in de tijd, als bij eerste orde vergelijkingen in de tijd.

Semi-discretisatie kan men op twee manieren realiseren, namelijk via de methode der *eindige differenties* en via de methode der *eindige elementen.* In sectie 2.2 van het eerste deel van het colloquium is de methode der eindige differenties toegepast. Daar is ook verwezen naar SINCOVEC & MADSEN [1975], die een programma gepubliceerd hebben dat voor een vrij algemene

partiële differentiaal-operator, via eindige differenties, de semi-discretisatie uitvoert. Zo'n programma noemt men een *interface*. In dit hoofdstuk wordt niet ingegaan op de interface van Sincovec en Madsen. Daarentegen zullen we semi-discretisatie met behulp van eindige elementen bespreken. Een interface, gebaseerd op eindige elementen, is in ontwikkeling op het Mathematisch Centrum.

Na het uitvoeren van semi-discretisatie, hetzij via eindige differenties, hetzij via eindige elementen, resulteert er een stelsel gewone differentiaalvergelijkingen. Dit stelsel kan een eerste orde stelsel of een hogere orde stelsel zijn. In beide gevallen geldt dat dit resulterende stelsel geïntegreerd kan worden met behulp van een integratiemethode voor een gewone differentiaalvergelijking. We zullen in dit hoofdstuk deze *tijdsintegratie* behandelen voor drie klassen van problemen. Namelijk, eerste orde parabolische problemen, en eerste en tweede orde hyperbolische problemen. Voor alle klassen van problemen beperken wij ons tot integratietechnieken, welke op het Mathematisch Centrum ontwikkeld, of nog in ontwikkeling zijn. De toepasbaarheid van deze technieken zal geïllustreerd worden aan de hand van numerieke voorbeelden.

3.2. Semi-discretisering door middel van de methode der eindige elementen

3.2.1. Inleiding

Hoewel er uiterlijk veel overeenkomsten bestaan tussen de methode der lijnen en de methode der eindige elementen, zeker als men afgaat op beider resultaten, liggen aan de methoden verschillende theorieën ten grondslag. De methode der lijnen gaat ervan uit dat de benaderende oplossing slechts op een aantal discrete lijnen is gedefinieerd. De partiële afgeleiden naar de ruimtevariabele x worden benaderd door middel van differentiequotiënten, bv.

$$u_x(x,t) \approx \frac{u(x+h,t)-u(x-h,t)}{2h} \quad ;$$

$$u_{xx}(x,t) \approx \frac{u(x+h,t)-2u(x,t)+u(x-h,t)}{h^2} \quad .$$

De methode der eindige elementen gaat uit van een benaderende oplossing die continu is in zowel x als t. Deze approximatie is van de vorm

$$(3.2.1) \qquad U(x,t) = \sum_{j=1}^{M} a_j(t)\, Q_j(x),$$

waarbij $Q_1(x),\ldots,Q_M(x)$ geschikt gekozen funkties zijn. Het is evident dat U vast ligt op het tijdstip t, als $(a_1,\ldots,a_M)$ bekend is. De eindige elementenmethode is er nu op gericht een begin-randwaardeprobleem van de vorm

$$(3.2.2) \qquad L(t, \frac{\partial}{\partial t}, \frac{\partial^2}{\partial t^2},\ldots)u_i = \rho_i(x,t,\vec{u})\frac{d}{dx}F_i(x,t,\vec{u},\frac{\partial\vec{u}}{\partial x}) - G_i(x,t,\vec{u},\frac{\partial\vec{u}}{\partial x}),$$

$$i = 1,\ldots,NPDE,$$

om te zetten in een zuiver beginwaardeprobleem van de vorm

$$(3.2.3) \qquad L(t,\frac{d}{dt}, \frac{d^2}{dt^2},\ldots)a_{ij} = H_{ij}(t,a_{11},\ldots,a_{1NPDE},\ldots,a_{M1},\ldots,a_{MNPDE}),$$

$$i = 1,\ldots,M,\ j = 1,\ldots,NPDE,$$

waarbij L een (partiële) differentiaaloperator in t aanduidt. Is (3.2.3) opgelost, dan kan in principe de oplossing van (3.2.2) in ieder punt op het x-interval worden benaderd m.b.v. formule (3.2.1).

In de subroutine PDEFEM wordt de semi-discretisering van het rechterlid van (3.2.2) gerealiseerd. In de §§ 3.2.2 t/m 3.2.5 zal ingegaan worden op de theoretische achtergronden van PDEFEM, terwijl in 3.2.6 de subroutine zelf beknopt zal worden besproken.

3.2.2. Galerkin's methode

We illustreren de in de subroutine PDEFEM gebezigde methode aan de hand van het modelprobleem

$$(3.2.4) \qquad \begin{aligned} &u_t = [p(x)u_x]_x - q(x)u,\ x \in [0,1] = I,\ t \geq 0,\\ &u(0,t) = 0;\ u_x(1,t) = u(1,t);\\ &u(x,0) = u_0(x). \end{aligned}$$

In linkereindpunt x = 0 heeft u een essentiële of Dirichlet-randvoorwaarde, in het rechtereindpunt x = 1 een natuurlijke randvoorwaarde.

Uit de klassieke theorie van de begin-randwaardeproblemen is bekend dat de oplossing van de vorm

$$(3.2.5) \qquad u(x,t) = \sum_{j=1}^{\infty} a_j(t)\varphi_j(x)$$

is. Hierbij zijn $\varphi_1,\ldots,\varphi_n,\ldots$ funkties die alle aan de randvoorwaarden $\varphi_j(0) = 0$ en $\varphi_j'(1) = \varphi_j(1)$ voldoen. Nemen we nu het inprodukt van het linker- en rechterlid van (3.2.4) met φ_i $(i=1,\ldots)$, dan krijgen we na partiële integratie

$$(u_t,\varphi_i) + (pu_x,\varphi_i') + (qu,\varphi_i) = u(1,t)\varphi_i(1)p(1), \qquad i = 1,\ldots .$$

Passen we dit toe op (3.2.5), dan krijgen we voor $a_1,\ldots,a_n,\ldots$ de *gewone* differentiaalvergelijking

$$(3.2.6a) \qquad \sum_{j=1}^{\infty} \left[(\varphi_j,\varphi_i)\frac{da_j}{dt} + \{(p\varphi_j',\varphi_i') + (q\varphi_j,\varphi_i) - p(1)\varphi_j(1)\varphi_i(1)\}a_j\right] = 0, \quad i = 1,\ldots .$$

De beginvoorwaarden van (3.2.6) worden dan gegeven door de vergelijking

$$(3.2.6b) \qquad \sum_{j=1}^{\infty} a_j(0)\varphi_j(x) = u_0(x).$$

De klassieke Galerkin-methode beslaat hierin dat de reeksen (3.2.5) en (3.2.6) worden afgekapt bij i = n waardoor een vectorbeginwaardeprobleem ontstaat van de vorm

$$(3.2.7) \qquad \begin{aligned} &M\frac{d}{dt}\vec{a} + K\vec{a} = (a_0\varphi_i(1)); \quad \vec{a}(0) = \vec{a}_0; \\ &m_{ij} = (\varphi_i,\varphi_j); \quad k_{ij} = B(\varphi_i,\varphi_j) = (p\varphi_i',\varphi_j') + (q\varphi_i,\varphi_j) - p(1)\varphi_i(1)\varphi_j(1), \\ &\qquad\qquad i,j = 1,\ldots,n. \end{aligned}$$

Aan deze methode kleeft het nadeel dat (3.2.7) in het algemeen slecht geconditioneerd is. De eindige elementenmethode geeft voor dit manko de volgende oplossing. De funkties $\varphi_1,\varphi_2,\ldots$ spannen een bepaalde ruimte V op. De klassieke aanpak komt erop neer dat van deze V een n-dimensionale deelruimte V_n wordt genomen, eenvoudigweg door de basis af te breken bij n + 1. De methode der eindige elementen komt erop neer dat van V een eindig-dimensionale deelruimte wordt genomen die *niet* door een deel van $\{\varphi_i\}_{i=1}^{\infty}$ wordt opgespannen. In de volgende sectie geven we een beschrijving van zo'n ruimte.

3.2.3. Continue stuksgewijs kwadratische funkties

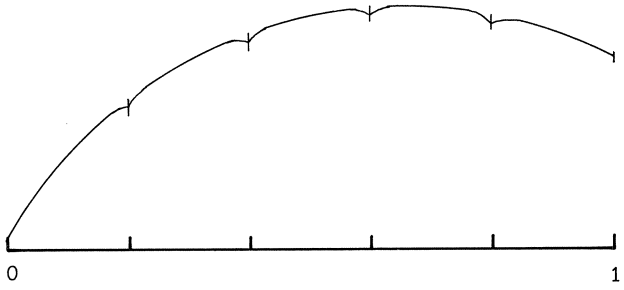
We verdelen [0,1] in N segmentjes van gelijke lengte h = 1/N. We geven

de verdeling aan met

(3.2.8) $\Delta = \{0 = x_0 < x_1 < \ldots < x_j = jh < \ldots < x_N = 1\}$.

De segmenten geven we aan met $I_j = [x_{j-1}, x_j]$; we noemen h de maaswijdte van Δ. We definiëren nu $S_2(\Delta)$ als de ruimte van funkties die

1^o continu zijn op I;
2^o op ieder segment I_j een polynoom van de graad ≤ 2 zijn;
3^o in 0 de waarde 0 hebben.



Figuur 3.2.1. Een element uit $S_2(\Delta)$.

N.B. Het zal de lezer zijn opgevallen dat de rechterrandvoorwaarde $v'(1) = v(1)$ *niet* voorkomt in de definitie van $S_2(\Delta)$. We kunnen niet ingaan op de theoretische achtergronden van de verdwijning van deze randvoorwaarde, maar verwijzen hiervoor naar STRANG & FIX [1973, hoofdstuk 1].

3.2.3.1. Basis van $S_2(\Delta)$

Een van de eigenschappen van de ruimte $S_2(\Delta)$ is dat we de basisfunkties zo kunnen kiezen dat ieder van hen op ten hoogste twee segmentjes van Δ niet identiek 0 is. We definiëren de punten $x_{j+\frac{1}{2}}$, $j = 0,\ldots,N-1$, door

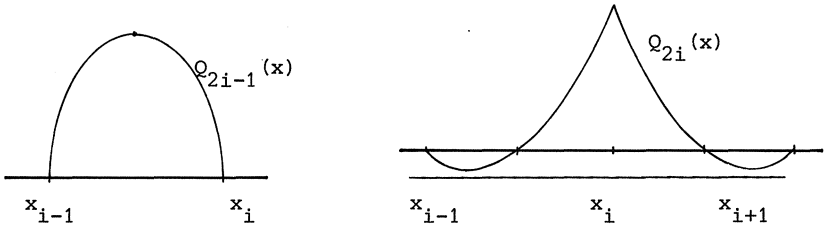
(3.2.9) $x_{j+\frac{1}{2}} = \frac{1}{2}(x_j + x_{j+1}) = (j+\frac{1}{2})h$.

We construeren nu de basis van $S_2(\Delta)$ als volgt: Iedere $Q \in S_2(\Delta)$ is geheel bepaald door de funktiewaarden in de punten $x_{\frac{1}{2}}, x_1, \ldots, x_{N-\frac{1}{2}}, x_N$. De basis bestaat derhalve uit 2N elementen. We definiëren de basisfunkties nu als volgt

$$Q_{2i-1}(x) = \begin{cases} 0 & , \quad \text{als } x \notin I_i, \\ \dfrac{4(x-x_{i-1})(x_i-x)}{h^2} & , \quad \text{als } x \in I_i, \\ & i = 1,\ldots,N; \end{cases}$$

(3.2.10)
$$Q_{2i}(x) = \begin{cases} 0 & , \quad x \notin I_i,\ x \notin I_{i+1}, \\ \dfrac{2(x-x_{i-\frac{1}{2}})(x-x_{i-1})}{h^2} & , \quad x \in I_i, \\ \dfrac{2(x-x_{i+\frac{1}{2}})(x-x_{i+1})}{h^2} & , \quad x \in I_{i+1}, \\ & i = 1,\ldots,N-1; \end{cases}$$

$$Q_{2N}(x) = \begin{cases} 0 & , \quad x \notin I_N, \\ \dfrac{2(x-x_{N-\frac{1}{2}})(x-x_{N-1})}{h^2} & , \quad x \in I_N. \end{cases}$$

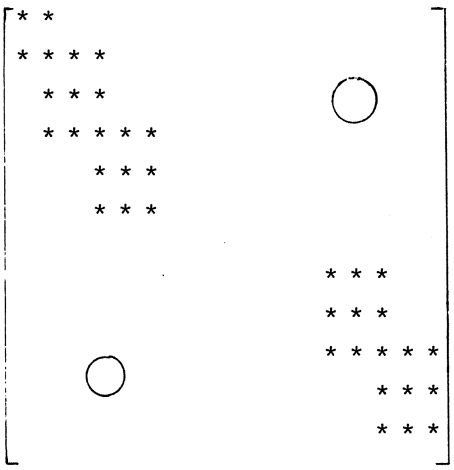


Figuur 3.2.2. Grafiek van Q_{2i-1} en Q_{2i}

Zie verder BAKKER e.a. [1976, hoofdstuk 3] voor een verdere beschrijving van $S_2(\Delta)$.

3.2.4. IJlheid van de Jacobiaan

Een van de voordelen van $S_2(\Delta)$ is dat de matrices $B(Q_i,Q_j)$ en (Q_i,Q_j) ijl zijn. Deze ijlheid is een regelrecht gevolg van het feit dat de basisfunkties op slechts twee segmenten van Δ niet identiek 0 zijn. Ze hebben derhalve de struktuur



Een ander voordeel van $S_2(\Delta)$ is dat $B(Q_i,Q_j)$ en (Q_i,Q_j) segmentsgewijs opgebouwd kunnen worden (zie STRANG & FIX [1973, hoofdstuk 1]).

3.2.5. Numerieke kwadratuur

In het algemeen kan de matrix $B(Q_i,Q_j)$ niet exakt geëvalueerd worden maar moet er een kwadratuurregel worden toegepast. We kiezen hiervoor de *geregen* regel van Simpson voor de approximatie van het inprodukt (α,β):

$$(\alpha,\beta) \simeq (\alpha,\beta)^* = \sum_{\ell=1}^{N} (\alpha,\beta)_\ell^* \ ;$$

(3.2.11)

$$(\alpha,\beta)_\ell^* = \frac{h}{6}\left[\alpha(x_{\ell-1})\beta(x_{\ell-1})+\alpha(x_\ell)\beta(x_\ell)\right]$$

$$+ \frac{2}{3} h\alpha(x_{\ell-\frac{1}{2}})\beta(x_{\ell-\frac{1}{2}}), \quad \ell = 1,\ldots,N;$$

De toepassing van deze kwadratuurregel heeft drie voordelen:

1° De discretiseringsfout blijft van dezelfde grootte orde, nl. h^3 (zie bv. RAVIART [1973]);

2° De ijlheid van de Jacobiaan blijft gehandhaafd;

3° Indien (Q_i,Q_j) eveneens wordt benaderd door de geregen regel van Simpson, is het resultaat een diagonaalmatrix, zodat een *expliciet* beginwaardeprobleem wordt verkregen. Men kan dus, *als men wil*, het beginwaardeprobleem integreren m.b.v. expliciete methoden.

Definiëren we nu

$$(3.2.12)\qquad B^*(Q_i,Q_j) = \sum_{\ell=1}^{N} [(pQ_i',Q_j')^*_\ell] - p(1)\delta_{i2N}\delta_{j2N},$$

dan wordt het beginwaardeprobleem (3.2.6) omgezet in

$$(3.2.13)\qquad \lambda_i \frac{da_i}{dt} + \sum_{j=1}^{2N} B^*(Q_j,Q_i)a_j = 0,$$

$$a_i = u_0(\tfrac{1}{2}hi),\ t = 0; \qquad i = 1,\ldots,2N,$$

waarbij $\lambda_i = (Q_i,Q_i)^*$ gedefinieerd is door

$$(3.2.14)\qquad \lambda_{2\ell} = \frac{1}{3}h;$$

$$\lambda_{2\ell-1} = \frac{2}{3}h.$$

Voorbeeld

De partiële differentiaalvergelijking

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \qquad x \in [0,1], \quad t \geq 0,$$

$$u(0,t) = 0, \qquad u_x(1,t) = u(1,t),$$

$$u(x,0) = x,$$

gaat, na semi-discretisering, over in

$$\frac{d}{dt}a_1 = \frac{4}{h^2}(-2a_1+a_2);$$

$$\frac{d}{dt}a_{2i-1} = \frac{4}{h^2}(a_{2i-2}-2a_{2i-1}+a_{2i}), \qquad i = 2,\ldots,N;$$

$$\frac{d}{dt}a_{2i} = -\frac{1}{h^2}(14a_{2i}-8(a_{2i-1}+a_{2i+1})+a_{2i-2}+a_{2i+2}), \qquad i = 1,\ldots,N-1;$$

$$\frac{d}{dt}a_{2N} = -\frac{2}{h^2}(7a_{2N}-8a_{2N-1}+a_{2N-2})+\frac{6a_{2N}}{h}, \qquad t \geq 0,$$

en

$$a_j = \frac{jh}{2}, \qquad t = 0.$$

3.2.6 De subroutine PDEFEM

PDEFEM is een subroutine die het beginwaardeprobleem (3.2.2) met randvoorwaarden

$$\alpha_{i\ell}u_i(a,t) + \beta_{i\ell}\frac{\partial u_i}{\partial x}(a,t) = \lambda_{i\ell}(t,\vec{u}(a,t)),$$

$$\alpha_{ir}u_i(b,t) + \beta_{ir}\frac{\partial u_i}{\partial x}(b,t) = \lambda_{ir}(t,u(b,t)), \qquad (3.2.15)$$

en beginvoorwaarde

$$u_i(x,0) = v_i(x),$$

omzet in een beginwaardeprobleem van de vorm

$$\frac{d}{dt}a_{ij} = H_{ij}(t,a_{11},\ldots,a_{1NPDE},\ldots,a_{NX1},\ldots,a_{NXNPDE}),$$

$$a_{ij}(0) = v_i(x_j), \qquad (3.2.16)$$

$$i = 1,\ldots,NX,$$

$$j = 1,\ldots,NPDE,$$

waarbij x_j door de gebruiker op te geven punten op het interval [a,b] zijn.
Men dient ervoor te zorgen dat NX *oneven* is.

We illustreren het gebruik van PDEFEM aan de hand van een uit de industrie afkomstig probleem (zie paragraaf 3.1.5 van het eerste deel van dit colloquium):

$$\frac{\partial u}{\partial t} = \varepsilon p\frac{\partial^2 u}{\partial x^2} - g(u-v);$$

$$\frac{\partial v}{\partial t} = p\frac{\partial^2 v}{\partial x^2} + g(u-v);$$

$$\frac{\partial u}{\partial x} = v = 0, \qquad x = 0; \qquad (3.2.17)$$

$$u = 1,\ \frac{\partial v}{\partial x} = 0, \qquad x = 1;$$

$$u(x,0) = 1, \qquad v(x,0) = 0;$$

$$g(z) = \exp(\tfrac{1}{3}\eta z) - \exp(-\tfrac{2}{3}\eta z);$$

$$\varepsilon = 0.143;\ p = 0.1743;\ \eta = 17.19.$$

Bij gegeven punten $x_1 = 0$, $x_2 = h, \ldots, x_{NX} = (NX-1)h$; zet de subroutine PDEFEM (3.2.17) om in het volgende beginwaardeprobleem

$$\frac{d}{dt} u_{2i} = -\frac{\varepsilon p}{h^2}[2u_{2i} - u_{2i-1} - u_{2i+1}] - g(u_{2i} - v_{2i}), \quad i = 1, \ldots, \frac{NX-1}{2};$$

$$\frac{d}{dt} u_{2i+1} = \frac{-\varepsilon p}{4h^2}[14u_{2i+1} - 8(u_{2i+2} + u_{2i}) + u_{2i+3} + u_{2i-1}]$$

$$- g(u_{2i+1} - v_{2i+1}), \quad i = 1, \ldots, \frac{NX-3}{2};$$

$$\frac{du_1}{dt} = \frac{-2\varepsilon p}{h^2}[7u_1 - 8u_2 + u_3] - g(u_1 - v_1)$$

$$\frac{du_{NX}}{dt} = 0;$$

(3.2.18)

$$\frac{dv_1}{dt} = 0;$$

$$\frac{dv_{2i}}{dt} = \frac{-p}{h^2}[2v_{2i} - v_{2i-1} - v_{2i+1}] - g(u_{2i} - v_{2i}); \quad i = 1, \ldots, \frac{NX-1}{2};$$

$$\frac{dv_{2i+1}}{dt} = \frac{-p}{uh^2}[14v_{2i+1} - 8(v_{2i} + v_{2i+2}) + v_{2i+3} + v_{2i-1}]$$

$$+ 9(u_{2i+1} - v_{2i+1}), \quad i = 1, \ldots, \frac{NX-3}{2};$$

$$\frac{dv_{NX}}{dt} = \frac{-2p}{h^2}[7v_{NX} - 8v_{NX-1} + v_{NX-2}] + g(u_{NX} - v_{NX}).$$

$$u_i = 1, \qquad v_i = 0, \qquad t = 0, \qquad i = 1, \ldots, NX.$$

Deze omzetting wordt gerealiseerd door de aanroep

CALL PDEFEM (X,T,A,NX,NPDE,F,G,RHO,BLEFT,BRIGHT),

waarbij de parameters de volgende betekenis hebben

X(NX): de partitie van [0,1];
X(i) = (i-1)/(NX-1.), i = 1,...,NX

T : de tijdsvariabele

A(NX,NPDE) :
IN : A(i,1) is een approximatie van u(x(i),t)
A(i,2) is een approximatie van v(x(i),t)
UIT: het rechterlid van (3.2.18), aangeschreven op A(NX,2)

F: de funktie corresponderend met F_i uit (3.2.2); F is van de vorm

```
FUNCTION F (I,X,T,U,UX)
DIMENSION U(2),UX(2)
F = 0.1743*UX(I)
RETURN
END
```

G: de funktie corresponderend met G_i uit (3.2.2); G heeft hier de vorm

```
FUNCTION G(I,X,T,U,UX)
DIMENSION U(2),UX(2)
Z = U(1)-U(2)
G = (EXP(5.73*Z)-EXP(-11.46*Z))*(-1)**(I-1)
RETURN
END
```

RHO: de funktie corresponderend met ρ_i uit (3.2.2); RHO is van de vorm

```
FUNCTION RHO (I,X,T,U)
DIMENSION U(2)
RHO = 1.0
IF (I.EQ.1) RHO = 0.143
RETURN
END
```

BLEFT: de subroutine die de randvoorwaarden in x = 0 verwerkt in de parameters α_ℓ, β_ℓ en γ_ℓ uit (3.2.15):

```
SUBROUTINE BLEFT (T,ALFA,BETA,GAMMA,U)
DIMENSION ALFA(2), BETA(2), GAMMA(2), U(2)
ALFA(2) = 1.0 $ BETA(2) = GAMMA(2) = 0.0
```

```
      ALFA(1) = GAMMA(1) = 0.0 $ BETA(1) = 1.0
      RETURN
      END
```

BRIGHT: de subroutine die de rechterrandvoorwaarden verwerkt in de parameters α_r, β_r en γ_r:

```
      SUBROUTINE BRIGHT(T,ALFA,BETA,GAMMA,U)
      DIMENSION ALFA(2), BETA(2), GAMMA(2),U(2)
      ALFA(1) = GAMMA(1) = 1.0 $ BETA(1) = 0.0
      ALFA(2) = GAMMA(2) = 0.0 $ BETA(2) = 1.0
      RETURN
      END
```

In de volgende paragraaf zal de tijdsintegratie van (3.2.17) worden besproken.

3.3. Een klasse van gestabiliseerde driestaps-Runge-Kuttamethoden voor de tijdsintegratie van parabolische vergelijkingen

We behandelen in deze sectie de *tijdsintegratie van parabolische differentiaalvergelijkingen* nadat deze door middel van de *methode der lijnen* tot stelsels gewone differentiaalvergelijkingen zijn herleid. De numeriek op te lossen stelsels gewone differentiaalvergelijkingen veronderstellen we van de autonome vorm

$$y'(x) = f(y(x)), \tag{3.3.1}$$

waarbij we opmerken dat x nu in feite de tijdsvariabele voorstelt (vgl. (3.2.3)). Een belangrijke eigenschap van stelsels gewone differentiaalvergelijkingen, welke ontstaan zijn uit semi-discretisatie van parabolische vergelijkingen, is dat de *eigenwaarden van de Jacobiaan* van de vectorfunctie meestal gelegen zijn in *een lange, smalle strook langs de negatieve as* van het complexe vlak (zie bv. RICHTMEYER & MORTON [1967]). In deze paragraaf wordt er van uitgegaan dat (3.3.1) deze eigenschap bezit.

De integratietechniek die we hier bespreken berust op *expliciete gestabiliseerde formules*. Een voordeel van het expliciet zijn van de formules is dat, in principe, een zeer grote klasse van parabolische vergelijkingen indirect op *uniforme wijze* numeriek opgelost kan worden. De enige beperking die we opleggen is dat de eigenwaarden van de matrix der partiële afgeleiden reëel of bijna reëel zijn. In de praktijk doet deze beperking zich nauwelijks gelden. Een beperking in het toepassen van expliciete formules, die zich in de praktijk wel kan doen gelden, ontstaat indien de spectraalradius van de Jacobiaan uitzonderlijk groot is. In dit geval moeten we, ondanks het gestabiliseerde karakter, zeer kleine integratiestappen nemen om stabiliteit te garanderen. Deze beperking doet zich uiteraard vooral gelden indien het integratie-interval groot is. Het is dan aan te bevelen om (3.3.1) met een *impliciete* techniek op te lossen, welke voor willekeurig grote staplengten stabiel is. Aan het gebruik van een impliciete techniek zijn echter ook nadelen verbonden, met name bij het oplossen van meer-dimensionale problemen. (zie bv. RICHTMEYER & MORTON [1967]).

3.3.1. De integratiemethode

De integratiemethode welke we behandelen is gedefinieerd door de volgende formule:

$$y_{n+1}^{(0)} = y_n,$$

$$y_{n+1}^{(1)} = (1-b_1)y_n+b_1y_{n-1}+c_1hf(y_{n-1})+\lambda_{1,0}hf(y_n),$$

$$y_{n+1}^{(j)} = (1-b_j)y_n+b_jy_{n-1}+c_jhf(y_{n-1})+\lambda_{j,0}hf(y_n)+\lambda_{j,j-1}hf(y_{n+1}^{(j-1)}), \qquad (3.3.2)$$

$$j = 2,\ldots,m, \quad m \geq 2,$$

$$y_{n+1} = dy_{n+1}^{(m)}+(1-d)y_{n-2}, \quad n \geq 2.$$

Hierin is de vector y_n altijd een numerieke benadering voor de analytische oplossing $y(x)$, in $x = x_n$, van stelsel (3.3.1). De punten x_j, $j = n-2,\ldots,n+1$ zijn de referentiepunten van de driestaps-formule en h geeft de staplengte aan, d.w.z. $h = x_{n+1}-x_n$. Tenzij anders vermeld, wordt h constant beschouwd. Voor het toepassen van (3.3.2) zijn drie startvectoren nodig, namelijk y_0, y_1 en y_2. Voor de startvector y_0 kiezen we steeds de gegeven beginwaardevector van (3.3.1). De startvectoren y_1 en y_2 worden berekend met een eenstapsformule.

Methode (3.3.2) is een *driestapsmethode* welke behoort tot de zeer grote klasse van de *meerstaps-Runge-Kuttamethoden*. Deze methoden zijn voor het eerst besproken door GEAR [1964], die van *hybride methoden* spreekt. Hybridisch omdat het lineaire meerstapsidee gecombineerd is met het niet-lineaire eenstapsidee. Een theoretische analyse van meerstaps-Runge-Kutta-methoden is gegeven door WATT [1967], terwijl in VAN DER HOUWEN [1977] een aantal praktische toepassingen worden besproken. Wij zullen schema (3.3.2) ook een *driestaps-schema van de graad* m noemen, waarbij m dan het aantal functie-evaluaties per integratiestap aangeeft.

Indien $d = 1$, $b_j = 0$ en $c_j = 0$ voor $j = 1,\ldots,m$ en $\lambda_{j,0} = 0$ voor $j = 2,\ldots,m$, gaat (3.3.2) over in een eenstapsformule. *Gestabiliseerde eenstapsformules* van dit type zijn beschreven in VAN DER HOUWEN [1977]. Deze eenstapsformules zullen wij gebruiken om de startvectoren y_1 en y_2 te berekenen.

3.3.1.1. Voorwaarden voor convergentie en consistentie

Aangezien we hier een praktische toepassing van methode (3.3.2) behandelen, zullen we weinig of geen aandacht schenken aan zuiver theoretische aspecten van meerstaps-Runge-Kuttamethoden. Hiervoor verwijzen we naar WATT [1967], waar ook een convergentie-stelling bewezen wordt. Een goede inleiding in de algemene theorie van numerieke integratie kan men vinden in LAMBERT [1973], waarnaar we ook verwijzen voor de strenge definities van convergentie, consistentie en stabiliteit.

We zullen eerst de consistentievoorwaarden geven voor (3.3.2). Daartoe is het handig om met ons driestapsschema de niet-lineaire operator $E[y_n, y_{n-1}, y_{n-2}]$ te associëren, d.w.z. we schrijven

$$y_{n+1} = E[y_n, y_{n-1}, y_{n-2}].$$

Veronderstel nu even dat y(x) een voldoende vaak differentieerbare functie is. Hierdoor is het mogelijk om een Taylorontwikkeling voor $E[y(x_n), y(x_{n-1}), y(x_{n-2})]$ om $x = x_n$ op te stellen. Drukken we vervolgens de hogere afgeleiden van y uit in termen van f en zijn hogere afgeleiden, en maken we gebruik van de tensornotatie in de ontwikkeling van Taylor voor functies van meer variabelen (zie HENRICI [1962], p.118), dan vinden we

$$(3.3.3)\qquad y(x_{n+1}) - E[y(x_n), y(x_{n-1}), y(x_{n-2})] =$$

$$C_1 h f + C_2 h^2 f_j f^j + C_{31} h^3 f_j f^j_k f^k + C_{32} h^3 f_{jk} f^j f^k + O(h^4),$$

waarbij

$$C_1 = 1 - \{d(-b_m + c_m + \lambda_{m,0} + \lambda_{m,m-1}) - 2(1-d)\},$$

$$(3.3.4)\qquad C_2 = \tfrac{1}{2} - \{d(\tfrac{1}{2}b_m - c_m + \lambda_{m,m-1}(-b_{m-1} + c_{m-1} + \lambda_{m-1,0} + \lambda_{m-1,m-2})) + 2(1-d)\},$$

$$C_{31} = \frac{1}{6} - \{d(-\frac{1}{6}b_m + \tfrac{1}{2}c_m + \tfrac{1}{2}\lambda_{m,m-1}(b_{m-1} - 2c_{m-1} + 2\lambda_{m-1,m-2}(-b_{m-2} + c_{m-2} + \lambda_{m-2,0} + \lambda_{m-2,m-3}))) - \frac{8}{6}(1-d)\},$$

$$C_{32} = \frac{1}{6} - \{d(-\frac{1}{6}b_m + \tfrac{1}{2}c_m + \tfrac{1}{2}\lambda_{m,m-1}(-b_{m-1} + c_{m-1} + \lambda_{m-1,0} + \lambda_{m-1,m-2})^2) - \frac{8}{6}(1-d)\}.$$

Schema (3.3.2) heet *consistent* (*nauwkeurig*) van de orde 1 indien $C_1 = 0$, consistent van de orde 2 als bovendien $C_2 = 0$, en consistent van de orde 3 als bovendien $C_{31} = C_{32} = 0$.

Zij y(x) nu weer een oplossing van de differentiaalvergelijking, dan is

$$(3.3.5) \qquad y(x_{n+1})-E[y(x_n),y(x_{n-1}),y(x_{n-2})]$$

de *lokale afbreekfout*. Deze fout is lokaal in de zin dat hier de "lokaliserende" aanname is gemaakt dat de benaderingen y_n, y_{n-1} en y_{n-2} alle op de oplossing y(x) liggen (zie LAMBERT [1973] voor een meer uitgebreide behandeling hiervan). Wij zullen de lokale afbreekfout gebruiken voor een stapkeuzemechanisme.

Consistentievoorwaarden voor orde groter dan drie leiden we niet af. We zullen ons zelfs verder beperken tot het behandelen van schema's van de orde p = 1 en p = 2. Voor zeer veel praktische toepassingen bij partiële differentiaalvergelijkingen is zo'n lage orde voldoende. De consistentievoorwaarden voor orde p = 3 zullen later gebruikt worden om een schatting van de lokale afbreekfout voor tweede orde formules mogelijk te maken.

Een noodzakelijke voorwaarde voor *convergentie* van meerstaps-Runge-Kuttaformules is de voorwaarde van *nulstabiliteit,* zoals dit het geval is bij lineaire meerstapsformules (zie LAMBERT [1973]). In feite geldt hier ook de bekende *convergentiestelling*: *de methode is convergent d.e.s.d.a de methode nulstabiel is en consistent van de orde* $p \geq 1$. We zullen ons hier verder beperken tot het vermelden van de voorwaarde voor nulstabiliteit. Deze luidt (zie VERWER [1976a])

$$d(c_m+\lambda_{m,0}+\lambda_{m,m-1}) \neq 0.$$

Voor onze formules zullen we steeds eisen dat

$$(3.3.6) \qquad d(c_m+\lambda_{m,0}+\lambda_{m,m-1}) = 1.$$

Dit betekent dat de zogenaamde genormaliseerde foutconstanten gelijk zijn aan de echte foutconstanten. Deze conditie van convergentie wordt ook gesteld bij lineaire meerstapsformules. Voor een uitgebreide discussie hierover verwijzen we naar HENRICI [1962] en WATT [1967].

3.3.1.2. Absolute stabiliteitseigenschappen

De absolute stabiliteitseigenschappen van een integratiemethode worden geanalyseerd aan de hand van de *scalaire lineaire modelvergelijking*

$$y'(x) = \delta y(x), \quad \delta \in \mathbb{C}, \quad \mathrm{Re}(\delta) < 0. \tag{3.3.7}$$

We gaan hier niet in op de afleiding en het belang van deze vergelijking voor de stabiliteit. Een uitgebreide behandeling hierover kan men vinden in nagenoeg alle inleidingen over numerieke integratie. Wel merken we op dat de complexe δ staat voor een *eigenwaarde van de Jacobiaan* van de functie f. Hier komen we straks op terug. Passen we schema (3.3.2) toe op vergelijking (3.3.7) dan vinden we de *lineaire recursievergelijking*

$$y_{n+1} = dS(z)y_n + dP(z)y_{n-1} + (1-d)y_{n-2}, \tag{3.3.8}$$

waarin S(z) en P(z) polynomen zijn van de graad m in $z = h\delta$. S en P noemen we de *stabiliteitspolynomen* behorende bij schema (3.3.2). Indien we schrijven

$$S(z) = \sum_{i=0}^{m} s_i z^i \quad \text{en} \quad P(z) = \sum_{i=0}^{m} p_i z^i, \tag{3.3.9}$$

dan geldt

$$\begin{aligned}
s_0 &= 1 - b_m, \\
s_1 &= \lambda_{m,0} + \lambda_{m,m-1}(1-b_{m-1}), \\
s_i &= \prod_{j=m-i+2}^{m} \lambda_{j,j-1}(\lambda_{m-i+1,0} + \lambda_{m-i+1,m-i}(1-b_{m-i})), \quad i = 2,\ldots,m-1,
\end{aligned} \tag{3.3.10}$$

en

$$\begin{aligned}
p_0 &= b_m, \\
p_1 &= \lambda_{m,m-1} b_{m-1} + c_m, \\
p_i &= \left(\prod_{j=m-i+1}^{m} \lambda_{j,j-1}\right) b_{m-i} + \left(\prod_{j=m-i+2}^{m} \lambda_{j,j-1}\right) c_{m-i+1}, \quad i = 2,\ldots,m-1, \\
p_m &= \left(\prod_{j=2}^{m} \lambda_{j,j-1}\right) c_1.
\end{aligned} \tag{3.3.11}$$

We voeren hier de volgende *definitie van absolute stabiliteit* in: schema (3.3.2) heet absoluut stabiel voor een zekere $z = h\delta$, indien voor die z geldt, $y_n \to 0$ als $n \to \infty$, waarbij y_n gegeven wordt door (3.3.8). Het *absolute stabiliteitsgebied* is gedefinieerd als het gebied in het z-vlak waar absolute stabiliteit optreedt. Deze definitie van stabiliteit sluit aan bij het gedrag van de oplossing $y(x) = y_0 e^{\delta x}$, y_0 gegeven beginwaarde, van de modelvergelijking (3.3.7).

De oplossing van de recursievergelijking (3.3.8) is samengesteld uit de wortels van zijn zogenaamde *karakteristieke vergelijking* (zie LAMBERT [1973])

$$\alpha^3 - dS(z)\alpha^2 - dP(z)\alpha - 1 + d = 0. \tag{3.3.12}$$

Men kan aantonen dat bij willekeurige startwaarden y_0, y_1 en y_2 $\lim_{n\to\infty} y_n = 0$, indien de wortels $\alpha_i(z)$, i = 1,2,3, van (3.3.12) binnen de eenheidscirkel liggen. Met andere woorden, schema (3.3.2) is absoluut stabiel voor zekere z, indien voor die z geldt $|\alpha_i(z)| < 1$. De wortels $\alpha_i(z)$ noemt men ook *amplificatiefactoren*.

Zoals opgemerkt stelt δ een eigenwaarde voor van de Jacobiaan van f. Aangezien we schema (3.3.2) hier gebruiken voor de integratie van parabolische problemen, veronderstellen we δ in een lange smalle strook langs de negatieve as. We zijn daarom geinteresseerd in stabiliteitspolynomen S(z) en P(z) die, tezamen met de parameter d, een absoluut stabiliteitsgebied geven wat zo'n lange smalle strook langs de negatieve as bevat. Bovendien moeten de parameter d en de polynomen S(z) en P(z) voldoen aan de consistentievoorwaarden (zie sectie 3.3.1.1) en de conditie van convergentie (3.3.6). Met behulp van relaties (3.3.10) en (3.3.11) vinden we voor S(z) en P(z) de consistentierelaties

$$\begin{array}{c|l} & s_0 + p_0 = 1, \\ p=1 & s_1 - p_0 + p_1 = (3-2d)/d; \\ \hline p=2 & s_2 + \tfrac{1}{2}p_0 - p_1 + p_2 = (-3/2+2d)/d; \end{array} \tag{3.3.13}$$

De conditie van convergentie (3.3.6) kunnen wij schrijven als

$$p_0 = \frac{2(d-1)}{d}. \tag{3.3.14}$$

Polynomen S en P van de graad twee tot en met twaalf, waarvoor voldaan is aan (3.3.13) en (3.3.14), en waarvoor het bijbehorende absolute stabiliteitsgebied een lange smalle strook langs de negatieve as bevat, zijn geconstrueerd in VERWER [1976b]. De *reële absolute stabiliteitsgrenzen* β hierbij zijn

$$\begin{aligned} &\beta(m) \simeq 5.15m^2, \quad p = 1, \\ &\beta(m) \simeq 2.29m^2, \quad p = 2. \end{aligned} \tag{3.3.15}$$

met $m = 2,\ldots,12$. De polynomen $S(z)$ en $P(z)$ zijn zo geconstrueerd dat $|\alpha_i(z)| \lesssim 0.9$, $-\beta \leq z \leq -1.5$. Een gevolg hiervan is dat er een *sterke demping* optreedt *voor de hogere harmonischen*. De stabiliteitsgrenzen (3.3.15) zijn bijna driemaal zo groot als de grenzen van gestabiliseerde eenstapsmethoden met dezelfde dempingseigenschappen (zie VAN DER HOUWEN [1977]).

3.3.1.3. Een klasse gestabiliseerde formules van orde een en twee

Nu de coefficiënten s_i en p_i van de stabiliteitspolynomen $S(z)$ en $P(z)$ gegeven zijn, kunnen de parameters van het eigenlijke rekenschema (3.3.2) bepaald worden uit de relaties (3.3.10) en (3.3.11). Uit deze relaties volgt onmiddellijk dat we hierbij een grote vrijheid hebben. We zullen deze vrijheid gebruiken om het rekenwerk te reduceren, door zoveel mogelijk parameters nul te kiezen, en om te voldoen aan de gelijkheid (zie (3.3.4))

$$c_{31} = c_{32}. \tag{3.3.16}$$

Voor het tweede orde geval geldt dan, vanwege de gelijkheid

$$y''' = f_j f_k^j f^k + f_{jk} f^j f^k,$$

dat de lokale afbreekfout (3.3.5) gegeven wordt door

$$y(x_{n+1}) - E[y(x_n), y(x_{n-1}), y(x_{n-2})] = c_3 h^3 y^{(3)}(x_n) + O(h^4), \tag{3.3.17}$$

C_3 constant. Deze vorm van de lokale afbreekfout is nodig voor het schatten van de lokale fout middels interpolatie op terugwaartse waarden van y en y'. Wij gaan op deze manier de lokale fout schatten om de schema's te voorzien van een stapkeuzemechanisme. Merk op dat voor het eerste orde geval de hoofdterm van de lokale afbreekfout altijd gegeven wordt door

$$C_2 h^2 y^{(2)}(x_n),\ C_2 \text{ constante.}$$

Niettemin zullen we, voor de uniformiteit in de formulering, ook voor het eerste orde geval voldoen aan (3.3.16).

Aangezien we alle parameters uit het schema uitdrukken in de coefficiënten s_i en p_i worden de *foutconstanten* C_2 en C_3 volledig bepaald door s_i en p_i. Het blijkt dat C_2 en C_3 nagenoeg onafhankelijk zijn van de graad m van de schema's. Dit betekent dat ook de nauwkeurigheid van de schema's bij benadering onafhankelijk is van m. De foutconstanten zijn tamelijk groot en worden gegeven door

(3.3.18) $\quad C_2 \simeq 1.27, \quad C_3 \simeq 0.44.$

Tenslotte geven we de parameters, uitgedrukt in s_i en p_i, die onze klasse van gestabiliseerde driestaps-methoden van de graad twee tot en met twaalf definiëren. De orde wordt dan bepaald door de keuze van s_i en p_i. Deze zijn gegeven in VERWER [1976b]. De expressies voor de parameters zijn:

$$b_i = 0, \quad i = 1,\ldots,m-2, \quad b_{m-1} = \frac{p_1 - c_m}{\lambda_{m,m-1}}, \quad b_m = p_0;$$

$$c_i = \frac{p_{m+1-i}}{s_{m-i}}, \quad i = 1,\ldots,m-2.$$

$$c_{m-1} = \frac{p_2}{\lambda_{m,m-1}},$$

(3.3.19) $$c_m = \frac{(1-\frac{1}{2}p_0)(p_1-2p_2+2p_3+2s_3) - (\frac{1}{2}+\frac{1}{4}p_0)^2}{2+p_1-2p_2+2p_3+2s_3},$$

$$\lambda_{i,0} = 0, \quad i = 2,\ldots,m,$$

$$\lambda_{i,i-1} = \frac{s_{m+1-i}}{s_{m-i}}, \quad i = 1,\ldots,m-2,$$

$$\lambda_{m-1,m-2} = \frac{s_2}{\lambda_{m,m-1}},$$

$$\lambda_{m,m-1} = 1 - \tfrac{1}{2}p_0 - c_m.$$

Opmerking. Bij het opstellen van een integratieschema is het aan te bevelen om zo mogelijk alle parameters positief te kiezen om het wegvallen van cijfers tegen te gaan. Helaas is dat voor onze tweede orde formules niet mogelijk, aangezien dan alle p_i negatief zijn, en alle s_i positief. Dit betekent dat voor de tweede orde formules de $\lambda_{j,j-1}$ parameters positief zijn en $d < 1$, terwijl b_m, b_{m-1} en de c_j parameters negatief zijn. In het eerste orde geval zijn de coefficiënten s_i en p_i alle positief. Dit betekent dat het mogelijk is om $y_{n+1}^{(m)}$ met positieve parameters te berekenen. Aangezien voor $p = 1$ de parameter $d = 1.375$, is de coefficiënt van y_{n-2} altijd negatief. Door de eerste orde methode te definiëren volgens (3.3.19) wordt bovendien b_{m-1} negatief gemaakt.

3.3.1.4. Interne stabiliteitseigenschappen

Een belangrijk verschijnsel, dat zich voordoet bij gestabiliseerde methoden van het Runge-Kuttatype, is de *accumulatie van afrondfouten,* welke optreedt per integratiestap. Deze geaccumuleerde afrondfouten kunnen zelfs zo groot worden dat zij van invloed zijn op de lokale nauwkeurigheid. Aangezien dit verschijnsel zich niet, of in veel mindere mate, voordoet bij niet-gestabiliseerde methoden van het Runge-Kuttatype, en daarom nieuw zal zijn voor vele lezers, zullen we er hier enige aandacht aan schenken.

In VAN DER HOUWEN [1977, sectie 2.6.10], wordt dit verschijnsel geanalyseerd voor de klasse van gestabiliseerde eenstapsmethoden welke bevat is in klasse (3.3.2). Hij definieert daar een functie welke bij benadering deze foutenopbouw beschrijft. Deze functie wordt de *interne stabiliteitsfunctie* genoemd. We zullen laten zien dat wij voor onze driestapsmethode dezelfde interne stabiliteitsfunctie kunnen definiëren.

Zij $\rho_{n+1}^{(j)}$ de lokale fout die optreedt bij de berekening van $y_{n+1}^{(j)}$. Zij $\varepsilon_{n+1}^{(j)}$ de geaccumuleerde lokale fout na de berekening van $y_{n+1}^{(j)}$. Verder, laat $\bar{y}_{n+1}^{(j)}$ de verstoring van $y_{n+1}^{(j)}$ zijn, dat wil zeggen $\varepsilon_{n+1}^{(j)} = \bar{y}_{n+1}^{(j)} - y_{n+1}^{(j)}$.
In plaats van (3.3.2) hebben we dan

$$\bar{y}_{n+1}^{(0)} = y_n,$$

$$\bar{y}_{n+1}^{(1)} = (1-b_1)y_n + b_1 y_{n-1} + c_1 hf(y_{n-1}) + \lambda_{1,0} hf(y_n) + \rho_{n+1}^{(1)},$$

$$\bar{y}_{n+1}^{(j)} = (1-b_j)y_n + b_j y_{n-1} + c_j hf(y_{n-1}) + \lambda_{j,0} hf(y_n) +$$

$$+ \lambda_{j,j-1} hf(\bar{y}_{n+1}^{(j-1)}) + \rho_{n+1}^{(j)}, \quad j = 2,\ldots,m; \quad m \geq 2,$$

$$\bar{y}_{n+1} = d\bar{y}_{n+1}^{(m)} + (1-d)y_{n-2}, \quad n \geq 2.$$

De fouten $\varepsilon_{n+1}^{(j)}$ voldoen aan

$$\varepsilon_{n+1}^{(1)} = \rho_{n+1}^{(1)},$$

$$\varepsilon_{n+1}^{(j)} = \lambda_{j,j-1} h[f(y_{n+1}^{(j-1)} + \varepsilon_{n+1}^{(j-1)}) - f(y_{n+1}^{(j-1)})] + \rho_{n+1}^{(j)}, \quad j = 2,\ldots,m,$$

$$\varepsilon_{n+1} = d\varepsilon_{n+1}^{(m)},$$

waarbij $\varepsilon_{n+1} = \bar{y}_{n+1} - y_{n+1}$. Door aan te nemen dat de Jacobiaan, zeg $J(y)$, langzaam varieert tijdens een integratiestap, geldt bij benadering

$$\varepsilon_{n+1}^{(j)} \simeq \lambda_{j,j-1} hJ(y_n)\varepsilon_{n+1}^{(j-1)} + \rho_{n+1}^{(j)}, \quad j = 2,\ldots,m.$$

Na enig elementair rekenwerk komen wij dan bij de afschatting

$$\|\varepsilon_{n+1}\| \leq [d + \sum_{k=1}^{m-1} d \prod_{j=m+1-k}^{m} |\lambda_{j,j-1}| \|(hJ(y_n))^k\|] \max_{1\leq k\leq m} \|\rho_{n+1}^{(k)}\|,$$

waarbij we als norm de spectraalnorm kiezen. De *interne stabiliteits-functie* wordt nu gedefinieerd door

$$(3.3.20) \quad Q(z) = d + \sum_{k=1}^{m-1} d \prod_{j=m+1-k}^{m} |\lambda_{j,j-1}| \, |z|^k.$$

In geval $J(y_n)$ een normale matrix is, geldt er dan

$$(3.3.21) \quad \|\varepsilon_{n+1}\| \leq Q(h\sigma(J(y_n))) \max_{1\leq k\leq m} \|\rho_{n+1}^{(k)}\|,$$

waarbij σ de spectraalradius aangeeft. De betekenis van deze ongelijkheid

wordt duidelijk indien men zich realiseert dat Q een sterk stijgende functie is en dat hσ zeer groot kan zijn. Het is dus gewenst om, in het geval van een normale matrix $J(y_n)$, de staplengte h zo te kiezen dat tenminste voldaan is aan

$$(3.3.22) \qquad Q(h\sigma(J(y_n))) \leq \frac{\text{maximaal lokale afbreekfout}}{\text{rekenprecisie}} \quad .$$

Dan zal, bij benadering, de lokale geaccumuleerde afrondfout niet van invloed zijn op de lokale afbreekfout. Indien echter wegvallen van cijfers optreedt kan voorwaarde (3.3.22) te optimistisch zijn.

Voorwaarde (3.3.22) wordt de *interne stabiliteitsvoorwaarde* genoemd, en voor willekeurige functies f toegepast. Uit praktische ervaring blijkt dat indien aan (3.3.22) voldaan is, de interne foutenopbouw in het algemeen onder controle blijft. De waarden Q(β) zijn gegeven in table 3.3.1, en gelden zowel voor de eerste orde, als voor de tweede orde formules.

Om de betekenis van de interne stabiliteitsvoorwaarde te illustreren hebben we een experiment uitgevoerd met de eenvoudige lineaire vergelijking

$$\begin{aligned} y_1' &= (-2y_1+y_2+1)*10000, \\ (3.3.23) \qquad y_j' &= (y_{j-1}-2y_j+y_{j+1})*10000, \qquad j = 2,\ldots,99. \\ y_{100}' &= (y_{99}-2y_{100}+1)*10000. \end{aligned}$$

De Jacobiaan hiervan is symmetrisch met $\sigma \simeq 40000$. Door de beginwaarden $y_j(0) = 1$, $j = 1,\ldots,100$, voor te schrijven hebben we de oplossing $y_j(x) \equiv 1$, $x \geq 0$, $j = 1,\ldots,100$. Indien we exact rekenen, d.w.z. zonder afrondfouten, zal een integratieschema waarvan de parameters exact representeerbaar zijn, de analytische oplossing geven.

Wij hebben één integratiestap met $h = \beta/\sigma$ uitgevoerd met onze driestapsschema's van de graad $m = 3,\ldots,12$, en van orde $p = 1$ en $p = 2$. Het experiment is uitgevoerd op een CDC73/28 computer met een rekenprecisie van ongeveer 14 cijfers. De benodigde startwaarden zijn daarom gekozen als $1 + rn*_{10}-14$, waarbij rn een random getal is tussen -1 en +1. De resultaten van dit experiment zijn vermeld in tabel 3.3.1. In deze tabel is gegeven $10^{14} * \text{fout}_p(m)$, met $\text{fout}_p(m) = \max_j(y_j-1)$,

m	$Q(\beta)$	$10^{14}*fout_1(m)$	$10^{14}*fout_2(m)$
3	$.1_{10}3$	$.9_{10}1$	$.9_{10}2$
4	$.7_{10}3$	$.6_{10}2$	$.4_{10}3$
5	$.4_{10}4$	$.4_{10}3$	$.1_{10}5$
6	$.3_{10}5$	$.1_{10}4$	$.8_{10}5$
7	$.2_{10}6$	$.7_{10}4$	$.6_{10}6$
8	$.9_{10}6$	$.3_{10}5$	$.1_{10}7$
9	$.5_{10}7$	$.2_{10}6$	$.9_{10}7$
10	$.3_{10}8$	$.2_{10}7$	$.2_{10}9$
11	$.2_{10}9$	$.3_{10}7$	$.4_{10}10$
12	$.1_{10}10$	$.4_{10}8$	$.4_{10}11$

Tabel 3.3.1

De resultaten van dit experiment bevestigen dat het gedrag van de accumulatie van afrondfouten met toenemende m in overeenstemming is met de foutenanalyse zoals boven gegeven. Voor de eerste orde formules is de interne stabiliteitsvoorwaarde voor stelsel (3.3.23) iets te pessimistisch. Daarentegen is de interne stabiliteitsvoorwaarde hier voor de tweede orde formules van hogere graad duidelijk te optimistisch. Er treedt hier wegvallen van cijfers op wat veroorzaakt wordt door het voorkomen van positieve en negatieve parameters. Dit betekent dat wij de tweede orde formules van hogere graad met enige voorzichtigheid toe moeten passen.

3.3.2. De implementatie van driestaps-Runge-Kuttamethoden

Een veel gebruikte techniek bij het implementeren van lineaire meerstaps-formules is de techniek van NORDSIECK [1962]. Deze techniek bestaat hierin dat niet de y en y' vectoren opgeslagen worden, maar een lineaire transformatie van deze vectoren. De lineaire transformatie wordt dan zo gekozen dat men rekent met vectoren welke bij iedere integratiestap beschouwd kunnen worden als geschaalde hogere afgeleiden van het interpolerende polynoom door de y en y' waarden. Anders gezegd, de lineaire transformatie wordt zo gekozen dat men lokaal met een eenstapsformule werkt.

Hierdoor kan men zeer makkelijk van *staplengte veranderen,* en met de geschaalde hogere afgeleiden kan men direct *lokale fouten schatten.* Bovendien is op deze manier een *ordestrategie* en een *startmechanisme* zonder veel moeite te realiseren. Voor een eenvoudig en efficiënt gebruik van meerstapsmethoden is het gewenst dat een implementatie deze faciliteiten bezit (voor meer informatie hierover, zie GEAR [1971]).

Vanwege het specifieke karakter van onze gestabiliseerde driestapsformules is een Nordsieck-implementatie ervan echter niet aan te bevelen. Dit om twee redenen. Ten eerste, een Nordsieck-implementatie is verhoudingsgewijs duur voor stelsels met veel vergelijkingen. Ten tweede, indien de y en y' vectoren getransformeerd worden naar een geschaalde afgeleiden-representatie moet men de parameters mee transformeren. Dit heeft tot gevolg dat veel van de parameters, die in de Lagrangerepresentatie (3.3.2) nul zijn, ongelijk aan nul worden. We zullen daarom een implementatie bespreken welke gebaseerd is op de Lagrangeformulering. In deze implementatie zijn opgenomen eerste en tweede orde schema's van de graad twee tot en met twaalf. Buiten de bovenvermelde controlemechanismen bevat deze implementatie een mechanisme wat de graad van de schema's zo laag mogelijk houdt. Dit om het grote aantal evaluaties van de afgeleide zo klein mogelijk te houden.

Een FORTRAN-versie van de hier te bespreken implementatie is in ontwikkeling. Met nadruk wordt er dan ook op gewezen, dat veel van wat hier besproken wordt zich nog in een *experimentele fase* bevindt. Niettemin zullen we in sectie 3.3.3 een aantal numerieke resultaten geven van deze integratieroutine, welke M3RK genoemd is.

3.3.2.1. Het starten van het proces

Voor het bepalen van de twee extra startvectoren y_1 en y_2 passen we een 2de orde gestabiliseerd eenstapsschema toe, welke wij verkrijgen uit (3.3.2) door de keuze: $d = 1$, $b_j = c_j = 0$, $j = 1,\ldots,m$; $\lambda_{j,0} = 0$, $j = 2,\ldots,m$. De resterende parameters $\lambda_{j,j-1}$ worden gekozen als

$$\lambda_{j,j-1} = \bar{s}_{m+1-j} \; / \; \bar{s}_{m-j}, \quad j = 1,\ldots,m-2,$$

$$\lambda_{m-1,m-2} = \tfrac{1}{2}, \qquad \lambda_{m,m-1} = 1,$$

waarbij $\bar{s}_j$, $j = 3,\ldots,m$, de coefficiënten zijn van de sterk stabiele stabiliteitsfunctie

$$R_m^{(2)}(z) = 1 + z + \tfrac{1}{2}z^2 + \bar{s}_3 z^3 + \ldots + \bar{s}_m z^m,$$

welke gegeven is door VAN DER HOUWEN [1977, tabel 2.6.7'']. De extrema van $R_m^{(2)}$ worden begrensd door .95. Hierbij ligt de graad m tussen reële stabiliteitsgrenzen voor deze stabiliteitsfunctie zijn:

m	3	4	5	6	7	8	9	10	11	12
$\beta(m)/m^2$	.68	.73	.76	.77	.78	.79	.79	.79	.79	.80

De eerste twee integratiestappen worden uitgevoerd met staplengte $h = h_{min}$, welke door de gebruiker dient te worden gespecificeerd. Deze beginstap, welke tevens als minimale staplengte voor het hele proces geldt, moet zo klein zijn dat voldaan is aan

$$h_{min} * \sigma \leq \beta(m_{max}).$$

De bepaling van σ en de verklaring van m_{max} worden besproken in sectie 3.3.2.5. Indien niet voldaan is aan de bovenstaande stabiliteitsvoorwaarde kan niet worden gestart.

3.3.2.2. Het schatten van de lokale fout

Wat men zou wensen bij numerieke integratie is bij iedere integratiestap een goede schatting te hebben van de globale fout, dat wil zeggen van de fout van de differentieoplossing ten opzichte van de globale analytische oplossing. In het algemeen is dit echter een ondoenlijke zaak om zo'n schatting te bepalen. Daarom moeten wij ons tevreden stellen met *schattingen van de lokale fout*, dat wil zeggen van de fout van de differentieoplossing ten opzichte van de lokaal analytische oplossing door $y = y_n$. Overeenkomstig (3.3.17) kunnen we de lokale afbreekfout in eerste benadering schatten met

$$C_{p+1} h^{p+1} y^{(p+1)}(x).$$

Na de n-de integratiestap, n > 2, schatten wij nu de lokale fout, welke gemaakt is bij de berekening van y_{n+1}, door als volgt te interpoleren:

$$C_2 h^2 y^{(2)}(x_n) \simeq E_2 = \frac{C_2}{C_2+\frac{1}{2}}(y_{n+1}-2y_n+y_{n-1}), \qquad p = 1,$$

$$C_3 h^3 y^{(3)}(x_n) \simeq E_3 = \frac{C_3}{C_3+\frac{5}{6}}(y_{n+1}-3(y_n-y_{n-1}) - y_{n-2}), \qquad p = 2.$$

Vervolgens gaan we na of voldaan is aan het *foutencriterium*

$$\text{(3.3.24)} \qquad tol + tol * \|y_{n+1}\| \leq C_{p+1}\|E_{p+1}\| ,$$

waarin $\|\cdot\|$ de gedeelde Euclidische norm voorstelt, en tol een door de gebruiker *gespecificeerde lokale tolerantie*. Het bovenstaande criterium noemt men een *gemengd criterium,* omdat voor grote waarden van y het criterium relatief werkt, en voor kleine waarden van y absoluut.

Indien niet aan het criterium voldaan is wordt de integratiestap verworpen en een kleinere staplengte bepaald. Indien wel aan het criterium voldaan is wordt de integratiestap geaccepteerd en zo nodig een grotere staplengte bepaald.

Wellicht ten overvloede merken wij nogmaals op dat wij met (3.3.24) de lokale fout onder controle houden. Deze lokale fouten kunnen op verschillende manieren accumuleren tot een globale fout, zodat na het beëindigen van de integratie de globale fout groter kan zijn dan de gespecificeerde tolerantie. In de praktijk werken lokale foutschatters echter zeer bevredigend.

3.3.2.3. Het variëren van de staplengte

Voordat we ingaan op de berekening van een nieuwe staplengte, bespreken we eerst hoe voor schema (3.3.2) het variëren van de staplengte gerealiseerd kan worden.

In sectie 3.3.1 hebben wij verondersteld dat h constant is. Dit betekent dat indien wij van staplengte willen veranderen, dit buiten het integratieschema (3.3.2) dient te gebeuren. Anders gezegd, indien wij met een nieuwe h verder willen integreren moeten we het integratieschema voorzien van nieuwe startwaarden. Het berekenen van de nieuwe startwaarden kan op twee manieren gebeuren. Ten eerste, door het toepassen van een

eenstapsformule. Dit is echter omslachtig en dus inefficiënt. Ten tweede, door het toepassen van *interpolatie* op reeds berekende waarden, wat duidelijk de voorkeur verdient boven het toepassen van een eenstapsformule. Wij zullen dan ook interpolatie toepassen, doch alleen voor de y-waarden.

Zij αh de nieuwe staplengte, en h de oude. De gebruikte interpolatieformule luidt dan

$$(3.3.25) \quad y(x-\bar{\alpha}h) \simeq \tfrac{1}{2}\bar{\alpha}(\bar{\alpha}-1)y(x-2h)+\bar{\alpha}(2-\bar{\alpha})y(x-h)+\tfrac{1}{2}(2-\bar{\alpha})(1-\bar{\alpha})y(x),$$

waarbij $\bar{\alpha} = \alpha$ of $\bar{\alpha} = 2\alpha$.

Formule (3.3.25) is kwadratisch. De geinduceerde fout, ten gevolge van de interpolatie, is voor de tweede orde schema's dus in de orde van de lokale afbreekfout. Formules (3.3.25) worden ook toegepast voor de eerste orde schema's. De afgeleide in $x = x_n - \alpha h$ wordt direct bepaald en niet met interpolatie berekend.

Aangezien het variëren van de staplengte buiten het rekenschema om gebeurt, zal duidelijk zijn dat dit niet bij iedere integratiestap mag optreden. Indien het te vaak gebeurt kan de stabiliteit van het proces verloren gaan. Een vuistregel die bij k-stapformules wordt toegepast, en die wij ook zullen hanteren, luidt: na een verandering van de staplengte minstens k + 1 stappen met constante staplengte uitvoeren, tenzij er stapverwerping optreedt. Stapverwerping wordt dus altijd toegestaan, hoewel dit aanleiding kan geven tot instabiliteiten. We komen hierop terug in de volgende subparagraaf.

3.3.2.4. De stapkeuze- en ordestrategie

Stapkeuze- en orde-strategieën zijn veelal grotendeels gebaseerd op *heuristische* beschouwingen. Dit geldt ook voor hetgeen hier besproken wordt. We gaan hier niet in op alle aspecten en zullen slechts de gevoerde strategie bespreken. Voor een meer uitvoerige beschouwing verwijzen we naar GEAR [1971] en VAN DER HOUWEN [1977].

Zij h de oude, en αh de nieuwe staplengte. Definieer (zie (3.3.24))

$$\bar{\alpha} = \left(\frac{\mathrm{tol} + \mathrm{tol}*\|y_{n+1}\|}{C_{p+1}\|E_{p+1}\|} \right)^{\frac{1}{p+1}} .$$

Deze wortelformule is gebaseerd op de evenredigheid van de hoofdterm van de lokale afbreekfout met h^{p+1}, en gaat uit van de veronderstelling dat de hogere afgeleiden alle constant zijn. Aangezien deze veronderstelling in het algemeen slechts bij benadering juist is, voegen we numerieke drempels toe:

$$\alpha = \begin{cases} \bar{\alpha}/3.0, & p = 1, \\ \bar{\alpha}/1.3, & p = 2. \end{cases}$$

Om marginale veranderingen in de staplengte te voorkomen wordt er geen verandering uitgevoerd, indien $.9 < \alpha < 1.1$. Tenslotte, om grote variaties in de staplengte te vermijden, wordt α aangepast aan de bovengrens 3. Merk op dat, vanwege de numerieke drempels, $\alpha < 1$ niet altijd impliceert dat niet voldaan is aan (3.3.24).

Met uitzondering van de eerste twee integratiestappen wordt er na iedere stap gecontroleerd of voldaan is aan het foutencriterium. Echter een stapverandering vindt plaats, minstens 4 stappen na een laatste verandering van de staplengte (zie sectie 3.3.2.3). Een uitzondering op deze regel wordt gemaakt indien er stapverwerping optreedt, dan wordt de staplengte altijd verkleind. Aangezien herhaald stapverwerpen kan duiden op instabiliteiten, wordt het integratieproces na drie verwerpingen achter elkaar onderbroken en opnieuw gestart (zie sectie 3.3.2.1) met de minimale staplengte $h = h_{min}$.

De minimale staplengte h_{min} dient te worden gegeven door de gebruiker. Hier wordt het proces ook mee gestart. Indien tijdens het proces $h = h_{min}$, en niet is voldaan aan het foutencriterium, wordt de integratie toch voortgezet. In deze situatie wordt namelijk in het algemeen wel een bruikbaar resultaat afgeleverd. Als deze situatie zich voordoet, wordt daarvan melding gegeven aan de gebruiker. Het programma berekent zelf een maximale staplengte, namelijk

$$h_{max}(p) = \beta(m_{max})/\sigma ,$$

waarin p de orde van het schema aangeeft. De bepaling van σ en de verklaring van m_{max} worden besproken in de volgende subparagraaf.

Tot slot van deze subsectie de ordestrategie, die zeer eenvoudig is gehouden. Het proces wordt gestart met een tweede orde eenstapsschema. Na twee integratiestappen, wordt er verder gegaan met het tweede orde driestapsschema. Zodra de staplengte $h = h_{max}(2)$ bereikt is, wordt een α berekend voor het

eerste orde geval. Indien deze $\alpha > 1.1$ gaan we over op de eerste orde schema's die een veel grotere staplengte toelaten. Als tijdens het proces $p = 1$ en $h \leq h_{max}(2)$, gaan we weer terug naar $p = 2$. De bedoeling is dat deze orde-strategie nog verfijnd wordt.

3.3.2.5. Het bepalen van de graad

In subparagraaf 3.3.1.4 hebben wij gezien dat de opbouw van afrondfouten per integratiestap de lokale nauwkeurigheid kan verstoren indien, bij benadering, niet voldaan is aan de interne stabiliteitsvoorwaarde (3.3.22). Daarom is het gewenst dat bij een gegeven tolerantie en rekenprecisie, een *maximale graad,* zeg m_{max}, wordt vastgesteld. Dit gebeurt in het programma. Uit tabel 3.3.1 kan men aflezen hoe groot m_{max} is bij een gegeven tolerantie en rekenprecisie, uitgaande van een maximale h.

Een eigenschap van de geïmplementeerde schema's is dat de nauwkeurigheid nagenoeg onafhankelijk is van de graad (zie subparagraaf 3.3.1.3). Dus het is toegestaan om de graad m zo te kiezen dat

(3.3.26) $h\sigma \leq \beta(m)$, m minimaal,

bij gegeven h en σ. Indien m steeds gekozen wordt overeenkomstig (3.3.26), minimaliseert men het aantal uit te voeren functie-evaluaties.

Om deze keuze te kunnen maken hebben we een goede *bovenschatting van de spectraalradius* nodig. Als de gebruiker zo'n bovenschatting niet kan of wil geven, wordt σ benaderd met behulp van een, voor niet-lineaire functies aangepaste, *power-methode*. Deze power-methode laat zich als volgt beschrijven (cf. LINDBERG [1972]): Zij $J(y_0)$ de Jacobiaan van de vectorfunctie f(y) in $y = y_0$, waarvan $\sigma(J(y_0))$ wordt gevraagd. Zij r_i een random getal in het interval $[-\varepsilon,\varepsilon]$, $\varepsilon > 0$, en zij x_0 een random verstoring van de vector y_0, die componentsgewijs gedefinieerd is door

$$x_{0,i} = \begin{cases} y_{0,i} + r_i y_{0,i}, & y_{0,i} \neq 0, \\ r_i, & y_{0,i} = 0. \end{cases}$$

De power-methode is dan gedefinieerd door de iteratie

$$x_{i+1} = x_0 + \varepsilon\|x_0\| \frac{f(x_i)-f(x_0)}{\|f(x_i)-f(x_0)\|} , \qquad i = 1,2,\ldots,$$

(3.3.27)

$$\rho_{i+1} = \frac{\|f(x_{i+1})-f(x_0)\|}{\varepsilon\|x_0\|} , \qquad i = 1,2,\ldots,$$

waarbij $x_1 = y_0$ en waarbij $\|\cdot\|$ de Euclidische norm voorstelt. Als f(y) een lineaire functie is, d.w.z. J(y) is constant, dan zal ρ_i convergeren naar de grootste eigenwaarde van de constante Jacobiaan. Bij een niet constante Jacobiaan zijn de ρ_i's schattingen van de Lipschitzconstante

$$L = \sup_{y\in S(x_0,\varepsilon)} \|J(y)\| .$$

Echter, door ε voldoende klein te kiezen mogen we J(y) constant beschouwen in $S(x_0,\varepsilon)$, en zal ook voor niet-lineaire functies ρ_i convergeren naar de spectraalradius van $J(y_0)$.

Voor ε hebben we gekozen 10^{-8}. De iteratie wordt afgebroken indien een relatief verschil is bereikt van 10^{-3}. In het algemeen zal het iteratieproces langzaam convergeren omdat de functies f(y) de eigenschap bezitten dat de eigenwaarden van de Jacobiaan vrij dicht bij elkaar liggen. Om een veilige bovenschatting te verkrijgen, vermeerderen wij het resultaat van het iteratieproces daarom nog eens met 10%. Indien f niet lineair is, zal met het voortschrijden van het integratieproces de spectraalradius in het algemeen veranderen. Daarom moet na een bepaald aantal integratiestappen σ opnieuw benaderd worden. In de huidige implementatie is dit aantal op 25 gezet. Bovendien wordt σ opnieuw uitgerekend na een stapverwerping, vooropgesteld dat f niet lineair is. Stapverwerping kan namelijk veroorzaakt worden door instabiliteiten.

3.3.3. Numerieke voorbeelden

We geven in deze sectie numerieke resultaten van M3RK toegepast op een tweetal, met behulp van PDEFEM, semi-gediscretiseerde parabolische problemen. Het gebruik van PDEFEM is reeds toegelicht in de vorige paragraaf. Het gebruik van M3RK zullen wij hier niet toelichten, aangezien deze routine nog te zeer in ontwikkeling is. We zullen volstaan met die informatie welke aansluit bij wat in paragraaf 3.3.2 besproken is.

Voorbeeld 3.3.1. Het eerste voorbeeld is het een-dimensionale stelsel van twee vergelijkingen (3.2.17), waarvoor het gebruik van PDEFEM geillustreerd is. Het probleem is twee keer opgelost. De eerste keer voor een equidistante verdeling van het interval [0,1] met 21 steunpunten, zeg N = 21, en de tweede keer voor een equidistante verdeling met 41 steunpunten, zeg N = 41. Aangezien er twee vergelijkingen zijn, heeft het eerste te integreren stelsel gewone differentiaalvergelijkingen 42 componenten, terwijl het tweede stelsel 82 componenten heeft. Beide stelsels zijn geintegreerd met M3RK over het interval [0,20]. Voor t = 20 is nagenoeg de stationaire oplossing bereikt. M3RK is beide keren aangeroepen met $h_{min} = 10^{-5}$ en tol = 10^{-4}, en heeft zelf σ bijgehouden. In de tabellen 3.3.2 en 3.3.3 is in een aantal punten de berekende oplossing gegeven voor respectievelijk N = 21 en N = 41. Het vergelijken hiervan geeft een betrouwbare indruk van de nauwkeurigheid van de resultaten.

Om tevens een indruk te krijgen van het verloop van het integratieproces geven wij tabel 3.3.4 en 3.3.5. Hierin is int het aantal uitgevoerde integratiestappen, fev het aantal evaluaties van de afgeleide f, sig het aantal evaluaties van f gebruikte om de schatting sigma van de spectraalradius σ te bepalen. Er zijn geen integratiestappen verworpen.

U-component

t \ x	0.0	0.2	0.4	0.6	0.8	1.0
0.1	.2134	.4740	.5105	.5125	.5258	1.0000
1.0	.0411	.1991	.3690	.5150	.6561	1.0000
5.0	.0323	.1649	.3252	.4850	.6448	1.0000
10.0	.0319	.1634	.3226	.4820	.6422	1.0000
20.0	.0320	.1635	.3229	.4823	.6425	1.0000

V-component

t \ x	0.0	0.2	0.4	0.6	0.8	1.0
0.1	.0000	.4087	.4829	.4898	.5164	.6421
1.0	.0000	.1963	.3663	.5144	.6564	.7789
5.0	.0000	.1647	.3251	.4849	.6438	.7742
10.0	.0000	.1633	.3226	.4820	.6413	.7725
20.0	.0000	.1634	.3229	.4823	.6416	.7727

Tabel 3.3.2. N = 21.

U-component

t \ x	0.0	0.2	0.4	0.6	0.8	1.0
0.1	.2211	.4743	.5105	.5124	.5234	1.0000
1.0	.0424	.1985	.3670	.5110	.6498	1.0000
5.0	.0331	.1631	.3215	.4795	.6377	1.0000
10.0	.0329	.1619	.3195	.4772	.6357	1.0000
20.0	.0329	.1619	.3195	.4772	.6358	1.0000

V-component

t \ x	0.0	0.2	0.4	0.6	0.8	1.0
0.1	.0000	.4090	.4828	.4893	.5116	.6217
1.0	.0000	.1956	.3643	.5103	.6501	.7696
5.0	.0000	.1629	.3214	.4794	.6367	.7642
10.0	.0000	.1618	.3195	.4772	.6347	.7629
20.0	.0000	.1618	.3195	.4772	.6348	.7629

Tabel 3.3.3. N = 41.

t	int	fev	sig	sigma
0.0	0	6	6	4002.9
0.1	56	226	40	480.3
1.0	75	326	61	466.2
5.0	89	445	61	466.2
10.0	94	516	61	466.2
20.0	100	606	70	478.3

Tabel 3.3.4. N = 21

t	int	fev	sig	sigma
0.0	0	19	19	5037.5
0.1	52	236	62	1783.5
1.0	70	361	62	1783.5
5.0	92	635	79	1839.9
10.0	106	832	103	1845.0
20.0	130	1143	113	1847.5

Tabel 3.3.5. N = 41

Voorbeeld 3.3.2. Het tweede voorbeeld is het niet-lineaire, een-dimensionale diffusieprobleem (cf. SINCOVEC & MADSEN [1975])

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(u \frac{\partial u}{\partial x}\right) - u^2, \qquad 0 \le x \le 1, \quad t > 0,$$

$$u(0,x) = 50, \qquad 0 \le x \le 1,$$

$$u(t,0) = 50, \qquad u_x(t,1) = 1 - \sin(u), \qquad t > 0.$$

De interface PDEFEM is hier eenvoudig op toe te passen. Het probleem is weer opgelost voor twee equidistante verdelingen van het interval [0,1], namelijk met N = 21 en N = 41, waarbij N het aantal steunpunten aangeeft. De beide te integreren stelsels hebben dus respectievelijk 21 en 41 componenten. Het integratie-interval is [0,0.1]. Voor t = 0.1 is de stationaire oplossing bereikt. M3RK is beide keren aangeroepen met $h_{min} = 10^{-5}$ en $tol = 10^{-3}$, terwijl de routine zelf weer σ berekent. In de tabellen 3.3.6 en 3.3.7 is weer de berekende oplossing gegeven voor respectievelijk N = 21 en N = 41. Het vergelijken hiervan geeft een betrouwbare indruk van de nauwkeurigheid van de resultaten. De tabellen 3.3.8 en 3.3.9 geven informatie over het verloop van het integratieproces. Er zijn geen stappen verworpen.

t \ x	0.0	0.2	0.4	0.6	0.8	1.0
.010	50.00	45.09	41.47	39.05	37.72	37.46
.025	50.00	44.51	40.27	37.28	35.60	35.25
.050	50.00	44.40	40.03	36.89	35.07	34.59
.100	50.00	44.38	39.98	36.81	34.94	34.43

Tabel 3.3.6 N = 21.

t \ x	0.0	0.2	0.4	0.6	0.8	1.0
.010	50.00	45.10	41.47	39.05	37.72	37.45
.025	50.00	44.50	40.24	37.24	35.55	35.19
.050	50.00	44.40	40.02	36.87	35.04	34.55
.100	50.00	44.38	39.98	36.81	34.95	34.44

Tabel 3.3.7 N = 41.

t	int	fev	sig	sigma
.000	0	21	21	129450.7
.010	32	201	46	117531.8
.025	42	303	46	117531.8
.050	50	436	69	112822.8
.100	63	581	69	112822.8

Tabel 3.3.8 N = 21

t	int	fev	sig	sigma
.000	0	16	16	515854.3
.010	41	322	32	495133.2
.025	61	582	56	478309.5
.050	99	968	126	475009.3
.100	141	1456	149	475556.3

Tabel 3.3.9 N = 41

3.4. Gestabiliseerde Runge-Kuttamethoden voor de tijdsintegratie van hyperbolische differentiaalvergelijkingen

Na in de voorgaande paragraaf de integratie van *parabolische* differentiaalvergelijkingen aan de orde gesteld te hebben, besteden we nu aandacht aan *hyperbolische* differentiaalvergelijkingen. Eenvoudige voorbeelden van hyperbolische vergelijkingen zijn:

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} ,$$

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} ,$$

$$\frac{\partial^2 u}{\partial t^2} = -\frac{\partial^4 u}{\partial x^4} + \frac{\partial^2 u}{\partial x^2} .$$

Voorbeelden van minder "schoolse" hyperbolische differentiaalvergelijkingen zijn in deel 1a van dit colloquium (zie paragraaf 2.2) behandeld. Door de inmiddels vertrouwd geworden semi-discretisatietechnieken, kunnen we vele hyperbolische differentiaalvergelijkingen, na toevoeging van begin- en randvoorwaarden, terugbrengen tot beginwaardeproblemen voor gewone differentiaalvergelijkingen van de vorm (in de notatie zoals gebruikelijk bij gewone differentiaalvergelijkingen)

(3.4.1) $\quad y'(x) = f(x,y(x))$

of

(3.4.2) $\quad y''(x) = f(x,y(x))$

waarin y en f vectorfuncties voorstellen waarvan het aantal componenten gelijk is aan hetzij het aantal roosterpunten gebruikt in de eindige differentiemethode, hetzij het aantal basisfuncties gebruikt in de eindige elementenmethode. We hebben derhalve te maken met in het algemeen *zeer grote* stelsels differentiaalvergelijkingen. Een tweede kenmerk van de door semi-discretisatie verkregen stelsels is dat, indien afkomstig van hyperbolische vergelijkingen, de eigenwaarden van de Jacobiaan van de rechterlidfunctie f (aan te geven met $\partial f/\partial y$) min of meer *imaginair* zijn in het geval (3.4.1) en min of meer *negatief* zijn in het geval (3.4.2). In het vervolg zullen we kortheidshalve gewone differentiaalvergelijkingen met de zojuist genoemde kenmerken *hyperbolische vergelijkingen* van de eerste respectievelijk tweede orde noemen.

Het is van belang om op te merken dat vergelijking (3.4.2) geschreven kan worden als een vergelijking van de vorm (3.4.1). Immers de nieuwe variabele w = y' voert (3.4.2) over in

$$(3.4.3) \qquad \begin{aligned} y'(x) &= w(x) \\ w'(x) &= f(x,y(x)) \end{aligned}$$

hetgeen een eerste orde vergelijking voor $\binom{y}{w}$ voorstelt en dus van de vorm (3.4.1) is. Verder geldt dat het spectrum van de Jacobiaan van (3.4.3) *imaginair* is wanneer dat van (3.4.2) *negatief* is. Dit volgt direct uit de matrix van partiële afgeleiden van het rechterlid van (3.4.3):

$$\begin{pmatrix} 0 & I \\ \frac{\partial f}{\partial y} & 0 \end{pmatrix} .$$

De eigenwaarden δ van deze matrix worden gegeven door

$$(3.4.4) \qquad \delta^2 - \tilde{\delta} = 0,$$

waarin $\tilde{\delta}$ de eigenwaarden van $\partial f/\partial y$ doorloopt. Aangezien deze laatste negatief zijn, zullen de eigenwaarden δ imaginair zijn. Hyperbolische differentiaalvergelijkingen die tot stelsels van de vorm (3.4.2) aanleiding geven zijn blijkbaar bijzondere gevallen van hyperbolische differentiaalvergelijkingen die tot de vorm (3.4.1) gediscretiseerd kunnen worden.

In deze paragraaf gaan we in op de numerieke integratie van hyperbolische vergelijkingen van de eerste en tweede orde.

3.4.1. Hyperbolische vergelijkingen van de eerste orde

Zoals uiteengezet in paragraaf 3.1 is het mogelijk via semi-discretisatie en met behulp van *expliciete* integratieformules vrij algemeen toepasbare programmatuur voor partiele differentiaalvergelijkingen te ontwikkelen. Voor hyperbolische vergelijkingen van de eerste orde komen eenstaps-Runge-Kuttamethoden met gereduceerd geheugengebruik in aanmerking. Deze formules vallen onder de klasse (3.3.2) en ontstaan voor $d = 1$, $b_1 = c_1 = b_j = c_j = \lambda_{j,0} = 0$ voor $j = 2,3,\ldots,m$. Dit betekent dat de voor de klasse (3.3.2) afgeleide consistentievoorwaarden ook voor de hier beschouwde eenstapsformules

gelden. Met weglating van de details komen we uiteindelijk tot de volgende algoritmen: voor p = 1 (geschreven in autonome vorm, d.w.z. f alleen afhankelijk van y)

$$
\begin{aligned}
y_{n+1}^{(1)} &= y_n + \lambda_{10} hf(y_n), \\
y_{n+1}^{(2)} &= y_n + \lambda_{21} hf(y_{n+1}^{(1)}), \\
&\quad \cdots \\
y_{n+1}^{(m)} &= y_n + hf(y_{n+1}^{(m-1)}), \\
y_{n+1} &= y_{n+1}^{(m)};
\end{aligned}
\tag{3.4.5}
$$

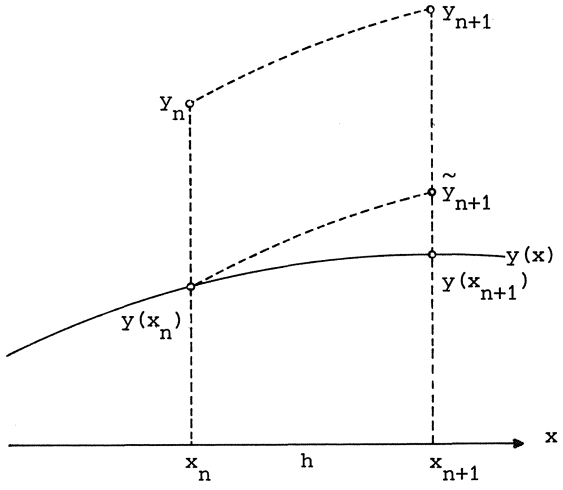
voor p = 2

$$
\begin{aligned}
y_{n+1}^{(1)} &= y_n + \lambda_{10} hf(y_n), \\
&\quad \cdots \\
y_{n+1}^{(m-1)} &= y_n + \tfrac{1}{2} hf(y_{n+1}^{(m-2)}), \\
y_{n+1}^{(m)} &= y_n + hf(y_{n+1}^{(m-1)}), \\
y_{n+1} &= y_{n+1}^{(m)};
\end{aligned}
\tag{3.4.6}
$$

en tenslotte voor p = 3

$$
\begin{aligned}
y_{n+1}^{(1)} &= y_n + \lambda_{10} hf(y_n), \\
y_{n+1}^{(2)} &= y_n + \tfrac{1}{4} hf(y_n) + \lambda_{21} hf(y_{n+1}^{(1)}), \\
y_{n+1}^{(3)} &= y_n + \tfrac{1}{4} hf(y_n) + \lambda_{32} hf(y_{n+1}^{(2)}), \\
&\quad \cdots \\
y_{n+1}^{(m-2)} &= y_n + \tfrac{1}{4} hf(y_n) + \frac{17}{60} hf(y_{n+1}^{(m-3)}), \\
y_{n+1}^{(m-1)} &= y_n + \tfrac{1}{4} hf(y_n) + \frac{5}{12} hf(y_{n+1}^{(m-2)}), \\
y_{n+1}^{(m)} &= y_n + \tfrac{1}{4} hf(y_n) + \tfrac{3}{4} hf(y_{n+1}^{(m-1)}), \\
y_{n+1} &= y_{n+1}^{(m)}.
\end{aligned}
\tag{3.4.7}
$$

In deze rekenschema's zijn de voorkomende parameters $\lambda_{j,j-1}$ nog vrij te kiezen. Deze vrijheden benutten we door de algoritmen aan te passen aan het op te lossen probleem, in dit geval een hyperbolische vergelijking. We illustreren dit aan de hand van het rekenschema dat uit (3.4.5) voor m = 3 ontstaat. In figuur 3.4.1 is de situatie in het punt x_n geschetst.



Figuur 3.4.1. Afbreekfout en globale fout

Voor een scalaire vergelijking is de numerieke oplossing y_n en y_{n+1} in x_n en x_{n+1} getekend en de numerieke oplossing $\tilde{y}_{n+1}$ die men met het rekenschema in x_{n+1} zou vinden wanneer dit schema niet op y_n maar op de analytische oplossing $y(x_n)$ toegepast zou worden ($y(x)$ stelt de analytische oplossing voor). Het verschil tussen de numerieke oplossing y_{n+1} en de gezochte oplossing $y(x_{n+1})$ de zogenaamde *globale fout*, bestaat uit de fout $y_{n+1} - \tilde{y}_{n+1}$ en de *afbreekfout* $\tilde{y}_{n+1} - y(x_{n+1})$ die men zou krijgen wanneer men de numerieke integratieformule zou toepassen in het punt $(x_n, y(x_n))$. De afbreekfout wordt in belangrijke mate door de orde van nauwkeurigheid van het schema bepaald en mag in elk geval verwacht worden voldoende klein te zijn voor $h \to 0$. De tweede component in de globale fout zullen we nu wat nader bekijken. Laten we $y(x_n)$ even met $\tilde{y}_n$ aangeven en alle met bovengenoemde rekenschema's hieruit gevonden tussenpunten met $\tilde{y}^{(j)}_{n+1}$. In het geval (3.4.5) met $m = 3$ vinden we dan

$$
\begin{aligned}
y^{(1)}_{n+1} - \tilde{y}^{(1)}_{n+1} &= y_n - \tilde{y}_n + \lambda_{10} h[f(y_n) - f(\tilde{y}_n)] , \\
y^{(2)}_{n+1} - \tilde{y}^{(2)}_{n+1} &= y_n - \tilde{y}_n + \lambda_{21} h[f(y^{(1)}_{n+1}) - f(\tilde{y}^{(1)}_{n+1})] , \\
y_{n+1} - \tilde{y}_{n+1} &= y_n - \tilde{y}_n + h[f(y^{(2)}_{n+1}) - f(\tilde{y}^{(2)}_{n+1})] .
\end{aligned}
\tag{3.4.8}
$$

Voor voldoend kleine waarden van de fout $y_n - \tilde{y}_n$ kan dit geschreven worden als

$$(3.4.8') \qquad y_{n+1} - \tilde{y}_{n+1} \simeq \left(1+hJ^{(2)}_{n+1}(1+\lambda_{21}hJ^{(1)}_{n+1}(1+\lambda_{10}hJ^{(0)}_{n+1}))\right)(y_n-\tilde{y}_n),$$

waarin $J^{(j)}_{n+1}$ de Jacobiaan van f is in het punt $y^{(j)}_{n+1}$. Behalve het klein zijn van $y_n - \tilde{y}_n$ veronderstellen we nu ook nog dat $J^{(j)}_{n+1}$ voor alle j dichtbij een alleen van n afhangende matrix J_n ligt. Dit betekent dat (3.4.8') overgaat in

$$(3.4.8'') \qquad y_{n+1} - \tilde{y}_{n+1} = P_3(hJ_n)(y_n-\tilde{y}_n),$$

waarin P_3 een derdegraads polynoom in J_n is gegeven door

$$(3.4.9) \qquad P_3(z) = 1 + z + \lambda_{21}z^2 + \lambda_{10}\lambda_{21}z^3.$$

$P_3(z)$ wordt het stabiliteitspolynoom van het rekenschema genoemd. Indien men $\|y_{n+1}-\tilde{y}_{n+1}\|$ niet groter dan $\|y_n-\tilde{y}_n\|$ wil laten worden, dan moet men de nog vrije parameters h,λ_{10} en λ_{21} zo trachten te kiezen dat $\|P_3(hJ_n\|$ kleiner of gelijk 1 is. Een *nodige* voorwaarde daartoe is dat alle al eerder geintroduceerde amplificatiefactoren $P_3(h\delta)$, δ eigenwaarde van J_n, binnen of op de eenheidscirkel liggen. In ons geval, waarin de differentiaalvergelijking hyperbolisch is en dus de eigenwaarden δ imaginair zijn, betekent dit dat $P_3(z)$ op een zeker segment van de imaginaire as waarden binnen of op de eenheidscirkel moet aannemen. Stel dat dit segment het interval $[-i\beta,i\beta]$, β positief, is dan geldt dat de amplificatiefactoren $|P_3(h\delta)| \leq 1$ zijn wanneer (stabiliteitsvoorwaarde)

$$(3.4.10) \qquad h \leq \frac{\beta}{|\delta|_{max}}.$$

Voor hyperbolische vergelijkingen, en algemener voor semi-gediscretiseerde partiele differentiaalvergelijkingen, is $|\delta|_{max}$ zeer groot zodat men $P_3(z)$ zodanig tracht te kiezen dat β maximaal is. Zo niet dan zou voorwaarde (3.4.10) veel kleinere stappen voorschrijven dan de nauwkeurigheid voorschrijft. Het is een niet al te moeilijke opgave om te bewijzen dat het polynoom

$$(3.4.11) \qquad P_3(z) = 1 + z + \tfrac{1}{2}z^2 + \tfrac{1}{4}z^3$$

van alle polynomen van de vorm (3.4.9) de grootste β oplevert en wel $\beta = 2$. De parameters λ_{10} en λ_{21} volgen direct uit identificatie van (3.4.9) en (3.4.11):

$$\lambda_{10} = \lambda_{21} = \tfrac{1}{2}. \tag{3.4.12}$$

De hierboven geschetste constructie is algemeen toepasbaar. Zo verkrijgen we altijd de "variatie-vergelijking"

$$y_{n+1} - \tilde{y}_{n+1} \cong P_m(hJ_n)(y_n - \tilde{y}_n),$$

waarin P_m een polynoom van de graad m in z is waarvan de begintermen gegeven worden door

$$P_m(z) = 1 + z + \ldots + \frac{1}{j!} z^j + \ldots + \frac{1}{p!} z^p + \ldots , \quad p = \text{orde},$$

en waarin de coefficienten van de hogere graadstermen van de Runge-Kutta-parameters λ_{jl} afhangen. Zodra het optimaliseringsprobleem voor $P_m(z)$ opgelost is, d.w.z. het maximaliseren van het imaginaire interval waarop $P_m(z)$ waarden binnen of op de eenheidscirkel aanneemt, kunnen de Runge-Kutta-parameters bepaald worden en ligt het rekenschema vast. Voor *p = 1* en *p = 2* zijn de optimale polynomen voor *oneven* waarden van m in gesloten vorm bekend en blijken identiek te zijn. Ze worden gegeven door

$$P_m(z) = T_k\left(1 + \frac{z^2}{2k^2}\right) + \frac{z}{k}\left(1 + \frac{z^2}{4k^2}\right) U_{k-1}\left(1 + \frac{z^2}{2k^2}\right), \quad m = 2k+1, \tag{3.4.13}$$

waarin T_k en U_k Chebyshev-polynomen van eerste en tweede soort zijn. De β-waarde voor deze polynomen wordt gegeven door

$$\beta = m-1. \tag{3.4.14}$$

Voor *even* waarden van m zijn analytisch slechts enkele lagere graads polynomen bekend. Verdere details van deze optimaliseringsproblemen kan men in VAN DER HOUWEN [1977] vinden.

De voorwaarde voor de staplengte h in schema's gegenereerd door de polynomen (3.4.13) wordt krachtens (3.4.10):

$$h < \frac{m-1}{|\delta|_{max}} \quad . \tag{3.4.10'}$$

Hieruit volgt dat de "staplengte per functie-evaluatie" gegeven wordt door $(m-1)/m|\delta|_{max}$. Dus voor m = 5 is 80% van de maximaal haalbare staplengte per functie-evaluatie bereikt. In de praktijk zal men m dan ook meestal de waarde 5 of 7 geven, omdat hogere waarden nauwelijks tot grotere *effectieve* stappen leidt.

In verband met de stabiliteitsvoorwaarde (3.4.10') merken we op dat in gevallen waarin de *Jacobiaan* $\partial f/\partial y$ *beschikbaar is en zonder al te veel rekenwerk te evalueren is,* een kleine modificatie van de schema's (3.4.5)-(3.4.7) tot schema's leidt waar de effectieve integratiestap wel snel met m toeneemt. De bedoelde modificaties zijn de volgende: in geval van schema (3.4.5)

$$
\begin{aligned}
y_{n+1}^{(1)} &= y_n + \lambda_{10} hJ_n y_n, \\
y_{n+1}^{(j)} &= y_n + \lambda_{jj-1} hJ_n y_{n+1}^{(j-1)}, \qquad j = 2,\ldots,m-1, \\
y_{n+1} &= y_n + hf(y_{n+1}^{(m-1)}),
\end{aligned}
\tag{3.4.15}
$$

waarin J_n een benadering is voor de Jacobianen $J_{n+1}^{(j)}$; schema (3.4.6) gaat over in

$$
\begin{aligned}
y_{n+1}^{(1)} &= y_n + \lambda_{10} hJ_n y_n \\
y_{n+1}^{(j)} &= y_n + \lambda_{jj-1} hJ_n y_{n+1}^{(j-1)}, \qquad j = 2,\ldots,m-2 \\
y_{n+1}^{(m-1)} &= y_n + \tfrac{1}{2} hf(y_{n+1}^{(m-2)}), \\
y_{n+1} &= y_n + hf(y_{n+1}^{(m-1)});
\end{aligned}
\tag{3.4.16}
$$

en tenslotte de modificatie van schema (3.4.7)

$$
\begin{aligned}
y_{n+1}^{(1)} &= y_n + \lambda_{10} hf(y_n), \\
y_{n+1}^{(j)} &= y_n + \tfrac{1}{4} hf(y_n) + \lambda_{jj-1} hJ_n y_{n+1}^{(j-1)}, \qquad j = 2,3,\ldots,m-3 \\
y_{n+1}^{(m-2)} &= y_n + \tfrac{1}{4} hf(y_n) + \frac{17}{60} hf(y_{n+1}^{(m-3)}), \\
y_{n+1}^{(m-1)} &= y_n + \tfrac{1}{4} hf(y_n) + \frac{5}{12} hf(y_{n+1}^{(m-2)}), \\
y_{n+1} &= y_n + \tfrac{1}{4} hf(y_n) + \tfrac{3}{4} hf(y_{n+1}^{(m-1)}).
\end{aligned}
\tag{3.4.17}
$$

Deze gemodificeerde schema's zijn respectievelijk eerste, tweede en derde orde nauwkeurig voor *elke gekozen benadering* J_n *van de Jacobiaan* $\partial f/\partial y$. De stabiliteitsvoorwaarden zijn echter alleen gelijk aan die van de oorspronkelijke schema's wanneer J_n een *redelijk goede benadering* voor de Jacobiaan is. Wat de bewerkelijkheid betreft, bovenstaande modificaties vragen respectievelijk 1, 2 en 4 rechterlid-evaluaties, een aantal matrix-vermenigvuldigingen en "zo nu en dan" de evaluatie van de Jacobiaan. Voor numerieke experimenten met de in deze paragraaf besproken methoden verwijzen we naar paragraaf 3.4.3.

3.4.2. Hyperbolische vergelijkingen van de tweede orde

Voor vergelijkingen van de vorm (3.4.2) komen formules van de vorm (1.1.2) uit hoofdstuk 1 in aanmerking. In het bijzonder zullen we de volgende klassen van algoritmen beschouwen: voor p = 1 (weer in autonome vorm geschreven)

$$
\begin{aligned}
y_{n+1}^{(1)} &= y_n + \mu_1 h y_n' \\
y_{n+1}^{(j)} &= y_n + \mu_j h y_n' + \lambda_{jj-1} h^2 f(y_{n+1}^{(j-1)}), \qquad j = 2,3,\ldots,m, \\
y_{n+1}^{(m-1)} &= y_n + \tfrac{1}{2} h y_n' + \lambda_{m-1m-2} h^2 f(y_{n+1}^{(m-2)}), \qquad (3.4.18) \\
y_{n+1} &= y_n + h y_n' + \lambda_{mm-1} h^2 f(y_{n+1}^{(m-1)}), \\
y_{n+1}' &= y_n' + h f(y_{n+1}^{(m-1)});
\end{aligned}
$$

en voor p = 2 hetzelfde schema waarin

$$\lambda_{mm-1} = \tfrac{1}{2} \qquad (3.4.19)$$

gesteld is. Derde orde schema's zijn nog onderwerp van onderzoek op het Mathematisch Centrum (zie GERRITSEN [1977]).

In bovenstaand rekenschema zijn de parameters μ_j en λ_{jj-1}, $j = 1,\ldots,m-2$ ter beschikking om de stabiliteit op te voeren. Op dezelfde wijze als in paragraaf 3.4.1, kunnen we verstoringen van y_{n+1} en y_{n+1}' uitdrukken in de verstoringen van y_n en y_n'. Zoals in hoofdstuk 1 al aangestipt, vinden we een relatie van de vorm

$$(3.4.20)\qquad \begin{pmatrix} y_{n+1} - \tilde{y}_{n+1} \\ h(y'_{n+1} - \tilde{y}'_{n+1}) \end{pmatrix} = R^{(m)}(h^2 J_n) \begin{pmatrix} y_n - \tilde{y}_n \\ h(y'_n - \tilde{y}'_n) \end{pmatrix} ,$$

waarin $R^{(m)}$ een (2*2)-matrix-functie is waarvan de elementen polynomen van de graad m zijn (zie paragraaf 1.3), en waarin J_n weer een benadering van $\partial f/\partial y$ voorstelt. Het optimaliseringsprobleem bestaat nu uit het kiezen van de nog vrije parameters μ_j en λ_{jj-1} zodanig dat de eigenwaarden van de matrices $R^{(m)}(h^2\delta)$, waarin δ de (negatieve) eigenwaarden van J_n doorloopt, binnen of op de eenheidscirkel liggen voor de grootst mogelijke waarde van h. Dit probleem kan analytisch opgelost worden (zie VAN DER HOUWEN [1976]). Zowel voor p = 1 als p = 2 vindt men

$$(3.4.21)\qquad h < 2\,\frac{m-1}{\sqrt{|\delta|_{max}}} .$$

Om dit resultaat te kunnen interpreteren, gaan we na welke integratiestappen genomen zouden kunnen worden wanneer we vergelijking (3.4.2) eerst tot eerste orde vorm gereduceerd zouden hebben (zie (3.4.3)) en hierop de methoden uit de vorige paragraaf zouden toepassen. Omdat volgens (3.4.4) de eigenwaarden van het gereduceerde eerste orde stelsel de wortels zijn uit de eigenwaarden δ van de tweede orde vergelijking, leidt voorwaarde (3.4.10') voor de schema's (3.4.4) en (3.4.5) tot

$$(3.4.10'')\qquad h < \frac{m-1}{\sqrt{|\delta|_{max}}} , \qquad m \text{ oneven.}$$

Bedenken we verder dat deze schema's m rechterlid evaluaties vergen, en schema (3.4.18) slechts m-1 evaluaties, dan laat laatstgenoemd rekenschema *meer dan tweemaal grotere integratiestappen* toe.

Deze winst wordt echter enigszins te niet gedaan door een andere, minder prettige eigenschap van schema (3.4.18). Berekent men namelijk de eigenwaarden $\alpha(h^2\delta)$ van $R^{(m)}(h^2\delta)$, dan blijkt dat deze allen precies op de eenheidscirkel liggen; derhalve wordt geen enkele component van de eigenvectorontwikkeling van de verstoring van (y_n, hy'_n) in het interval $[-h_{max}|\delta|_{max}, 0]$ uitgedempt. Dit was de aanleiding om een gewijzigd optimaliseringsprobleem te formuleren waarin gevraagd wordt de Runge-Kuttaparameters zo te kiezen dat voor de grootst mogelijke h (aan te geven met h_{max}) de eigenwaarden $\alpha(h^2\delta)$ binnen of op een cirkel met straal $\sqrt{\rho(h^2\delta)}$ liggen waarin ρ een zekere

functie is die op het interval $[-h^2|\delta|_{max},0]$ waarden kleiner dan 1 aanneemt. Wanneer ρ niet te ver van 1 afwijkt, is dit optimaliseringsprobleem in goede benadering op te lossen. We vermelden hier een eerste en een tweede orde formule met respectivelijk m = 2 en m = 3:

(3.4.22) $p = 1$: $y_{n+1} = y_n + hy'_n + \frac{1}{2}\frac{4-\varepsilon}{4-3\varepsilon}h^2 f(y_n+\frac{1}{2}hy'_n)$,

$$y'_{n+1} = y'_n + hf(y_n+\tfrac{1}{2}hy'_n),$$

$$h < h_{max} = \sqrt{\frac{4-3\varepsilon}{|\delta|_{max}}}\ ,\ \rho(h^2\delta) = 1 + \frac{\varepsilon}{4-3\varepsilon}h^2\delta;$$

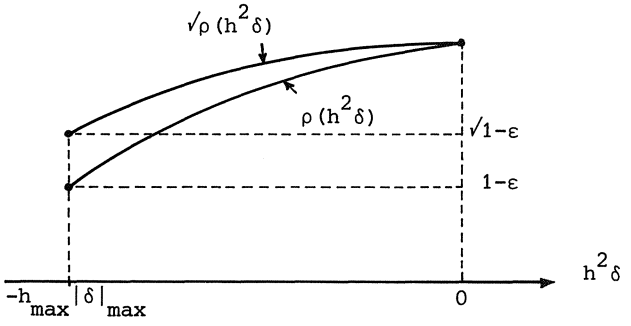
(3.4.23) $p = 2$: $y_{n+1} = y_n + hy'_n + \frac{1}{2}h^2 f(y_n+\frac{1}{2}hy'_n+(\sigma_2-\pi_2)h^2 f(y_n+\frac{1}{2}\frac{\sigma_2+\pi_2}{\sigma_2-\pi_2}hy'_n))$,

$$y'_{n+1} = 2\frac{y_{n+1}-y_n}{h} - y'_n\ ,$$

$$\sigma_2 = \frac{\beta-2\varepsilon}{\beta^2}\ ,\ \pi_2 = -\frac{\varepsilon}{\beta^2}\ ,\ \beta = 8(1+\sqrt{1-\varepsilon}).$$

$$h < h_{max} = \sqrt{\frac{\beta}{|\delta|_{max}}}\ ,\ \rho(h^2\delta) = 1 - \frac{\varepsilon}{\beta^2}h^4\delta^2.$$

De parameter ε geeft de afwijking van 1 van de functie ρ in het punt $-h^2_{max}|\delta|_{max}$ aan (zie figuur 3.4.2). Hieruit valt te concluderen dat voor relatief kleine concessies aan de maximale integratiestap een aanvaardbare demping verkregen kan worden (ε=1/10 of 1/5).



Figuur 3.4.2. Functies $\rho(h^2\delta)$ en $\sqrt{\rho(h^2\delta)}$

Tenslotte merken we op dat ook hier een soortgelijke modificatie mogelijk is als beschreven in paragraaf 3.4.1; zonder aantasting van de *orde* van nauwkeurigheid mag men alle rechterlid-evaluaties $(f(y_{n+1}^{(j-1)})$ in schema (3.4.18) welke door een nog vrije parameter λ_{jj-1} vooraf wordt gegaan, vervangen door $J_n y_{n+1}^{(j-1)}$.

3.4.3. Numerieke experimenten

Aan de hand van het beginwaardeprobleem

$$\frac{\partial^2 u}{\partial t^2} = gh\,\frac{\partial^2 u}{\partial x^2} + \tfrac{1}{4}\lambda^2 u + e^{\frac{1}{2}\lambda t}w,\quad g = 9.81,\quad \lambda = 25_{10}{}^{-6}$$

(3.4.24)

$$u(0,x) = \frac{\partial u}{\partial t}(0,x) = \frac{\partial u}{\partial x}(t,0) = \frac{\partial u}{\partial x}(t,b) = 0$$

welke de waterhoogten $z = -e^{-\frac{1}{2}\lambda t}u$ in een rivier met lengte b en diepte h(x) beschrijft tengevolge van een windveld w(t,x), zullen we enkele van de in voorgaande paragrafen besproken formules illustreren. In tabel 3.4.1 zijn de aantallen *correcte cijfers* van de numeriek berekende waarden voor z, en tussen haakjes het aantal benodigde *rechterlid-evaluaties*, opgenomen, welke verkregen zijn door toepassing van de formules op probleem (3.4.24), waarin $\partial^2/\partial x^2$ door centrale differenties vervangen zijn en waarin

$$(3.4.25)\qquad h(x) = 10(2+\cos(\frac{2\pi x}{10^5})),\ w(t,x) = 10^{-3}\sin(\frac{\pi x}{10^5}),\ b = 10^5.$$

Deze resultaten werden door P.A. Beentjes en W.J. Gerritsen verkregen.

Bij de specificatie van de gebruikte formule is het aantal rechterlid-evaluaties per stap met $\tilde{m}$ aangegeven; de gebruikte integratiestap Δt kan dan berekend worden door $3600\tilde{m}$ te delen door het totaal benodigde functie-evaluaties zoals dit in de tabel achter het aantal correcte cijfers tussen haakjes aangegeven is. Het eerst genoemde aantal correcte cijfers voor iedere formule is berekend met de grootste nog stabiele integratiestap. Tenslotte merken we nog op dat de referentie-oplossingen van de semi-gediscretiseerde stelsels differentiaalvergelijkingen verkregen zijn door toepassing van een hogere orde Runge-Kuttaformule met $\Delta t = 100$ voor $\Delta x = 10^4$ en $\Delta t = 10$ voor $\Delta x = 10^3$.

Tabel 3.4.1. Aantallen correcte cijfers in z en aantallen benodigde functie-evaluaties ten tijde t = 3600 voor 2 waarden van de discretisatieparameter Δx

Formule	p	$\Delta x = 10^{-4}$		$\Delta x = 10^{-3}$	
(3.4.5) , $m = \tilde{m} = 2$	1	0.8(26)		1.8(248)	
(3.4.6) , $m = \tilde{m} = 3$	2	2.3(21)		4.2(186)	
(3.4.6) , $m = \tilde{m} = 5$	2	1.9(21)		4.2(156)	
(3.4.6) , $m = \tilde{m} = 7$	2	1.9(22)		4.2(148)	
(3.4.16) , $m = 5, \tilde{m} = 2$	2	0.7(8)		2.8(62)	
(3.4.16) , $m = 7, \tilde{m} = 2$	2	0.7(6)		2.8(42)	
(3.4.22) , $\tilde{m} = 1, \varepsilon = .1$	1	1.8(6)	2.1(13)	2.8(65)	3.1(124)
(3.4.23) , $\tilde{m} = 2, \varepsilon = .1$	2	2.4(8)	3.1(14)	4.4(64)	5.0(124)

LITERATUUR

BAKKER, M., e.a. [1976], *Colloquium discretiseringsmethoden,* MC Syllabus 27, Mathematisch Centrum, Amsterdam.

GEAR, C.W. [1964], *Hybrid methods for initial value problems in ordinary differential equations,* J. SIAM Numer. Anal., Ser. B, 2, 69-86.

GEAR, C.W. [1971], *Numerical initial value problems in ordinary differential equations,* Prentice Hall, Englewood Cliffs, New Jersey.

GERRITSEN, W.J. [1977], *Experiments with stabilized Runge-Kutta methods for second order differential equations without first derivatives,* te verschijnen in de NW-serie van het Mathematisch Centrum, Amsterdam.

HENRICI, P. [1962], *Discrete variable methods in ordinary differential equations,* John Wiley & Sons, New York.

LAMBERT, J.D. [1973], *Computational methods in ordinary differential equations,* John Wiley & Sons, London.

LINDBERG, B. [1972], *Impex, A program package for solution of systems of stiff differential equations,* Report NA 72-50, Department of Information Processing, The Royal Institute of Technology, Stockholm.

NORDSIECK, A. [1962], *On the numerical integration of ordinary differential equations,* Math. Comp. 16, 22-49.

RAVIART, P.A. [1973], *The use of numerical integration in finite element methods for solving parabolic equations,* in: MILLER, J.J.H. (ed.), *Topics in numerical analysis,* Proceedings of the Royal Irish Academy Conference on Numerical Analysis 1972, Academic Press, London.

RICHTMEYER, R.D. & K.W. MORTON [1967], *Difference methods for initial value problems,* Interscience, New York.

SINCOVEC, R.F. & N.K. MADSEN [1975], *Software for nonlinear partial differential equations,* ACM Transactions on mathematical software 1, 232-260.

STRANG, G. & G.J. FIX [1973], *An analysis of the finite element method,* Prentice-Hall, Englewood Cliffs, New Jersey.

VAN DER HOUWEN, P.J. [1976], *Stabilized Runge-Kutta methods for second order differential equations without first derivatives,* Report NW 26/76, Mathematisch Centrum, Amsterdam.

VAN DER HOUWEN, P.J. [1977], *Construction of integration formulas for initial value problems,* North-Holland Publishing Company, Amsterdam.

VERWER, J.G. [1976a], *Multipoint multistep Runge-Kutta methods I: On a class of two-step methods for parabolic equations,* Report NW 30/76, Mathematisch Centrum, Amsterdam.

VERWER, J.G. [1976b], *Multipoint multistep Runge-Kutta methods II: The construction of a class of stabilized three-step methods for parabolic equations,* Report NW 31/76, Mathematisch Centrum, Amsterdam.

WATT, J.M. [1967], *The asymptotic discretization error of a class of methods for solving ordinary differential equations,* Proc. Camb. Phil. Soc. 61, 461-472.

4. LINEAIRE PROGRAMMERING

door J.M. Anthonisse
(Mathematisch Centrum)

Samenvatting

Lineaire programmering (LP) houdt zich bezig met het optimaliseren van een lineaire criteriumfunctie onder een aantal lineaire restricties, waarbij de variabelen in het algemeen slechts niet-negatieve waarden mogen aannemen. Vele practische beslissingssituaties blijken als LP probleem te kunnen worden geformuleerd, een optimale oplossing van het LP probleem is dan een basis voor de te nemen beslissing. LP problemen treden ook op in een meer theoretische context, zoals bij het benaderen van functies, of als sub-probleem bij het oplossen van niet-lineaire programmeringsproblemen.

De meest bekende en gebruikte techniek voor het oplossen van LP problemen is de simplexmethode. GASS [1975] geeft een inleiding tot de gehele theorie. ANTHONISSE c.s. [1973] beschrijven de methode zeer beknopt, DEKKER [1971] behandelt de numerieke aspecten van enkele varianten van de simplexmethode.

In de literatuur zijn diverse routines voor het oplossen van LP problemen gepubliceerd: JOSEFSEN [1964], AIRD [1966], SALASAR & SEN [1968], BARTELS c.s. [1969], KRONSJO [1970], BARTELS c.s. [1971], LAND c.s. [1973]. De vier eerstgenoemde routines zijn door BROCKLEHURST & DENNIS [1974] onderzocht. Het voordeel van in ALGOL of FORTRAN gepubliceerde routines is, dat ze gemakkelijk kunnen worden ingepast in een programmatheek en in andere routines. Ook kunnen ze aan speciale wensen worden aangepast. Nadelen zijn echter, dat zij niet expliciet voor het oplossen van omvangrijke problemen zijn ontwikkeld en daarvoor dan ook ongeschikt zijn en dat de ervaring ermee doorgaans gering is. Ook de verzorging van in- en uitvoer laat in het algemeen te wensen over.

De pakketten APEX III van Control Data en MPSX van IBM zijn voorbeelden van systemen, die speciaal voor het goedkoop en routinematig oplossen

van in de practijk optredende, vaak grote, LP problemen zijn ontwikkeld. Zij worden op ruime schaal gebruikt. Dit soort pakketten biedt de gebruiker vele opties, zowel met betrekking tot invoerverzorging, probleemdefinitie, oplossingsstrategie als uitvoerverzorging. De routines zijn echter black boxes, die ook niet of nauwelijks in andere routines kunnen worden ingepast.

De in SARA-verband toegankelijke programmatheken CERN, IMSL, NAG en OPERAL (van de afdeling Mathematische Besliskunde van het Mathematisch Centrum) bevatten routines voor het oplossen van LP problemen.

Literatuur

AIRD, T.J. [1966], *The mutual primal-dual method,* Comm. ACM 9, 326.

ANTHONISSE, J.M. & J.K. LENSTRA [1973], *Lineaire programmering,* Rapport BC 7/73, Mathematisch Centrum, Amsterdam.

BARTELS, R.H. & G.H. GOLUB [1969], *Simplex method procedure employing LU decomposition,* Comm. ACM 12, 275-278.

BARTELS, R.M., J. STOER & CH. ZENGER [1971], *A realization of the simplex method based on triangular decompositions,* in: WILKINSON, J.H. & C. REINSCH (eds.), *Linear Algebra,* Springer, Berlin.

BEALE, E.M.L. [1971], *Mathematical programming in practice,* Pitman, London.

BEALE, E.M.L. [1975], *The current algorithmic scope of mathematical programming,* in: BALINSKI, M.L. & E. HELLERMAN (eds.), *Mathematical Programming Study 4,* North-Holland, Amsterdam.

BROCKLEHURST, E.R. & K. DENNIS [1974], *A comparison of six algorithms for dense linear programs,* NPL Report NAC 51.

DEKKER, T.J. [1971], *Numerieke Algebra,* MC Syllabus 12, Mathematisch Centrum, Amsterdam.

GASS, S.I. [1975], *Linear programming,* McGraw-Hill, New York.

JOSEFSEN, H. [1964], *Linear programming by the modified simplex method,* BIT 4, 189-196.

KRONSJO, T.O.M. [1970], *An illustrative primal simplex linear program,* Computer Journal 13, 426-437.

LAND, A.H. & S. POWELL [1973], *FORTRAN codes for mathematical programming; linear, quadratic and discrete,* Wiley, London.

SALAZAR, R.C. & S.K. SEN [1968], *Minit algorithm for linear programming,* Comm. ACM 11, 437-440.

WHITE, W.W. [1973], *A status report on computing algorithms for mathematical programming,* Computing Surveys 5, 135-166.

5. NONLINMIN, EEN PROCEDURE VOOR HET MINIMALISEREN VAN NIET-LINEAIRE FUNCTIES ONDER NIET-LINEAIRE NEVENVOORWAARDEN

door J.L. de Jong
(Technische Hogeschool Eindhoven)

5.1. Inleiding

De bestaande methoden voor het numeriek oplossen van niet-lineaire optimaliseringsproblemen met nevenvoorwaarden kunnen ruwweg worden ingedeeld in drie categorieën (vgl. LUENBERGER [1973]), t.w. boetefunctiemethoden, primale methoden en duale of Lagrangemethoden. De eerste categorie bestaat uit die methoden, waarbij geprobeerd wordt aan de beperkingen te voldoen door aan het overschrijden van de beperkingen "boeten" op te leggen, die opgeteld worden bij de te minimaliseren objectfunctie. De tweede categorie betreft die methoden die iteratief nieuwe en betere schattingen van het optimum zoeken langs speciale aan de beperkingen aangepaste zoekrichtingen. De derde categorie omvat die methoden die expliciet gebruik maken van de eigenschappen van de Lagrangefunctie van het niet-lineaire minimaliseringsprobleem.

Voor het oplossen van niet-lineaire minimaliseringsproblemen werd enige jaren geleden door F.A. Lootsma de Algolprocedure MINFUN (LOOTSMA [1972]) ontwikkeld waarin diverse boetefunctiemethoden zijn gerealiseerd. Min of meer in navolging daarvan werd door C.J.B. Dirkx en de schrijver in 1975 een begin gemaakt met de ontwikkeling van de in deze voordracht te bespreken procedure NONLINMIN (DIRKX [1975]), waarbinnen op analoge wijze een aantal verschillende methoden uit de tweede categorie zijn samengebracht. Bij deze ontwikkeling werd in het bijzonder gestreefd naar een opzet waarbij enerzijds de gebruiker de keuzemogelijkheid heeft uit een aantal verschillende algoritmen, terwijl anderzijds deze algoritmen zoveel mogelijk gebruik maken van dezelfde hulpprocedures. Een dergelijke opzet heeft als extra voordeel dat met slechts kleine aanpassingen andere soortgelijke algoritmen kunnen worden ingepast en dat nieuwe ideeën kunnen worden uitgeprobeerd op een

goed vergelijkbare manier.

De procedure NONLINMIN is op dit moment nog slechts beperkt operationeel. Gereed zijn alle (hulp-)procedures waaruit de procedure NONLINMIN zelf weer is opgebouwd. Sommige van deze bestaan echter nog slechts in hun meest elementaire vorm. Gegeven de modulaire opbouw van het geheel, is het mogelijk deze (hulp-)procedures nog individueel te verfijnen en daarmee de totale procedure naar wens en behoefte uit te bouwen.

In de hierna te geven beschrijving zal eerst worden ingegaan op de aan de totale procedure ten grondslag liggende algoritme. Vervolgens zal aandacht worden besteed aan de theoretische achtergrond van enkele stappen binnen deze algoritme en tenslotte zal een korte discussie worden gewijd aan de practische uitvoering van diezelfde stappen. Voor programmatische details, listings etc. van de procedure moet worden verwezen naar een in de toekomst nog te schrijven handleiding voor het gebruik van de procedure.

5.2. Probleemstelling en algoritme

5.2.1. Probleem, notatie en definities

De hierna te beschrijven Algolprocedure NONLINMIN heeft betrekking op het oplossen van niet-lineaire optimaliseringsproblemen van de vorm

$$(5.2.1) \qquad \min\{f(x) \mid c_j(x) = 0,\ j = 1,\ldots,m_1;\ c_j(x) \geq 0,\ j = m_1+1,\ldots,m,\ x \in \mathbb{R}^n\}$$

waar $f: \mathbb{R}^n \to \mathbb{R}^1$ en $c_j: \mathbb{R}^n \to \mathbb{R}^1$, $j = 1,\ldots,m$. Verondersteld wordt dat zowel $f \in C^1$ als $c_j \in C^1$, $j = 1,\ldots,m$. Een rol bij de beschrijving hierna spelen de volgende probleemgrootheden, genoteerd in een grotendeels uit GILL & MURRAY [1974] overgenomen notatie:

$\hat{x}$: optimale punt, oplossing van het probleem (5.2.1)

$x^{(k)}$: k-de benadering voor $\hat{x}$, startpunt (k+1)-de iteratiestap

$g := \nabla f(x) = (\frac{\partial f}{\partial x_1} \ldots \frac{\partial f}{\partial x_n})^T$: gradiënt van de objectfunctie

$g^{(k)}$: gradiënt van de objectfunctie in het punt $x^{(k)}$

$d^{(k)}$: zoekrichting vanuit $x^{(k)}$

$\alpha^{(k)}$: stapgrootte(factor) in de richting $d^{(k)}$

$G(x) := \nabla^2 f(x)$: Hessiaan van de objectfunctie

$a_j^{(k)} := \nabla c_j(x^{(k)})$: gradiënt van de j-de beperking in het punt $x^{(k)}$

$A^{(k)} := [a_1^{(k)} \ldots a_m^{(k)}]$: $n \times m$-matrix met als kolommen de gradiënten van de beperkingen in het punt $x^{(k)}$

$N^{(k)}$:= $[n_1^{(k)} \ldots n_q^{(k)}]$: n × q-matrix met als kolommen q gradiënten (of normalen) van actieve beperkingen in $x^{(k)}$ die samen een basis vormen voor de deelruimte opgespannen door de normalen van alle actieve beperkingen

$N^{(k)+}$:= $(N^{(k)T}N^{(k)})^{-1}N^{(k)T}$: pseudo-of gegeneraliseerde inverse van $N^{(k)}$

$P^{(k)}$:= $N^{(k)}N^{(k)+}$: matrix van projectie op de normalen van de actieve beperkingen

$\bar{P}^{(k)}$:= $I - N^{(k)}N^{(k)+}$: matrix van projectie op de doorsnede van de raakvlakken aan de actieve beperkingen

$\lambda^{(k)}$: j-de component van een benadering voor de oplossing van het stelsel $N^{(k)}\lambda - \nabla f(x^{(k)}) = 0$, of, equivalent, benadering voor de Lagrange multiplicator corresponderend met de j-de beperking in het optimale punt in de doorsnede van de actieve beperkingen in het punt $x^{(k)}$.

Van belang voor het navolgende zijn ook de volgende begrippen:

- Een punt x heet een *toegelaten* (feasible) *punt* als in x aan alle beperkingen voldaan wordt, d.w.z. indien geldt

$$c_j(x) = 0 \quad (j=1,\ldots,m_1) \qquad c_j(x) \geq 0 \quad (j=m_1+1,\ldots,m).$$

- Een beperking $c_j(x)$ heet in het punt x respectievelijk

actief	als	$c_j(x) = 0$	en	$j \in \{1,\ldots,m\}$
passief	als	$c_j(x) > 0$	en	$j \in \{m_1+1,\ldots,m\}$
overschreden	als	$c_j(x) \neq 0$	en	$j \in \{1,\ldots,m_1\}$ of $c_j(x) < 0$ en $j = \{m_1+1,\ldots,m\}$.

- Coördinaatbeperkingen van de vorm

$$a_i \leq x_i \quad \text{of} \quad x_i \leq b_i ,$$

waarvan de gradiënten de corresponderende eenheidsvectoren zijn, worden *triviale beperkingen* genoemd.

5.2.2. Basisalgoritme

Bij de meeste primale methoden bestaat de eerste stap naar de oplossing van het probleem (5.2.1) uit het bepalen van een toegelaten startpunt. In dat punt wordt gekeken welke beperkingen actief zijn, waarna de objectfunctie wordt geminimaliseerd over de doorsnede van de raakvlakken aan

deze actieve beperkingen. De passieve beperkingen leveren een begrenzing van het zoekgebied. Er zijn dan 2 mogelijkheden: ofwel het minimum wordt aangenomen op de rand van het zoekgebied, ofwel het minimum wordt gevonden in de doorsnede van de raakvlakken aan de actief veronderstelde beperkingen. In het nieuw gevonden punt worden de beperkingen geëvalueerd en wordt nagegaan of het gevonden punt een toegelaten punt is. Zo nee, dan wordt uitgaande van het gevonden punt een nieuw toegelaten punt gegenereerd (dit proces wordt het *restauratieproces* genoemd). Zo ja, en dat is steeds (bij benadering) het geval bij uitsluitend lineaire beperkingen, dan wordt opnieuw vooraan begonnen. Kan daarna geen verbetering meer worden bewerkstelligd doordat in het betreffende punt het minimum gevonden wordt van de restrictie van de objectfunctie tot de doorsnede van de actieve beperkingen, dan wordt gekeken of het mogelijk is de objectfunctie verder te verlagen door het passief veronderstellen van een van de actieve beperkingen. Als dit niet kan dan is de oplossing van het probleem gevonden; als het wel kan, dan wordt verder gezocht over de doorsnede van de nieuwe verzameling actieve beperkingen.

Dit verbaal beschreven proces wordt in meer detail weergegeven door de volgende basisalgoritme voor de primale methoden in de procedure NONLINMIN:

(0) Zet $\bar{x}^{(0)}$:= gegeven startpunt; zet k := 0.

(i) Met $\bar{x}^{(k)}$ als uitgangspunt bepaal een toegelaten punt $x^{(k)}$, bepaal welke beperkingen actief zijn in $x^{(k)}$ en evalueer de corresponderende normalen $a_j^{(k)} := \nabla c_j(x^{(k)})$.

(ii) Bepaal de functiewaarde $f(x^{(k)})$, de gradiënt $g^{(k)}$ en zo nodig de Hessiaan $G^{(k)}$ van de objectfunctie in $x^{(k)}$.

(iii) Bepaal een met de actieve beperkingen corresponderende matrix $N^{(k)}$ van lineair onafhankelijke normalen en de restrictie van de gradiënt tot de daarmee corresponderende doorsnede van raakvlakken, d.i. of de geprojecteerde gradiënt (vgl. (5.3.7))

$$\bar{P}^{(k)} \nabla f(x^{(k)})$$

of de gereduceerde gradiënt (vgl. (5.3.24))

$$\bar{R}^{(k)} \nabla f(x^{(k)}).$$

(iv) Ga na of het punt $x^{(k)}$ het optimale punt is in de huidige actieve set; zo nee, dan ga door naar stap (v) en start een nieuwe iteratie; zo ja

dan evalueer het teken van de Lagrangemultiplicatoren die corresponderen met de actieve ongelijkheidsbeperkingen en die voldoen aan (5.3.16)

$$\lambda_j^{(k)} := ((N^{(k)T} N^{(k)})^{-1} N^{(k)T} \nabla f(x^{(k)}))_j$$

of (5.3.30)

$$\lambda_j^{(k)} := (B^{(k)-T} \nabla_{x_B} f(x^{(k)}))_j .$$

Als geen van deze $\lambda_j^{(k)}$ (met $j \in \{m_1+1,\ldots,m\} \cap I_A(x^{(k)})$ waar $I_A(x^{(k)})$ de indexverzameling is van de actieve beperkingen in $x^{(k)}$) negatief is dan *klaar*; zo niet, dan veronderstel een van de met de negatieve $\lambda_j^{(k)}$'s corresponderende beperkingen passief en ga terug naar stap (iii).

(v) Bepaal een nieuwe zoekrichting $d^{(k)}$.

(vi) Bepaal een maximale stapgroottefactor $\bar{\alpha}^{(k)}$.

(vii) Bepaal een stapgroottefactor $\alpha^{(k)}$ met $0 \leq \alpha^{(k)} \leq \bar{\alpha}^{(k)}$ zodat

$$f(x^{(k)}+\alpha^{(k)}d^{(k)}) < f(x^{(k)}). \qquad (5.2.2)$$

(viii) Bepaal een nieuw punt

$$\bar{x}^{(k+1)} := x^{(k)} + \alpha^{(k)}d^{(k)}, \qquad (5.2.3)$$

zet k := k+1 en ga terug naar stap (i).

Een flowdiagram van de op deze algoritme gebaseerde procedure NONLINMIN is weergegeven in Figuur 5.4.1 op blz.113. In deze figuur is bij alle subprocedures aangegeven met welke stap uit de hier gegeven algoritme deze corresponderen. In de volgende paragraaf wordt nader ingegaan op de details van de stappen (i), (v) en (vi).

Van belang om op te merken is dat de in stap (iv) tot uiting komende strategie voor de behandeling van de beperkingen in de literatuur bekend is als de *actieve-set-strategie* (vgl. GILL & MURRAY [1974], p. 50). Deze strategie bestaat daaruit dat men veronderstelt dat de beperkingen die actief zijn in $x^{(k)}$ ook daarna actief blijven en waarbij eerst dan een beperking uit de verzameling van actieve beperkingen wordt verwijderd wanneer het minimum over de doorsnede van de raakvlakken aan de actieve beperkingen is bereikt. Men voorkomt hiermee het zg. "zigzagging"- of "jamming"-verschijnsel, waarbij steeds tussen twee of meer dezelfde beperkingen op en neer

wordt gesprongen en dat zelfs convergentie naar een niet optimaal punt tot gevolg kan hebben.

5.3. Theoretische details

5.3.1. Bepaling van een toegelaten startpunt

Voor de bepaling van een toegelaten startpunt zijn een aantal methoden bekend in de literatuur: Indien alle beperkingen lineair zijn is het mogelijk gebruik te maken van de z.g. Fase 1-methode voor lineaire programmering (vgl. LUENBERGER [1973]). Indien ook niet-lineaire beperkingen aanwezig zijn, is het mogelijk op analoge wijze te werk te gaan door de minimalisering van in te voeren artificiële variabelen dan wel door de minimalisering van een kwadratische of andersoortige boetefunctie. (Voor deze onbeperkte minimaliseringsproblemen kan eventueel het bestaande programma zelf worden gebruikt). Voor het bepalen van een toegelaten startpunt in de procedure NONLINMIN werd een speciale algoritme ontwikkeld, die tevens kon worden gebruikt voor het bepalen van toegelaten punten in latere fasen van het minimaliseringsproces. Deze algoritme bestaat daaruit dat in een aantal opvolgende stappen telkens de minimum-norm kleinste-kwadraten oplossing wordt bepaald van het stelsel lineaire vergelijkingen gegenereerd door linearisatie van de lokale overschreden (eng.: violated) en actieve beperkingen in het laatste gevonden punt. Wordt dit stelsel

$$(5.3.1) \qquad c_j(\bar{x}^{(k)}) + \nabla c_j^T(\bar{x}^{(k)})(x-\bar{x}^{(k)}) = 0, \quad j \in (I_A(\bar{x}^{(k)}) \cup I_V(\bar{x}^{(k)})),$$

herschreven in de vorm

$$(5.3.2) \qquad \bar{A}^T \Delta x + \bar{c} = 0$$

dan kan de minimum-norm kleinste-kwadraten oplossing met behulp van de pseudo-inverse $(\bar{A}^T)^+$ van de matrix $\bar{A}^T$ worden weergegeven als

$$(5.3.3) \qquad \Delta\hat{x} = - (\bar{A}^T)^+ \bar{c}.$$

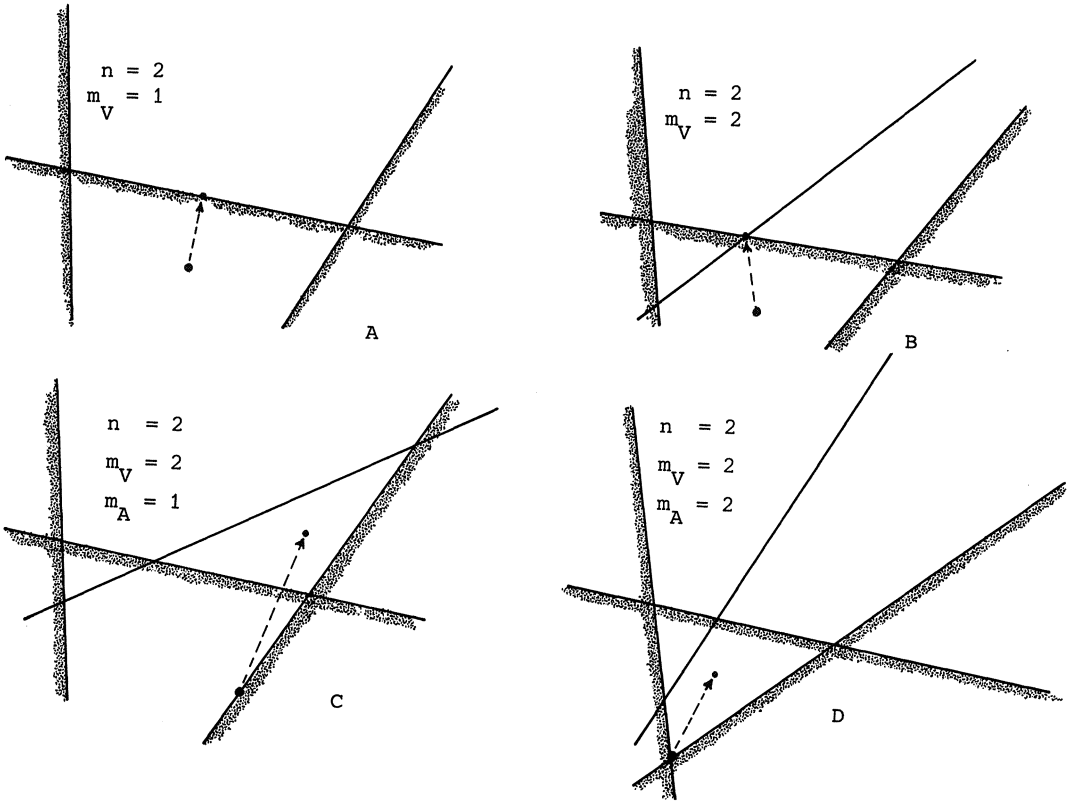
Als $\bar{A}^T$ een $(p \times n)$-matrix is en verondersteld wordt dat $\bar{A}^T$ maximale rang bezit dan volgt bij uitwerking van de pseudo-inverse in het geval dat $p < n$ (onderbepaald stelsel) dat

(5.3.3) $\quad \Delta\hat{\underline{x}} = - \underline{\bar{A}}(\underline{\bar{A}}^T\underline{\bar{A}})^{-1}c$

en in het geval dat p ≥ n (overbepaald stelsel) dat

(5.3.4) $\quad \hat{\underline{x}} = - (\underline{\bar{A}}\underline{\bar{A}}^T)^{-1}\underline{\bar{A}}c.$

In het eerste geval is de correctie een lineaire combinatie van de gradiënten van de beperkingen, in het tweede geval is de correctie zodanig dat slechts in kleinste-kwadraten zin aan de vergelijkingen wordt voldaan. Ten grondslag aan dit proces ligt de veronderstelling dat er na herhaalde bepaling van de minimum-norm kleinste-kwadraten oplossing een regulier of onderbepaald stelsel overblijft. De werking van de algoritme is geïllustreerd in Figuur 5.3.1 waarin voor het geval van $\mathbb{R}^2$ enkele verschillende mogelijkheden met betrekking tot de overschreden (m_V) en actieve (m_A) beperkingen zijn weergegeven.



Figuur 5.3.1. Voorbeelden van enige mogelijkheden voor niet toegelaten punten.

5.3.2. Eerste-orde zoekrichtingen

Uitgangspunt voor de afleiding van zowel eerste- als hoger-orde zoekrichtingen voor primale methoden is de lokale linearisatie van de actieve beperkingen in het toegelaten punt $x^{(k)}$ waarmee het originele probleem (5.2.1) overgaat in het probleem

(5.3.5) $\quad \min\{f(x) \mid N^{(k)T}(x-x^{(k)}) = 0,\ x \in \mathbb{R}^n\}.$

Voor dit basisprobleem bestaan drie equivalente formuleringen die aanleiding vormen voor evenzovele formuleringen voor eerste- en hoger-orde zoekrichtingen. De eerst-orde zoekrichtingen volgen bij minimalisering van de gelineariseerde objectfunctie over een hyperbol. De meest bekende eerste-orde zoekrichting volgt als oplossing van het probleem

(5.3.6) $\quad \min\{f(x^{(k)}+\nabla^T f(x^{(k)})(x-x^{(k)}) \mid N^{(k)T}(x-x^{(k)}) = 0,\ \|x-x^{(k)}\| \leq r\}$

en staat in de literatuur bekend als de *geprojecteerde-gradient* (van ROSEN [1960])

(5.3.7) $\quad d^{(k)} := -(I-N^{(k)}(N^{(k)T}N^{(k)})^{-1}N^{(k)T})\nabla f(x^{(k)}) =: -\bar{P}^{(k)}\nabla f(x^{(k)}).$

Een andere formulering van het basisprobleem (5.3.5) resulteert, wanneer gebruik wordt gemaakt van een $nx(n-q)$-matrix $Z^{(k)}$ waarvan de kolommen een orthonormale basis vormen van het orthogonale complement van de deelruimte opgespannen door de kolommen van $N^{(k)}$, d.i. de normalen van de actieve beperkingen in $x^{(k)}$. Voor deze matrix $Z^{(k)}$ geldt

(5.3.8) $\quad N^{(k)T}Z^{(k)} = O_{q\times(n-q)}$

en

(5.3.9) $\quad Z^{(k)T}Z^{(k)} = I_{(n-q)\times(n-q)}.$

Het basisprobleem (5.3.5) kan daarmee worden geschreven als een onbeperkt minimaliseringsprobleem

(5.3.10) $\quad \min\{f(x^{(k)}+Z^{(k)}w) \mid w \in \mathbb{R}^{n-q}\}$

en het aan (5.3.6) analoge probleem

$$(5.3.11) \quad \min\{f(x^{(k)}) + \nabla^T f(x^{(k)}) Z^{(k)} w \mid \|w\| < r\}$$

levert dan in termen van w de zoekrichting

$$(5.3.12) \quad d_w^{(k)} := - Z^{(k)T} \nabla f(x^{(k)})$$

en in termen van het originele probleem de zoekrichting

$$(5.3.13) \quad d^{(k)} := - Z^{(k)} Z^{(k)T} \nabla f(x^{(k)}).$$

Aangezien zowel

$$(I-N^{(k)} (N^{(k)T} N^{(k)})^{-1} N^{(k)T}) [N^{(k)} \mid Z^{(k)}] = [0 \mid Z^{(k)}]$$

als

$$Z^{(k)} Z^{(k)T} [N^{(k)} \mid Z^{(k)}] = [0 \mid Z^{(k)}]$$

volgt onmiddellijk dat

$$(5.3.14) \quad Z^{(k)} Z^{(k)T} = (I-N^{(k)} (N^{(k)T} N^{(k)})^{-1} N^{(k)}) =: \bar{P}^{(k)} .$$

Dit laatste impliceert dat beide uitdrukkingen (5.3.7) en (5.3.13) exact dezelfde zoekrichting representeren. De formulering (5.3.10) van het basisprobleem (5.3.5) werd vooral door GILL & MURRAY [1974] benut voor de afleiding van een aantal verschillende hoger-orde zoekrichtingen. In verband daarmee wordt de formulering (5.3.13) van de geprojecteerde gradiënt met behulp van de projectiematrix $Z^{(k)} Z^{(k)T}$ hierna ook wel aangeduid als de geprojecteerde gradiënt van GILL en MURRAY of wel de *Gill-Murray-gradiënt*.

Het zoekproces met de geprojecteerde gradiënt (van Rosen of van Gill en Murray) stopt indien geen nieuwe zoekrichting meer kan worden gegenereerd d.i. als

$$(5.3.15) \quad d^{(k)} := - \bar{P}^{(k)} \nabla f(x^{(k)}) = - \nabla f(x^{(k)}) + N^{(k)} (N^{(k)T} N^{(k)})^{-1} N^{(k)T} \nabla f(x^{(k)}) = 0.$$

Daaruit volgt dan onmiddellijk dat in dat geval

$$(5.3.16) \qquad \lambda^{(k)} = (N^{(k)T}N^{(k)})^{-1}N^{(k)T}\nabla f(x^{(k)}) = N^{(k)+}\nabla f(x^{(k)}).$$

Een derde formulering van het basisprobleem (5.3.5) resulteert, indien gebruik wordt gemaakt van de mogelijkheid om de q lineair onafhankelijke gelijkheidsvoorwaarden

$$(5.3.17) \qquad N^{(k)T}x = N^{(k)T}x^{(k)} =: b^{(k)}$$

te gebruiken om q variabelen op te lossen als functie van de n-q overige variabelen. De vector x wordt in dat geval gesplitst in een q-vector x_B die de afhankelijke of *basisvariabelen* bevat en een (n-q)-vector x_D die de onafhankelijke of *niet-basisvariabelen* bevat. De matrix $N^{(k)T}$ wordt analoog gesplitst in een reguliere q × q-matrix $B^{(k)}$ en een q × (n-q)-matrix $D^{(k)}$. Het stelsel (5.3.17) is dan te schrijven als

$$(5.3.18) \qquad B^{(k)}x_B + D^{(k)}x_D = b^{(k)}$$

waaruit direct volgt dat

$$(5.3.19) \qquad x_B = B^{(k)-1}b^{(k)} - B^{(k)-1}D^{(k)}x_D.$$

De objectfunctie van het probleem (5.3.5) is daarmee te schrijven als een functie van alleen de vector x_D

$$(5.3.20) \qquad f(x) = f(x_B,x_D) = f(B^{-1}b-B^{-1}Dx_D,x_D) =: \tilde{f}(x_D)$$

(in welke uitdrukking, evenals in de hierna volgende, de indices (k) achterwege werden gelaten). Het basisprobleem (5.3.5) kan opnieuw worden beschouwd als een onbeperkt minimaliseringsprobleem, ditmaal in de variabele x_D (de nevenvoorwaarde (5.3.18) fungeert in dit geval uitsluitend als een voorschrift voor de bepaling van de afhankelijke variabele x_B)

$$(5.3.21) \qquad \min\{\tilde{f}(x_D) \mid x_D \in \mathbb{R}^{n-q}\}.$$

Minimalisering van de linearisering van de objectfunctie $\tilde{f}(x_D)$ rond $x_D^{(k)}$ over een hyperbol in $\mathbb{R}^{n-q}$ geeft als zoekrichting de negatieve gradiënt van de functie $\tilde{f}(x_D)$ naar x_D. Deze gradient, die in de literatuur bekend staat als de *gereduceerde gradient* (vgl. WOLFE [1963]) wordt gegeven door

$$\nabla_{x_D} f = - D^T B^{-T} \nabla_{x_B} f + \nabla_{x_D} f. \tag{5.3.22}$$

Als zoekrichting in de originele ruimte volgt daaruit

$$\begin{aligned} d_B &:= - B^{-1} D(-\nabla_{x_D} \tilde{f}) = - B^{-1} D D^T B^{-T} \nabla_{x_B} f + B^{-1} D \nabla_{x_D} f \\ d_D &:= -\nabla_{x_D} \tilde{f} = D^T B^{-T} \nabla_{x_B} f - \nabla_{x_D} f \end{aligned} \tag{5.3.23}$$

of in matrix-vector notatie

$$d := - \begin{bmatrix} B^{-1} D D^T B^{-T} & -B^{-1} D \\ -D^T B^{-T} & I \end{bmatrix} \begin{bmatrix} \nabla_{x_B} f \\ \nabla_{x_D} f \end{bmatrix} =: - \bar{R}^{(k)} \nabla f(x^{(k)}) \tag{5.3.24}$$

of, nog anders, met de matrix

$$M := \begin{bmatrix} -B^{-1} D \\ I \end{bmatrix} \tag{5.3.25}$$

uitgeschreven

$$d := - M M^T \nabla f. \tag{5.3.26}$$

Deze laatste uitdrukking vertoont grote overeenkomst met de uitdrukking voor de Gill-Murray-gradiënt (5.3.13), een overeenkomst die nog wordt versterkt door de observatie dat

$$N^{(k)T} M = [B|D] \begin{bmatrix} -B^{-1} D \\ I \end{bmatrix} = O_{q \times (n-q)}. \tag{5.3.27}$$

Het zoekproces met de gereduceerde gradiënt stopt indien

$$\nabla_{x_D} \tilde{f} = - D^T B^{-T} \nabla_{x_B} f + \nabla_{x_D} f = 0. \tag{5.3.28}$$

In dat geval geldt

$$(5.3.29) \quad - \begin{bmatrix} B^T \\ \\ D^T \end{bmatrix} B^{-T} \nabla_{x_B} f + \begin{bmatrix} \nabla_{x_B} f \\ \\ \nabla_{x_D} f \end{bmatrix} = 0$$

en daaruit volgt weer

$$(5.3.30) \quad \lambda = B^{-T} \nabla_{x_B} f \ .$$

De gereduceerde gradiënt werd voor het eerst als zoekrichting gesuggereerd door WOLFE [1963]. De (gegeneraliseerde) gereduceerde-gradiënt methode die er op gebaseerd is werd vooral verder ontwikkeld door ABADIE & CARPENTIER [1969]. De hierboven weergegeven afleiding wijkt af van de gebruikelijke doordat alleen de actieve beperkingen in de matrix $N^{(k)T} = [B^{(k)} | D^{(k)}]$ worden opgenomen. In de gebruikelijke situatie worden, in navolging van de simplexmethode voor lineaire programmering, alle passieve beperkingen met behulp van "slack"-variabelen actief gemaakt, waarna als uitgangsprobleem fungeert een probleem van de vorm

$$(5.3.31) \quad \min\{f(x) \mid A^T x - b = 0, \quad x \geq 0\}.$$

In deze laatste situatie worden alléen die componenten van de gereduceerde gradiënt gebruikt als zoekrichting die geen overschrijding van de coördinaatbeperkingen tot gevolg hebben. D.w.z. de niet-basiscomponenten van de zoekrichting worden gedefinieerd als

$$(5.3.32) \quad \begin{aligned} \bar{d}_{D,i} &:= - (\nabla_{x_D} \tilde{f})_i \quad \text{als} \quad x_i \neq 0 \vee (\nabla_{x_D} \tilde{f})_i < 0 \\ &:= 0 \qquad\qquad\qquad\quad x_i = 0 \wedge (\nabla_{x_D} \tilde{f})_i > 0. \end{aligned}$$

Waarna als basiscomponenten dan volgen

$$(5.3.33) \quad \bar{d}_{B,j} := -(B^{-1} D \bar{d}_D)_j .$$

Wanneer alle beperkingen lineair zijn dan zullen de verzamelingen van actieve beperkingen in opvolgende iteraties slechts in één (of geen) element verschillen en wel doordat of een eerdere passieve beperking actief wordt of een eerdere actieve beperking passief verondersteld wordt. Het is in dit

geval niet nodig om de totale berekeningen die samenhangen met de matrices $\bar{P}^{(k)}$ (5.3.7), $Z^{(k)}Z^{(k)T}$ (5.3.14) en $\bar{R}^{(k)}$ (5.3.24) geheel opnieuw van voor af aan uit te voeren. Door diverse auteurs, waaronder o.a. ROSEN [1960] en GILL & MURRAY [1974] werden aanpassingsformules opgesteld die er voor zorgen dat een efficient gebruik gemaakt wordt van de aanwezige informatie over de actieve beperkingen die actief blijven.

5.3.3. Hoger-orde zoekrichtingen

Tweede-orde of Newton-zoekrichtingen kunnen worden afgeleid door gebruik te maken van tweede-orde benaderingen van de objectfunctie van het basisprobleem (5.3.5). Quasi-Newton-zoekinrichtingen kunnen daar weer van worden afgeleid door gebruik te maken van benaderingen in de uitdrukkingen van de Newton-zoekrichtingen. In deze paragraaf zullen alleen voor het meest simpele geval (positief definiete Hessiaan van de objectfunctie) twee tweede-orde of Newton-zoekrichtingen worden besproken om een indruk te geven van de mogelijkheden in dit verband. Voor meer gedetailleerde discussies over modificaties van deze zoekrichtingen en van daarvan afgeleide quasi-Newton-zoekrichtingen wordt de lezer verwezen naar elders (bv. GILL & MURRAY [1974], DE JONG [1976], DIRKX [1975]).

Analoog aan de situatie bij de eerste orde zoekrichtingen resulteren de eerder besproken verschillende formuleringen van het basisprobleem in evenzo vele verschillende formuleringen van Newton-zoekrichtingen. In de eerste, gebruikelijke formulering wordt de tweede-orde benadering van de objectfunctie rond het punt $x^{(k)}$ gegeven door de uitdrukking

$$f(x^{(k)})+\nabla^T f(x^{(k)})(x-x^{(k)})+\tfrac{1}{2}(x-x^{(k)})^T G(x^{(k)})(x-x^{(k)}). \tag{5.3.34}$$

Het minimum $x^{(k+1)}$ van deze kwadratische functie onder de nevenvoorwaarde

$$N^{(k)T}(x-x^{(k)}) = 0 \tag{5.3.35}$$

kan worden gevonden met behulp van de noodzakelijke voorwaarde dat in het optimale punt $x^{(k+1)}$ moet gelden

$$\nabla f(x^{(k+1)}) - N^{(k)}\lambda^{(k+1)} = 0 \tag{5.3.36}$$

waar

(5.3.37) $\nabla f(x^{(k+1)}) = \nabla f(x^{(k)}) + G(x^{(k)})(x^{(k+1)}-x^{(k)})$.

In het geval dat $G(x^{(k)})$ niet singulier is volgt daaruit dat

(5.3.38ä) $x^{(k+1)}-x^{(k)} = G^{-1}(x^{(k)})(N^{(k)}\lambda^{(k+1)}-\nabla f(x^{(k)}))$

of met een (ook hierna toe te passen) vereenvoudigde notatie

(5.3.38b) $x^* - x = G^{-1}(N\lambda^*-\nabla f)$.

Substitutie van dit resultaat in de nevenvoorwaarde (5.3.35) geeft de mogelijkheid de onbekende parameter λ^* te bepalen uit

(5.3.39) $N^TG^{-1}N\lambda^* - N^TG^{-1}\nabla f = 0$

met als resultaat de *Newton-zoekrichting I*

(5.3.40) $d^{(k)} := -(I-G^{(k)-1}N^{(k)}(N^{(k)T}G^{(k)-1}N^{(k)})N^{(k)T})G^{(k)-1}\nabla f(x^{(k)})$.

Een tweede-orde benadering van de alternatieve formulering (5.3.10) van het basisprobleem met behulp van de matrix $Z^{(k)}$ (5.3.8) resulteert in het onbeperkte kwadratische minimaliseringsprobleem

(5.3.41) $\min\{f(x^{(k)})+\nabla^T f(x^{(k)})Z^{(k)}w+\frac{1}{2}w^TZ^{(k)T}G(x^{(k)})Z^{(k)}w \mid w \in \mathbb{R}^{n-q}\}$

waarvan de oplossing, indien $Z^{(k)T}G(x^{(k)})Z^{(k)}$ positief definiet is, gevonden wordt voor

(5.3.42) $w^{(k+1)} := -(Z^{(k)T}G(x^{(k)})Z^{(k)})^{-1}Z^{(k)T}\nabla f(x^{(k)})$.

In de originele coördinaten levert dit direct de *Newton-zoekrichting II*

(5.3.43) $d^{(k)} := -Z^{(k)}(Z^{(k)T}G(x^{(k)})Z^{(k)})^{-1}Z^{(k)T}\nabla f(x^{(k)})$.

Juist als in het geval bij de eerste orde zoekrichtingen kan worden aangetoond dat, als $G(x^{(k)})$ niet singulier is, de beide uitdrukkingen (5.3.40) en (5.3.43) dezelfde zoekrichting representeren. Het verschil tussen beide zit voornamelijk in de grotere toepasbaarheid van de laatste formulering

die een gevolg is van het feit dat alleen de eigenschappen van de matrix $Z^{(k)T}G^{(k)}Z^{(k)}$ meespelen en niet die van de gehele originele Hessiaan $G(x^{(k)})$. De tweede formulering leent zich ook beter voor modificaties in het geval de matrix $Z^{(k)T}G^{(k)}Z^{(k)}$ niet positief definiet is, als ook voor de ontwikkeling van quasi-Newtonmethoden (vgl. GILL & MURRAY [1974]).

5.3.4. Bepaling van de maximale stapgroottefactor

Voor de bepaling van de maximale stapgrootte wordt uitgegaan van de lokale linearisering van alle beperkingen, zowel actieve als passieve, in het toegelaten punt $x^{(k)}$. De in de voorgaande punten besproken zoekrichtingen werden zo geconstrueerd dat alle gelineariseerde actieve beperkingen actief blijven ongeacht de grootte van de stap. Bepalend voor de maximale stapgrootte zijn daarom uitsluitend de gelineariseerde passieve beperkingen

$$(5.3.44) \qquad c_j(x^{(k)})+a_j^{(k)T}(x-x^{(k)}) \geq 0, \quad j \in I_p(x^{(k)}).$$

Als voorwaarde volgt daaruit voor de maximale stap $\bar{\alpha}$ dat

$$(5.3.45) \qquad \bar{\alpha}a_j^{(k)T}d^{(k)} \geq - c_j(x^{(k)})$$

hetgeen onmiddellijk leidt tot de uitdrukking

$$(5.3.46) \qquad \bar{\alpha} := \min_j \left\{ \frac{-c_j(x^{(k)})}{a_j^{(k)T}d^{(k)}} \;\middle|\; j \in I_p(x^{(k)}) \wedge a_j^{(k)T}d^{(k)} < 0 \right\}.$$

5.3.5. Bepaling van toegelaten punten tijdens het iteratieproces

In het geval dat alle beperkingen lineair zijn en gebruik wordt gemaakt van de in het voorgaande besproken zoekrichtingen en stapgroottebegrenzingen zullen in theorie (d.i. indien wordt afgezien van eventuele numerieke onnauwkeurigheden) als het startpunt toegelaten is, ook alle volgende iteratiepunten toegelaten zijn. Bij aanwezigheid van niet-lineaire beperkingen verandert die situatie omdat de actieve beperkingen niet noodzakelijk actief blijven en omdat de met linearisaties berekende stapgroottebegrenzingen passieve beperkingen niet noodzakelijk actief maken. Het voorlopige eindresultaat van iedere iteratiestap

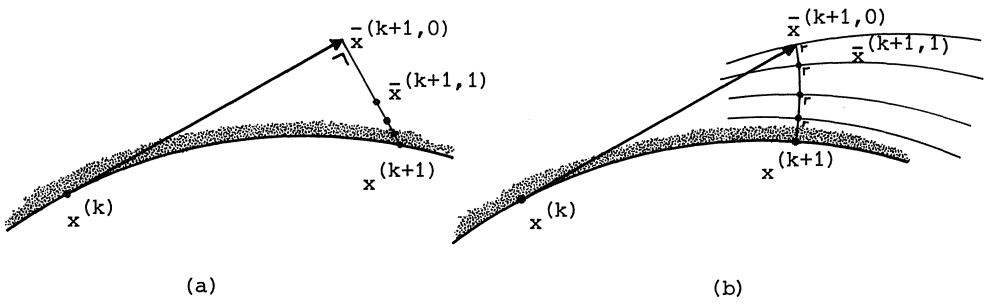
$$\bar{x}^{(k+1,0)} = x^{(k)} + \alpha^{(k)} d^{(k)} \tag{5.3.47}$$

zal in het algemeen dan ook een niet-toegelaten punt zijn en speciale procedures zijn nodig om uitgaande van het niet-toegelaten punt (5.3.47) een toegelaten punt te bepalen. Zoals gezegd in par. 5.2.2 worden deze procedures aangeduid als *restauratieprocedures*. Men kan bij het bepalen van een toegelaten punt vanuit het laatst bereikte niet-toegelaten punt uitgaan van twee essentieel verschillende ideeën over het verkrijgen van de benodigde informatie in het niet-toegelaten punt:

a) Men gaat uit van de informatie, zoals deze bepaald is in het laatst bepaalde toegelaten punt.

b) Men doet voor elke restauratiestap alsof het niet-toegelaten punt een eerste niet-toegelaten punt is en bepaalt in dat punt alle nodige informatie.

Ook allerlei tussenvormen zijn mogelijk. Bijvoorbeeld door een deel van de oude informatie te gebruiken en de rest opnieuw te bepalen of door alleen in het eerste niet-toegelaten punt in de betreffende iteratieslag alle informatie opnieuw te bepalen en deze in eventuele volgende restauratiestappen te gebruiken. De mogelijkheden genoemd onder a en b worden geïllustreerd in Figuur 5.3.2.



Figuur 5.3.2. Twee mogelijkheden om vanuit een niet toegelaten punt $\bar{x}^{(k+1,0)}$ een toegelaten punt $x^{(k+1)}$ te vinden.

In het geval men alleen gebruik wil maken van de informatie die in het laatste toegelaten punt bekend is, maakt het verschil of de zoekrichting $d^{(k)}$ bepaald is met de gereduceerde-gradiëntmethode of met de geprojekteerde-gradiëntmethode. Wanneer $d^{(k)}$ bepaald is met de gereduceerde-gradiëntmethode dan kan men de niet-basiscomponenten van $\bar{x}^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$ vasthouden en alleen de basiscomponenten aanpassen aan de niet-lineaire beperkingen. Dat wil zeggen dat de restauratie alleen via de basisvariabelen wordt uitgevoerd. Als geldt dat tijdens het hele restauratieproces geen beperkingen overschreden worden, behalve diegene die in $x^{(k)}$ actief waren, kan de restauratiestap $s^{(k+1,r)}$ met een Newton-achtige methode worden berekend uit

$$\begin{aligned} & B^{(k)} s_B^{(k+1,r)} = - c(\bar{x}^{(k+1,r)}) \\ & s_D^{(k+1,r)} = 0 \end{aligned} \tag{5.3.48}$$

waar $s^{(k+1,r)} = \bar{x}^{(k+1,r+1)} - \bar{x}^{(k+1,r)}$ de (r+1)-de restauratiestap is, $\bar{x}^{(k+1,r)}$ de r-de benadering is voor het toegelaten punt $x^{(k+1)}$, en $B^{(k)}$ de in het toegelaten punt $x^{(k)}$ bepaalde basismatrix. Als wel nieuwe beperkingen overschreden worden, heeft het weinig nut vast te houden aan de oude splitsing in basisvariabelen en niet-basisvariabelen en kan de restauratie evengoed worden uitgevoerd via alle variabelen.

Als $d^{(k)}$ bepaald is met de geprojecteerde-gradiëntmethode kan men een toegelaten punt zoeken door een lineaire combinatie te bepalen van de normalen van de actieve en overschreden beperkingen, berekend in het laatst bepaalde toegelaten punt $x^{(k)}$. In dat geval kan de restauratiestap worden bepaald uit het Newton-achtige stelsel (vgl. (5.3.2))

$$\bar{N}^{(k)} T_s^{(k+1,r)} = \bar{N}^{(k)} T_{\bar{N}}^{(k)} \kappa^{(k+1,r)} = - c(\bar{x}^{(k+1,r)}) \tag{5.3.49}$$

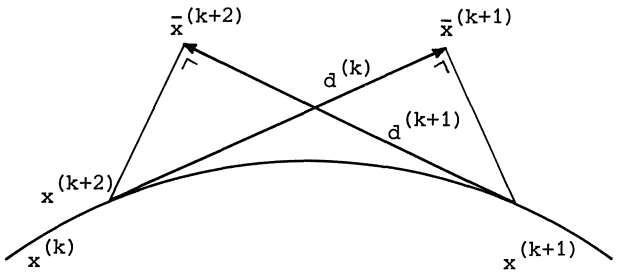
waaruit volgt

$$s^{(k+1,r)} = - \bar{N}^{(k)} (\bar{N}^{(k)} T_{\bar{N}}^{(k)})^{-1} c(\bar{x}^{(k+1,r)}). \tag{5.3.50}$$

De matrix $\bar{N}^{(k)}$ die vergelijkbaar is met de in par.5.3.1 besproken matrix $\bar{A}$ bevat in eerste instantie de normalen van de actieve beperkingen in $x^{(k)}$. Als tijdens de hele restauratieprocedure geen andere beperkingen overschreden worden dan die, die in $x^{(k)}$ actief waren, kan daarvoor de eerder

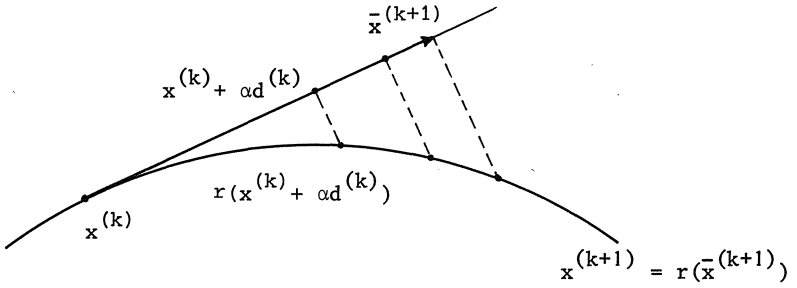
toegepaste matrix $N^{(k)}$ worden gebruikt. In andere gevallen moet de matrix $\bar{N}^{(k)}$ aangepast worden, zo mogelijk door gebruik te maken van de normalen van de beperkingen zoals deze berekend zijn in het punt $x^{(k)}$. In het geval we restauratie willen uitvoeren door in elk bereikt niet-toegelaten punt alle informatie opnieuw te bepalen wordt in elke stap precies gehandeld zoals in par. 5.3.1 beschreven voor het geval van een toegelaten startpunt.

Een heel ander probleem dat op kan treden ten gevolge van het niet-lineair zijn van de beperkingen (met de daaruit voortvloeiende noodzaak tot restauratie) is het volgende. De meeste gebruikte methoden garanderen, dat in het gevonden niet-toegelaten punt $\bar{x}^{(k+1)}$ geldt $f(\bar{x}^{(k+1)}) < f(x^{(k)})$. Tengevolge van de restauratie is het echter mogelijk dat in het uit $\bar{x}^{(k+1)}$ bepaalde toegelaten punt $x^{(k+1)}$ geldt $f(x^{(k+1)}) > f(x^{(k)})$. Dit houdt de mogelijkheid tot cykelen in (zie Figuur 5.3.3). Voorkomen kan men dat door in de gebruikte lijnminimaliseringsprocedure een andere strategie



Figuur 5.3.3. Een mogelijk geval van cykelen

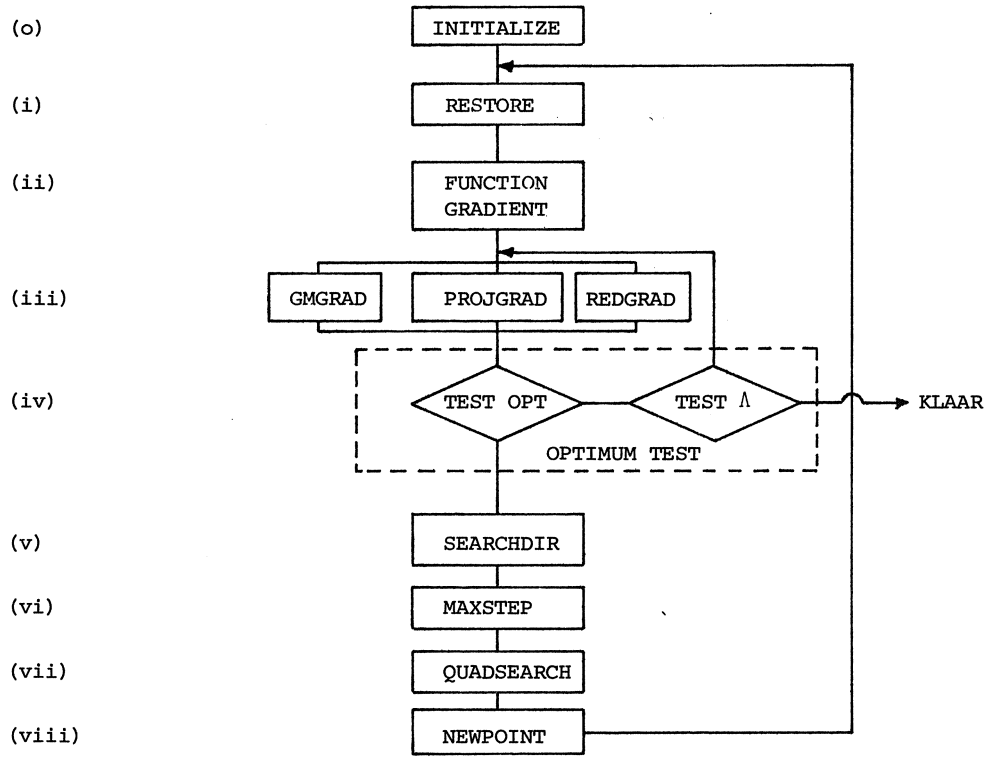
toe te passen zoals geïllustreerd in Figuur 5.3.4. Zij r(x) het toegelaten punt, dat gevonden wordt bij restauratie uitgaande van het punt x. Dan moeten we in de gebruikte lijnminimaliseringsprocedure het minimum bepalen van $f(r(x^{(k)}+\alpha d^{(k)}))$ in plaats van van $f(x^{(k)}+\alpha d^{(k)})$.



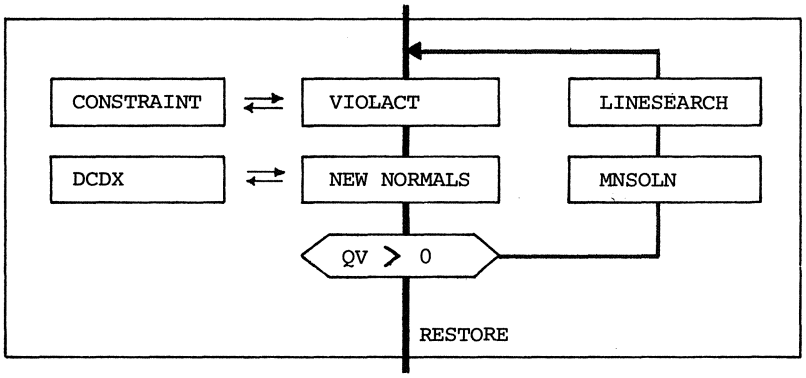
Figuur 5.3.4. Lijnminimaliseringsmethode om cykelen te voorkomen.

5.4. Practische details: Beschrijving van de belangrijkste procedures

In deze paragraaf zullen de belangrijkste procedures beschreven worden waaruit de Algol-procedure NONLINMIN is opgebouwd. De plaats die deze procedures innemen in het totaal volgt direct uit het flowdiagram van de procedure NONLINMIN dat weergegeven is in Figuur 5.4.1. De procedure RESTORE bepaalt zowel voor de eerste als voor alle andere iteraties een toegelaten punt $x^{(k)}$ als startpunt voor de komende iteratie. Tevens bepaalt de procedure welke beperkingen actief zijn in het betreffende toegelaten punt. De procedure bestaat zelf uit een viertal subprocedures a) VIOLACT, b) NEW NORMALS, c) MNSOLN en d) LINESEARCH, welke aan elkaar gekoppeld zijn volgens het in Figuur 5.4.2 geschetste flowdiagram, waarin QV het in de procedure VIOLACT gevonden aantal overschreden beperkingen voorstelt. De functies van deze subprocedures zijn als volgt:
De procedure VIOLACT bepaalt de status van alle beperkingen (actief, passief dan wel overschreden) in de actuele benadering x van de oplossing. Hierbij wordt onderscheid gemaakt tussen de triviale en de niet-triviale beperkingen. Eerst worden alle triviale beperkingen bekeken en bij overschrijding van een ervan, wordt het punt x aangepast door de betreffende component op de dichtsbijliggende grenswaarde te zetten. Daarmee wordt deze triviale beperking actief. Daarna worden in het, eventueel aangepaste, punt x alle niet-triviale beperkingen geëvalueerd en genoteerd of deze actief, passief dan wel overschreden zijn. Ook worden de aantallen overschreden, actieve triviale en actieve niet-triviale beperkingen bijgehouden.



Figuur 5.4.1. Flowdiagram van de procedure NONLINMIN (De nummers tussen ( ) corresponderen met de stappen in de basisalgoritme (pt. 5.2.2)).



Figuur 5.4.2. De procedure RESTORE

De procedure NEW NORMALS vult de in par. 5.3.1 en par. 5.3.5 besproken matrices $\bar{A}$ of $\bar{N}^{(k)}$. Afhankelijk van de gevolgde strategie (vgl. par. 5.3.5) worden bij de methoden, die gebruik maken van een geprojecteerde gradiënt, de normalen van de niet-lineaire beperkingen gehandhaafd, dan wel opnieuw geëvalueerd voor iedere nieuwe benadering $x^{(k,r)}$ van het toegelaten punt $x^{(k)}$. Bij de methoden, die gebruik maken van de gereduceerde gradiënt wordt in de procedure NEW NORMALS of (1) de basismatrix gehandhaafd, of (2) er vindt een basiswisseling plaats, of (3) van het onderscheid tussen basis- en niet-basisvariabelen wordt geheel afgezien. Deze drie mogelijkheden doen zich respectievelijk voor (1) wanneer de strategie het handhaven van de oude normalen voorschrijft en geen, eerder passieve triviale of passieve niet-triviale beperking actief wordt; (2) wanneer de strategie het handhaven van de oude normalen voorschrijft en er een triviale beperking actief wordt en (3) in alle andere gevallen. In het laatste geval gaat de restauratieprocedure verder als bij de methoden die gebruik maken van de geprojecteerde gradiënt.

De procedure MNSOLN bepaalt, uitgaande van een niet-toegelaten punt en met behulp van de matrix van normalen van actieve en overschreden beperkingen, de stap naar een toegelaten punt, dan wel de stap naar een punt met minder overschreden beperkingen. Afhankelijk van de omstandigheid of de som van het aantal actieve en het aantal overschreden beperkingen groter of gelijk, dan wel kleiner is dan het aantal variabelen (vgl. par. 5.3.1) bepaalt de procedure de kleinste-kwadraten dan wel de minimumnorm oplossing van de vergelijking (5.3.2)

$$\bar{A}^T \Delta x = - \bar{c}$$

of, equivalent, van de vergelijking (5.3.49)

$$\bar{N}^{(k)} T_s^{(k+1,r)} = - c(\bar{x}^{(k+1,r)}).$$

De gevonden stap wordt als zoekrichting gebruikt in de procedure LINESEARCH. De procedure RESTORE wordt verlaten indien een toegelaten punt is gevonden, dan wel indien meer dan een vooraf aangegeven aantal restauratiestappen zijn gedaan zonder resultaat. In dit laatste geval wordt het iteratieproces gestopt.

In de procedure NONLINMIN zijn drie procedures, te weten: PROJGRAD, GMGRAD en REDGRAD, opgenomen, die evenzoveel transformaties van de gradiënt van de objectfunctie genereren. Deze getransformeerde gradiënten worden zowel gebruikt voor het genereren van een zoekrichting in de procedure SEARCHDIR als, samen met de in dezelfde procedures bepaalde Lagrangemultiplicatoren, in de test of een lokaal optimaal punt is bereikt. Dit gebeurt in de procedure OPTIMUM TEST. De drie procedures voor het genereren van een transformatie van de gradiënt van de objectfunctie fungeren in meer detail als volgt: De procedure GMGRAD berekent de geprojecteerde gradiënt volgens de methode van Gill en Murray. Gebruik wordt gemaakt van een matrix Z, waarvan de kolommen een orthonormale basis vormen van het orthogonale complement van de deelruimte opgespannen door de normalen van actieve beperkingen. Zij A de $(n \times m_A)$-matrix met in de kolommen de normalen van de actieve beperkingen, dan geeft QR-decompositie, als k de rang is van de matrix A, het resultaat:

$$(5.4.1) \qquad A = Q \begin{bmatrix} R & D \\ 0 & 0 \end{bmatrix} P^T$$

waar R een k × k bovendriehoeksmatrix is, Q een n × n matrix waarvoor $Q^T Q = I$, P een $m_A \times m_A$ permutatiematrix en D een $k \times (m_A - k)$ matrix. Splitsen we Q in een n × k matrix $Q^{(1)}$ en een n × (n-k) matrix $Q^{(2)}$, dan volgt

$$(5.4.2) \qquad \begin{aligned} {Q^{(1)}}^T A &= [R|D]P^T \\ {Q^{(2)}}^T A &= 0. \end{aligned}$$

Hieruit en uit

$$(5.4.3) \qquad \begin{bmatrix} Q^{(1)^T} \\ \hline Q^{(2)^T} \end{bmatrix} (Q^{(1)} \vdots Q^{(2)}) = \begin{bmatrix} I_k & 0 \\ 0 & I_{n-k} \end{bmatrix}$$

blijkt dat $Q^{(2)}$ een goede keuze is voor de in par. 5.3.2 gepostuleerde matrix Z. De Gill-Murray-gradiënt (5.3.13) wordt met de matrix $Q^{(2)}$ gemakkelijk gevonden als

$$(5.4.4) \qquad g_{GM}^{(k)} = Q^{(2)} Q^{(2)^T} \nabla f(x^{(k)}).$$

De vector $\lambda^{(k)}$ van Lagrangemultiplicatoren wordt gevonden als de kleinste-kwadraten-oplossing van

$$(5.4.5) \qquad A\lambda^{(k)} - \nabla f(x^{(k)}) = 0.$$

Met behulp van de resultaten van de QR-decompositie van A wordt hiervoor gevonden

$$(5.4.6) \qquad \lambda^{(k)} = P \begin{bmatrix} R^{-1} Q^{(1)^T} \\ \hline 0 \end{bmatrix} \nabla f(x^{(k)})$$

als we de Lagrangemultiplicatoren voor de afhankelijke actieve beperkingen a priori gelijk aan nul kiezen.

De procedure PROJGRAD bepaalt de in par. 5.3.2 besproken geprojecteerde gradiënt eveneens met behulp van de QR-decompositie (5.4.1) van de matrix A. Na splitsing van Q op dezelfde manier als in GMGRAD blijkt dat de projectiematrix (5.3.14)

$$\bar{P}^{(k)} = (I - N^{(k)} (N^{(k)^T} N^{(k)})^{-1} N^{(k)^T})$$

gegeven wordt door de uitdrukking

$$(5.4.7) \qquad \bar{P}^{(k)} = I - Q^{(1)} Q^{(1)^T}.$$

Daarmee is de geprojecteerde gradiënt te schrijven als

$$(5.4.8) \qquad g_{PROJ}^{(k)} = \nabla f(x^{(k)}) - Q^{(1)}Q^{(1)^T}\nabla f(x^{(k)}).$$

De vector van Lagrangemultiplicatoren wordt weer gevonden uit de hierboven bij de procedure GMGRAD besproken relatie (5.4.6)

$$\lambda^{(k)} = P \begin{bmatrix} R^{-1}\, Q^{(1)^T} \\ \hline 0 \end{bmatrix} \nabla f(x^{(k)}) \ .$$

De procedure REDGRAD bepaalt een gereduceerde gradiënt op de manier, zoals besproken in par. 5.3.2. Begonnen wordt met het maken van een keuze voor een basis. Het criterium daarbij is, dat geen van de basisvariabelen op zijn onder- of bovengrens ligt. Vervolgens wordt nagegaan of de matrix B niet-singulier is. Is dit wel het geval, dan wordt een basiswisseling uitgevoerd. Zoniet, dan wordt een vector λ van Lagrangemultiplicatoren berekend (vgl. (5.3.30))

$$(5.4.9) \qquad \lambda = B^{-T}\nabla_{x_B} f$$

en daarmee de gereduceerde gradiënt

$$(5.4.10) \qquad \nabla_{x_D}\tilde{f} = \nabla_{x_D} f - D^T\lambda.$$

Daarna wordt afgeweken van de in par. 5.3.2 beschreven formulering en wordt gelet op de mogelijkheid om de componenten van $-\nabla_{x_D}\tilde{f}$ voor de niet-basisvariabelen te gebruiken als componenten van de zoekrichting. Voor de niet-basisvariabelen waarvoor deze componenten van teken zo zijn, dat actieve triviale beperkingen overschreden dreigen te worden, wordt de overeenkomstige component van de aangepaste gereduceerde gradiënt $[\nabla_{x_D}\tilde{f}]$ op nul gesteld. De andere componenten worden ongewijzigd gehandhaafd. Tenslotte worden de basiscomponenten van de aangepaste gereduceerde gradiënt bepaald uit (5.3.33)

$$g_{R,B} = -B^{-1}D[\nabla_{x_D}\tilde{f}]$$

waarmee als "gereduceerde gradiënt" resulteert

$$(5.4.11)\qquad g_R = \begin{bmatrix} g_{R,B} \\ \hline g_{R,D} \end{bmatrix} = \begin{bmatrix} -B^{-1}D[\nabla_{x_D}\tilde{f}] \\ \hline [\nabla_{x_D}\tilde{f}] \end{bmatrix} .$$

In de procedure OPTIMUM TEST wordt gecontroleerd of het laatst bepaalde toegelaten punt het optimale punt is. Hierbij wordt gebruik gemaakt van drie verschillende criteria

a) (1): $\|g^{(k)}_{GM/PROJ/R}\| < \varepsilon_1\|\nabla f(x^{(0)})\| + \varepsilon_2$ en (2): $\lambda \geq 0$

b) $\|x^{(k)} - x^{(k-1)}\| \leq \varepsilon_3$

c) $|f(x^{(k)}) - f(x^{(k-1)})| \leq \varepsilon_4|f(x^{(k-1)})| + \varepsilon_5.$

Het eerst wordt nagegaan of aan criterium a) (1) voldaan is. Is dat het geval, dan wordt gecontroleerd of alle Lagrangemultiplicatoren behorend bij de actieve ongelijkheidsbeperkingen groter dan of gelijk aan nul zijn. Wanneer dit niet het geval is, wordt de beperking die overeenkomt met de meest negatieve Lagrangemultiplicator passief gemaakt en wordt zonder de andere convergentiecriteria te beschouwen teruggegaan naar een der procedures GMRAD, PROJGRAD of REDGRAD waar een nieuwe getransformeerde gradiënt wordt bepaald. In het andere geval worden beide andere convergentiecriteria gecontroleerd. Meegegeven moet worden of men het proces convergent noemt zodra behalve aan criterium a) ook aan nul, één of twee van de onder b) en c) genoemde criteria voldaan is.

In de procedure SEARCHDIR wordt de zoekrichting $d^{(k)}$ bepaald. Op dit moment wordt in deze procedure nog niets anders gedaan, dan de zoekrichting gelijkstellen aan de negatieve getransformeerde gradiënt

$$(5.4.12)\qquad d^{(k)} = - g^{(k)}_{GM/PROJ/R}.$$

De procedure MAXSTEP bepaalt de maximale toegelaten stap vanuit het punt $x^{(k)}$ in de zoekrichting $d^{(k)}$, binnen het door de passieve gelineariseerde beperkingen bepaald gebied. Dit wordt gedaan volgens de methode beschreven in par. 5.3.4, d.w.z. door evaluatie van de uitdrukking (5.3.46)

$$\alpha_{max} = \min_j \left\{ - \frac{c_j(x^{(k)})}{a_j^{(k)T} d^{(k)}} \;\middle|\; j \in I_p(x^{(k)}),\ a_j^{(k)T} d^{(k)} < 0 \right\} .$$

In het geval geen enkele beperking passief is, wordt α_{max} gelijkgesteld aan een meegegeven constante.

De procedure QUADSEARCH tenslotte bepaalt een benadering voor het minimum van de objectfunctie langs de zoekrichting $d^{(k)}$. Daartoe worden, binnen het door α_{max} beperkte gebied, drie punten op deze lijn gezocht, zodanig dat het mogelijk is door de drie bijbehorende functiewaarden een dalparabool te construeren. Het minimum van deze dalparabool wordt gebruikt als benadering voor het gezochte minimum van de objectfunctie. Een meer gedetailleerde beschrijving van deze procedure wordt gegeven door EILERS [1975].

5.5. Slotopmerking

In het voorgaande is een beknopte beschrijving gegeven van de opbouw van een algemene Algolprocedure voor het minimaliseren van niet-lineaire functies onder niet-lineaire nevenvoorwaarden. De ontwikkeling daarvan is nog in volle gang. Numerieke ervaring werd slechts in zeer beperkte mate opgedaan. De gekozen modulaire opbouw van de procedure is tot nu toe geschikt gebleken voor de beoogde ontwikkeling en diverse deelprocedures werden inmiddels reeds verder verfijnd en uitgebouwd. De belangrijkste ervaring tot op dit moment is de enorm grote hoeveelheid tijd en energie die nodig is voor het ontwikkelen en operationeel maken van dit soort universele programma's. Een woord van dank aan R. Kool van de Onderafdeling der Wiskunde, die samen met C.J.B. Dirkx en de schrijver hier reeds vele uren aan besteedde, lijkt in dit verband zeker op zijn plaats.

LITERATUUR

ABADIE, J. & J. CARPENTIER, [1969]: *Generalization of the Wolfe reduced gradient to the case of nonlinear constraints*, in: Fletcher, J. (ed.): *Optimization*, Academic Press, New York.

DIRKX, C.J.B. [1975]: *NONLINMIN, een Algol-procedure voor het minimaliseren van niet-lineaire functies onder niet-lineaire nevenvoorwaarden*, Afstudeerverslag, Onderafdeling der Wiskunde, Technische Hogeschool Eindhoven.

EILERS, G.A.M. [1975]: *Het minimaliseren van sommen van kwadraten van niet-lineaire functies*, Memorandum COSOR 75.08 , Onderafdeling der Wiskunde, Technische Hogeschool Eindhoven.

GILL, P.E. & W. MURRAY, [1974]: *Numerical methods for constrained optimization*, Academic Press, London.

DE JONG, J.L. [1976]: *Numerieke algoritmen voor niet-lineaire optimaliseringsproblemen*, Syllabus by het college Capita Optimaliseringsmethoden, voorjaar 1976, Onderafdeling der Wiskunde, Technische Hogeschool Eindhoven.

LOOTSMA, F.A. [1972]: *The Algol-procedure MINIFUN for solving non-linear optimization problems*, Report 4761, Philips Research Laboratories, Eindhoven.

LUENBERGER, D.G. [1973]: *Introduction to linear and nonlinear programming*, Addison Wesley Publ. Cy., Reading, Mass.

ROSEN, J.B. [1960]: *The gradient projection method for nonlinear programming, Part I:Linear constraints*, SIAM J. Appl. Math. 8, pp. 181-217.

WOLFE, PH. [1963]: *Methods of nonlinear programming*, in: GRAVES, R.L. & PH. WOLFE, (eds.) *Recent advances in mathematical programming*, McGraw-Hill, New York.

6. VOLTERRA-INTEGRAALVERGELIJKINGEN VAN DE TWEEDE SOORT

door P.J. van der Houwen
(Mathematisch Centrum)

6.1. Inleiding

Volterra-vergelijkingen vormen samen met de Fredholm-vergelijkingen de belangrijkste klassen van integraalvergelijkingen in de toegepaste wiskunde. Hierbij onderscheidt men vergelijkingen van de eerste en van de tweede soort. In dit hoofdstuk beperken we ons tot *Volterra-vergelijkingen van de tweede soort*; in hoofdstuk 7 zullen integraalvergelijkingen van de *eerste soort*, zowel van het Volterra- als het Fredholm-type aan de orde komen. Voor de bespreking van programmatuur voor Fredholm-vergelijkingen van de tweede soort verwijzen we naar ATKINSON [1976].
De algemene (niet-lineaire) Volterra-vergelijking van de tweede soort heeft de vorm

$$f(x) = g(x) + \int_{x_0}^{x} K(x,\xi,f(\xi))d\xi, \tag{6.1.1}$$

waarin f de onbekende functie voorstelt en g en K gegeven functies zijn. Voor eenduidigheid en existentie van oplossingen verwijzen we naar MILLER [1971]. Hier zullen we steeds aannemen dat (6.1.1) een eenduidige oplossing bezit.

Integraalvergelijkingen van het type (6.1.1) zijn te schrijven als beginwaardeproblemen voor differentiaalvergelijkingen "met een verleden" ofwel integro-differentiaalvergelijkingen. Laat de functie K, de zogenaamde *kernfunctie*, en de functie g differentieerbaar zijn naar x, dan kan (6.1.1) geschreven worden als

$$(6.1.1') \qquad \frac{df(x)}{dx} = \frac{dg(x)}{dx} + K(x,x,f(x)) + \int_{x_0}^{x} \frac{\partial}{\partial x} K(x,\xi,f(\xi))d\xi,$$

met de beginvoorwaarde $f(x_0) = g(x_0)$.
De laatste term in deze "differentiaalvergelijking" betekent dat het rechterlid afhangt van de waarden van de oplossing in het gehele voorafgaande interval $[x_0,x]$, dit in tegenstelling tot "normale" differentiaalvergelijkingen. Alleen wanneer K *niet* van x afhangt is een Volterra-vergelijking van de tweede soort te schrijven als een differentiaalvergelijking "zonder verleden". Omgekeerd kan elk beginwaardeprobleem voor een gewone differentiaalvergelijking geschreven worden in de vorm (6.1.1): het probleem

$$(6.1.2) \qquad \frac{df}{dx} = K(x,f), \qquad f(x_0) = g(x_0)$$

is te schrijven als

$$(6.1.2') \qquad f(x) = g(x_0) + \int_{x_0}^{x} K(\xi,f(\xi))d\xi.$$

Deze relatie tot beginwaardeproblemen voor gewone differentiaalvergelijkingen suggereert om te onderzoeken in hoeverre numerieke berekeningsmethoden voor beginwaardeproblemen toepasbaar zijn op Volterra-vergelijkingen. Kenmerkend voor deze berekeningsmethoden is het "stap-voor-stap" karakter: men veronderstelt dat de oplossing in het interval $[x_0,x_n]$ berekend is en men stelt een formule op voor de oplossing in het punt $x_{n+1} = x_n + h$. Iets dergelijks doet men ook bij integraalvergelijkingen van het type (6.1.1). Daartoe schrijven we (6.1.1) in de vorm

$$(6.1.3) \qquad f(x_{n+1}) = \left[g(x_{n+1}) + \int_{x_0}^{x_n} K(x_{n+1},\xi,f(\xi))d\xi\right] + \int_{x_n}^{x_{n+1}} K(x_{n+1},\xi,f(\xi))d\xi,$$

dat wil zeggen, we splitsen het rechterlid in een gedeelte waarin alleen de tot en met x_n berekende oplossing f(x) voorkomt en in een gedeelte dat nog niet bekende f-waarden bevat. Hieruit volgt direct de structuur van de numerieke berekeningsmethode. Het eerste gedeelte in (6.1.3), dat we eenvoudigheidshalve met $F_n(x_{n+1})$ aan zullen geven, kan met een standaard quadratuurformule berekend worden gebaseerd op de functie-waarden $K(x_{n+1},x_j,f_j)$, $j = 0,1,\ldots,n$ (f_j stelt hier de gevonden benadering voor $f(x_j)$ voor). Het tweede gedeelte kan ook weer met een standaard quadratuur-

formule gebaseerd op de waarden $K(x_{n+1},x_j,f_j)$, $j = n+1,n,n-1,\ldots n+1-k$, berekend worden waarmee een formule ontstaat waarin de K-waarden allen *lineair* voorkomen. Dergelijke formules worden naar analogie met de differentiaalvergelijkingen *lineaire k-stapsmethoden* genoemd. Een alternatief is om het tweede gedeelte met een quadratuurformule gebaseerd op zogenaamde "non-step points" te berekenen: in de quadratuurformule treden dan K-waarden op die in punten (x,ξ,f) geëvalueerd worden waarbij $x \neq x_j$, $\xi \neq x_j$ en $f \neq f_j$. De resulterende formule heeft veel gemeen met de Runge-Kuttamethoden voor differentiaalvergelijkingen en wordt dan ook een *Runge-Kuttaformule* genoemd. In deze syllabus beperken we ons tot deze twee klassen van methoden. We merken nog op dat de evaluatie van de eerste term in formule (6.1.3), dus de berekening van $F_n(x_{n+1})$, het meest bewerkelijke gedeelte vormt.

Lineaire meerstapsmethoden en Runge-Kuttamethoden leveren inderdaad algoritmen voor Volterra-vergelijkingen die "op papier" vrij algemeen toepasbaar zijn. Dat implementaties van deze algoritmen nauwelijks opgenomen zijn in de grotere programmatheken laat zich niet verklaren door een niet van voldoend belang zijn van Volterra-vergelijkingen. Dit type komt veelvuldig voor in systeem-theorie (cf. MILLER [1971,p.63]), parabolische differentiaalvergelijkingen met niet-lineaire randvoorwaarden, kernreactor-kinetica en uiteraard in populatieproblemen in de biologie. Er is dan ook een duidelijke behoefte aan programmatuur op dit gebied.

6.2. Lineaire meerstapsmethoden

Laten we de numerieke benadering van de "functie van het verleden" $F_n(x)$ met $\tilde{F}_n(x)$ aangeven. Een lineaire k-stapsmethode voor de integraalvergelijking (6.1.1) wordt dan gedefinieerd door

$$f_{n+1} = \tilde{F}_n(x_{n+1}) + h \sum_{\ell=0}^{k} \lambda_{n,\ell} K(x_{n+1},x_{n+1-\ell},f_{n+1-\ell}). \tag{6.2.1}$$

De parameters $\lambda_{n,\ell}$ moeten zodanig gekozen worden dat de tweede term in deze formule een voldoend nauwkeurige benadering is van de tweede term in formule (6.1.3). Ontwikkeling in Taylor-reeksen leert dat deze tweede term tot op orde h^{p+1} nauwkeurig is wanneer de coëfficiënten $\lambda_{n,\ell}$ voldoen aan de relaties

$$(6.2.2)\qquad \sum_{\ell=0}^{k} \lambda_{n,\ell} \left(\frac{x_{n-\ell+1}-x_n}{h}\right)^r = \frac{1}{r+1}\,, \qquad r = 0,1,\ldots,p-1.$$

We zullen de methode *consistent* van de orde p noemen wanneer aan (6.2.2) voldaan is.

Ter illustratie beschouwen we een 3-stapsformule met uniforme integratiestappen, i.e.

$$\frac{x_{n-\ell+1}-x_n}{h} = -(\ell-1).$$

We krijgen dan p lineaire consistentie-voorwaarden in 4 onbekenden $\lambda_{n,\ell}$. Stellen we p = 4 dan vinden we de formule

$$(6.2.3)\qquad f_{n+1} = \tilde{F}_n(x_{n+1}) + \frac{1}{24}\,h[9K_{n+1,n+1} + 19K_{n+1,n} - 5K_{n+1,n-1} + K_{n+1,n-2}]$$

waarin we kortheidshalve $K(x_{n+1},x_{n+1-\ell},f_{n+1-\ell}) = K_{n+1,n+1-\ell}$ gesteld hebben. Deze formule vertoont sterke gelijkenis met de 3-staps Adams-Moultonformule voor gewone differentiaalvergelijkingen. Voor vergelijking (6.1.2) zou deze formule luiden

$$(6.2.4)\qquad f_{n+1} = f_n + \frac{1}{24}\,h[9K_{n+1} + 19K_n - 5K_{n-1} + K_{n-2}],$$

waarin $K_{n+1-\ell} = K(x_{n+1-\ell},f_{n+1-\ell})$. We merken hier bij op dat toepassing van formule (6.2.3) op de met het beginwaardeprobleem (6.1.2) equivalente integraalvergelijking (6.1.2') *niet* noodzakelijk tot formule (6.2.4) zal leiden. Dit is alleen het geval wanneer voor vergelijking (6.1.2') zou gelden

$$(6.2.5)\qquad \tilde{F}_n(x_{n+1}) = f_n,$$

hetgeen in het algemeen niet het geval zal zijn, alhoewel dit wel consistent zou zijn met de integraalvergelijking, immers uit de definitie van $F_n(x)$ volgt voor vergelijking (6.1.3')

$$(6.2.5')\qquad F_n(x) = g(x) + \int_{x_0}^{x_n} K(x,\xi,f(\xi))d\xi = f(x_n).$$

In paragraaf 6.5, waar de stabiliteit van de verschillende formules besproken zal worden, komen we op deze kwestie nog terug.

We besluiten deze paragraaf met enkele voorbeelden van meerstapsmethoden. Om ruimte te sparen noteren we de formules door middel van de matrix

$W_{n+1} = (w_{ij})$, $i = 1,\ldots,n+1$, $j = 0,\ldots,i$, waarin w_{ij} het gewicht voorstelt van de functiewaarde $K(x_i,x_j,f_j)$ in de formule voor f_i. Verder nemen we steeds aan dat de coefficienten $\lambda_{n,\ell}$ volgen uit

(6.2.6) $\quad h\lambda_{n,\ell} = w_{n+1,n+1-\ell} - w_{n,n+1-\ell}$.

Trapeziumregel + Herhaald Simpson (cf. DELVES & WALSH [1974,p.155])

$$(w_{ij}) = h \begin{pmatrix} 1/2 & 1/2 & 0 & & & & & & & 0 \\ 1/3 & 4/3 & 1/3 & & & & & & & \\ 1/2 & 5/6 & 4/3 & 1/3 & & & & & & \\ 1/3 & 4/3 & 2/3 & 4/3 & 1/3 & & & & & \\ 1/2 & 5/6 & 4/3 & 2/3 & 4/3 & 1/3 & & & & \\ & & & \ldots & & & & & & \\ 1/3 & 4/3 & 2/3 & 4/3 & \ldots & & 2/3 & 4/3 & 1/3 & 0 \\ 1/2 & 5/6 & 4/3 & 2/3 & 4/3 & \ldots & & 2/3 & 4/3 & 1/3 \\ & & & \ldots & & & & & & \end{pmatrix}$$

Herhaald Simpson + Trapeziumregel

$$(w_{ij}) = h \begin{pmatrix} 1/2 & 1/2 & 0 & & & & & & & 0 \\ 1/3 & 4/3 & 1/3 & & & & & & & \\ 1/3 & 4/3 & 5/6 & 1/2 & & & & & & \\ 1/3 & 4/3 & 2/3 & 4/3 & 1/3 & & & & & \\ 1/3 & 4/3 & 2/3 & 4/3 & 5/6 & 1/2 & & & & \\ & & & \ldots & & & & & & \\ 1/3 & 4/3 & 2/3 & 4/3 & & \ldots & 2/3 & 4/3 & 1/3 & 0 \\ 1/3 & 4/3 & 2/3 & 4/3 & & \ldots & 2/3 & 4/3 & 5/6 & 1/2 \\ & & & & \ldots & & & & & \end{pmatrix}$$

Herhaald Simpson + 3/8- regel

$$(w_{ij}) = h \begin{pmatrix} \cdots & & & & & & & & & & \\ 1/3 & 4/3 & 1/3 & 0 & & & & & & & 0 \\ 3/8 & 9/8 & 9/8 & 3/8 & & & & & & & \\ 1/3 & 4/3 & 2/3 & 4/3 & 1/3 & & & & & & \\ 1/3 & 4/3 & 17/24 & 9/8 & 9/8 & 3/8 & & & & & \\ 1/3 & 4/3 & 2/3 & 4/3 & 2/3 & 4/3 & 1/3 & & & & \\ 1/3 & 4/3 & 2/3 & 4/3 & 17/24 & 9/8 & 9/8 & 3/8 & & & \\ & & & \cdots & & & & & & & \\ 1/3 & 4/3 & & \cdots & & & & & 4/3 & 2/3 & 4/3 & 1/3 & 0 \\ 1/3 & 4/3 & & \cdots & & & & & 4/3 & 17/24 & 9/8 & 9/8 & 3/8 \\ & & & \cdots & & & & & & & & & \end{pmatrix}$$

In deze formule zijn de gewichten voor f_1 niet gespecificeerd: de trapeziumregel is niet voldoende nauwkeurig zodat óf een meerstapsformule óf een hogere orde eenstapsformule voor f_1 gebruikt moet worden. In het eerste geval betekent dit dat de formules voor f_1 en f_2 simultaan opgelost moeten worden omdat ze beiden impliciet zijn in f_1 en f_2. In het tweede geval kan men zijn toevlucht tot Runge-Kutta-achtige methoden nemen die in de volgende paragraaf aan de orde komen.

6.3. Runge-Kuttamethoden

Evenals bij beginwaardeproblemen voor gewone differentiaalvergelijkingen kunnen de (eenstaps-) Runge-Kuttamethoden dienen om meerstapsmethoden te starten. Hiernaast hebben ze echter zeker ook een zelfstandige betekenis voor de numerieke oplossing van Volterra-vergelijkingen. De algemene m-punts-Runge-Kuttamethode heeft de vorm van een "stap-voor-stap-formule binnen een stap-voor-stap-proces":

$$f_{n+1}^{(0)} = f_n,$$

$$(6.3.1) \qquad f_{n+1}^{(j)} = \tilde{F}_n(x_n+\mu_j h) + h \sum_{\ell=0}^{m} \lambda_{j,\ell} K(x_n+\theta_{j,\ell} h, x_n+\nu_{j,\ell} h, f_{n+1}^{(\ell)})$$

$$f_{n+1} = \tilde{F}_n(x_{n+1}) + h \sum_{\ell=0}^{m} \lambda_{m,\ell} K(x_n+\theta_{m,\ell} h, x_n+\nu_{m,\ell} h, f_{n+1}^{(\ell)}).$$

Hierin dienen de parameters $\mu_j, \lambda_{j,\ell}, \theta_{j,\ell}$ en $\nu_{j,\ell}$ zodanig gekozen worden dat de tweede term in de formule voor f_{n+1} een voldoend nauwkeurige benadering is van de tweede term in formule (6.1.3) (vergelijk de overeenkomstige voorwaarde voor de lineaire meerstapsmethoden (6.2.1)). Ontwikkeling in Taylorreeksen levert weer de relaties voor deze parameters. In het geval van Runge-Kuttamethoden zijn deze relaties echter *niet lineair* en groter in aantal: 2, 6 en 20 relaties voor respectievelijk een benaderingsfout van de orde 2, 3 en 4 in h, d.w.z. een orde van consistentie 1, 2 en 3.

De eerste in de gangbare literatuur bekende algoritme, gebaseerd op het Runge-Kutta-idee, werd in 1960 door Pouzet gegeven. In termen van de parametermatrices $(\lambda_{j,\ell})$, $(\theta_{j,\ell})$, $(\nu_{j,\ell})$ en (μ_j) wordt zijn algoritme gekarakteriseerd door (j = 1(1)m, ℓ = 0(1)m)

$$(\lambda_{j,\ell}) = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1/6 & 1/3 & 1/3 & 1/6 & 0 \end{pmatrix}, \quad (\theta_{j,\ell}) = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix},$$

(6.3.2)

$$(\nu_{j,\ell}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 1 & 0 \end{pmatrix}, \qquad (\mu_j) = \begin{pmatrix} 1/2 \\ 1/2 \\ 1 \\ 1 \end{pmatrix}.$$

Dit schema is consistent van de orde 3. In de practijk is dit schema, en in het algemeen Runge-Kuttamethoden met een orde van consistentie *groter dan 2*, betrekkelijk duur omdat *minstens twee* $\tilde{F}_n$-evaluaties nodig zijn en, zoals in de inleiding al opgemerkt, de $\tilde{F}_n$-evaluaties vormen het hoofdbestanddeel in het rekenschema. De voordelen van Runge-Kuttamethoden boven lineaire meerstapsmethoden moeten gezocht worden in:

(a) Geen "startproblemen"

(b) Flexibele verandering van de stapgrootte h

(c) Vermoedelijk grotere stabiliteitsgebieden (zie §6.5).

In de literatuur is tot dusver weinig aandacht besteed aan Runge-Kuttamethoden voor Volterravergelijkingen. Aan DELVES & WALSH [1974] ontlenen we de referenties: NOBLE [1964], BELTJUKOV [1966] en DAY [1968].

Met name de stabiliteitseigenschappen zijn nauwelijks of niet onderzocht, vermoedelijk omdat men op het standpunt staat dat ze alleen als startformules een functie hebben.

Tenslotte merken we op dat, evenals bij de lineaire meerstapsmethoden, het schema overgaat in een klassieke methode voor *gewone differentiaalvergelijkingen,* in dit geval een Runge-Kuttamethode, indien $\tilde{F}_n(x)$ aan voorwaarde (6.2.5) voldoet en het schema toegepast wordt op de klasse van vergelijkingen (6.1.2). Het schema van Pouzet gaat dan over in de standaard 4e orde methode.

6.4. Convergentie

Laat $\tilde{f}(x)$ een functie zijn die voldoet aan

$$\tilde{f}(x_j) = f_j, \qquad j = 0,1,\ldots . \tag{6.4.1}$$

De in de twee voorgaande paragrafen besproken methoden kunnen dan geschreven worden als

$$f_{n+1} = \tilde{F}_n(x_{n+1}) + \Phi_n(K(x,\xi,\tilde{f}(\xi))), \tag{6.4.2}$$

waarin Φ_n een operator is die uit de functie $K(x,\xi,\tilde{f}(\xi))$ een benadering voor de integraal over het interval $[x_n,x_{n+1}]$ betekent (vergelijk formule (6.1.3)).

STELLING. *Wanneer* $\tilde{F}_n$ en Φ_n *respectievelijk voldoen aan*

$$\tilde{F}_n(x) = g(x) + \int_{x_0}^{x_n} K(x,\xi,(\xi))d\xi + E_n(h)$$

$$\Phi_n(K(x,\xi,f(\xi))) = \int_{x_n}^{x_{n+1}} K(x_{n+1},\xi,f(\xi))d\xi + T_n(h) \tag{6.4.3}$$

waarin $E_n(h) \to 0$ *en* $T_n(h) \to 0$ *als* $h \to 0$, *en wanneer* K *en* Φ_n *respectievelijk voldoen aan*

$$|K(x,\xi,f) - K(x,\xi,\tilde{f})| \le L_1|f-f|$$

$$|\Phi_n(K(x,\xi,f))-\Phi_n(K(x,\xi,\tilde{f}))| \le L_2 h \sum_{\ell=0}^{k} |f(x_{n+1-\ell})-f_{n+1-\ell}| , \tag{6.4.4}$$

waarin L_1 *en* L_2 *de Lipschitz-constanten zijn, dan geldt*

$$(6.4.5) \qquad |f(x_{n+1})-f_{n+1}| \le L_1 \int_{x_0}^{x_n} |f(\xi)-\tilde{f}(\xi)|d\xi + L_2 h \sum_{\ell=0}^{k} |f(x_{n+1-\ell})-\tilde{f}_{n+1-\ell}| + |E_n(h) + T_n(h)|.$$

Voorwaarden (6.4.3) *betekenen dat de integraal "over het verleden" en de integraal"over de toekomst"in formule* (6.1.3) *met de respectieve benaderingsfouten* $E_n(h)$ *en* $T_n(h)$ *berekend worden. Gebruikt men voor de berekening van* $\tilde{F}_n(x)$ *een quadratuurformule van de orde* q *en is het rekenschema consistent van de orde* p, *dan vindt men dus*

$$(6.4.6) \qquad E_n(h) = O(h^{q+1}), \quad T_n(h) = O(h^{p+1}) \qquad \text{als } h \to 0.$$

Uit de stelling kan dan worden afgeleid dat

$$(6.4.7) \qquad |f(x_{n+1})-f_{n+1}| \le O(h^{q+1})+O(h^{p+1}).$$

Wat de voorwaarden (6.4.4) *betreft, deze impliceren dat de kernfunctie* K *voldoende "glad" moet zijn.*

Als voorbeeld passen we het resultaat (6.4.7) toe op de in het voorgaande gegeven formules. Zo vinden we dat de Trapezium- + Herhaalde Simpsonregel een fout van de orde h^3 zal opleveren; dit geldt ook voor de Herhaalde Simpson + Trapezium-regel. De Simpson + 3/8-regel levert een fout van de orde h^4 evenals het Pouzet-schema wanneer $\tilde{F}_n(x)$ hierin met een quadratuurformule met een fout $O(h^4)$ wordt berekend (dus bijvoorbeeld de Simpson + 3/8-regel).

6.5. Stabiliteit

Allereerst onderzoeken we wanneer de integraalvergelijking zélf stabiel is, dat wil zeggen wat gebeurt er met $f(x_{n+1})$, zoals gegeven door formule (6.1.3), wanneer we de functie f(x) op het interval $x_0 \le x < x_{n+1}$ verstoren met een bedrag $\Delta f(x)$. Uit (6.1.3) volgt dat $f(x_{n+1})$ dan verstoord wordt met het bedrag

$$(6.5.1)\qquad \Delta f(x_{n+1}) \cong \int_{x_0}^{x_n} \frac{\partial K}{\partial f}(x_{n+1},\xi,f(\xi))\Delta f(\xi)d\xi + \int_{x_n}^{x_{n+1}} \frac{\partial K}{\partial f}(x_{n+1},\xi,f(\xi))\Delta f(\xi)d\xi$$

$$\cong \int_{x_0}^{x_n} \frac{\partial K}{\partial f}(x_{n+1},\xi,f(\xi))\Delta f(\xi) + h\frac{\partial K}{\partial f}(x_{n+1},x_n,f(x_n))\Delta f(x_n)$$

$$- \int_{x_0}^{x_n} \frac{\partial K}{\partial f}(x_n,\xi,f(\xi))\Delta f(\xi)d\xi + \Delta f(x_n)$$

$$\cong [1+h\frac{\partial K}{\partial f}(x_{n+1},x_n,f(x_n))]\Delta f_n + h\int_{x_0}^{x_n} \frac{\partial^2 K}{\partial f \partial x}(x_n,\xi,f(\xi))\Delta f(\xi)d\xi.$$

Hierin is $\Delta f(x)$ voldoende klein en K voldoende differentieerbaar verondersteld. Uit deze relatie volgt direct dat een nodige voorwaarde voor stabiliteit is

$$(6.5.2)\qquad \frac{\partial K}{\partial f}(x,x,f(x)) \leq 0.$$

Deze voorwaarde is juist de stabiliteitsvoorwaarde voor de differentiaalvergelijking (6.1.1') indien $\partial K/\partial x \equiv 0$.

Wat de stabiliteit van de diverse rekenschema's betreft, in de literatuur is vooral aandacht besteed aan de meerstapsmethoden (MAYERS [1962], KOBAYASI [1966], LINZ [1968], NOBLE [1969]). Hierbij wordt uitgegaan van de modelvergelijking (zie DELVES & WALSH [1974, p.155])

$$(6.5.3)\qquad f(x) = 1 - a\int_0^x f(\xi)d\xi, \qquad a > 0.$$

Het rekenschema vereenvoudigt zich dan tot een lineaire betrekking tussen de waarden f_j, $j = 0,1,\ldots,n+1$ en wanneer met uniforme stapgrootte gewerkt wordt, kan deze betrekking teruggebracht worden tot een relatie waarin nog slechts de "laatste f_j's" voorkomen. Bijvoorbeeld de in paragraaf 6.2 gegeven Trapeziumregel + Herhaald Simpson levert voor vergelijking (6.5.3) (cf. DELVES & WALSH [1974, p.155]) de relatie

$$(6.5.4)\qquad (1+\tfrac{1}{3}ah)f_{n+1} + \tfrac{4}{3}ahf_n - (1-\tfrac{1}{3}ah)f_{n-1} = 0,$$

en dus dezelfde betrekking voor de verstoringen $\Delta f_{n-1}, \Delta f_n$ en Δf_{n+1}. Het is eenvoudig na te gaan dat de hierbij behorende karakteristieke vergelijking één van zijn wortels buiten de eenheidscirkel heeft liggen zodat de methode voor

de modelvergelijking onvoorwaardelijk instabiel is.

De hierboven geschetste aanpak kan ook uitgevoerd worden onder de minder inperkende voorwaarden dat de functie

$$(6.5.5) \qquad J = \frac{\partial K}{\partial f}(x,\xi,f)$$

langzaam varieert in de omgeving van de punten (x_j,x_j,f_j) en dat

$$(6.5.6) \qquad h\left|\frac{\partial^2 K}{\partial x \partial f}(x_j,x_j,f_j)\right| << \left|\frac{\partial K}{\partial f}(x_j,x_j,f_j)\right|,$$

voor $j = 0,1,\ldots,n-1$. Voor lineaire meerstapsmethoden is deze modificatie triviaal, voor Runge-Kuttamethoden is de afleiding nogal technisch en valt buiten het kader van dit colloquium. We volstaan hier met de vermelding van de eindresultaten.

Lineaire meerstapsmethoden

Onder de voorwaarden (6.5.5) en (6.5.6) geldt

$$(6.5.7) \qquad (1-w_{n+1,n+1}\,J_n)\Delta f_{n+1} = [1+(w_{n+1,n}-w_{n,n})\,J_n]\Delta f_n + \sum_{j=0}^{n-1}(w_{n+1,j}-w_{n,j})\,J_n\Delta f_j,$$

waarin h constant verondersteld is en

$$(6.5.8) \qquad J_n = \frac{\partial K}{\partial f}(x_n,x_n,f_n).$$

Ter illustratie passen we (6.5.7) toe op de Herhaalde Simpson + Trapezium-regel. Voor *even* waarden van n volgt uit de matrix W_{n+1} dat

$$(w_{n,j}) = (\tfrac{1}{3}, \tfrac{4}{3}, \tfrac{2}{3}, \ldots, \tfrac{2}{3}, \tfrac{4}{3}, \tfrac{1}{3}, 0, \ldots 0)h$$

$$(w_{n+1,j}) = (\tfrac{1}{3}, \tfrac{4}{3}, \tfrac{2}{3}, \ldots, \tfrac{2}{3}, \tfrac{4}{3}, \tfrac{5}{6}, \tfrac{1}{2}, 0, \ldots 0)h.$$

Hieruit vinden we de recurrente betrekking

$$\Delta f_{n+1} = \frac{1+\frac{1}{2}hJ_n}{1-\frac{1}{2}hJ_n}\Delta f_n,$$

welke voor alle negatieve waarden van hJ_n stabiel is.
Voor *oneven* waarden van n vinden we met gebruikmaking van bovenstaande relatie

$$(1-\tfrac{1}{3}hJ_n)\Delta f_{n+1} = (1+\tfrac{5}{6}hJ_n)\Delta f_n - \tfrac{1}{6}hJ_n\Delta f_{n-1} =$$

$$= \left[1+\tfrac{5}{6}hJ_n - \tfrac{1}{6}hJ_n\,\frac{1-\frac{1}{2}hJ_n}{1+\frac{1}{2}hJ_n}\right]\Delta f_n,$$

ofwel

$$\Delta f_{n+1} = \frac{6+7hJ_n+3h^2J_n^2}{6+hJ_n-h^2J_n^2}\,\Delta f_n.$$

Deze amplificatiefactor is in modulus kleiner dan 1 wanneer (cf. DELVES & WALSH [1974.p.156])

$$-\tfrac{3}{2} < hJ_n < 0.$$

We merken nog op dat voor *oneven* n óók geldt

$$\Delta f_{n+1} = \frac{6+7hJ_n+3h^2J_n^2}{6-5hJ+h^2J_n^2}\,\Delta f_{n-1}$$

en deze relatie is stabiel voor

$$-6 < hJ_n < 0.$$

Runge-Kuttamethoden

Onder de voorwaarden (6.5.5) en (6.5.6) geldt

(6.5.9)
$$\Delta f_{n+1} = Q_m(hJ_n)\Delta f_n + R_m(hJ_n)\Delta\tilde{F}_n(x_n) =$$

$$= [1+Q_m(hJ_n) + w_{n,n}\,J_nR_m(hJ_n)]\Delta f_n +$$

$$+ [(w_{n,n-1}-w_{n-1,n-1})\,J_nR_m(hJ_n) - Q_m(hJ_n)]\Delta f_{n-1} +$$

$$+ \sum_{j=0}^{n-2}(w_{n,j}-w_{n-1,j})\,J_nR_m(hJ_n)\Delta f_j,$$

waarin de functies R_m en Q_m gedefinieerd zijn volgens het schema

$$Q_0(z) = 1, \quad Q_j(z) = \sum_{\ell=0}^{m} \lambda_{j,\ell} z Q_\ell(z)$$

(6.5.10) $$,j = 1,2,\ldots,m.$$

$$R_0(z) = 0, \quad R_j(z) = 1 + \sum_{\ell=0}^{m} \lambda_{j,\ell} z R_\ell(z)$$

Als toepassing beschouwen we de algoritme van Pouzet. Het is eenvoudig te verifieren dat

$$R_4(z) = 1 + \frac{5}{6} z + \frac{1}{3} z^2 + \frac{1}{12} z^3$$

(6.5.10')

$$Q_4(z) = \frac{1}{6} z(1+z+ \frac{1}{2} z^2 + \frac{1}{4} z^3).$$

Substitutie in (6.5.9) en vaststelling van de quadratuurformule voor $\tilde{F}_n(x)$ levert de voor het Pouzet-schema geldende foutenformule. Voor de Herhaalde Simpson + 3/8-regel is door RECKERS [1977] aangetoond dat relatie (6.5.9) stabiel is indien

(6.5.11) $$- 2 < hJ_n < 0.$$

Ter vergelijking zij vermeld dat de standaard 4e orde Runge-Kuttamethode het stabiliteitsinterval $- 2.78 < hJ_n < 0$ bezit.

6.6. Numerieke experimenten

Tot dusver zijn op het Mathematisch Centrum de voor Volterra-vergelijkingen van de tweede soort uitgevoerde experimenten gedaan in het kader van het testen van de stabiliteitstheorie. De verkregen resultaten zijn nog zeer onvolledig; zo zijn alleen nog Runge-Kuttamethoden getest voor een beperkt aantal vergelijkingen. We bespreken hier een viertal methoden toegepast op een viertal vergelijkingen. Hierin werd $\tilde{F}_n(x)$ steeds met de herhaalde trapeziumregel berekend en de stapgrootte constant gehouden. De experimenten zijn uitgevoerd door F.J. Reckers en J. Schilder.

I. *Expliciete, 2-puntsformule van de tweede orde*

$$(\lambda_{j,\ell}) = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 0 \end{pmatrix}, \quad (\theta_{j,\ell}) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \quad (\nu_{j,\ell}) = \begin{pmatrix} 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 \end{pmatrix}, \quad (\mu_j) = \begin{pmatrix} 1 \\ 1 \end{pmatrix};$$

$E_n(h) = O(h^2)$, $T_n(h) = O(h^3)$;
Stabiliteitsvoorwaarde $-2 < hJ_n < 0$.

II. *Expliciete, 3-puntsformule van de tweede orde*

$$(\lambda_{j,\ell}) = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 1-\lambda_2 & \lambda_2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \end{pmatrix}, \quad (\theta_{j,\ell}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \quad (\nu_{j,\ell}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$(\mu_j) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix};$$

$E_n(h) = O(h^2)$, $T_n(h) = O(h^3)$;
Stabiliteitsvoorwaarden:

a: $\lambda_1 = \frac{2}{3} \quad \lambda_2 = \frac{1}{2}$: $\quad -1.74 < hJ_n < 0$.

b: $\lambda_1 = \dfrac{1+hJ_n+\frac{1}{2}h^2J_n^2}{hJ_n(1+\frac{1}{2}hJ_n)}$, $\quad \lambda_2 = -\dfrac{2+hJ_n}{h^2J_n^2}$: $-\infty < hJ_n < 0$.

III. *Zwak impliciete, 2-puntsformule van de tweede orde*

$$(\lambda_{j,\ell}) = \begin{pmatrix} 1/6 & 1/6 & 0 \\ 0 & 3/4 & 1/4 \end{pmatrix}, \quad (\theta_{j,\ell}) = \begin{pmatrix} 1 & -1/3 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad (\nu_{j,\ell}) = \begin{pmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 1 \end{pmatrix},$$

$$(\mu_j) = \begin{pmatrix} 1/3 \\ 1 \end{pmatrix};$$

$E_n(h) = O(h^2)$, $T_n(h) = O(h^4)$;
Stabiliteitsvoorwaarde: $-2 < hJ_n < 0$.

We merken hierbij nog op dat de orde van deze formules *één* hoger wordt in de gevallen I en II en *twee* hoger wordt in het geval III, wanneer $\tilde{F}_n(x)$ met de Simpson + 3/8-regel berekend zou worden.

De gekozen vergelijkingen zijn

A: $f(x) = x^2 + \frac{1}{7}x^7 - \int_0^x f^3(\xi)d\xi$; oplossing $f(x) = x^2$.

B: $f(x) = \frac{1}{10}x^2 + \frac{1}{7}x^7 - 1000\int_0^x f^3(\xi)d\xi$; oplossing $f(x) = \frac{1}{10}x^2$.

C: $f(x) = x^2 + \frac{1}{7}x^8 - x\int_0^x f^3(\xi)d\xi$; oplossing $f(x) = x^2$.

$$D:\ f(x) = \frac{1}{10}x^2 + \frac{1}{7}x^8 - 1000\,x\int_0^x f^3(\xi)d\xi;\ \text{oplossing } f(x) = \frac{1}{10}x^2.$$

De eerst twee problemen zijn equivalent met een beginwaardeprobleem voor een gewone differentiaalvergelijking (cf. vergelijking (6.1.1')). Het derde en vierde probleem zijn modificaties van de eerste twee, maar niet meer te reduceren tot gewone differentiaalvergelijkingen. Problemen B en D zou men *stijve* integraalvergelijkingen kunnen noemen ($J = \frac{\partial K}{\partial f} << -1$) naar analogie met stijve differentiaalvergelijkingen. In tabel 6.6.1 zijn de stabiliteitsvoorwaarden voor deze problemen en de methoden I, IIa,b en III bijeen gezet, waarbij er van gebruik is gemaakt dat we de oplossing van deze testproblemen kennen. Uit deze tabel volgt dat voor constante h het integratieproces na een zekere waarde van x instabiel zou moeten worden. In tabel 6.6.2 zijn de experimenteel verkregen resultaten opgenomen uitgedrukt in het aantal correcte cijfers. De van een * voorziene waarden in deze tabel zijn de laatste resultaten die volgens voorgaande theorie nog stabiel zijn; daarna wordt het proces instabiel.

Tabel 6.6.1.

Stabiliteitsvoorwaarden

Methode	A	B	C	D
I	$hx^4 < 2/3$	$hx^4 < 2/30$	$hx^5 < 2/3$	$hx^5 < 2/30$
IIa	$hx^4 < 1.74/30$	$hx^4 < 1.74/30$	$hx^5 < 1.74/30$	$hx^5 < 1.74/30$
IIb	$hx < \infty$	$hx < \infty$	$hx < \infty$	$hx < \infty$
III	$hx^4 < 2/3$	$hx^4 < 2/30$	$hx^5 < 2/3$	$hx^5 < 2/30$

Tabel 6.6.2.

Numerieke resultaten

x	I				IIa				IIb				III			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
h = 1/10																
.6	2.9	1.8	3.2	2.1	3.0	2.2	3.2	2.3	2.8	1.1	3.1	1.6	3.4	2.5	3.6	2.6
.8	2.5	1.1*	2.6	1.3*	2.7	1.3*	2.7	1.7*	2.2	-	2.4	-	2.9	2.4*	3.0	2.4*
1.0	2.2	-	2.1	-	2.6	1.2	2.5	.9	1.5	-	1.5	-	2.7	2.6	2.7	2.6
1.2	1.8	-	1.7	-	2.6	.2	2.1	.7	.7	-	.4	-	2.6	1.5	2.6	1.4
1.4	1.4	-	1.0*	-	1.5*	-	1.1*	-	-	-	-	-	2.7	.6	2.9*	.6
1.6	.6*	-	-	-	1.2	-	1.2	-	-	-	-	-	3.5*	.1	2.4	.3
1.8	-	-	-	-	1.2	-	1.2	-	-	-	-	-	2.4	-	1.5	-
2.0	-	-	-	-	.9	-	-	-	-	-	-	-	1.6	-	.6	-
2.4	-	-	-	-	-	-	-	-	-	-	-	-	.5	-	-	-
h = 1/20																
.8	3.2	2.0	3.3	2.1	3.2	2.5	3.3	3.1	3.0	.7	3.1	1.1	3.4	2.7	3.4	2.7
1.0	2.9	1.2*	2.9	1.2*	3.1	1.4*	3.1	1.4*	2.4	-	2.4	-	3.1	3.0*	3.1	3.1*
1.2	2.6	-	2.5	-	3.5	1.4	4.1	1.4	1.8	-	1.5	-	3.1	2.4	3.0	2.1
1.4	2.4	-	2.1	-	2.5	.9	2.0	-	.8	-	-	-	3.1	.9	3.1	.7
1.6	2.0	-	1.3*	-	1.9	-	1.6	-	-	-	-	-	3.2	.1	3.6*	-
1.8	1.4*	-	-	-	1.7*	-	1.5*	-	-	-	-	-	3.6*	-	2.7	-
2.0	-	-	-	-	1.6	-	.9	-	-	-	-	-	3.0	-	1.2	-
2.2	-	-	-	-	1.7	-	-	-	-	-	-	-	1.9	-	.7	-
2.4	-	-	-	-	0.9	-	-	-	-	-	-	-	.9	-	-	-

6.7. Alternatieve integratieformules

In de bespreking van meerstaps- en Runge-Kuttaformules hebben we al opgemerkt dat deze formules, indien toegepast op de klasse van vergelijkingen (6.1.2), reduceren tot de klassieke meerstaps- en Runge-Kuttaformules voor gewone differentiaalvergelijkingen op voorwaarde dat de formule voor $\tilde{F}_n(x)$ voldoet aan

$$\tilde{F}_n(x_{n+1}) = f_n. \tag{6.7.1}$$

Indien we dergelijke formules voor $\tilde{F}_n(x)$ zouden gebruiken, mogen we verwachten dat de stabiliteitsvoorwaarden identiek zullen zijn met de stabiliteitsvoorwaarden van het gereduceerde schema. Past men het Pouzet-schema toe met een $\tilde{F}_n(x)$-formule die aan (6.7.1) voldoet, dan valt te verwachten dat het stabiliteitsinterval gelijk is aan dat van de klassieke standaard 4e orde formule, derhalve (-2.78,0) in plaats van (-2.0,0) zoals gegeven in (6.5.11). Onderzoek van de andere Runge-Kuttamethoden die in dit hoofdstuk ter sprake zijn gekomen (de methoden I, IIa, IIb en III uit de vorige paragraaf) laat zien dat ze onder voorwaarde (6.7.1) voor de klasse van vergelijkingen (6.1.2') reduceren tot Runge-Kuttamethoden met gelijke of grotere stabiliteitsintervallen; we vinden (-2,0), (-2.51,0), ($-\infty$,0) en (-11,0) in plaats van de intervallen (-2,0), (-1.74,0), ($-\infty$,0) en (-2,0) zoals genoemd in paragraaf 6.6. Ook andere methoden laten steeds tenminste gelijke, meestal grotere stabiliteitsintervallen zien. Zo heeft RECKERS [1977] aangetoond dat de formule van BELTJUKOV (cf. DELVES & WALSH [1974, p.153]) en formule IIa waarin $\tilde{F}_n(x)$ met de Herhaalde Simpson + 3/8-regel geëvalueerd wordt de respectieve stabiliteitsintervallen (-1.27,0) en (-1.52,0) bezitten waar voorwaarde (6.7.1) in beide gevallen aanleiding geeft tot het stabiliteitsinterval (-2.51,0).

Bovenstaand getallenmateriaal was voor ons voldoende reden om na te gaan of er formules voor $\tilde{F}_n(x)$ te vinden zijn die aan (6.7.1) voldoen en wat de consequenties daarvan zijn. Met weglating van alle meer technische details zullen we nu een overzicht van de resultaten van deze analyse geven.

(a) Laat (w_{ij}) weer de matrix van gewichten zijn geassocieerd aan een quadratuurformule voor $K(x,\xi,\tilde{f}(\xi))$ over het interval $x_0 \le \xi \le x_i$, dan is

$$\tilde{F}_n(x) = f_n + g(x)-g(x_n) + \sum_{m=0}^{n} w_{nj}[K(x,x_j,f_j) - K(x_n,x_j,f_j)] \tag{6.7.2}$$

een consistente benadering van $F_n(x)$.

(b) Indien de in (6.7.2) gebruikte quadratuurformule een fout $O(h^{q+1})$ heeft en de fout $T_n(h)$ van de orde h^{p+1} is (zie (6.4.3)) dan volgt uit stelling 6.4.1

$$|f(x_{n+1}) - f_{n+1}| \leq O(h^{q+1}) + O(h^p). \tag{6.7.3}$$

(c) Indien voorwaarde (6.5.5) geldt en indien

$$h\left|\frac{\partial^2 K}{\partial x \partial f}(x, x_j, f_j)\right| << 1 \tag{6.7.4}$$

voor $j = 0,1,\ldots,n-1$ en x in de omgeving van x_n, dan geldt voor de lineaire meerstapsmethode (6.2.1)

$$(1-\lambda_{m,0}hJ_n)\Delta f_{n+1} = (1+\lambda_{n,1}hJ_n)\Delta f_n + \sum_{\ell=2}^{k} \lambda_{n,\ell} hJ_n \Delta f_{n+1-\ell} \tag{6.7.5}$$

en voor de Runge-Kuttamethode (6.3.1)

$$\Delta f_{n+1} = [Q_m(hJ_n) + R_m(hJ_n)]\Delta f_n, \tag{6.7.6}$$

waarin Q_m en R_m door (6.5.10) gegeven worden (de relaties (6.7.5) en (6.7.6) zijn inderdaad identiek met de relaties die men voor de corresponderende meerstaps- en Runge-Kuttaformules voor gewone differentiaalvergelijkingen vindt).

In tabel 6.7.1 zijn de resultaten weergegeven van dezelfde experimenten als besproken in de vorige paragraaf (zie tabel 6.6.2), alleen met dit verschil dat $\tilde{F}_n(x)$ volgens formule (6.7.2) werd berekend waarin de gewichten w_{ij} echter weer die van de Herhaalde Trapeziumregel zijn.

Tabel 6.7.1.

Numerieke resultaten

x	I				IIa				IIb				III			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
h = 1/10																
.6	2.9	1.8	3.2	2.1	3.0	2.2	3.2	2.3	2.8	1.1	3.1	1.6	4.3	3.4	4.1	3.1
.8	2.5	1.0*	2.6	1.2	2.7	1.2*	2.7	1.6*	2.1	.0	2.3	.2	4.1	3.7	3.6	3.0
1.0	2.1	.1	2.1	.2	2.6	.4	2.6	.4	1.4	-	1.5	-	4.0	3.7	3.3	2.6*
1.2	1.6	-	1.5	-	2.2	-	1.9	-	.7	-	.5	-	4.2	3.2*	3.2	2.2
1.4	1.2	-	1.0*	-	1.2	-	.9*	-	.2	-	.0	-	5.6	2.9	3.6	2.2
1.6	.7*	-	.1	-	.7*	-	.7	-	-	-	-	-	4.1	2.6	3.6	1.8
1.8	.2	-	-	-	.6	-	.1	-	-	-	-	-	3.8	2.2	3.0	1.4
2.0	-	-	-	-	1.2	-	-	-	-	-	-	-	3.6	1.8	2.7*	0.9
2.4	-	-	-	-	-	-	-	-	-	-	-	-	3.2*	.8	2.0	-
h = 1/20																
.8	3.1	1.7	3.2	1.9	3.2	2.5	3.3	2.6	2.8	.8	3.0	1.0	5.0	4.7	4.1	3.5
1.0	2.7	1.1*	2.7	1.1*	3.1	1.4*	3.1	.9*	2.1	.1	2.2	.1	5.0	4.4	3.8	3.9
1.2	2.3	-	2.2	-	3.3	1.4	2.9	.6	1.5	-	1.3	-	5.1	4.0	3.8	3.6
1.4	2.0	-	1.8	-	2.1	.9	1.7	-	.8	-	.5	-	5.7	3.7	3.9	3.0*
1.6	1.6	-	1.2*	-	1.4	-	.9*	-	.4	-	.1	-	4.9	3.4*	4.5	2.6
1.8	1.2*	-	.2	-	.9	-	.7	-	.1	-	-	-	4.7	3.0	4.2	1.9
2.0	.5	-	-	-	.7*	-	.5	-	-	-	-	-	4.5	2.4	3.7	.9
2.2	-	-	-	-	-	-	-	-	-	-	-	-	4.3	1.6	3.4*	.2
2.6	-	-	-	-	-	-	-	-	-	-	-	-	4.0*	.2	2.5	-
3.0	-	-	-	-	-	-	-	-	-	-	-	-	3.6	-	1.0	-

REFERENTIES

ATKINSON, K. [1976], *An automatic program for linear Fredholm integral equations of the second kind,* ACM Transactions on Mathematical Software 2, 154-171.

BELTJUKOV, B.A. [1966], *An analogue of the Runge-Kutta method for the solution of nonlinear integral equations of Volterra type,* Differential Equations 1, 417-26.

DAY, J.T. [1968], *On the numerical solution of Volterra integral equations,* BIT 8, 134-37.

DELVES, L.M. & J. WALSH (ed.) [1974], *Numerical solution of integral equations,* Clarendon Press, Oxford.

KOBAYASI, M. [1966], *On numerical solution of the Volterra integral equations of the second kind by linear multistep methods,* Rep. Stat. App. Res. JUSE 13, 1-21.

LINZ, P. [1968], *The numerical solution of Volterra integral equations by finite difference methods,* M.R.C. Tech. Rept. 825.

MAYERS, D.F. [1962], *Integral equations of Volterra type,* in: FOX, L. (ed.), *Numerical solution of ordinary and partial differential equations,* Pergamom Press, 165-173.

MILLER, R.K. [1971], *Nonlinear Volterra integral equations,* W.A. Benjamin, Inc., California.

NOBLE, B. [1964], *The numerical solution of nonlinear integral equations and related topics,* in: ANSELONE, P.M. (ed.), *Nonlinear integral equations,* Madison, The University of Wisconsin Press, 215-318.

NOBLE, B. [1969], *Instability when solving Volterra integral equations of the second kind by multistep methods,* in: MORRIS, J.Ll. (ed.), *Conference on the numerical solution of differential equations,* Springer, 22-39.

POUZET, P. [1960], *Méthode d'intégration numérique des équations intégrales et intégro-différentielles du type Volterra de seconde espèce, Formules de Runge-Kutta,* in: *Symposium on the Numerical treatment of ordinary differential equations, integral and integro-differential equations,* Birkhäuser Verlag, 362-368.

RECKERS, F.J. [1977], *Stabiliteit van Runge-Kuttamethoden voor Volterra-integraalvergelijkingen van de tweede soort*, te verschijnen in de NN-serie van het Mathematisch Centrum, Amsterdam.

7. REGULARISATIEMETHODEN VOOR INTEGRAALVERGELIJKINGEN VAN DE EERSTE SOORT

door H.J.J. te Riele
(Mathematisch Centrum)

7.1. Inleiding

Integraalvergelijkingen van de eerste soort zijn functionaalvergelijkingen waarin de onbekende functie onder een integraalteken voorkomt, en nergens anders in de vergelijking. We beschouwen hier twee belangrijke klassen van integraalvergelijkingen van de eerste soort, nl. de lineaire *Fredholmvergelijking*

$$(7.1.1) \qquad \int_a^b K(x,y)f(y)dy = g(x) \qquad (c\leq x\leq d),$$

en de lineaire *Volterravergelijking*

$$(7.1.2) \qquad \int_a^x K(x,y)f(y)dy = g(x) \qquad (a\leq x\leq b);$$

hierin is f de onbekende functie, terwijl K (de *kern*) en g zijn gegeven. In het vervolg zullen we g soms de *datafunctie* noemen. In (7.1.1) is de kern gedefinieerd op de rechthoek $c \leq x \leq d$, $a \leq y \leq b$ van het (x,y)-vlak, in (7.1.2) op de driehoek $a \leq y \leq x \leq b$. Wanneer men voor de kern in (7.1.2) definieert

$$K_1(x,y) = \begin{cases} K(x,y) & (a\leq y\leq x\leq b), \\ 0 & (a\leq x\leq y\leq b), \end{cases}$$

dan kan men de Volterravergelijking als een speciale Fredholmvergelijking met kern K_1 beschouwen. In het algemeen zal echter K_1 discontinu zijn op de

lijn x = y, zoal niet zelf, dan toch in een van zijn afgeleiden; daarom maken wij onderscheid tussen (7.1.1) en (7.1.2).

Het is soms van nut om (7.1.1) en (7.1.2) in operatorvorm te schrijven: zij X en Y gegeven ruimten, K: X → Y een lineaire integraaloperator zoals in (7.1.1) of (7.1.2), en g ∈ Y, dan zoeken we f ∈ X zó dat geldt

(7.1.3) $\quad Kf = g.$

Indien K(x,x) ≠ 0 kan men (7.1.2) door differentiëren omzetten in een integraalvergelijking van Volterra van de tweede soort:

$$(7.1.4) \qquad K(x,x)f(x) + \int_a^x K'_x(x,y)f(y)dy = g'(x) \qquad (a \le x \le b).$$

Indien K(x,x) ≡ 0, dan is (7.1.4) weer van de vorm (7.1.2) en men kan het proces van differentiëren herhalen tot er een r ∈ ℕ gevonden is, waarvoor geldt

$$\left[\left(\frac{\partial}{\partial x}\right)^r K(x,y)\right]_{y=x} \neq 0;$$

voor practische toepassingen is deze methode niet zo geschikt, omdat g(x) in het algemeen niet exact bekend is; vaak zelfs is g(x) alleen in tabelvorm met inexacte data gegeven. We zullen ons hier dan ook met name beperken tot die methoden voor het numeriek oplossen van (7.1.1) en (7.1.2), die rekening houden met de mogelijkheid *dat g in tabelvorm en met beperkte nauwkeurigheid gegeven is.*

De programmatheken ACCULIB, IMSL, NAG en NUMAL bevatten geen routines voor de hier beschreven problemen. Wel worden in de literatuur vele algoritmen voorgesteld. Sommige auteurs geven enkele resultaten van (soms oppervlakkige) numerieke experimenten, anderen laten deze zelfs weg en volstaan met de opmerking dat "de resultaten van numerieke experimenten met de voorgestelde algoritme bemoedigend zijn"!

Uit de rijstebrij van methoden hebben we de zgn. regularisatiemethode van Phillips en Tihonov gekozen en deze in ALGOL 60 geprogrammeerd. De tekst van de procedure, alsmede een korte beschrijving vindt men in de bijlagen bij dit hoofdstuk.

In §7.2 zullen we iets vertellen over de herkomst van integraalvergelijkingen van de eerste soort. Deze vergelijkingen behoren tot de klasse van zgn. onjuist gestelde problemen, waarover §7.3 handelt. In §7.4 wordt de

regularisatiemethode van Phillips en Tihonov besproken. Andere methoden met verwijzingen naar de literatuur behandelen we in §7.5. §7.6 bevat resultaten van experimenten met de regularisatiemethode van Phillips en Tihonov.

7.2. Herkomst van integraalvergelijkingen van de eerste soort

Integraalvergelijkingen van de eerste soort treden veelvuldig op bij de mathematische analyse van problemen uit verschillende terreinen van de fysica en de biologie, zoals de astrofysica, atoomfysica, biofysica, fysiologische fysica, geofysica, hydrodynamica, kernfysica, plasmafysica, fysica van de vaste stof, en de statistische mechanica. Opsommingen van concrete problemen, met verwijzingen naar de bronnen, zijn bv. te vinden in HILGERS [1974] en NEDELKOV [1972].

Een belangrijke klasse van Volterravergelijkingen is die waarbij de kern een functie is van x - y. Men noemt deze integraalvergelijkingen van het *convolutie*type. Een voorbeeld hiervan, uit de fysiologische fysica, is het probleem van het bepalen van de impedantie Z van het arteriële systeem, als functie van de tijd t, door het meten van de druk P(t) en de stroom F(t). Het verband wordt gegeven door de vergelijking

$$\int_0^t F(t-\tau)Z(\tau)d\tau = P(t) \qquad (0\leq t\leq T).$$

Een zeer belangrijk aspect hierbij is het feit dat de nauwkeurigheid van de metingen nadelig wordt beïnvloed door de gebruikte instrumenten. In de practijk houdt men hier rekening met een meetonnauwkeurigheid van enkele procenten. Numerieke berekeningen aan dit probleem vindt men in §7.6.

Een tweede voorbeeld (uit de meteorologie) is het volgende (zie SHIFRIN c.s. [1966]). Men zoekt naar de verdeling van de stralen van (bolvormig verondersteld) deeltjes in een of andere suspensie. Men meet nu de verdeling van de intensiteit van verstrooid licht dat op de suspensie valt: deze is gerelateerd aan de verdeling van de stralen van de deeltjes door een Fredholm-integraalvergelijking van de eerste soort met kern

$$K(x,y) = \left(\frac{\sin xy}{xy} - \cos xy\right)^2.$$

Numerieke berekeningen aan dit probleem vindt men in §7.6.

Een derde voorbeeld is de numerieke berekening van de afgeleide van een functie, die experimenteel wordt gemeten. Dit probleem kan men formuleren

als een Volterravergelijking van de eerste soort in de vorm

$$\int_a^x f(y)dy = g(x);$$

de kern is hier dus identiek gelijk aan 1. Er moet natuurlijk gelden g(a) = 0, maar dat is geen beperking van de algemeenheid. CULLUM [1971] formuleert het probleem als een *Fredholm*vergelijking van de eerste soort in de vorm

$$\int_0^1 h(x-y)f(y)dy = g(x) - g(0) \qquad (0\leq x\leq 1),$$

waarbij h(x) de eenheidsstapfunctie is.

7.3. Onjuist gestelde problemen

Een belangrijk aspect van integraalvergelijkingen van de eerste soort is het feit dat ze tot de klasse van *onjuist gestelde* problemen (Engels: ill-posed, improperly posed) behoren. Om dit nader toe te lichten geven we eerst de definitie van een *juist gesteld* probleem (in de zin van Hadamard, zie LAVRENTIEV [1967]).

DEFINITIE 7.3.1. Zij X en Y volledige metrische ruimten, K: X → Y een operator van X in Y en zij g ∈ Y gegeven. Het probleem: "*Bepaal* f *uit* X *waarvoor* Kf = g" heet juist gesteld als voor iedere g ∈ Y geldt:

(i) f bestaat,

(ii) f is uniek in X, en

(iii) de oplossing van de vergelijking Kf = g hangt continu af van de data, m.a.w. de inverse operator K^{-1}: R(K) → X is begrensd.

Een onjuist gesteld probleem is een probleem dat niet juist gesteld is in de zin van definitie 7.3.1.

Beschouw nu bv. de Fredholmvergelijking (7.1.1) en stel dat deze een unieke oplossing f_0 bezit. Volgens een stelling van Riemann-Lebesgue (zie NATANSON [1956, p.281]) geldt voor iedere Lebesgue-integreerbare functie K(x,y) dat

$$\lim_{m\to\infty} \int_a^b K(x,y)\sin my\,dy = 0.$$

Substitueren we nu in (7.1.1) $f_0(y)$ + C sin my dan kunnen we dus de afwijking

van het rechterlid t.o.v. g(x) zo klein maken als we maar willen, als we m maar groot genoeg kiezen. Een willekeurig kleine verstoring van het rechterlid kan dus afkomstig zijn van iedere eindige verstoring van f_0, m.a.w. de oplossing van (7.1.1) hangt niet continu af van de datafunctie g.

Een ander belangrijk aspect is het feit, zoals al eerder betoogd, dat we, wanneer we (7.1.1) of (7.1.2) numeriek willen oplossen, altijd te maken hebben met inexacte data. Normaal gesproken tengevolge van meetonnauwkeurigheid van de meetinstrumenten, in het ideale geval ten gevolge van het introduceren van een afbreekfout bij het (vroeg of laat) discretiseren van de integraal. Het feit dat eerste soort integraalvergelijkingen onjuist gesteld zijn maakt daarom het numeriek oplossen ervan zo moeilijk. Bij het behandelen van enkele methoden zullen we ons met name interesseren voor die methoden die nog acceptabele resultaten afleveren bij een meetonnauwkeurigheid van 1-5% in de datafunctie.

7.4. De regularisatiemethode van Phillips en Tihonov

Onafhankelijk van elkaar hebben PHILLIPS [1962] een TIHONOV [1963a, 1963b] een methode voorgesteld voor het numeriek oplossen van (7.1.1). Hoewel de uitgangspunten verschillen zijn de twee methoden in essentie gelijk. Tihonov's aanpak is theoretisch beter gefundeerd, Phillips benadrukt meer, aan de hand van numerieke voorbeelden, het practische nut van zijn aanpak. TWOMEY [1963] verbeterde de efficientie van Phillips' aanpak aanzienlijk, door twee benodigde matrixinversies terug te brengen tot één.

Beschouw het volgende, *onjuist* gestelde probleem:

(7.4.1)
$$\begin{cases} \text{bepaal } f(x) \text{ uit } \int_a^b K(x,y)f(y)dy = g(x), \qquad c \le x \le d, \\ \text{waarbij } K \text{ gegeven is; de functie } g \text{ is niet exact bekend, wel} \\ \text{is gegeven een benadering } \tilde{g} \text{ van } g \text{ en een getal } \delta > 0 \text{ zó dat} \\ \|\tilde{g}-g\| \le \delta. \end{cases}$$

De hierbij gebruikte norm is de L_2-norm, d.w.z. $\|\phi\| = (\int_0^1 \phi^2(x)dx)^{\frac{1}{2}}$. De regularisatiemethode van Phillips en Tihonov bestaat in essentie hieruit, dat probleem (7.4.1) wordt vervangen door het volgende, *juist* gestelde probleem:

(7.4.2)
$$\begin{cases} \text{minimaliseer de kwadratische functionaal} \\ \Phi_\alpha(f) = \|Kf-\tilde{g}\|^2 + \alpha\|Lf\|^2 \\ \text{over alle } f \text{ uit een of andere } \textit{compacte} \text{ verzameling,} \\ \text{waarvoor } \|Kf-\tilde{g}\| \le \delta. \end{cases}$$

Hierbij is $(Kf)(x) = \int_a^b K(x,y)f(y)dy$ voor $x \in [c,d]$, α is een vast positief getal, de zgn. *regularisatieparameter* en L is een of andere lineaire operator, bv. Lf = f, f' of f", of, als een benadering $\hat{f}$ van f bekend is, $Lf = f - \hat{f}$. Algemener kan men $\|Lf\|^2$ in (7.4.2) vervangen door een lineaire combinatie van de kwadraten van de normen van verschillende afgeleiden van f,

$$\sum_{j=0}^{p} \int_a^b a_j(x) \left[\frac{d^j f(x)}{dx^j}\right]^2 dx,$$

waarbij $p \geq 0$ en $a_j(x) \geq 0$ $\forall x \in [a,b]$. Hierin noemt men p wel de *orde* van het regularisatieproces.

Onder bepaalde, milde voorwaarden heeft (7.4.2) een unieke oplossing, die we f_α noemen. Bovendien zal, als δ naar nul gaat en als α voldoet aan

$$(7.4.3) \qquad C_1\delta^2 < \alpha < C_2\delta^2,$$

waarbij C_1 en C_2 positieve getallen zijn, f_α uniform op [a,b] naar de oplossing van de vergelijking $Kf = g$ convergeren. Helaas is g niet exact bekend en doordat (7.4.1) een onjuist gesteld probleem is, zal de oplossing van de vergelijking $Kf = \tilde{g}$ (d.w.z. de oplossing van probleem (7.4.2) voor $\alpha \to 0$) in het algemeen sterk oscilleren rond de oplossing van de vergelijking $Kf = g$. Een toename van α zal in het algemeen een toename van het residu $\|Kf_\alpha - \tilde{g}\|$ tot gevolg hebben, en een afname van de grootheid $\|Lf_\alpha\|$; en omgekeerd. Wanneer L geschikt gekozen wordt zal $\|Lf_\alpha\|$ bij stijgende α een toenemend dempend effect op de ongewenste oscillaties van f_α uitoefenen. Dit werpt de vraag op: hoe groot moet α worden gekozen? De keuze (7.4.3) is voor de theorie weliswaar van belang, in de practijk is deze formule niet goed bruikbaar. In ieder geval zal α zó gekozen moeten worden dat de waarden van het residu $\|Kf_\alpha - \tilde{g}\|$ en van de grootheid $\|Lf_\alpha\|$ (die bv. de gladheid van f meet) voor de gebruiker *beide* acceptabel zijn. In een later stadium komen we hier nog op terug (zie §7.6).

Teneinde (7.4.2) numeriek op te lossen voeren we de volgende discretisatie uit: zij $x_i = c + ih_2$ $(i = 0,1,\ldots,N_2;\ h_2 = (d-c)/N_2)$ en $y_j = a + jh_1$ $(j = 0,1,\ldots,N_1;\ h_1 = (b-a)/N_1)$ en zij

$$(7.4.4) \qquad \int_a^b K(x,y)f(y)dy \simeq \sum_{j=0}^{N_1} w_j K(x,y_j)f(y_j),$$

waarbij w_j $(j = 0,1,\ldots,N_1)$ de gewichten zijn van een of andere geschikt gekozen kwadratuurformule. Zij verder $(Kf-\tilde{g})(x) = \varepsilon(x)$, $\varepsilon(x_i) = \varepsilon_i$,

$f(y_j) = f_j$, $\hat{f}(y_j) = \hat{f}_j$, $\tilde{g}(x_i) = g_i$ en $w_j K(x_i,y_j) = K_{ij}$. Voor Lf kiezen we

$$(7.4.5) \qquad Lf := a_0(f-\hat{f}) + a_1 f' + a_2 f''$$

waarbij $a_i = 0$ of 1 en $\hat{f}$ is een a priori bekende benadering van de gezochte f. Als deze niet bekend is kieze men $\hat{f} \equiv 0$. We vervangen probleem (7.4.2) nu door het volgende:

$$(7.4.6) \qquad \begin{cases} \text{minimaliseer de kwadratische functionaal} \\ \bar{\Phi}_\alpha(\vec{f}) = \sum_{i=0}^{N_2} \varepsilon_i^2 + \alpha\Big\{a_0 \sum_{i=0}^{N_1} (f_i-\hat{f}_i)^2 + a_1 \sum_{i=0}^{N_1-1} (f_{i+1}-f_i)^2 + \\ \qquad\qquad + a_2 \sum_{i=1}^{N_1-1} (f_{i+1}-2f_i+f_{i-1})^2\Big\} \\ (\text{met } \varepsilon_i = \sum_{j=0}^{N_1} K_{ij} f_j - g_i) \quad \text{over alle vectoren} \\ \vec{f} = [f_0,f_1,\ldots,f_{N_1}]^T \in \mathbb{R}^{N_1+1}, \text{waarvoor } \sum_{i=0}^{N_2} \varepsilon_i^2 \le \delta^2 \text{ is.} \end{cases}$$

Uit de voorwaarde $\partial\bar{\Phi}_\alpha/\partial f_k = 0$ $(k = 0,1,\ldots,N_1)$ volgt na enig rekenwerk de matrix-vectorvergelijking

$$(7.4.7) \qquad [K^T K + \alpha(a_0 H_0 + a_1 H_1 + a_2 H_2)]\vec{f} = K^T\vec{g} + \alpha a_0 \vec{\hat{f}},$$

waarbij

$$\vec{g} = [g_0,g_1,\ldots,g_{N_2}]^T, \quad \vec{\hat{f}} = [\hat{f}_0,\hat{f}_1,\ldots,\hat{f}_{N_1}]^T,$$

$$K = (K_{ij}), \quad K^T = (K_{ji}), \quad H_0 = I_{N_1} \text{ (de } N_1 \times N_1\text{-eenheidsmatrix)},$$

$$H_1 = \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & \bigcirc & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & \bigcirc & & & -1 & 1 \end{bmatrix}_{N_1\times N_1} \text{en } H_2 = \begin{bmatrix} 1 & -2 & 1 & & & \\ -2 & 5 & -4 & 1 & & \bigcirc \\ 1 & -4 & 6 & -4 & 1 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & 1 & -4 & 6 & -4 & 1 \\ & \bigcirc & & 1 & -4 & 5 & -2 \\ & & & & 1 & -2 & 1 \end{bmatrix}_{N_1\times N_1}.$$

Resultaten van numerieke experimenten met schema (7.4.7) kan men vinden in §7.6. Hoewel Phillips en Tihonov hun methode hebben afgeleid voor *Fredholm*-vergelijkingen van de eerste soort, kan de methode met enige modificaties ook toegepast worden op *Volterra*-vergelijkingen van de eerste soort, dus (7.1.2). Ook dan resulteert weer het schema (7.4.7), met het verschil dat de matrix K nu een onderdriehoeksmatrix is, en dat de intervallen [a,b] en [c,d] samenvallen. Het is gebruikelijk om dan ook $N_2 = N_1$ te kiezen. Resultaten van numerieke experimenten met (7.4.7) voor Volterra-vergelijkingen vindt men in §7.6.

De ALGOL 60-implementatie, alsmede een korte beschrijving van het gebruik hiervan, is opgenomen in de bijlagen bij dit hoofdstuk.

Sinds de artikelen van Phillips en Tihonov is er een geweldige hoeveelheid literatuur verschenen over regularisatiemethoden voor operatorvergelijkingen van de eerste soort, in het bijzonder integraalvergelijkingen van Fredholm en ook van Volterra. Met name de Russen (Bakushinskii, Morozov, Arsenin, Ivanov, Savelova, Tanana, Vinokurov en vele anderen) hebben veel theoretisch onderzoek verricht, gericht op constructie van regularisatiemethoden en optimale keuze van de regularisatieparameter. Wij volstaan hier met de twee belangrijkste bronnen te vermelden, nl. de tijdschriften Dokl. Akad. Nauk SSSR en Zh. vychisl. Mat. i mat. Fiz. (in het Engels vertaald in Soviet Mathematics Doklady, resp. USSR Computational Mathematics and Mathematical Physics). Een goed recent overzicht van (Russische) literatuur over onjuist gestelde problemen wordt gegeven door TIHONOV c.s. [1976].

Belangwekkend theoretisch onderzoek van regularisatiemethoden is ook verricht door NASHED [1976] en anderen, waarbij gebruik wordt gemaakt van de theorie der gegeneraliseerde inversen (van algemene operatoren) en de theorie der zgn. kernreproducerende Hilbertruimten. Een opmerkelijk uitgebreide becommentarieerde bibliografie over gegeneraliseerde inversen en toepassingen hiervan (waaronder regularisatie van onjuist gestelde problemen) kan men vinden in NASHED c.s. [1976], met 1776 (!) referenties.

Toepassingen van en resultaten van numerieke experimenten met regularisatiemethoden vindt men in TIHONOV c.s. [1964,1965], RIBIÈRE [1967], STALLMANN [1970], CULLUM [1971], ANDERSSEN c.s. [1974], FRANKLIN [1974], HILGERS [1974] en LEWIS [1975]. Met name de laatste twee artikelen zijn interessant vanwege de vergelijking van de regularisatiemethode met andere methoden. Het is is opmerkelijk dat alle numerieke experimenten in de hier

gegeven referenties zijn uitgevoerd op *Fredholm*vergelijkingen. Experimenten met regularisatiemethoden, rechtstreeks toegepast op *Volterra*vergelijkingen zijn in de literatuur niet te vinden.

7.5. Andere methoden

Naast regularisatiemethoden zijn er ook vele andere methoden ontwikkeld voor het numeriek oplossen van integraalvergelijkingen van de eerste soort. DELVES c.s. [1974] geven een tot 1973 tamelijk volledig overzicht van de literatuur. We geven hier een kort résumée, aangevuld met enkele referenties na 1972.

7.5.1. Volterravergelijkingen

De bestaande methoden voor Volterravergelijkingen van de eerste soort zijn in het algemeen gebaseerd op die voor Volterravergelijkingen van de tweede soort (zie hoofdstuk 6). Deze methoden zijn echter alleen geschikt voor vergelijkingen van de eerste soort met *exacte* data (afgezien van machineprecisie en discretisatiefout). Er wordt bij deze methoden ook steeds verondersteld dat $K(x,x) \neq 0$. We kunnen globaal de volgende indeling maken:

1. Directe toepassing van kwadratuur op de integraalvergelijking, gevolgd door oplossen van het resulterende lineaire stelsel: JONES [1961], LINZ [1967,1969], KOBAYASHI [1967], DE HOOG c.s. [1973a,1973b], GLADWIN [1972,1973], HOLYHEAD c.s. [1975,1976], RECKERS [1977].
2. Productintegratie; het eenvoudigste geval hiervan is de herhaalde toepassing op (7.1.2) van de benadering
$$\int_{\alpha}^{\beta} K(x,y)f(y)dy \simeq f((\alpha+\beta)/2) \int_{\alpha}^{\beta} K(x,y)dy,$$
ook wel de *gewijzigde midpointregel* genoemd. Zie ANDERSSEN c.s. [1971], LINZ [1971].
3. Continue (globale) benadering van de gezochte oplossing m.b.v. splinefuncties: BRUNNER [1973], EL THOM [1976], en andere functies: BRUNNER [1974,1975a,1975b].

De enige methode die rekening houdt met inexacte data is die van SCHMAEDEKE [1968,1969]. Het is een soort regularisatiemethode, maar helaas geeft Schmaedeke geen resultaten van numerieke experimenten. Door ons uitgevoerde numerieke experimenten met de methode van Schmaedeke geven slechte

resultaten, vergeleken met experimenten met de in §7.4 beschreven methode.

7.5.2. Fredholmvergelijkingen

Voor Fredholmvergelijkingen zijn er verschillende methoden die rekening houden met *inexacte* data en maar zeer weinig die rekening houden met *exacte* data, een situatie die precies omgekeerd is vergeleken met die bij Volterravergelijkingen. De reden hiervan is dat directe methoden, gebaseerd op exacte data, bij Volterravergelijkingen in speciale gevallen nog wel eens succes hebben, terwijl deze bij Fredholmvergelijkingen meestal onbruikbare resultaten geven.

Afgezien van de regularisatiemethode, beschreven in §7.4, kan men globaal de volgende indeling maken:

1. Ontwikkeling van de oplossing in singuliere functies. Voor de theorie hiervan zie men SMITHIES [1958]. BAKER c.s. [1964] waren de eersten die deze methode voorstelden; hun methode werd later uitgebreid en verder onderzocht door HANSON [1971,1972], VAINSTEIN [1972a,1972b] en CRONE [1972]. Deze methode is theoretisch aantrekkelijk door het inzicht in het probleem dat men ermee krijgt, vanuit practisch standpunt gezien is hij tamelijk bewerkelijk en duur.
2. Continue benadering van de oplossing m.b.v. lineaire splinefuncties, gevolgd door collocatie en oplossing van het gestabiliseerde lineaire stelsel. Zo nodig kan de methode geïtereerd worden teneinde de nauwkeurigheid van de verkregen oplossing te verbeteren. Zie HANSON c.s. [1975]. Deze veelbelovende methode is enigszins verwant aan de regularisatiemethode. Ook benaderingen van de oplossing met andere basisfuncties dan splines zijn mogelijk, in het bijzonder wanneer bv. door kennis van de fysische achtergronden van het probleem iets bekend is over de basisfuncties waaruit de oplossing zou kunnen zijn opgebouwd.
3. Iteratieve methoden. De eenvoudigste iteratieve methode, afkomstig van LANDWEBER [1951] heeft de vorm

$$f^{(n+1)} = f^{(n)} + K^T(g-Kf^{(n)}), \qquad n = 0,1,2,\ldots,$$

met startwaarde $f^{(0)} = 0$. Hierbij is K de operator behorende bij (7.1.1). Onder bepaalde voorwaarden convergeert $f^{(n)}$ naar de oplossing van (7.1.1). Landweber's methode is later op verschillende manieren gegeneraliseerd

en verfijnd. Zie MARČUK c.s. [1970] en STRAND [1974,1976]. Ook iteratief toepassen van de regularisatiemethode is onderzocht, zie SHAW [1972] en STRAND [1974,1976].

4. Statistische methoden. Het probleem (7.1.1) wordt hierbij beschouwd vanuit statistisch standpunt. Belangrijk hierbij is dat f en g vervangen worden door stationaire random processen. Het blijkt dat men ook enige *a priori* informatie over f nodig heeft, om verder te kunnen komen. Zie STRAND c.s. [1968], TURČIN [1967,1968] en TURČIN c.s. [1973].

7.6. Numerieke experimenten met de regularisatiemethode van Phillips en Tihonov

De regularisatiemethode van Phillips en Tihonov is geïmplementeerd in een ALGOL 60-procedure genaamd REGULAR. De gebruiker dient bij aanroep van de procedure de volgende grootheden op te geven:

- frevol, met waarde 1 als het gaat om een Fredholmvergelijking, waarde 2 bij een Volterravergelijking;
- twee functieprocedures kernel en g, die bij aanroep resp. de waarde van de kern en die van de datafunctie (in het opgegeven argument) opleveren;
- a_0, a_1 en a_2, met waarden 0 of 1 (zie (7.4.5));
- als $a_0 = 1$, een beginschatting $\vec{\hat{f}}$ van de uit (7.4.7) te berekenen $\vec{f}_\alpha$;
- α, de regularisatieparameter;
- c, d, N_1, a, b en N_2 (zie de tekst boven (7.4.4)).

Na beëindiging van de procedure is de berekende oplossing $\vec{f}_\alpha$ overschreven op $\vec{\hat{f}}$; verder heeft de gebruiker in het array res[0:3] de beschikking over de bij $\vec{f}_\alpha$ behorende waarden van het residu $(\Sigma\varepsilon_i^2)^{\frac{1}{2}}$ en de grootheden (vgl. 7.4.6)

$$\left\{\sum_{i=0}^{N_1}(f_i-\hat{f}_i)^2\right\}^{\frac{1}{2}},\quad \left\{\sum_{i=0}^{N_1-1}(f_{i+1}-f_i)^2\right\}^{\frac{1}{2}},\quad \left\{\sum_{i=1}^{N_1-1}(f_{i+1}-2f_i+f_{i-1})^2\right\}^{\frac{1}{2}}.$$

Voor de kwadratuur is de herhaalde trapeziumregel gekozen, omdat een nauwkeuriger kwadratuurregel bij data met een onnauwkeurigheid van 1 tot 5% zinloos is. Voor de Fredholmvergelijking wordt de matrix K uit (7.4.7) dan:

$$K_{FRED} = h_2 \begin{bmatrix} K_{00}/2 & K_{01} & \cdots & K_{0N_1-1} & K_{0N_1}/2 \\ K_{10}/2 & K_{11} & \cdots & K_{1N_1-1} & K_{1N_1}/2 \\ \vdots & \vdots & & \vdots & \vdots \\ K_{N_2 0}/2 & K_{N_2 1} & \cdots & K_{N_2 N_1-1} & K_{N_2 N_1}/2 \end{bmatrix}.$$

Voor de Volterravergelijking krijgen we (onder de aannamen [c,d] = [a,b], $N_2 = N_1 = N$, $h_2 = h_1 = h$):

$$K_{VOLT} = h \begin{bmatrix} 0 & & & & \\ K_{10}/2 & K_{11}/2 & & & \\ K_{20}/2 & K_{21} & K_{22}/2 & & \\ \vdots & \vdots & \vdots & \ddots & \\ K_{N0}/2 & K_{N1} & K_{N2} & \cdots & K_{NN}/2 \end{bmatrix}$$

REGULAR is op een aantal voorbeelden getest, steeds met exacte en met inexacte data. Inexacte data werden als volgt geconstrueerd: de datafunctie g werd vermenigvuldigd met

$$1 + AMP* (2*RANDOM-1),$$

waarbij RANDOM een at random gekozen getal tussen 0 en 1 voorstelt, en AMP = 0.01. Voor exacte data geldt AMP = 0.

Een korte beschrijving van de procedure REGULAR, alsmede de source-tekst vindt men in de bijlagen bij dit hoofdstuk.

VOORBEELD 1. Volterravergelijking van de eerste soort, *numerieke differentiatie*:

$$\int_0^x K(x,y)f(y)dy = g(x) \qquad (0 \le x \le 1),$$

met $K \equiv 1$ en $g(x) = \sin 2\pi x$; oplossing $f(x) = g'(x) = 2\pi \cos 2\pi x$. Verder kozen we $a_0 = a_1 = 0$ en $a_2 = 1$ (dus 2e orde regularisatie); $N = N_1 = N_2 = 20$, dus $h = h_1 = h_2 = \frac{1}{20}$. Een selectie van de resultaten vindt men in tabel 7.6.1. Hierin is res[3] = $\|K\vec{f}_\alpha - \vec{g}\|$, waarin $\vec{f}_\alpha$ de berekende oplossingsvector is en K en $\vec{g}$ onder (7.4.7) gedefinieerd zijn. In de kolom # staat het gemiddelde aantal correcte cijfers in de berekende oplossing.

Tabel 7.6.1

Numerieke resultaten bij voorbeeld 1

	AMP = 0		AMP = 0.01	
$-^{10}\log\,\alpha$	res[3]	#	res[3]	#
10	2.3E-8	1.61	9.7E-7	0.44
8	2.3E-6	1.61	9.5E-5	0.45
6	1.5E-4	1.62	4.1E-3	0.69
5	7.5E-4	1.88*	9.1E-3	1.21
4	3.3E-3	1.59	1.4E-2	1.40*
3	1.3E-2	1.51	2.2E-2	1.37
2	4.9E-2	1.20	5.3E-2	1.29
1	1.7E-1	0.46	1.7E-1	0.46

Voor AMP = 0 vindt men bij $\alpha = 10^{-5}$ het beste resultaat, bij AMP = 0.01 vindt men $\alpha = 10^{-4}$. Meer informatie over deze twee gevallen vindt men in de bijlagen bij dit hoofdstuk.

VOORBEELD 2. Volterravergelijking van de eerste soort van het *convolutietype*:

$$\int_0^x K(x-y)f(y)\,dy = g(x) \qquad (0\le x\le 1),$$

met

$$K(u) = \begin{cases} \sin 2\pi u, & 0 \le u \le \tfrac{1}{2} \\ 0, & \tfrac{1}{2} < u \le 1, \end{cases}$$

en

$$g(x) = \begin{cases} \sin 2\pi x + \dfrac{2\pi}{1+4\pi^2}\left[e^{-x} + \dfrac{1}{2\pi}\sin 2\pi x - \cos 2\pi x\right], & 0 \le x \le \tfrac{1}{2}, \\ \dfrac{2\pi}{1+4\pi^2}(1+\sqrt{e})\,e^{-x}, & \tfrac{1}{2} < x \le 1. \end{cases}$$

Oplossing: $f(x) = \delta(x) + e^{-x}$, met δ de Dirac-functie. Dit is een geïdealiseerde versie van het in §7.2 besproken fysiologische probleem. In werkelijkheid is de functionaalvorm van K en g niet bekend, en zijn K en g slechts in tabelvorm gegeven.

Overige grootheden: $a_0 = a_2 = 1$, $a_1 = 0$; $N = N_1 = N_2 = 20$, $h = h_1 = h_2 = \frac{1}{20}$.
Als beginschatting van $\vec{f}$ kozen we $(\vec{f})_0 = 10$ en $(\vec{f})_i = 0$ voor $i = 1,2,\ldots,20$.

De piek in f(0), veroorzaakt door de δ-functie, is op fysische gronden bekend, de hoogte ervan evenwel niet. In het geval AMP = 0 zijn de resultaten bevredigend, AMP = 0.01 levert geen bruikbare resultaten. Iteratie met de regularisatiemethode is misschien een remedie, een andere mogelijkheid is het gebruik van Fourieranalyse (men zie hiervoor hoofdstuk 4.2 uit de syllabus van deel 1 van dit colloquium).

Meer informatie over het geval AMP = 0, $\alpha = 10^{-10}$ vindt men in de bijlagen bij dit hoofdstuk.

Tabel 7.6.2.

Numerieke resultaten bij voorbeeld 2

	AMP = 0		AMP = 0.01	
$-^{10}\log\ \alpha$	res[3]	#	res[3]	#
10	2.4E-6	2.33*	3.0E-6	0.48
8	2.4E-4	2.13	3.0E-4	0.46
6	1.5E-2	1.11	1.6E-2	0.39
5	6.2E-2	0.50	6.2E-2	0.57
4	1.6E-1	0.21	1.7E-1	0.26
3	3.4E-1	0.35	3.4E-1	0.37
2	7.5E-1	0.73	7.4E-1	0.70
1	1.5	0.47	1.5	0.47

VOORBEELD 3. Fredholmvergelijking van de eerste soort:

$$\int_0^5 K(x,y)f(y)dy = g(x) \qquad (0\le x\le 5),$$

met $K(x,y) = \left(\frac{\sin xy}{xy} - \cos xy\right)^2$, en oplossing

$$f(y) = \begin{cases} \frac{1}{4}(y-3)^2y^2, & 0 \le y \le 3, \\ 0, & 3 < y \le 5. \end{cases}$$

Met behulp van de procedure QADRAT uit de NUMAL-programmatheek is de datafunctie g(x) in de punten $x_i = i/4$ (i = 0,1,...,20) in 8 cijfers nauwkeurig berekend. De berekende waarden dienden als invoergegevens voor de procedure REGULAR. Voorts kozen we [a,b] = [c,d] = [0,5], $N_1 = N_2 = 20$, $a_0 = a_1 = 1$ en $a_2 = 0$. Als beginschatting voor $\vec{f}$ kozen we $\vec{f} = \vec{0}$. Dit is het in §7.2 genoemde, uit de meteorologie afkomstige, probleem. De belangrijkste resultaten

van de berekeningen vindt men in tabel 7.6.3. Meer informatie over de gevallen AMP = 0, $\alpha = 10^{-6}$ en AMP = 0,01, $\alpha = 10^{-2}$ vindt men in de bijlagen bij dit hoofdstuk.

Tabel 7.6.3.

Numerieke resultaten bij voorbeeld 3

	AMP = 0		AMP = 0.01	
$-^{10}\log \alpha$	res[3]	#	res[3]	#
10	8.3E-8	3.26	9.2E-3	---
8	1.8E-7	3.31	1.1E-2	0.42
6	7.3E-6	3.42*	1.1E-2	1.41
5	4.8E-5	3.09	1.2E-2	1.64
4	4.5E-4	2.99	1.2E-2	1.92
3	4.4E-3	2.94	1.3E-2	1.95
2	4.3E-2	1.93	4.6E-2	2.01*
1	3.2E-1	1.08	3.2E-1	1.07

Wat betreft de *keuze van de waarde van de regularisatieparameter* α merken we tot slot nog het volgende op. Er bestaat geen simpel voorschrift om die waarde van α te bepalen, waarvoor vgl. (7.4.7) de "beste" oplossing oplevert. In de practijk verdient het aanbeveling het probleem met een aantal verschillende waarden van α te draaien en die oplossingen te selecteren waarbij het residu $\|K\vec{f}-\vec{g}\|$ binnen de maximaal toegelaten grens ligt. Vergelijk deze onderling, bv. door ze te plotten, en kies de "beste" eruit, daarbij zoveel mogelijk informatie vanuit de fysische achtergrond van het probleem gebruikend. Draai het probleem ook nog eens met een kleinere stap h; los het probleem ook nog eens op met 0^e orde regularisatie en de geselecteerde oplossing als beginschatting.

LITERATUUR

ANDERSSEN, A.S. & E.T. WHITE, [1971], *Improved numerical methods for Volterra integral equations of the first kind,* The Computer Journal 14, 442-443.

ANDERSSEN, R.S. & P. BLOOMFIELD, [1974], *Numerical differentiation procedures for non-exact data,* Numer. Math. 22, 157-182.

BAKER, C.T.H., L. FOX, D.F. MAYERS & K. WRIGHT, [1964], *Numerical solution of Fredholm integral equations of first kind,* The Computer Journal 7, 141-148.

BRUNNER, H., [1973], *The solution of Volterra integral equations of the first kind by piecewise polynomials,* J. Inst. Maths. Applics 12, 295-302.

BRUNNER, H., [1974], *On the approximate solution of first-kind integral equations of Volterra type,* Computing 13, 67-79.

BRUNNER, H., [1975a], *The approximate solution of linear and nonlinear first-kind integral equations of Volterra type,* in: WATSON, G.A. (ed.), *Numerical Analysis,* Dundee 1975, Lecture Notes in Mathematics #506, Springer, 15-27.

BRUNNER, H., [1975b], *Projection methods for the approximate solution of integral equations of the first kind,* Proc. Fifth Manitoba Conf. on Numer. Math., 3-23.

CRONE, L., [1972], *The singular value decomposition of matrices and cheap numerical filtering of systems of linear equations,* J. Franklin Inst. 294, 133-136.

CULLUM, J., [1971], *Numerical differentiation and regularization,* SIAM J. Numer. Anal. 8, 254-265.

DELVES, L.M. & J. WALSH, [1974], *Numerical solution of integral equations,* Clarendon Press, Oxford.

EL TOM, M.E.A., [1976], *Application of spline functions to systems of Volterra integral equations of the first and second kinds,* J. Inst. Maths. Applics 17, 295-310.

FRANKLIN, J.N., [1974], *On Tihonov's method for ill-posed problems,* Math. Comp. 28, 889-907.

GLADWIN, C.J., [1972], *Methods of high order for the numerical solution of first kind Volterra integral equations,*Proc. Second Manitoba Conf. on Numer. Math., 179-193.

GLADWIN, C.J., [1973], *Some remarks on the numerical solution of first kind Volterra integral equations,* Proc. Third Manitoba Conf. on Numer. Math., 223-237.

HANSON, R.J., [1971], *A numerical method for solving Fredholm integral equations of the first kind using singular values,* SIAM J. Numer. Anal. 8, 616-622.

HANSON, R.J., [1972], *Integral equations of immunology,* Comm. ACM 15, 883-890.

HANSON, R.J. & J.L. PHILLIPS, [1975], *An adaptive numerical method for solving linear Fredholm integral equations of the first kind,* Numer. Math. 24, 291-307.

HILGERS, J.W., [1974], *Non-iterative methods for solving operator equations of the first kind,* MRC Tech. Summ. Rept. #1413, Univ. of Wisconsin, Madison, Wisconsin.

HOLYHEAD, P.A.W., S. McKEE & P.J. TAYLOR, [1975], *Multistep methods for solving linear Volterra integral equations of the first kind,* SIAM J. Numer. Anal. 12, 698-711.

HOLYHEAD, P.A.W. & S. McKEE, [1976], *Stability and convergence of multistep methods for linear Volterra integral equations of the first kind,* SIAM J. Numer. Anal. 13, 269-292.

DE HOOG, F. & R. WEISS, [1973a], *High order methods for Volterra integral equations of the first kind,* SIAM J. Numer. Anal. 10, 647-664.

DE HOOG, F. & R. WEISS, [1973b], *On the solution of Volterra integral equations of the first kind,* Numer. Math. 21, 22-32.

JONES, J.G., [1961], *On the numerical solution of convolution integral equations and systems of such equations,* Math. Comp. 15, 131-142.

KOBAYASHI, M., [1967], *On the numerical solution of the Volterra integral equations of the first kind by trapezoidal rule,* Rep. Stat. Appl. Res. JUSE 14, 65-78.

LANDWEBER, L., [1951], *An iteration formula for Fredholm integral equations of the first kind,* Amer. J. Math. 73, 615-624.

LAVRENTIEV, M.M., [1967], *Some improperly posed problems of mathematical physics,* Springer tracts in natural philosophy, Vol. 11, Springer, Berlin.

LEWIS, B.A., [1975], *On the numerical solution of Fredholm integral equations of the first kind,* J. Inst. Maths. Applics 16, 207-220.

LINZ, P., [1967], *The numerical solution of Volterra integral equations by finite difference methods,* MRC Tech. Summ. Rept. #825, Univ. of Wisconsin, Madison, Wisconsin.

LINZ, P., [1969], *Numerical methods for Volterra integral equations of the first kind,* The Computer Journal 12, 393-397.

LINZ, P., [1971], *Product integration methods for Volterra integral equations of the first kind,* BIT 11, 413-421.

MARČUK, G.I. & V.G. VASILIEV, [1970], *On an approximate solution for operator equations of the first kind,* Soviet Math. Dokl. 11, 1562-1566.

NASHED, M.Z., [1976], *Aspects of generalized inverses in analysis and regularization,* in: NASHED, M.Z. (ed.), *Generalized inverses and applications,* Acad. Press, 193-244.

NASHED, M.Z. & L.B. RALL, [1976], *Annotated bibliography on generalized inverses and applications,* in: NASHED, M.Z. (ed.), *Generalized inverses and applications,* Acad. Press, 771-1041.

NATANSON, I.P., [1956], *Theorie der Funktionen einer reellen Veränderlichen,* Akademie-Verlag, Berlin.

NEDELKOV, I.P., [1972], *Improper problems in computational physics,* Comput. Phys. Comm. 4, 157-164.

PHILLIPS, D.L., [1962], *A technique for the numerical solution of certain integral equations of the first kind,* J. ACM 9, 84-97.

RECKERS, F., [1977], *Kwadratuurmethoden voor het numeriek oplossen van lineaire Volterra integraalvergelijkingen van de eerste en tweede soort,* Report NN 10/77, Mathematisch Centrum, Amsterdam.

RIBIÈRE, G., [1967], *Regularisation d'opérateurs,* Rev. Franç. Inf. Rech. Opér. 1, 57-79.

SCHMAEDEKE, W.W., [1968], *Approximate solutions for Volterra integral equations of the first kind,* J. Math. Anal. Applics 23, 604-613.

SCHMAEDEKE, W.W., [1969], *A new approach to the solution of unstable problems using variational techniques,* J. Math. Anal. Applics 25, 272-275.

SHAW, C.B. Jr., [1972], *Improvement of the resolution of an instrument by numerical solution of an integral equation,* J. Math. Anal. Applics 37, 83-112.

SHIFRIN, K.S. & E.A. CHAYANOVA, [1966], *The determination of the vertical spectrum from the scattering formula,* Izv.Atm. and Oceanic Phys. 2.

SMITHIES, F., [1958], *Integral equations,* Cambridge.

STALLMANN, F.W., [1970], *Numerical solution of integral equations,* Numer. Math. 15, 297-305.

STRAND, O.N., [1974], *Theory and methods related to the singular-function expansion and Landweber's interation for integral equations of the first kind,* SIAM J. Numer. Anal. 4, 798-825.

STRAND, O.N., [1976], *Some aspects of the behavior of regularized solutions as the amount of smoothing is varied,* Comp. and Math. 2, 181-187.

STRAND, O.N. & E.R. WESTWATER, [1968], *Statistical estimation of the numerical solution of a Fredholm integral equation of the first kind,* J. ACM 15, 100-114.

TIHONOV, A.N., [1963a], *Solution of incorrectly formulated problems and the regularization method,* Soviet Math. Dokl. 4, 1035-1038.

TIHONOV, A.N., [1963b], *Regularization of incorrectly posed problems,* Soviet Math. Dokl. 4, 1624-1627.

TIHONOV, A.N. & V.B. GLASKO, [1964], *The approximate solution of Fredholm integral equations of the first kind,* USSR Comp. Math. and Math. Phys. 4, No. 3, 236-247.

TIHONOV, A.N. & V.B. GLASKO, [1965], *Use of the regularization method in non-linear problems,* USSR Comp. Math. and Math. Phys. 5, No. 3, 93-107.

TIHONOV, A.N., V.K. IVANOV & M.M. LAVRENTIEV, [1976], *Improperly posed problems,* Amer. Math. Soc. Transl., Series 2, 105, 313-332.

TURČIN, V.F., [1967], *Solution of the Fredholm equation of the first kind in a statistical ensemble of smooth functions,* USSR Comp. Math. and Math. Phys. 7, No. 6, 79-96.

TURČIN, V.F., [1968], *Selection of an ensemble of smooth functions for the solution of the inverse problem,* USSR Comp. Math. and Math. Phys. 8, No. 1, 328-339.

TURČIN, V.F. & L.S. TUROVCEVA, [1973], *The method of statistical regularization with an a priori estimate of the error in the initial data,* Soviet Math. Dokl. 14, 1430-1434.

TWOMEY, S., [1963], *On the numerical solution of Fredholm integral equations of the first kind by the inversion of the linear system produced by quadrature,* J. ACM 10, 97-101.

VAINSTEIN, L.A., [1972a], *Filtering of noise in a numerical solution of integral equations of the first kind,* Soviet Phys. Dokl. 17, 519-521.

VAINSTEIN, L.A., [1972b], *Numerical solution of integral equations of the first kind using a priori information on the function to be determined,* Soviet Phys. Dokl. 17, 532-534.

BIJLAGEN BIJ HOOFDSTUK 7 VAN HET
COLLOQUIUM NUMERIEKE PROGRAMMATUUR, DEEL 2

```
    THE HEADING OF THE PROCEDURE READS:
    "PROCEDURE" REGULAR(FREVOL, KERNEL, G, F, ALFA,
                        A, B, N1, C, D, N2, AA, RES);
    "VALUE" FREVOL, ALFA, A, B, N1, C, D, N2;
    "REAL" ALFA, A, B, C, D;
    "INTEGER" FREVOL, N1, N2;
    "INTEGER" "ARRAY" AA;
    "ARRAY" F, RES;
    "REAL" "PROCEDURE" KERNEL, G;

    REGULAR SOLVES AN INTEGRAL EQUATION OF THE FIRST KIND OF FREDHOLM
    OR VOLTERRA WITH THE REGULARIZATION METHOD OF PHILLIPS AND TIHONOV;

    THE MEANING OF THE FORMAL PARAMETERS IS;
    FREVOL: <ARITHMETIC EXPRESSION>;
            ENTRY; FREVOL=1 IN CASE OF A FREDHOLM EQUATION,
                   FREVOL=2 IN CASE OF A VOLTERRA EQUATION;
    KERNEL: <PROCEDURE IDENTIFIER>;
            "REAL" "PROCEDURE" KERNEL(X, Y); "VALUE" X, Y; "REAL" X, Y;
            A CALL OF THIS PROCEDURE MUST DELIVER THE VALUE OF THE
            KERNEL OF THE INTEGRAL EQUATION IN THE ARGUMENT (X, Y);
    G;      <PROCEDURE IDENTIFILR>;
            "REAL" "PROCEDURE" G(X); "VALUE" X; "REAL" X;
            A CALL OF THIS PROCEDURE MUST DELIVER THE VALUE OF THE
            DATAFUNCTION OF THE INTEGRAL EQUATION IN THE ARGUMENT X;
    F;      <ARRAY IDENTIFIER>;
            "ARRAY" F[0 : N1];
            ENTRY; IF AA[0] = 1 THEN F MUST CONTAIN AN INITIAL ESTIMATE
                   OF THE SOLUTION IN THE POINTS A + I * (B - A) / N1,
                   I = 0, 1, ... N1;
            EXIT;  THE COMPUTED SOLUTION;
    ALFA;   <ARITHMETIC EXPRESSION>;
            ENTRY; THE VALUE OF THE REGULARIZATION PARAMETER;
    A, B, N1, C, D, N2: <ARITHMETIC EXPRESSIONS>;
            ENTRY; [A,B] IS THE RANGE OF THE SECOND PARAMETER OF KERNEL
                   AND OF THE ARGUMENT OF THE SOLUTION FUNCTION OF THE
                   INTEGRAL EQUATION;
                   [C,D] IS THE RANGE OF THE FIRST PARAMETER OF KERNEL
                   AND OF THE PARAMETER OF G;
                   N1 + 1 IS THE NUMBER OF PARTITION POINTS OF [A,B];
                   N2 + 1 IS THE NUMBER OF PARTITION POINTS OF [C,D];
                   THE PARTITIONS ARE SUPPOSED TO BE EQUIDISTANT;
    AA;     <ARRAY IDENTIFIER>;
            "INTEGER" "ARRAY" AA[0 : 2];
            ENTRY; AA[I] = 1 IN CASE OF I-TH ORDER REGULARIZATION,
                   OTHERWISE AA[I] = 0 (I = 0, 1, 2);
    RES;    <ARRAY IDENTIFIER>;
            "ARRAY" RES[0 : 3];
            EXIT;  RES[0] = THE VALUE OF THE EUCLIDEAN NORM OF THE
                            DIFFERENCE OF THE COMPUTED SOLUTION AND
                            THE INITIAL ESTIMATE;
                   RES[1] = THE VALUE OF THE EUCLIDEAN NORM OF THE
                            VECTOR WITH COMPONENTS F[I+1] - F[I]
                            (I = 0, 1, ... N1-1);
                   RES[2] = THE VALUE OF THE EUCLIDEAN NORM OF THE
                            VECTOR WITH COMPONENTS F[I+1] - 2 * F[I]
                            + F[I-1] (I = 1, 2, ... N1-1);
                   RES[3] = THE VALUE OF THE EUCLIDEAN NORM OF THE
                            RESIDUE OF THE DISCRETIZED EQUATION;
```

SOURCE TEXT;

```
    "PROCEDURE" REGULAR(FREVOL, KERNEL, G, F, ALFA,
                        A, B, N1, C, D, N2, AA, RES);
    "VALUE" FREVOL, ALFA, A, B, N1, C, D, N2;
    "REAL" ALFA, A, B, C, D;
    "INTEGER" FREVOL, N1, N2;
    "INTEGER" "ARRAY" AA;
    "ARRAY" F, RES;
    "REAL" "PROCEDURE" KERNEL, G;
    "BEGIN"
       "INTEGER" I, J, IM1, N1M1, N1M2;
       "REAL" H1, H2, X, Y, ALFA2, ALFA4, ALFA5, ALFA6;
       "ARRAY" KMAT[0:N2, 0:N1], HMAT[0:N1+1, 0:N1+1],
               GVEC, H2VEC[0:N2], H1VEC[0:N1+1], HH[0:N1], AUX[2:3];

       "PROCEDURE" DUPVEC(L, U, SHIFT, A, B);          "CODE" 31030;
       "PROCEDURE" DUPVECCOL(L, U, I, A, B);           "CODE" 31033;
       "PROCEDURE" DUPCOLVEC(L, U, I, A, B);           "CODE" 31034;
       "PROCEDURE" RESVEC(LR, UR, LC, UC, A, B, C, X); "CODE" 31503;
       "PROCEDURE" FULTAMVEC(LR, UR, LC, UC, A, B, C); "CODE" 31501;
       "REAL" "PROCEDURE" VECVEC(L, U, SHIFT, A, B);   "CODE" 34010;
       "REAL" "PROCEDURE" MATVEC(L, U, I, A, B);       "CODE" 34011;
       "PROCEDURE" ELMVEC(L, U, SHIFT, A, B, X);       "CODE" 34020;
       "PROCEDURE" SYMDECSOL2(A, N, AUX, B);           "CODE" 34706;

       N1M1:= N1 - 1; N1M2:= N1 - 2;
       H1:= (B - A) / N1; H2:= (D - C) / N2;

       "COMMENT" FILL THE ARRAYS KMAT AND GVEC;
       "IF" FREVOL = 1 "THEN"
       "BEGIN" X:= C;
          "FOR" I:= 0 "STEP" 1 "UNTIL" N2 "DO"
          "BEGIN" Y:= A;
             KMAT[I, 0]:= KERNEL(X, Y) * H1 / 2; Y:= Y + H1;
             "FOR" J:= 1 "STEP" 1 "UNTIL" N1M1 "DO"
             "BEGIN" KMAT[I, J]:= KERNEL(X, Y) * H1; Y:= Y + H1 "END"J;
             KMAT[I, N1]:= KERNEL(X, Y) * H1 / 2;
             GVEC[I]:= G(X); X:= X + H2
          "END" I
       "END" FREVOL = 1 "ELSE"
       "BEGIN" "FOR" J:= 0 "STEP" 1 "UNTIL" N1 "DO" KMAT[0, J]:= 0;
          GVEC[0]:= G(C); X:= C + H2;
          "FOR" I:= 1 "STEP" 1 "UNTIL" N2 "DO"
          "BEGIN" Y:= A;
             KMAT[I, 0]:= KERNEL(X, Y) * H1 / 2; Y:= Y + H1;
             IM1:= I - 1;
             "FOR" J:= 1 "STEP" 1 "UNTIL" IM1 "DO"
             "BEGIN" KMAT[I, J]:= KERNEL(X, Y) * H1; Y:= Y + H1 "END"J;
             KMAT[I, I]:= KERNEL(X, Y) * H1 / 2;
             "FOR" J:= I + 1 "STEP" 1 "UNTIL" N1 "DO" KMAT[I, J]:= 0;
             GVEC[I]:= G(X); X:= X + H2
          "END" I
       "END" FREVOL = 2;
```

```
"COMMENT" FILL THE LOWER TRIANGLE OF THE AUXILIARY ARRAY HMAT
          WITH KMAT'KMAT;
"FOR" J:= 0 "STEP" 1 "UNTIL" N1 "DO"
"BEGIN" DUPVECCOL(0, N2, J, H2VEC, KMAT);
   FULTAMVEC(0, N2, J, N1, KMAT, H2VEC, H1VEC);
   DUPCOLVEC(J, N1, J, HMAT, H1VEC)
"END" J;

"COMMENT" ADD ALFA * (AA[0]*H0 + AA[1]*H1 + AA[2]*H2)
          TO THE LOWER TRIANGLE OF HMAT.
          H0, H1 AND H2 ARE SPECIAL CONSTANT BAND MATRICES;
"IF" AA[0] = 1 "THEN"
"FOR" I:= 0 "STEP" 1 "UNTIL" N1 "DO"
HMAT[I, I]:= HMAT[I, I] + ALFA;
"IF" AA[1] = 1 "THEN"
"BEGIN"
   ALFA2:= 2 * ALFA;
   HMAT[0, 0]:= HMAT[0, 0] + ALFA;
   HMAT[1, 0]:= HMAT[1, 0] - ALFA;
   HMAT[N1, N1]:= HMAT[N1, N1] + ALFA;
   "FOR" I:= 1 "STEP" 1 "UNTIL" N1M1 "DO"
   "BEGIN"
      HMAT[I, I]:= HMAT[I, I] + ALFA2;
      HMAT[I+1, I]:= HMAT[I+1, I] - ALFA
   "END" I
"END" AA[1] = 1;
"IF" AA[2] = 1 "THEN"
"BEGIN"
   ALFA2:= 2 * ALFA; ALFA4:= 4 * ALFA;
   ALFA5:= 5 * ALFA; ALFA6:= 6 * ALFA;
   HMAT[0, 0]:= HMAT[0, 0] + ALFA;
   HMAT[1, 0]:= HMAT[1, 0] - ALFA2;
   HMAT[2, 0]:= HMAT[2, 0] + ALFA;
   HMAT[1, 1]:= HMAT[1, 1] + ALFA5;
   HMAT[2, 1]:= HMAT[2, 1] - ALFA4;
   HMAT[3, 1]:= HMAT[3, 1] + ALFA;
   HMAT[N1, N1]:= HMAT[N1, N1] + ALFA;
   HMAT[N1, N1M1]:= HMAT[N1, N1M1] - ALFA2;
   HMAT[N1M1, N1M1]:= HMAT[N1M1, N1M1] + ALFA5;
   "FOR" I:= 2 "STEP" 1 "UNTIL" N1M2 "DO"
   "BEGIN"
      HMAT[I, I]:= HMAT[I, I] + ALFA6;
      HMAT[I+1, I]:= HMAT[I+1, I] - ALFA4;
      HMAT[I+2, I]:= HMAT[I+2, I] + ALFA
   "END" I;
"END" AA[2] = 1 ;

"COMMENT" FILL THE UPPER TRIANGLE OF THE SYMMETRIC ARRAY HMAT
          WITH ITS LOWER TRIANGLE;
"FOR" I:=0 "STEP" 1 "UNTIL" N1M1 "DO"
"FOR" J:= I + 1 "STEP" 1 "UNTIL" N1 "DO" HMAT[I, J]:= HMAT[J,I];

"COMMENT" FILL H1VEC WITH KMAT'GVEC + AA[0] * ALFA * F;
FULTAMVEC(0, N2, 0, N1, KMAT, GVEC, H1VEC);
"IF" AA[0] = 1  "THEN" ELMVEC(0, N1, 0, H1VEC, F, ALFA);
```

```
   "COMMENT" SOLUTION OF THE LINEAR SYSTEM WITH MATRIX HMAT AND
             R.H.S. H1VEC, THE SOLUTION IS OVERWRITTEN ON H1VEC;
   "FOR" I:= N1 "STEP" -1 "UNTIL" 0 "DO"
   "BEGIN" H1VEC[I+1]:= H1VEC[I];
      "FOR" J:= N1 "STEP" -1 "UNTIL" 0 "DO"
      HMAT[I+1, J+1]:= HMAT[I, J]
   "END" I;
   AUX[2]:="-12; SYMDECSOL2(HMAT, N1+1, AUX, H1VEC);
   "IF" ABS(AUX[3]) < N1+1 "THEN"
   "BEGIN" OUTPUT(61,"("" (" *** AUX[3] < N1+1 ***")",/,
    " (" *** EXECUTION OF REGULAR PREMATURELY ABORTED ***")",*")");
      "GOTO" STOP
   "END";
   DUPVEC(0, N1, 1, H1VEC, H1VEC);

   "COMMENT" COMPUTE RES[0 : 3];
   "FOR" I:= 0 "STEP" 1 "UNTIL" N1 "DO"
   HH[I]:= H1VEC[I] - ("IF" AA[0]=1 "THEN" F[I] "ELSE" 0);
   RES[0]:= SQRT(VECVEC(0, N1, 0, HH, HH));
   "FOR" I:= 0 "STEP" 1 "UNTIL" N1M1 "DO"
   HH[I]:= H1VEC[I + 1] - H1VEC[I];
   RES[1]:= SQRT(VECVEC(0, N1M1, 0, HH, HH));
   "FOR" I:= 1 "STEP" 1 "UNTIL" N1M1 "DO"
   HH[I]:= H1VEC[I + 1] - 2 * H1VEC[I] + H1VEC[I - 1];
   RES[2]:= SQRT(VECVEC(1, N1, 0, HH, HH));
   RESVEC(0, N2, 0, N1, KMAT, H1VEC, GVEC, -1);
   RES[3]:= SQRT(VECVEC(0, N2, 0, GVEC, GVEC));
   DUPVEC(0, N1, 0, F, H1VEC);
STOP:
"END" REGULAR;
```

COLLOQUIUM NUMERIEKE PROGRAMMATUUR, DEEL 2

VOORBEELDEN VAN HET GEBRUIK VAN PROCEDURE REGULAR
VOOR HET NUMERIEK OPLOSSEN VAN INTEGRAALVERGELIJKINGEN
VAN FREDHOLM EN VAN VOLTERRA VAN DE EERSTE SOORT

VOORBEELD 1, VOLTERRAVGL.
AMP= 0.0"+0 ALFA= 1.00"-05
A0=0 A1=0 A2=1

X	F(X)	F COMPUTED	# CORR. DIG.	G(X)
0.0000	+6.2831853071796"+000	+6.4010888764197"+000	0.93	+0.0000000000000"+000
0.0500	+5.9756643294831"+000	+5.9759862020533"+000	3.49	+3.0901699437495"-001
0.1000	+5.0832036923153"+000	+5.1457379769595"+000	1.20	+5.8778525229248"-001
0.1500	+3.6931636609809"+000	+3.7196140157655"+000	1.58	+8.0901699437495"-001
0.2000	+1.9416110387254"+000	+1.9578283276625"+000	1.79	+9.5105651629516"-001
0.2500	-5.5421161829392"-014	+3.3167811943713"-004	3.48	+1.0000000000000"+000
0.3000	-1.9416110387256"+000	-1.9578860957404"+000	1.79	+9.5105651629514"-001
0.3500	-3.6931636609810"+000	-3.7237885068466"+000	1.51	+8.0901699437494"-001
0.4000	-5.0832036923153"+000	-5.1254118299460"+000	1.37	+5.8778525229246"-001
0.4500	-5.9756643294832"+000	-6.0252812127524"+000	1.30	+3.0901699437493"-001
0.5000	-6.2831853071796"+000	-6.3353509927586"+000	1.28	-1.7641103714087"-014
0.5500	-5.9756643294831"+000	-6.0252812319973"+000	1.30	-3.0901699437497"-001
0.6000	-5.0832036923152"+000	-5.1254113334909"+000	1.37	-5.8778525229250"-001
0.6500	-3.6931636609809"+000	-3.7237898744441"+000	1.51	-8.0901699437495"-001
0.7000	-1.9416110387253"+000	-1.9578825989587"+000	1.79	-9.5105651629516"-001
0.7500	+7.6974051939158"-014	+3.2975077886022"-004	3.48	-1.0000000000000"+000
0.8000	+1.9416110387257"+000	+1.9577990261602"+000	1.79	-9.5105651629514"-001
0.8500	+3.6931636609811"+000	+3.7197873275308"+000	1.57	-8.0901699437492"-001
0.9000	+5.0832036923154"+000	+5.1452416067653"+000	1.21	-5.8778525229245"-001
0.9500	+5.9756643294832"+000	+5.9768593716450"+000	2.92	-3.0901699437493"-001
1.0000	+6.2831853071796"+000	+6.4064788826353"+000	0.91	+3.5282207428175"-014

AVE # CORR. DIGITS: 1.88
RESIDUES:
7.544467"-04
2.103395"+01 6.271433"+00 6.655979"+00

COLLOQUIUM NUMERIEKE PROGRAMMATUUR, DEEL 2

VOORBEELDEN VAN HET GEBRUIK VAN PROCEDURE REGULAR
VOOR HET NUMERIEK OPLOSSEN VAN INTEGRAALVERGELIJKINGEN
VAN FREDHOLM EN VAN VOLTERRA VAN DE EERSTE SOORT

VOORBEELD 1, VOLTERRAVGL.
AMP= 1.0"-2 ALFA= 1.00"-04
A0=0 A1=0 A2=1

X	F(X)	F COMPUTED	# CORR. DIG.	G(X)
0.0000	+6.2831853071796"+000	+6.4553696143562"+000	0.76	+0.0000000000000"+000
0.0500	+5.9756643294831"+000	+5.9374974383175"+000	1.42	+3.0661196447831"-001
0.1000	+5.0832036923153"+000	+5.0756609120926"+000	2.12	+5.8671218273997"-001
0.1500	+3.6931636609809"+000	+3.6403949447528"+000	1.28	+8.0403035307115"-001
0.2000	+1.9416110387254"+000	+1.9287747485167"+000	1.89	+9.4223492878460"-001
0.2500	-5.5421161829392"-014	+1.2944852980877"-001	0.89	+9.9000303438342"-001
0.3000	-1.9416110387256"+000	-1.9103566426447"+000	1.51	+9.5596711687162"-001
0.3500	-3.6931636609810"+000	-3.7397932253182"+000	1.33	+8.0242787231304"-001
0.4000	-5.0832036923153"+000	-5.0591788248154"+000	1.62	+5.9006205970673"-001
0.4500	-5.9756643294832"+000	-6.0251036852050"+000	1.31	+3.1110377707757"-001
0.5000	-6.2831853071796"+000	-6.3862858722190"+000	0.99	-1.7670254955750"-014
0.5500	-5.9756643294831"+000	-6.0587436495256"+000	1.08	-3.0873236545041"-001
0.6000	-5.0832036923152"+000	-5.0334033498993"+000	1.30	-5.9201114807400"-001
0.6500	-3.6931636609809"+000	-3.6772908591423"+000	1.80	-8.0247665098710"-001
0.7000	-1.9416110387253"+000	-2.1392893387799"+000	0.70	-9.4941543806800"-001
0.7500	+7.6974051939158"-014	-8.7583117561052"-002	1.06	-1.0099815898598"+000
0.8000	+1.9416110387257"+000	+2.1085058720236"+000	0.78	-9.5695632005654"-001
0.8500	+3.6931636609811"+000	+3.8013372862514"+000	0.97	-8.0428138381633"-001
0.9000	+5.0832036923154"+000	+5.0858064230838"+000	2.58	-5.9100644356276"-001
0.9500	+5.9756643294832"+000	+5.9624337188152"+000	1.88	-3.0614787701717"-001
1.0000	+6.2831853071796"+000	+6.4941530438526"+000	0.68	+3.5578797461891"-014

AVE # CORR. DIGITS: 1.40
RESIDUES:
1.448768"-02
2.103880"+01 6.285872"+00 6.756621"+00

COLLOQUIUM NUMERIEKE PROGRAMMATUUR, DEEL 2

VOORBEELDEN VAN HET GEBRUIK VAN PROCEDURE REGULAR
VOOR HET NUMERIEK OPLOSSEN VAN INTEGRAALVERGELIJKINGEN
VAN FREDHOLM EN VAN VOLTERRA VAN DE EERSTE SOORT

VOORBEELD 2, VOLTERRAVGL.
AMP= 0.0"+0 ALFA= 1.00"-10
A0=1 A1=0 A2=1

X	F(X)	F COMPUTED	# CORR. DIG.	G(X)	F INIT.EST.
0.0000	+1.0000000000000"+000	+4.0991376061139"+001		+0.0000000000000"+000	+1.0000000000000"+001
0.0500	+9.5122942450072"-001	+9.5966248353068"-001	2.07	+3.1667795117674"-001	+0.0000000000000"+000
0.1000	+9.0483741803596"-001	+9.1226410243928"-001	2.13	+6.1717974942912"-001	+0.0000000000000"+000
0.1500	+8.6070797642506"-001	+8.6813649468830"-001	2.13	+8.7136728359101"-001	+0.0000000000000"+000
0.2000	+8.1873075307798"-001	+8.2568779089371"-001	2.16	+1.0536712616749"+000	+0.0000000000000"+000
0.2500	+7.7880078307140"-001	+7.8543443473524"-001	2.18	+1.1455923918523"+000	+0.0000000000000"+000
0.3000	+7.4081822068172"-001	+7.4712827364670"-001	2.20	+1.1375105864188"+000	+0.0000000000000"+000
0.3500	+7.0468808971871"-001	+7.1069092268254"-001	2.22	+1.0296250871641"+000	+0.0000000000000"+000
0.4000	+6.7032004603564"-001	+6.7602850040992"-001	2.24	+8.3193348223470"-001	+0.0000000000000"+000
0.4500	+6.3762815162178"-001	+6.4306095140102"-001	2.26	+5.6325166476541"-001	+0.0000000000000"+000
0.5000	+6.0653065971264"-001	+6.1183532987502"-001	2.28	+2.4937066303583"-001	+0.0000000000000"+000
0.5500	+5.7694981038049"-001	+5.8153379195230"-001	2.34	+2.3720871228694"-001	+0.0000000000000"+000
0.6000	+5.4881163609403"-001	+5.5376680000819"-001	2.30	+2.2563990687526"-001	+0.0000000000000"+000
0.6500	+5.2204577676102"-001	+5.2639528068245"-001	2.36	+2.1463531876135"-001	+0.0000000000000"+000
0.7000	+4.9658530379141"-001	+5.0083183688200"-001	2.37	+2.0416743074288"-001	+0.0000000000000"+000
0.7500	+4.7236655274101"-001	+4.7639007687691"-001	2.40	+1.9421006764734"-001	+0.0000000000000"+000
0.8000	+4.4932896411722"-001	+4.5315640508512"-001	2.42	+1.8473833088043"-001	+0.0000000000000"+000
0.8500	+4.2741493194873"-001	+4.3105465644379"-001	2.44	+1.7572853616661"-001	+0.0000000000000"+000
0.9000	+4.0656965974060"-001	+4.1003510360551"-001	2.46	+1.6715815432612"-001	+0.0000000000000"+000
0.9500	+3.8674102345450"-001	+3.9003196620552"-001	2.48	+1.5900575494023"-001	+0.0000000000000"+000
1.0000	+3.6787944117144"-001	+1.8501441440277"-001	0.74	+1.5125095276410"-001	+0.0000000000000"+000

AVE # CORR. DIGITS: 2.33
RESIDUES:
2.394249"-06
3.112434"+01 4.003248"+01 3.998517"+01

COLLOQUIUM NUMERIEKE PROGRAMMATUUR, DEEL 2

VOORBEELDEN VAN HET GEBRUIK VAN PROCEDURE REGULAR
VOOR HET NUMERIEK OPLOSSEN VAN INTEGRAALVERGELIJKINGEN
VAN FREDHOLM EN VAN VOLTERRA VAN DE EERSTE SOORT

VOORBEELD 3, FREDHOLMVGL.
AMP= 0.0"+0 ALFA= 1.00"-06
A0=1 A1=1 A2=0

X	F(X)	F COMPUTED	# CORR. DIG.	G(X)	F INIT.EST.
0.0000	+0.0000000000000"+000	+8.2379350791826"-002	1.08	+0.0000000000000"+000	+0.0000000000000"+000
0.2500	+1.1816406250000"-001	+1.6475870158365"-001	1.33	+8.0030402450908"-003	+0.0000000000000"+000
0.5000	+3.9062500000000"-001	+3.8539557289551"-001	2.28	+1.0776376275059"-001	+0.0000000000000"+000
0.7500	+7.1191406250000"-001	+7.1076819580355"-001	2.94	+4.0871243002645"-001	+0.0000000000000"+000
1.0000	+1.0000000000000"+000	+1.0021685312185"+000	2.66	+8.6113364686250"-001	+0.0000000000000"+000
1.2500	+1.1962890625000"+000	+1.1945991744304"+000	2.77	+1.2525098608644"+000	+0.0000000000000"+000
1.5000	+1.2656250000000"+000	+1.2664206812889"+000	3.10	+1.4077257652313"+000	+0.0000000000000"+000
1.7500	+1.1962890625000"+000	+1.1962965466037"+000	5.13	+1.3480917671092"+000	+0.0000000000000"+000
2.0000	+1.0000000000000"+000	+9.9949220977078"-001	3.29	+1.2314892537374"+000	+0.0000000000000"+000
2.2500	+7.1191406250000"-001	+7.1248133277160"-001	3.25	+1.1741219698803"+000	+0.0000000000000"+000
2.5000	+3.9062500000000"-001	+3.9046986448710"-001	3.81	+1.1697677186058"+000	+0.0000000000000"+000
2.7500	+1.1816406250000"-001	+1.1739003043607"-001	3.11	+1.1609001509038"+000	+0.0000000000000"+000
3.0000	+0.0000000000000"+000	-1.4240462743326"-004	3.85	+1.1304087968395"+000	+0.0000000000000"+000
3.2500	+0.0000000000000"+000	+1.5998374300127"-003	2.80	+1.1035066101000"+000	+0.0000000000000"+000
3.5000	+0.0000000000000"+000	-8.0879038698160"-004	3.09	+1.0952852141459"+000	+0.0000000000000"+000
3.7500	+0.0000000000000"+000	+2.7546961646550"-004	3.56	+1.0927579464559"+000	+0.0000000000000"+000
4.0000	+0.0000000000000"+000	+2.2192602555299"-005	4.65	+1.0825260466124"+000	+0.0000000000000"+000
4.2500	+0.0000000000000"+000	-1.4085149947394"-004	3.85	+1.0695977532896"+000	+0.0000000000000"+000
4.5000	+0.0000000000000"+000	+1.4868047754613"-004	3.83	+1.0634934881270"+000	+0.0000000000000"+000
4.7500	+0.0000000000000"+000	-1.0046528407895"-004	4.00	+1.0621684639260"+000	+0.0000000000000"+000
5.0000	+0.0000000000000"+000	+9.0379063303906"-005	4.04	+1.0581882203072"+000	+0.0000000000000"+000

AVE # CORR. DIGITS: 3.42
RESIDUES:
7.295423"-06
2.797655"+00 7.788126"-01 3.768113"-01

COLLOQUIUM NUMERIEKE PROGRAMMATUUR, DEEL 2

VOORBEELDEN VAN HET GEBRUIK VAN PROCEDURE REGULAR
VOOR HET NUMERIEK OPLOSSEN VAN INTEGRAALVERGELIJKINGEN
VAN FREDHOLM EN VAN VOLTERRA VAN DE EERSTE SOORT

VOORBEELD 3, FREDHOLMVGL.
AMP= 1.0"-2 ALFA= 1.00"-02
A0=1 A1=1 A2=0

X	F(X)	F COMPUTED	# CORR. DIG.	G(X)	F INIT.EST.
0.0000	+0.0000000000000"+000	+8.754579468876 5"-002	1.06	+0.0000000000000"+000	+0.0000000000000"+000
0.2500	+1.1816406250000"-001	+1.7509158937753"-001	1.24	+7.9407538614816"-003	+0.0000000000000"+000
0.5000	+3.9062500000000"-001	+4.0261930062340"-001	1.92	+1.0756702761268"-001	+0.0000000000000"+000
0.7500	+7.1191406250000"-001	+7.1227796467974"-001	3.44	+4.0619319705716"-001	+0.0000000000000"+000
1.0000	+1.0000000000000"+000	+9.9219679543936"-001	2.11	+8.5314614486453"-001	+0.0000000000000"+000
1.2500	+1.1962890625000"+000	+1.1757104325096"+000	1.69	+1.2399885628509"+000	+0.0000000000000"+000
1.5000	+1.2656250000000"+000	+1.2330438757374"+000	1.49	+1.4149942911662"+000	+0.0000000000000"+000
1.7500	+1.1962890625000"+000	+1.1622336582344"+000	1.47	+1.5371120951544"+000	+0.0000000000000"+000
2.0000	+1.0000000000000"+000	+9.4994573494620"-001	1.30	+1.2362594718614"+000	+0.0000000000000"+000
2.2500	+7.1191406250000"-001	+6.9526401966952"-001	1.78	+1.1820507811177"+000	+0.0000000000000"+000
2.5000	+3.9062500000000"-001	+4.0103484332430"-001	1.98	+1.1717007145230"+000	+0.0000000000000"+000
2.7500	+1.1816406250000"-001	+1.1506916284806"-001	2.51	+1.1598308706782"+000	+0.0000000000000"+000
3.0000	+0.0000000000000"+000	+2.0901756850529"-002	1.68	+1.1385358972512"+000	+0.0000000000000"+000
3.2500	+0.0000000000000"+000	+1.3953739843568"-002	1.86	+1.0945855216544"+000	+0.0000000000000"+000
3.5000	+0.0000000000000"+000	+4.5852890551228"-003	2.34	+1.0933952647195"+000	+0.0000000000000"+000
3.7500	+0.0000000000000"+000	+4.1001836550656"-003	2.39	+1.1036654080935"+000	+0.0000000000000"+000
4.0000	+0.0000000000000"+000	+2.1006595515411"-002	1.68	+1.0892414111908"+000	+0.0000000000000"+000
4.2500	+0.0000000000000"+000	+5.0467750711566"-002	1.30	+1.0633368237305"+000	+0.0000000000000"+000
4.5000	+0.0000000000000"+000	+1.0309930503693"-003	2.99	+1.0693216641940"+000	+0.0000000000000"+000
4.7500	+0.0000000000000"+000	+6.7678826417506"-003	2.17	+1.0523065921448"+000	+0.0000000000000"+000
5.0000	+0.0000000000000"+000	+1.3109765035886"-002	1.88	+1.0670835843679"+000	+0.0000000000000"+000

AVE # CORR. DIGITS: 2.01
RESIDUES:
4.566032"-02
2.740334"+00 7.527320"-01 3.829967"-01

8. INTERPOLATIE, HET OPLOSSEN VAN VERGELIJKINGEN EN SOMMATIE

8.1. Interpolatie en het oplossen van vergelijkingen

8.1.1. Inleiding

Zij f(t) een reële functie in de reële variabele t. Zij $f^{(p)}(x)$ de p-de afgeleide van f in x en stel gegeven:

$$(8.1.1) \qquad f_i^p = f^{(p)}(x_i), \qquad p = 0,\ldots,\gamma-1, \qquad i = 0,1,\ldots,n.$$

Het interpolatieprobleem is nu het probleem een benaderende functie $\tilde{f}(t)$ te bepalen, zodat

$$\tilde{f}^{(p)}(x_i) = f_i^p, \qquad p = 0,\ldots,\gamma-1, \qquad i = 0,1,\ldots,n.$$

Het is duidelijk dat dit probleem oneindig veel oplossingen heeft zolang geen nadere specificatie wordt gegeven van de functie $\tilde{f}$ waarmee we interpoleren. Wij zullen in dit hoofdstuk kort ingaan op interpolatie met polynomen en met rationale functies. Vervolgens zullen wij enige iteratieve methoden voor het oplossen van vergelijkingen en het minimaliseren van functies bespreken, die gebruik maken van interpolatietechnieken.

8.1.2. Interpolatie met polynomen

We zoeken een polynoom P(t) zodat

$$(8.1.2) \qquad P^{(k)}(x_i) = f_i^k, \qquad k = 0,\ldots,\gamma-1, \qquad i = 0,1,\ldots,n.$$

Het is bekend dat er een uniek polynoom $P_{n,\gamma}$ van de graad $(n+1)\gamma-1$ bestaat zodat aan (8.1.2) is voldaan. Dit polynoom kan op verschillende manieren worden beschreven.

Newton's formulering

We gebruiken hierbij de volgende formulering voor confluente differentiequotienten (TRAUB [1964]):

$$f[x_i,x_{i-1}] = \frac{f_i-f_{i-1}}{x_i-x_{i-1}} \quad ,$$

$$f[x_i,\ldots,x_{i+k}] = \frac{f[x_{i+1},\ldots,x_{i+k}]-f[x_i,\ldots,x_{i+k-1}]}{x_{i+k}-x_i} \quad ,$$

$$f[x_i,1;x_{i+1},1;\ldots;x_{i+k},1] \equiv f[x_i,\ldots,x_{i+k}]$$

en algemeen voor γ_q en γ_r ongelijk aan nul met $0 \le q \le r \le k$:

$$\begin{aligned} &f[x_i,\gamma_0;\ldots;x_{i+k},\gamma_k] \\ &= \frac{1}{x_{i+q}-x_{i+r}} \Big\{ f[x_i,\gamma_0;\ldots;x_{i+q},\gamma_q;\ldots;x_{i+r},\gamma_r-1;\ldots;x_{i+k},\gamma_k] \\ &\quad -f[x_i,\gamma_0;\ldots;x_{i+q},\gamma_q-1;\ldots;x_{i+r},\gamma_r;\ldots;x_{i+k},\gamma_k] \Big\}. \end{aligned} \tag{8.1.3}$$

Dan kan het interpolerend polynoom $P_{n,\gamma}(t)$ dat voldoet aan (8.1.2) geschreven worden als

$$P_{n,\gamma}(t) = \sum_{j=0}^{n} \sum_{\ell=0}^{\gamma-1} C^{\gamma}_{\ell,j}(t) f[x_0,\gamma;x_1,\gamma;\ldots;x_j,\ell+1] \tag{8.1.4}$$

met

$$C^{\gamma}_{\ell,j}(t) = (t-x_j)^{\ell} \prod_{k=0}^{j-1} (t-x_k)^{\gamma}, \qquad \prod_{0}^{-1} = 1. \tag{8.1.5}$$

Voorbeelden:

$$P_{n,1}(t) = \sum_{j=0}^{n} f[x_0,x_1,\ldots,x_j] \prod_{k=0}^{j-1} (t-x_k).$$

In het bijzonder met $f_i^0 = f_i$ $(i = 0,\ldots,n)$

$$(8.1.6) \qquad P_{1,1}(t) = f_0 + (t-x_0)\,\frac{f_0-f_1}{x_0-x_1}\,.$$

Lagrange-Hermite formulering

Met gebruikmaking van de *Lagrange-polynomen*

$$(8.1.7) \qquad \ell_j(t) = \prod_{\substack{k=0 \\ k\neq j}}^{n} \left(\frac{t-x_k}{x_j-x_k}\right)$$

kunnen we het interpolerend polynoom ook schrijven als

$$(8.1.8) \qquad P_{n,\gamma}(t) = \sum_{j=0}^{n} \sum_{\ell=0}^{\gamma-1} A_{\ell,j}^{\gamma,n}(t) f_j^{\ell} = \sum_{\ell=0}^{\gamma-1} \sum_{j=0}^{n} A_{\ell,j}^{\gamma,n}(t) f_j^{\ell}\,,$$

waarin $A_{\ell,j}^{\gamma,n}$ onafhankelijk is van f en zijn afgeleiden. Voor $\gamma = 1$ krijgen we

$$(8.1.9) \qquad P_{n,1}(t) = \sum_{j=0}^{n} \ell_j(t) f_j.$$

Dit wordt meestal *Lagrange-interpolatie* genoemd. Voor $\gamma = 2$ krijgen we

$$(8.1.10) \qquad P_{n,2}(t) = \sum_{j=0}^{n} A_{0,j}^{2,n}(t) f_j + \sum_{j=0}^{n} A_{1,j}^{2,n}(t) f_j^1$$

met

$$A_{0,j}^{2,n}(t) = [1-2\ell_j'(x_j)(t-x_j)][\ell_j(t)]^2$$

$$A_{1,j}^{2,n}(t) = (t-x_j)[\ell_j(t)]^2\,.$$

Dit wordt meestal *Hermite-interpolatie* genoemd.

De *interpolatiefout* wordt gegeven door de formule

$$(8.1.11) \qquad f(t) - P_{n,\gamma}(t) = \frac{f^{(r)}[\xi(t)]}{r!} \prod_{k=0}^{n} (t-x_k)^{\gamma},$$

waarbij $r = \gamma(n+1)$ en $\xi(t)$ in het kleinste interval ligt dat de punten $x_0,\ldots,x_n$ en t bevat.

De Newton-interpolatie wordt in de practijk het meest gebruikt. Voor niet-equidistante basispunten is het gebruik van Lagrange-interpolatie inefficiënt. Als echter de basispunten equidistant zijn dan kunnen de Lagrange-coëfficienten veelal uit een tabel worden afgelezen zodat veel

rekenwerk kan worden bespaard.

8.1.3. Rationale interpolatie

Als interpolerende functie kiezen we

(8.1.12) $$R(t) = \frac{P(t)}{Q(t)}$$

waarbij P en Q polynomen zijn van graad p en q respectievelijk. We zullen er hier vanuit gaan dat γ in (8.1.1) gelijk is aan 1, dus dat geen afgeleiden gegeven zijn. Dan geldt dat voor vaste p en q zodat p+q = n er een unieke rationale functie R(t) bestaat die voldoet aan

(8.1.13) $$R(x_i) = f_i, \qquad i = 0,1,\ldots,n.$$

Schrijven we

$$P(t) = 1 - a_1 t - \ldots - a_p t^p$$

$$Q(t) = b_0 + b_1 t + \ldots + b_q t^q.$$

Dan zijn de constanten $a_1,\ldots,a_p$ en $b_0,\ldots,b_q$ bepaald door het lineaire stelsel:

$$\begin{pmatrix} x_0 & x_0^2 \ldots x_0^p f_0 & (x_0 f_0) \ldots (x_0^q f_0) \\ \vdots & & \vdots \\ x_n & x_n^2 \ldots x_n^p f_n & (x_n f_n) \ldots (x_n^q f_n) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_p \\ b_0 \\ \vdots \\ b_q \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} .$$

8.1.4. Gebruik van interpolatie voor het oplossen van vergelijkingen

Beschouwen we een reële functie in één variabele. Het probleem dat we ons stellen is een punt z te vinden zodat

$$f(z) = 0.$$

Wanneer de functie en zijn afgeleiden in een aantal punten zijn gegeven dan kunnen we een interpolerende functie bepalen. Berekening van het nulpunt van deze functie geeft ons een benadering van het nulpunt van de gegeven functie. Hoe goed deze benadering is wordt gegeven door (8.1.11).
Wanneer de functie echter voor willekeurige t berekenbaar is en niet slechts is gegeven in een aantal punten, dan zal men in de practijk het nulpunt bepalen door herhaald interpoleren met betrekkelijk eenvoudige interpolatiefuncties. De verzameling van interpolatiepunten wordt dan zo gekozen dat zij één of meer van de laatst berekende benaderingen bevat, teneinde de fout (zie (8.1.11)) zo klein mogelijk te maken.

Een typisch voorbeeld van een iteratieve methode voor de bepaling van een nulpunt van een functie die zo kan worden verkregen is *Newton's methode*. Deze methode maakt gebruik van de interpolerende functie

$$P_{0,2}(t) = f_0 + (t-x_0)f_0'.$$

We krijgen voor het nulpunt x_1 van deze lineaire functie:

$$x_1 = x_0 - \frac{f_0}{f_0'} .$$

Het iteratieve proces wordt dan, bij gegeven x_k, gedefinieerd door:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} , \qquad k = 0,1,\ldots . \tag{8.1.14}$$

Als z het nulpunt is van f, dan geldt voor x_k in voldoend kleine omgeving van z:

$$\varepsilon_{k+1} = O(\varepsilon_k^2), \qquad \text{voor } k \to \infty, \tag{8.1.15}$$

met $\varepsilon_i = x_i - z$. Dus het door (8.1.14) gedefinieerde proces is asymptotisch kwadratisch convergent. Het blijkt ook uit (8.1.14) dat het proces divergeert als $f'(x_k) = 0$ voor zekere k.

Een ander voorbeeld van een iteratieve methode, gebaseerd op herhaald interpoleren is de z.g. *sekantenmethode*. Deze is gebaseerd op de interpolerende functie (8.1.6). Het iteratieve proces wordt bij gegeven x_0 en x_1 gedefinieerd door:

$$x_{k+1} = x_k - \frac{f(x_k)(x_k-x_{k-1})}{f(x_k)-f(x_{k-1})} , \qquad k = 1,2,\ldots . \tag{8.1.16}$$

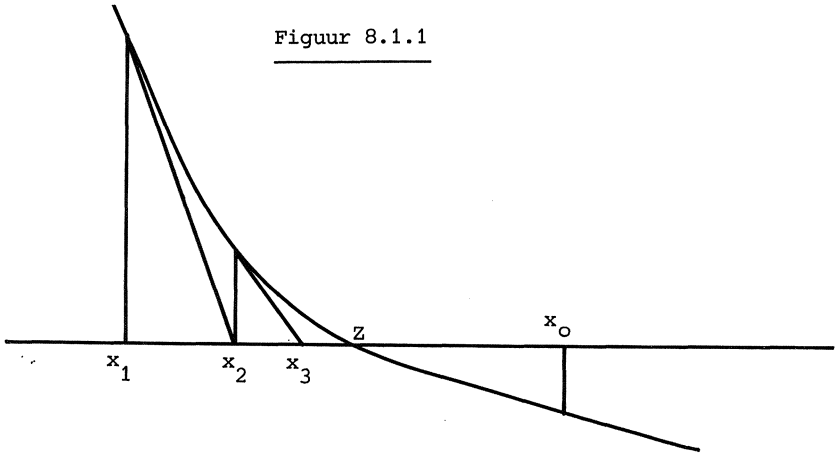
Hier geldt voor de fout asymptotisch:

$$\varepsilon_{k+1} = O(\varepsilon_k \varepsilon_{k-1}), \qquad \text{voor } k \to \infty. \tag{8.1.17}$$

De asymptotische orde van convergentie is hier 1.618. Het is duidelijk dat ook bij gebruik van (8.1.16) divergentie kan optreden.

Er zijn vele variaties bekend op de processen (8.1.14) en (8.1.16). We verwijzen hiervoor naar TRAUB [1964] en DOWELL & JARRAT [1971,1972]. Op één variatie zullen we hier nader ingaan. Als twee punten x_0 en x_1 zijn gegeven zodat $f(x_0)$ en $f(x_1)$ van teken verschillen, dan geldt, aannemend dat f continu is, dat er een nulpunt z ligt tussen x_0 en x_1. We kunnen nu het iteratieve proces zo veranderen dat in elke iteratiestap een interval wordt bijgehouden dat zo klein mogelijk is en waarvan de functiewaarden in de eindpunten van teken verschillen. De lengte van dit interval hoeft niet naar 0 te convergeren ook al convergeert het proces naar de oplossing. Dat blijkt uit figuur 8.1.1. Newton's methode genereert hier een rij intervallen $\{[x_k, x_0]\}_{k=1}^{\infty}$ welke convergeert naar $[z, x_0]$.

Figuur 8.1.1



Wil men toch een methode waarbij de lengte van het interval naar 0 convergeert dan zijn extra modificaties nodig. Bij numerieke berekening van een nulpunt zou het voldoende zijn om een ondergrens ε aan te geven voor de staplengte in een iteratiestap. Dus de iteratiestap wordt dan

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad \text{als} \quad \left|\frac{f(x_k)}{f'(x_k)}\right| \geq \varepsilon$$

$$= x_k + \text{sign}\left(\frac{f(x_k)}{f'(x_k)}\right)\varepsilon, \quad \text{anders.}$$

Methoden die er op gericht zijn om, uitgaande van twee punten met functiewaarden van verschillend teken, een rij intervallen te construeren die het nulpunt bevatten en waarvan de lengte naar 0 convergeert noemen we *insluitings-methoden*. In de volgende paragrafen geven we enkele voorbeelden van insluitings-methoden voor het berekenen van nulpunten en minima van functies van één variabele.

8.1.5. Insluitings-methoden voor het berekenen van nulpunten van functies van één variabele

De eenvoudigste insluitings-methode voor het bepalen van een nulpunt van een functie is de bisektiemethode. Gegeven een interval $[x_0,y_0]$ zodat $f(x_0)$ en $f(y_0)$ van teken verschillen, dan kan deze methode worden gedefinieerd door:

voor $k = 0,1,\ldots$

bereken $m = \frac{1}{2}(x_k+y_k)$

Als $f(m) \times f(x_k) \leq 0$ dan $y_{k+1} = m$

anders $x_{k+1} = m$.

Omdat in elke stap de lengte van het interval wordt gehalveerd is het aantal functie-evaluaties dat nodig is om het nulpunt in een precisie ε te bereiken gelijk aan

$$\sigma = \text{entier}\left(\frac{-^2\log\,\varepsilon}{1+^2\log|y_0-x_0|}\right) + 2. \tag{8.1.18}$$

Bijvoorbeeld voor $\varepsilon = 2^{-48}$ krijgen we $\sigma = 50$. In de practijk blijkt echter dat voor functies met enkelvoudige nulpunten eenvoudige lineaire interpolatie veel sneller tot resultaten leidt. Dikwijls zijn dan bij dezelfde precisie slechts een tiental functie-evaluaties nodig. Daarom lijkt het zinvol te proberen de snelle convergentie van methoden gebaseerd op interpolatie te combineren met de veiligheid van het bisektieproces. In BUS &

DEKKER [1975] worden twee algoritmen gegeven, die naast lineaire interpolatie en bisektie ook gebruik maken van rationale interpolatie (zie paragraaf 8.1.3). Als interpolerende functie wordt gekozen:

$$R(t) = \frac{1-at}{c+bt} .$$

Gegeven de interpolatiepunten x_{k-2}, x_{k-1} en x_k dan geldt voor het nulpunt x_{k+1} van $R(t)$:

$$(8.1.19) \qquad x_{k+1} = x_k - \frac{f[x_{k-1},x_{k-2}]f(x_k)(x_k-x_{k-1})}{f[x_{k-1},x_{k-2}]f(x_k)-f[x_k,x_{k-2}]f(x_{k-1})} .$$

Bij deze formule geldt voor de fout asymptotisch

$$(8.1.20) \qquad \varepsilon_{k+1} = O(\varepsilon_k\varepsilon_{k-1}\varepsilon_{k-2}), \qquad \text{voor } k \to \infty .$$

De asymptotische orde van convergentie van een proces uitsluitend gebaseerd op (8.1.19) is 1.839. Door een geschikte combinatie van (8.1.16) en (8.1.19) kan worden bereikt dat asymptotisch de functiewaarde in één van drie opvolgende iteratiepunten een ander teken heeft dan in de twee andere punten, mits het nulpunt enkelvoudig is. Indien een dergelijke tekenwisseling niet plaats vindt en in situaties waarin een iterand verkregen met interpolatie buiten het interval valt, wordt bisektie toegepast. Op deze wijze wordt bereikt dat de lengte van het interval tenminste wordt gehalveerd in 4 opvolgende iteratie-stappen. Voor een preciese definitie van deze algoritme zij verwezen naar BUS & DEKKER [1975] (Algoritme M). Hierin wordt ook een tweede algoritme gedefinieerd (algoritme R) die hoofdzakelijk gebruik maakt van (8.1.19). Voor deze algoritmen geldt dat het aantal functie-evaluaties dat nodig is voor berekening van een nulpunt in precisie ε, maximaal 4σ (algoritme M) resp. 5σ (algoritme R) bedraagt. De asymptotische orde van convergentie van deze algoritmen is 1.618 resp. 1.839 bij benadering. In bovengenoemde referentie worden ook de teksten gegeven van ALGOL 60 implementaties van deze algoritmen, alsmede een groot aantal testresultaten. Deze implementaties zijn opgenomen in NUMAL (zeroin en zeroinrat). Een routine die erg veel lijkt op deze procedures is opgenomen in IMSL (ZBRENT). Ter illustratie geven we in de bijlage een versie van zeroin in ALGOL 68.

8.1.6. Insluitings-methoden voor het minimaliseren van functies van één variabele

Een methode voor minimaliseren van functies van één variabele die vergelijkbaar is met de bisektiemethode voor bepaling van nulpunten is de *gulden snede-methode*. Voor deze methode geldt dat het interval dat het minimum bevat in elke stap wordt gereduceerd met een factor $1/\tau$ waarbij $\tau = \frac{1}{2}(1+\sqrt{5})$. Een verschil met nulpuntsbepaling is dat hier de functiewaarde in drie punten nodig is om te kunnen bepalen of er een minimum ligt in het interval dat deze punten bevat. Stel $x_0 < x_1 < x_2$, dan heeft f een minimum in (x_0,x_2) als

$$(8.1.21) \qquad f(x_1) < f(x_0) \quad \text{en} \quad f(x_1) < f(x_2).$$

De methode der gulden snede wordt nu, bij gegeven interval $[x_0,y_0]$ dat het minimum bevat, gedefinieerd door:

voor $k = 1,2,\ldots$

bereken $m_k = \frac{\tau-1}{\tau}(y_k-x_k) + x_k$

$$n_k = \frac{1}{\tau}(y_k-x_k) + x_k$$

Als $f(m_k) \leq f(n_k)$ dan $\quad x_{k+1} = x_k, \quad y_{k+1} = n_k,$

anders $x_{k+1} = m_k, \quad y_{k+1} = y_k.$

In feite is door de keuze van τ (er geldt $\tau^2 = \tau+1$) slechts één functie-evaluatie per stap nodig, afgezien van de eerste stap. Er geldt namelijk als $[x_{k+1},y_{k+1}] = [x_k,n_k]$ dat $m_k = n_{k+1}$ en als $[x_{k+1},y_{k+1}] = [m_k,y_k]$ dat $m_{k+1} = n_k$. Het totaal aantal functie-evaluaties dat nodig is voor de bepaling van een minimum met deze methode is bij gegeven precisie van te voren bekend. Bij gebruik van methoden die gebaseerd zijn op successieve interpolatie zal dit aantal echter meestal aanzienlijk kleiner zijn. We bespreken twee insluitings-methoden die gebaseerd zijn op interpolatie, gecombineerd met de gulden snede-methode in die stappen waarin interpolatie niet tot het gewenste resultaat leidt. Deze methoden worden gegeven in BRENT [1973].

Een methode die geen afgeleide gebruikt

Zij gegeven 3 punten x_k, x_{k-1} en x_{k-2}, waarin de functiewaarden f_k, f_{k-1} en f_{k-2}. Dan kan het interpolerend polynoom van graad 2 (zie (8.1.4)) worden geschreven als

$$P_{2,1}(t) = f_k + f[x_k,x_{k-1}](t-x_k) + f[x_k,x_{k-1},x_{k-2}](t-x_k)(t-x_{k-1}). \tag{8.1.23}$$

Voor het minimum van deze functie krijgen we

$$m = \tfrac{1}{2}(x_k+x_{k-1}) - \frac{f[x_k,x_{k-1}]}{2f[x_k,x_{k-1},x_{k-2}]}. \tag{8.1.24}$$

De algoritme is gebaseerd op deze formule. Als m minder dan de op te geven precisie van een vorig punt af ligt dan wordt een minimale stap genomen. Ligt m buiten het beschouwde interval of is in de drie laatste stappen de intervallengte niet voldoende gereduceerd dan wordt een punt gekozen gelijk aan één van de gulden snede-punten (cf. (8.1.22)). Het nieuwe interval wordt steeds zo gekozen dat het een minimum bevat (cf. criterium gegeven door (8.1.21)). Door de gevolgde strategie is de bovengrens voor het aantal functie-evaluaties redelijk (ongeveer s^2, met s het aantal nodig voor het gulden snede-proces) terwijl het proces onder zekere voorwaarden asymptotisch superlineair convergeert. Deze algoritme is beschikbaar in ALGOL 60 in NUMAL (minin).

Een methode die gebruik maakt van de afgeleide

We nemen aan dat de eerste afgeleide van de functie is gegeven. We kunnen nu bepalen of er een minimum ligt in het interval $[x_0,x_1]$ door het teken van de afgeleide in beide eindpunten te beschouwen. Als de afgeleide in x_0 negatief en in x_1 positief is dan ligt er een minimum in het interval. Zij nu gegeven twee punten x_k en x_{k-1} met daarin de functiewaarden f_k en f_{k-1} en de afgeleiden f'_k en f'_{k-1}. We kunnen dan een interpolerend polynoom van de graad 3 bepalen (cf. (8.1.4)):

$$\begin{aligned} P_{2,2}(t) = {} & f_0 + (t-x_0)f'_0 + (t-x_0)^2 f[x_0,2;x_1,1] \\ & + (t-x_1)(t-x_0)^2 f[x_0,2;x_1,2]. \end{aligned} \tag{8.1.25}$$

Voor het minimum van deze functie krijgen we:

$$(8.1.25) \qquad m = \frac{(f_1'+s-z)(x_1-x_0)}{f_1'-f_0'+2w} \quad ,$$

waarbij

$$z = -3f[x_0,x_1] + f_0' + f_1'$$

$$w = \sqrt{z^2-f_0'f_1'} \quad .$$

De formulering hier is die van DAVIDON [1959]. Analoge combinatie van deze interpolatieformule met gulden snede en de keuze van een minimale stap geeft een algoritme waarvoor de bovengrens voor het aantal functie-evaluaties eveneens ongeveer s^2 is en waarvan de asymptotische orde van convergentie hoger is dan van de voorgaande algoritme. Deze algoritme is geimplementeerd in ALGOL 60 en beschikbaar in NUMAL (mininder). De source tekst van deze procedure in ALGOL 60 wordt gegeven in de bijlage en vervangt de tekst welke is gegeven in de NUMAL handleiding.

```
'PROC' ZEROIN=('REF' 'REAL' X, Y, 'PROC' ('REAL') 'REAL' F, TOL)
'BOOL':
'BEGIN' 'INT' EXT:= 0;
    'REAL' FX:= F(X), FY:= F(Y), A:= Y, B;
    'REAL' FA:= FY, FB, TOLX, MX, M;

    'WHILE' ('IF' 'ABS' FY < 'ABS' FX 'THEN'
             'IF' Y 'NE' A 'THEN' B:= A; FB:= FA 'FI';
            A:= X; FA:= FX; X:= Y; FX:= FY; Y:= A; FY:= FA 'FI';
             'ABS'(MX:= (Y - X) * 0.5) > (TOLX:= TOL(X)))
    'DO' 'REAL' W:=
         'IF' EXT > 2 'THEN' MX 'ELSE'
              ('REAL' P:= (X - A) * FX, Q:=
              'IF' EXT <= 1 'THEN' FA - FX 'ELSE'
                   ('REAL' FBX:= (FB - FX) / (B - X),
                   FBA:= (FB - FA) / (B - A);
                   P:= FBA * P; FBX * FA - FBA * FX)
              'FI';
              'IF' P < 0 'THEN' P:= -P; Q:= -Q 'FI';
              'IF' P = 0 'OR' P <= Q * (TOLX *:= 'SIGN' MX)
              'THEN' TOLX
              'ELIF' P <= MX * Q
              'THEN' P / Q 'ELSE' MX
              'FI')
         'FI';
         B:= A; FB:= FA; A:= X; FA:= FX; FX:= F(X+:=W);
         EXT:= 'IF' 'IF' FY>= 0 'THEN' FX>= 0 'ELSE' FX<= 0 'FI'
               'THEN' Y:= A; FY:= FA; 0
               'ELIF' W = MX 'THEN' 0 'ELSE' EXT + 1
               'FI'
    'OD'; 'IF' FY >= 0 'THEN' FX <= 0 'ELSE' FX >= 0 'FI'
'END'
```

```
"REAL" "PROCEDURE" MININDER(X, Y, FX, DFX, TOLX);
"REAL" X, Y, FX, DFX, TOLX;
"BEGIN" "COMMENT" THE FUNCTION IS APPROXIMATED BY A CUBIC AS
    GIVEN BY DAVIDON, 1958, THE STRUCTURE IS SIMILAR TO THE
    STRUCTURE OF THE PROGRAM GIVEN BY BRENT, 1973, THIS IS
    A REVISION OF 760407;

    "INTEGER" SGN;
    "REAL" A, B, C, FA, FB, FU, DFA, DFB, DFU, E, D, TOL, BA,
    Z, P, Q, S;

    "IF" X <= Y "THEN"
    "BEGIN" A:= X; FA:= FX; DFA:= DFX;
        B:= X:= Y; FB:= FX; DFB:= DFX
    "END" "ELSE"
    "BEGIN" B:= X; FB:= FX; DFB:= DFX;
        A:= X:= Y; FA:= FX; DFA:= DFX
    "END";
    C:= (3 - SQRT(5)) / 2; D:= B - A; E:= D * 2; Z:= E * 2;
LOOP: BA:= B - A; TOL:= TOLX; "IF" BA >= TOL * 3 "THEN"
    "BEGIN" "IF" ABS(DFA) <= ABS(DFB) "THEN"
        "BEGIN" X:=A; SGN:= 1 "END" "ELSE"
        "BEGIN" X:= B; SGN:= -1 "END";
        "IF" DFA <= 0 "AND" DFB >= 0 "THEN"
        "BEGIN" Z:= (FA - FB) * 3 / BA + DFA + DFB;
            S:= SQRT(Z ** 2 - DFA * DFB);
            P:= "IF" SGN = 1 "THEN" DFA - S - Z "ELSE"
            DFB + S - Z; P:= P * BA;
            Q:= DFB - DFA + S * 2; Z:= E; E:= D;
            D:= "IF" ABS(P) <= ABS(Q) * TOL "THEN" TOL * SGN
            "ELSE" -P / Q
        "END" "ELSE" D:= BA;
        "IF" ABS(D) >= ABS(Z * 0.5) "OR" ABS(D) > BA * 0.5 "THEN"
        "BEGIN" E:= BA; D:= C * BA * SGN "END";
        X:= X + D; FU:= FX; DFU:= DFX;
        "IF" DFU >= 0 "OR" (FU >= FA "AND" DFA <= 0) "THEN"
        "BEGIN" B:= X; FB:= FU; DFB:= DFU "END" "ELSE"
        "BEGIN" A:= X; FA:= FU; DFA:= DFU "END";
        "GOTO" LOOP
    "END"; "IF" FA < FB "THEN"
    "BEGIN" X:= A; Y:= B; MININDER:= FA "END" "ELSE"
    "BEGIN" X:= B; Y:= A; MININDER:= FB "END"
"END" MININDER;
```

8.2. Sommatie van rijen

8.2.1. Inleiding

Bij sommatie van rijen (reeksen) is het streven een som van de gedaante

$$(8.2.1)\qquad S_I = \sum_{i\in I} f(i), \qquad f : I \to \mathbb{R}^m$$

te berekenen of met bekende nauwkeurigheid te benaderen, zonder alle termen van de rij te hoeven evalueren. Het veronderstellen van bepaalde regelmatigheden in de termen kan voldoende zijn om te volstaan met de berekening van "enkele" termen van de rij of van door bepaalde handelingen te verkrijgen andere rijen. Daarbij beperken we ons meestal tot afbeeldingen f van de vorm

$$(8.2.2)\qquad f: \{0,1,\ldots,n-1\} \to \mathbb{R},$$

waarbij f eenvoudig voort te zetten is tot

$$(8.2.3)\qquad f^*: [0,n) \to \mathbb{R},$$

en waarin n ook wel oneindig kan zijn. De gebruikelijke opgave is dus de bepaling van

$$(8.2.4)\qquad S_n = \sum_{i=0}^{n-1} f(i) \qquad (n \text{ eventueel } \infty).$$

Voor de meeste methoden is gewenst, dat f(i) voor willekeurige i berekend kan worden, desgewenst zonder dat eerst f(i-1) moet worden bepaald.

Aan dit probleem is door vrijwel alle groten der klassieke wiskunde gewerkt, vooral voor het bijzondere geval van oneindige reeksen waarbij $\lim_{i\to\infty} f(i) = 0$ een vereiste, en convergentie van de reeks een extra probleem zijn. Men onderscheidt hierin speciale gedaanten van f(i), zoals

$$f(i;x) = a_i x^i \qquad \text{voor gegeven} \qquad a_0, a_1, \ldots, a_n, \ldots,$$

hetgeen tot klassen van verwante reeksen (hier machtreeksen) leidt, waarbij convergentie van x afhangt. De uiteindelijke berekening van de som van zo'n reeks was meestal ondergeschikt aan de vraag, voor welke x convergentie van

de machtreeks optreedt, en of het antwoord transcendent is of niet.

De numerieke behandeling van sommen van rijen wordt gekenmerkt door het vervangen van (bijna) oneindige processen en formules door eindige processen en formules, met een acceptabel klein aantal handelingen of te berekenen termen. In de praktijk beperken we ons tot getalrijen; enkele methoden zijn ook voor vectorrijen toepasbaar.

8.2.2. Indeling van methoden

Voor de behandeling van de som (8.2.4) staan de volgende gereedschappen ter beschikking

a) *evaluatie van de analytische functie,* waarvan de reeks de machtreeks in een bepaald punt is.
   Voorbeeld: De reeks

$$S = \sum_{i=0}^{\infty} \frac{(-1)^i}{2i+1}$$

   komt van

$$\textit{arctan}\ x = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{2i+1}$$

   die voor $x = 1$ de waarde $S = \frac{\pi}{4}$ geeft. (Voor de evaluatie van deze analytische functies wordt verwezen naar o.m. hoofdstuk 9).

b) Er kan een *somfunctie* of primitieve functie F, behorend bij f(i) gevonden of benaderd worden. Zo'n functie heeft de eigenschap

$$F(i+1)-F(i) = f(i) \qquad \text{voor alle } i.$$

   Dan gaat de som van de rij over in

$$S_n = \sum_{i=0}^{n-1} \{F(i+1)-F(i)\} = F(n)-F(0),$$

   zodat met evaluatie of benadering van twee waarden van de somfunctie kan worden volstaan (zie §8.2.3).
   Voorbeeld:

$$\sum_{i=m}^{n-1} i^2 = F(n)-F(m)$$

   met $F(i) = \frac{1}{3} i^3 - \frac{1}{2} i^2 + \frac{1}{6} i.$

c) *directe sommatie*: de algemene term van de reeks nadert snel tot nul, zodat directe sommatie van een beginstuk van de rij een voldoend goede benadering geeft.

Voorbeeld:

$$\sum_{i=0}^{\infty} \frac{1}{i!} \rightarrow 2.718278 \text{ na } 8 \text{ termen}$$

De fout is nog $-3_{10}-6$.

d) *vervanging* van de reeks door een snel convergente reeks (§8.2.4) of *splitsing* van de reeks in een bekende reeks en een snel convergente reeks (§8.2.3).

Voorbeelden:

1. transformatie van Euler

2. $$\sum_{i=1}^{\infty} \frac{1}{i^2+1} = \sum_{i=1}^{\infty} \left(\frac{1}{i^2+1} - \frac{1}{i^2}\right) + \sum_{i=1}^{\infty} \frac{1}{i^2} = \sum_{i=1}^{\infty} \frac{-1}{i^2(i^2+1)} + \frac{\pi^2}{6} .$$

Voor sommatie van eindige getalrijen is hier alleen methode b) aangewezen. In een aantal gevallen is zo'n getalrij echter te zien als beginstuk van een oneindige te sommeren rij (nl. als de oneindige reeks convergeert). Dan kunnen ook de andere methoden op de reeksen

$$\sum_{i=0}^{\infty} f(i) \quad \text{en} \quad \sum_{i=n}^{\infty} f(i)$$

worden toegepast, waarna

$$\sum_{i=0}^{n-1} f(i) = \sum_{i=0}^{\infty} f(i) - \sum_{i=n}^{\infty} f(i)$$

volgt.

Heel gebruikelijk is verder, dat transformatie van een reeks (d) gevolgd wordt door toepassing van b) of c) of beide.

8.2.3. Eindige differentie-formules

Voor het gebruik van formules beperken we ons tot de introductie van één operator, nl. de *voorwaartse differentie-operator* Δ. Wie zich verder in dit gereedschap wil verdiepen, verwijzen we naar klassieke boeken als

HILDEBRAND [1956] en JORDAN [1947], waar ook de alternatieven met *achterwaartse en centrale differentie-operator* en met de *differentiaal-operator* gegeven worden.

DEFINITIE 8.2.1. Δ is een operator die aan functies $f: [a,b] \to \mathbb{R}$ nieuwe functies Δf toevoegt ($\Delta f: [a,b-h] \to \mathbb{R}$) op zodanige wijze, dat voor elke x_0 geldt:

$$\Delta f(x_0) = f(x_0+h)-f(x_0). \tag{8.2.5}$$

Hierin is h een globale constante, de basispuntafstand, die in onze toepassingen gelijk aan 1 is.

Als er nu een functie F bestaat, waarvoor geldt dat

$$f(i) = \Delta F(i) = F(i+1)-F(i)$$

voor elke term van de som (8.2.4), dan gaat deze formule over in

$$S_n = \sum_{i=0}^{n-1} \Delta F(i) = F(n)-F(0). \tag{8.2.6}$$

Duidelijk is, dat als $F_1(x)$ voldoet, dan ook $F_2(x) = F_1(x) + C$ voor willekeurige C. Een extra afspraak, zoals $F(0) = 0$ is minstens nodig. Bij F(x) kan echter ook elke periodieke functie met periode 1 opgeteld worden, dus ook eisen van continuïteit en gladheid zijn geboden. Voor het bruikbaar zijn van F(x) nemen we in het vervolg aan, dat hij *interpoleerbaar* is (indien er een tabel met afstand 1 van gemaakt zou worden).

Gezien de relatie $\Delta F(i) = f(i)$ zullen we voor F(i) in het vervolg $\Delta^{-1}f(i)$ schrijven. We noemen Δ^{-1} de *voorwaartse som-operator*.

De gevallen, waarin $\Delta^{-1}f$ een bekende functie is, zijn beperkt. Zo is bij de som $\sum_{i=0}^{n-1} i^2$ de somfunctie $\Delta^{-1}f(i) = \frac{1}{3} i^3 - \frac{1}{2} i^2 + \frac{1}{6} i$ te vinden. Veel afleidingen van somfuncties worden verkregen met *Stirling-getallen* en *Bernoulli-polynomen*. Een bruikbare relatie is

$$\Delta k^{(n)} = n.k^{(n-1)}$$

waardoor een symbolisch analogon van integratie- en differentiatie-methoden is verkregen ($k^{(n)} = k(k-1)...(k-n+1)$).

In de volgende paragrafen worden enkele manieren gegeven om de somfunctie $\Delta^{-1}f$ numeriek te benaderen.

8.2.3.1. Somfuncties

Zij E de *schuif-operator*, gedefinieerd door

DEFINITIE 8.2.2.

(8.2.7) $\quad Ef(i) = f(i+1).$

Dan geldt $E = \Delta + I$. We kunnen dan voor de som van een rij ook schrijven

$$(8.2.8) \qquad \sum_{i=0}^{n-1} f(i) = f(0)+f(1) + \ldots + f(n-1)$$

$$= f(0) + Ef(0) + \ldots + E^{n-1}f(0) = \sum_{i=0}^{n-1} E^i f(0).$$

Symbolische herleiding van de operator $\sum_{i=0}^{n-1} E^i$ geeft:

$$(8.2.9) \qquad \sum_{i=0}^{n-1} E^i = \frac{E^n-I}{E-I} = \frac{(I+\Delta)^n-I}{\Delta} = \sum_{j=1}^{n} \binom{n}{j}\Delta^{j-1}.$$

Daarmee vinden we dus voor Δ^{-1}

$$(8.2.10) \qquad \Delta^{-1}f(i) = \sum_{j=1}^{i} \binom{i}{j}\Delta^{j-1}f(0).$$

Hiermee is de som van functie-waarden herleid tot een som van differenties. Deze formule is bruikbaar als f(x) een polynoom is, daar dan de meeste differenties nul zijn.

Voorbeeld:

$$\sum_{i=0}^{n-1} i^2 = \sum_{j=1}^{n} \binom{n}{j}\Delta^{j-1}f(0).$$

Het differentie-schema is:

$f(i)$	$\Delta f(i)$	$\Delta^2 f(i)$	$\Delta^3 f(i)$
0			
	1		
1		2	
	3		0
4		2	⋮
	5	⋮	
9	⋮		

waaruit volgt:

$$\sum_{i=0}^{n-1} i^2 = \binom{n}{2} \cdot 1 + \binom{n}{3} . 2 = \frac{1}{6} n(n-1)(2n-1).$$

De formule (8.2.10) is niet bruikbaar voor oneindige reeksen. Wel kan een repertoire van bekende somfuncties nuttig zijn. Genoemd werd al

$$\Delta k^{(n)} = n k^{(n-1)} \quad \text{of} \quad \Delta^{-1} k^{(n)} = \frac{1}{n+1} k^{(n+1)}.$$

Hierbij wordt bovendien voor positieve n $k^{(-n)}$ gedefinieerd door

$$k^{(-n)} = \frac{1}{(k+n)^{(n)}} .$$

In dat geval geldt

(8.2.11) $\Delta^{-1} k^{(n)} = \frac{1}{n+1} k^{(n+1)}$ voor alle gehele $n \neq -1$.

<u>Voorbeeld</u>: Voor

$$\sum_{i=0}^{n-1} f(i) = \sum_{i=0}^{n-1} \frac{1}{(i+1)(i+2)(i+3)}$$

vinden we

$$\sum_{i=0}^{n-1} \frac{1}{(i+3)^{(3)}} = \sum_{i=0}^{n-1} i^{(-3)}$$

$$\Delta^{-1} f(i) = \frac{1}{-2} i^{(-2)} = -\frac{1}{2} \frac{1}{(i+2)^{(2)}} ,$$

en dus

$$\sum_{i=0}^{n-1} f(i) = -\frac{1}{2}\left\{\frac{1}{(n+2)^{(2)}} - \frac{1}{2^{(2)}}\right\} = \frac{1}{4} - \frac{1}{2(n+2)(n+1)} .$$

Deze somfunctie is ook voor oneindige reeksen bruikbaar. We vinden dan

$$\sum_{i=0}^{\infty} \frac{1}{(i+1)(i+2)(i+3)} = \frac{1}{4} .$$

Een andere bekende somfunctie vinden we voor *binomiaalcoefficienten*. Daar geldt

$$(8.2.12) \qquad \Delta\binom{k}{p+1} = \binom{k}{p} \qquad \text{voor vaste } p \geq 0,$$

volgt

$$(8.2.13) \qquad \sum_{k=0}^{n-1} \binom{k}{p} = \sum_{k=0}^{n-1} \Delta\binom{k}{p+1} = \binom{n}{p+1} - \binom{0}{p+1} = \binom{n}{p+1}.$$

Tenslotte is een bruikbaar gereedschap de *partiële sommatie* (als analogon van partiële integratie):

$$(8.2.14) \qquad \sum_{i=0}^{n-1} u_i.\Delta v_i = u_n v_n - u_0 v_0 - \sum_{i=0}^{n-1} v_{i+1}\Delta u_i$$

daar

$$\Delta(u_i v_i) = u_i \Delta v_i + v_{i+1}\Delta u_i.$$

Voorbeeld:

$$\sum_{i=0}^{n-1} i.x^i = n.\frac{x^n}{x-1} - \sum_{i=0}^{n-1} \frac{x^{i+1}}{x-1} = n.\frac{x^n}{x-1} - \frac{x}{x-1}\frac{x^n-1}{x-1} .$$

8.2.3.2. Benadering met integralen

Bij numerieke kwadratuur wordt een bepaalde integraal vaak benaderd door een som van oppervlakten van rechthoekjes, of, als de integrand door hogere-graadspolynomen stuksgewijze benaderd wordt, een som van bekende integralen. Een eenvoudige benadering is, als het integratie-interval (x_0, x_n) in n gelijke deelintervallen (x_j, x_{j+1}) is verdeeld, met

(8.2.15) $x_j = x_0 + j.h \quad$ waarbij $h = \frac{x_n - x_0}{n}$,

de *geregen trapeziumregel*:

(8.2.16) $$\frac{1}{h}\int_{x_0}^{x_n} f(x)dx = \tfrac{1}{2} f(x_0) + \sum_{j=1}^{n-1} f(x_j) + \tfrac{1}{2} f(x_n).$$

Als we de integrand benaderen met *interpolatiepolynomen van Newton,* is voor de integraal op één deelinterval de volgende benadering te maken:

(8.2.17) $$\frac{1}{h}\int_{x_j}^{x_{j+1}} f(x)dx = f_j + \frac{1}{2}\Delta_j - \frac{1}{12}\Delta_j^2 + \frac{1}{24}\Delta_j^3 - \ldots \quad (\Delta_j^i = \Delta^i f(x_j))$$

formule van Laplace

Sommatie over alle deelintervallen geeft (daar $\Delta_j^i = \Delta_{j+1}^{i-1} - \Delta_j^{i-1}$)

(8.2.18) $$\frac{1}{h}\int_{x_0}^{x_n} f(x)dx = (\Delta_n^{-1} - \Delta_0^{-1}) + \tfrac{1}{2}(f_n - f_0) - \frac{1}{12}(\Delta_n - \Delta_0) + \frac{1}{24}(\Delta_n^2 - \Delta_0^2) - \ldots$$

Blijkbaar geldt dan voor de onbepaalde integraal van a tot x_k, waarin a nog vast te kiezen is (zodat de integraal op een constante na bepaald is):

(8.2.19) $$\frac{1}{h}\int_{a}^{x_k} f(x)dx = \Delta_k^{-1} + \frac{1}{2}\Delta_k^0 - \frac{1}{12}\Delta_k^1 + \frac{1}{24}\Delta_k^2 - \ldots$$

Dit gereedschap kunnen we ook omdraaien, daar uit (8.2.19) een formule voor de som-operator volgt:

(8.2.20) $$\Delta_k^{-1} = \frac{1}{h}\int_{a}^{x_k} f(x)dx - \frac{1}{2}\Delta_k^0 + \frac{1}{12}\Delta_k - \frac{1}{24}\Delta_k^2 + \ldots$$

De waarde a is hier niet belangrijk, daar deze bij toepassing op een som wegvalt:

(8.2.21) $$\sum_{i=0}^{n-1} f(x_i) = \Delta_n^{-1} - \Delta_0^{-1} = \frac{1}{h}\int_{x_0}^{x_n} f(x)dx - \frac{1}{2}(f_n - f_0) + \frac{1}{12}(\Delta_n - \Delta_0) - \frac{1}{24}(\Delta_n^2 - \Delta_0^2) + \ldots$$

Als de integraal bekend is, is dus de som te benaderen door de bijbehorende integraal plus enige differentie-correcties. Voorwaarde hiervoor is, dat de integraal eindig is, en dat de differentie-termen snel verwaarloosbaar zijn.

Als de hogere orde differenties tam zijn, mag men hierop rekenen; bij het schatten wordt gebruik gemaakt van de relatie

$$\Delta_i^k = \frac{f^{(k)}(\xi)}{k!} h^k \quad \text{voor zekere} \quad \xi \in (x_i, x_{i+k}).$$

De afbreekfout is dan ruwweg de eerste verwaarloosde term:

$$\alpha_k \cdot (\Delta_n^k - \Delta_0^k).$$

De benadering (8.2.21) is uitstekend bruikbaar voor oneindige reeksen, indien de integraal bestaat en indien de functie en alle differenties naar nul gaan voor $x \to \infty$. Dan vinden we de relatie

$$(8.2.22) \qquad \sum_{i=0}^{\infty} f(x_i) = \Delta_\infty^{-1} - \Delta_0^{-1} = \frac{1}{h} \int_{x_0}^{\infty} f(x)dx + \frac{1}{2} f_0 - \frac{1}{12} \Delta_0 + \frac{1}{24} \Delta_0^2 - \ldots$$

Wel kan het wenselijk zijn (opdat de gebruikte hogere-orde-differentietermen inderdaad snel kleiner worden) om een kort beginstuk zonder meer te sommeren. We vinden dan:

$$(8.2.23) \qquad \sum_{i=0}^{\infty} f(x_i) = \sum_{i=0}^{k-1} f(x_i) + \sum_{i=k}^{\infty} f(x_i) =$$

$$= \sum_{i=0}^{k-1} f(x_i) + \frac{1}{h} \int_{x_k}^{\infty} f(x)dx + \frac{1}{2} f_k - \frac{1}{12} \Delta_k + \frac{1}{24} \Delta_k^2 - \ldots$$

<u>Voorbeeld</u>: Voor de reeks $\sum_{i=1}^{\infty} \frac{1}{i^2}$ berekenen we

$$1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \int_5^{\infty} \frac{dx}{x^2} + \frac{1}{2} f_5 - \frac{1}{12} \Delta_5 + \frac{1}{24} \Delta_5^2 - \frac{19}{720} \Delta_5^3 =$$

$$= 1.644892 \qquad \text{(laatste term = 0.000060)}.$$

Het echte antwoord is $\frac{\pi^2}{6} = 1.644934 \ldots$.

Een nadeel van de formule (8.2.21) kan zijn, dat aan het eind van het traject de hogere-orde-differenties functiewaarden buiten het interval bevatten. Dit bezwaar kan worden opgevangen door aan het eind een aantal functiewaarden zonder meer te sommeren waardoor de gebruikte differenties wel binnen het interval blijven. Andere differentieformules hebben dit bezwaar niet of minder, zoals de <u>formule van Gregory</u> (met voorwaartse en achterwaartse

differenties) en de formules van Gauss en Stirling (beide met centrale differenties).

De formules met differenties zijn ook te herleiden tot formules met afgeleiden, door alle differenties uit te drukken in de afgeleiden in de uiteinden. Zo vinden we de formule van Euler-Maclaurin

$$(8.2.24)\qquad \sum_{i=0}^{n-1} f(x_i) = \frac{1}{h}\int_{x_0}^{x_n} f(x)dx - \frac{1}{2}(f_n - f_0) + \frac{1}{12}h(f_n' - f_0') - \frac{h^3}{720}(f_n^{(3)} - f_0^{(3)}) + \frac{h^5}{30240}(f_n^{(5)} - f_0^{(5)}) - \ldots$$

8.2.4. Transformatie van reeksen

Naast extrapolatie-technieken voor iteratieve processen bestaan er nauwelijks versnellingsmethoden die speciaal voor het sommeren van reeksen gebruikt worden. De bekendste is afkomstig van *Euler*, en is met name geschikt voor langzaam convergerende alternerende reeksen. Bij reeksen met uitsluitend positieve termen is het schatten van de afbreekfout in feite de moeilijkste taak daar de som van de reeks willekeurig langzaam van een kant door de rij van partiële sommen benaderd wordt (bij alternerende reeksen wordt de som eenvoudig ingesloten). De oplossing hiervoor is het vinden van een scherpe majorante van de staart van de reeks. Voor praktisch werk (ook automatisch) is wel bruikbaar de *transformatie van Van Wijngaarden* (§8.2.4.2.).

8.2.4.1. Euler-transformatie

De transformatie van Euler is in wezen een hergroepering van te sommeren termen. Een afleiding in termen van differenties is de volgende. Zij de reeks

$$(8.2.25)\qquad S = \sum_{i=0}^{\infty} (-1)^i f(i) \quad \text{met} \quad f(i) \geq 0 \quad \text{en} \quad \lim_{i\to\infty} f(i) = 0.$$

Dan geldt:

$$S = f_0 - Ef_0 + E^2 f_0 - E^3 f_0 + \ldots = \sum_{i=0}^{\infty} (-1)^i E^i f_0 =$$

$$= (1+E)^{-1} f_0 = (2+\Delta)^{-1} f_0 = \frac{1}{2}(1+\frac{\Delta}{2})^{-1} f_0 =$$

$$= \frac{1}{2}[1 - \frac{\Delta}{2} + \frac{\Delta^2}{4} - \frac{\Delta^3}{8} + \ldots] f_0 = \frac{1}{2}\sum_{i=0}^{\infty} (-\frac{1}{2})^i \Delta^i f_0.$$

Hiervan verschijnen alle termen in een gewijzigd differentie-schema, waarin bij elke bewerking bovendien door 2 gedeeld wordt:

i	f_i	$\frac{1}{2}\Delta f_i$	$\frac{1}{4}\Delta^2 f_i$	$\frac{1}{8}\Delta^3 f_i$ ...
0	f_0			
		$\frac{1}{2}\Delta f_0$		
1	f_1		$\frac{1}{4}\Delta^2 f_0$	
		$\frac{1}{2}\Delta f_1$		$\frac{1}{8}\Delta^3 f_0$
2	f_2		$\frac{1}{4}\Delta^2 f_1$	⋮
		$\frac{1}{2}\Delta f_2$	⋮	
3	f_3	⋮		
⋮	⋮			

De rij getallen langs de bovenkant van de driehoek vormen nu (met de goede tekens) een nieuwe reeks die sneller convergeert dan de oude, indien de oorspronkelijke rij langzamer convergeert dan een meetkundige rij met reden $-\frac{1}{3}$.

Voorbeeld: De reeks

$$\sum_{i=0}^{\infty} (-\tfrac{4}{5})^i \quad \text{gaat, daar}$$

$$\tfrac{1}{2}\Delta f_i = -\tfrac{1}{10}(\tfrac{4}{5})^i = -\tfrac{1}{10} f_i, \quad \text{over in}$$

$$\tfrac{1}{2}(1+\tfrac{1}{10}+\tfrac{1}{100}+\tfrac{1}{1000}+\ldots) = 0.5555\ldots$$

8.2.4.2. Van Wijngaarden's transformatie

Langzaam convergerende reeksen met uitsluitend positieve termen zijn de vervelendste reeksen om te sommeren, daar de som niet door partiële sommen ingesloten wordt, terwijl moeilijk te schatten is hoever $\sum_{i=0}^{N} f(i)$ voor zekere N nog van $\sum_{i=0}^{\infty} f(i)$ afligt.

VAN WIJNGAARDEN [1965] (zie ook FRÖBERG [1966]) heeft voor deze gevallen

een methode ontworpen, die zo'n reeks splitst in oneindig veel snel convergerende reeksen, die zelf dus elk weer term zijn van een oneindige maar nu alternerende reeks. Deze methode is a.v. Zij de reeks $\sum_{k=1}^{\infty} u_k$. Laat v_k een nieuwe reeks zijn (voor elke k), nl.:

$$(8.2.26) \qquad v_k = u_k + 2u_{2k} + 4u_{4k} + 8u_{8k} + \ldots = \sum_{j=0}^{\infty} 2^j u_{2^j.k} \quad .$$

Hierbij noteren we de eigenschap

$$(8.2.27) \qquad v_k = u_k + 2v_{2k}.$$

Daarmee maken we van de oorspronkelijke reeks:

$$(8.2.28) \qquad \sum_{k=1}^{\infty} u_k = \sum_{k=1}^{\infty} (v_k - 2v_{2k}) = \sum_{k=1}^{\infty} (-1)^{k-1} v_k,$$

wat toegestaan is, als de oorspronkelijke reeks convergeert en de rij der (positieve) termen u_k niet-stijgend is. Dan convergeren alle reeksen v_k(8.2.26), terwijl de reeks (8.2.28, rechterlid) een convergente alternerende reeks is.

Voorbeeld: Voor de reeks $\sum_{k=1}^{\infty} \frac{1}{k^2}$ vinden we

$$v_k = \frac{1}{k^2}\left(1 + \frac{2}{4} + \frac{4}{16} + \frac{8}{64} + \ldots\right) = \frac{2}{k^2},$$

dus

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = 2 \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^2} \quad .$$

8.2.5. Programmatuur

Onderzocht zijn de programmatheken ACCULIB, CERNLIB, IMSL, MSL en NAG, waarin geen subroutines voor sommatie van getalrijen (reeksen) werden gevonden.

Alleen de bibliotheken NUMAL (ALGOL 60) en NUMPAS (PASCAL) bevatten de procedures *euler* (transformatie van Euler voor alternerende reeksen) en *sumposseries* (transformatie van Van Wijngaarden voor reeksen met positieve termen).

8.2.5.1. De procedure *euler*

De procedure *euler* is afgeleid van de gelijknamige procedure uit NAUR [1963]. Hierin wordt een aan *Van Wijngaarden* ontleende strategie gebruikt om te beslissen of de termen van de reeks zonder meer gesommeerd moeten worden of gebruikt moeten worden om hogere orde differenties voor de getransformeerde reeks te berekenen. Twee parameters (of in de NUMPAS-versie de velden van een record) *eps* en *tim* dienen om te stoppen: de sommatie wordt beëindigd als *tim* opeenvolgende termen van de (getransformeerde) reeks absoluut kleiner zijn dan *eps*.
Voorbeeld: (in PASCAL): De reeks

$$\sum_{i=0}^{\infty} \frac{(-1)^i}{i+1} \quad \text{wordt met } eps = 10^{-6}$$

berekend met het volgende programma:

```
PROGRAM eul(output);

    TYPE reca = RECORD eps : real;
                       tim, lasti : integer
                END;

    VAR aux : reca;

    FUNCTION fi(i : integer) : real;
    BEGIN fi:=  (1 - i MOD 2 * 2) / (i + 1) END;

BEGIN WITH aux DO BEGIN eps:= 1E-6; tim:= 4 END;

    writeln(' som = ', euler(fi, aux) : 20,'aantal termen : ', aux.
                    lasti : 5)
END.
```

De uitvoer van dit programma is:

```
som = 6.93147185524E-001 aantal termen : 15
(Vgl. echte waarde ln 2 = 0.6931471805...)
```

8.2.5.2. De procedure *sumposseries*

De Van Wijngaarden-transformatie is geïmplementeerd door *J.W. Daniel* (zie o.m. DANIEL [1969]) in de procedure *sumposseries*. Afhankelijk van de door de gebruiker gegeven tolerantie wordt de som benaderd door een beginstuk te sommeren of wordt de som van een getransformeerde reeks bepaald. Voor de getransformeerde reeks wordt de procedure *euler* aangeroepen, terwijl de reeksen v_k (8.2.26) berekend worden door *sumposseries* recursief aan te roepen. Meestal convergeren alle reeksen v_k, die voor euler nodig zijn, snel genoeg om door directe sommatie te berekenen. In verband met relatie (8.2.27) bevat *sumposseries* een geheugen (gesteld op 1000 plaatsen) om de berekende v_k te bewaren, zodat later v_{2k} gemakkelijk te vinden is; reeksen (8.2.26) hoeven dan alleen voor oneven k echt berekend te worden. Aan *sumposseries* worden toleranties *maxzero (max.zero)* en *tim (max.tim)* meegegeven die bestemd zijn voor *euler* (als *eps* en *tim*), terwijl de parameter *maxaddup (max.addup)* gebruikt wordt om te beslissen zoveel termen direct te sommeren of te transformeren. Tenslotte beperkt *maxrecurs (max.recurs)* de recursiediepte en *(max.)machexp* is de grens voor de exponent β, waarboven $2^{\beta}.u_{2^{\beta}}$ verwaarloosd mag worden (dit is nodig omdat 2^{β} overflow kan geven).

Voorbeeld: Door het volgende programma wordt $\sum_{k=1}^{\infty} \frac{1}{k^2}$ berekend met tolerantie $_{10}-7$, *max.tim* = 10 en *max.addup* = 20. Dit houdt in, dat er directe sommatie van 30 termen plaats vindt als de laatste 10 termen kleiner dan de tolerantie zijn.

```
PROGRAM sump(output);

    TYPE recmax = RECORD zero : real;
                    addup, recurs, machexp, tim : integer
                  END;

    VAR count : integer; max : recmax;

    FUNCTION fk(k : real) : real;
    BEGIN fk:= 1 / sqr(k); count:= count + 1 END;

BEGIN WITH max DO
   BEGIN addup:= 20; zero:= 1E-7; recurs:= 4;
      machexp:= 100; tim:= 10
   END;
```

```
    count:= 0; writeln(' som = ', sumposseries
      (fk, max) : 20, ' aantal termen :', count : 5)
END.
```

resultaat:

```
som = 1.64493406607E+000 aantal termen : 501.
```

Opmerking: het is economischer om *max.addup* groter te kiezen, omdat de recursiediepte dan afneemt.

8.2.6. Sommatie van vectorrijen

Laat het equivalent van (8.2.4) voor N-vectoren als volgt voorgesteld zijn:

$$(8.2.29) \qquad S = \sum_{i=0}^{n-1} F(i) \quad \text{met} \quad F(i) = (f_1(i),\ldots,f_N(i))^T,\ S = (S_1,\ldots,S_N)^T.$$

De bekende sommatie-methoden kunnen hiervoor componentsgewijs worden toegepast. Dit zal echter een weinig succesvolle manier zijn, vooral bij oneindige reeksen, omdat dan de vectornormen $\|F(i)\|$ voor toenemende i wel voldoende klein zijn, maar de verschillende componenten kunnen schijnbaar zeer onregelmatig veranderen daar F(i) om 0 heen draait. Indien deze draaiing zo regelmatig is, dat hij door een constante matrix kan worden voorgesteld (F(i+1) = A.F(i)), kan over de som wel meer gezegd worden doch dit is slechts een zeer speciaal geval (dat meer bij iteratieve processen thuishoort): dan geldt

$$\sum_{i=0}^{n-1} F(i) = \sum_{i=0}^{n-1} A^i F_0 = (I-A)^{-1}(I-A^n).F_0.$$

Voor $n \to \infty$ zijn de voorwaarden voor convergentie duidelijk: er moet gelden $\|A\| < 1$; dan volgt:

$$\sum_{i=0}^{\infty} A^i.F_0 = (I-A)^{-1}.F_0.$$

Bij andere iteratie-functies is eveneens een analytische herleiding denkbaar. Zo kennen we alle equivalenten van machtreeksen, bijv. $\sum_{i=0}^{\infty} \frac{A^i}{i!}.F_0 = \exp(A).F_0$.

Ook andere methoden voor het behandelen van sommen van getalrijen laten zich wel generaliseren voor vectorrijen, maar de te verrichten inspanning

neemt snel toe, terwijl de praktische waarde gering is.

Er is geen programmatuur voor het sommeren van vectorrijen.

LITERATUUR

BRENT, R.P. [1973], *Algorithms for minimization without derivatives*, Prentice Hall.

BUS, J.C.P. & T.J. DEKKER [1975], *Two efficient algorithms with guaranteed convergence for finding a zero of a function*, TOMS 1, 330-345.

DANIEL, J.W. [1969], *Summation of a series of positive terms by condensation transformations*, Math. of Comp. 23, 91-96.

DAVIDON, W.C. [1959], *Variable metric methods for minimization*, A.E.C. Res. and Development Report, ANL-5990.

DOWELL, M. & P. JARRAT [1971], *A modified regula falsi method for computing the root of an equation*, BIT 11, 168-174.

DOWELL, M. & P. JARRAT [1972], *The "Pegasus" method for computing the root of an equation*, BIT 12, 503-508.

FRÖBERG, C.E. [1966], *Introduction to numerical analysis*, Addison-Wesley.

HILDEBRAND, F.B. [1956], *Introduction to numerical analysis*, McGraw-Hill.

JORDAN, C. [1947], *Calculus of finite differences*, Chelsea (2nd ed.).

NAUR, P. (ed.) [1963], *Revised report on the algorithmic language ALGOL 60*, Springer.

SCHEID, F. [1968], *Theory and problems of numerical analysis*, McGraw-Hill.

TRAUB, J.F. [1964], *Iterative methods for the solution of equations*, Prentice Hall.

WIJNGAARDEN, A. VAN [1965], *Cursus Wetenschappelijk Rekenen B, Procesanalyse*, syllabus CR 18, Mathematisch Centrum.

9. APPROXIMATIE VAN FUNCTIES EN DATA

door C.G. van der Laan

(Rekencentrum, RU Groningen)

Inhoudsopgave pag.

9.0. Inleiding

Bij het samenstellen van deze bijdrage aan het Colloquium Numerieke Programmatuur hebben wij ons laten leiden door de vraag:

wat willen "gebruikers" met approximatie?

Onze ervaring gaf de antwoorden:

1) Parameters schatten (expliciet: zie colloquiumverslag deel 1b, sectie 5.2, 5.3;
impliciet: zie colloquiumverslag deel 1b, sectie 3.1);

2) Afgeleide bepalen (zie CULLUM [1971], ANDERSSEN c.s. [1974], of differentieer de benaderende spline);

3) Integraal bepalen (zie DAVIS c.s. [1975], of integreer de benaderende spline);

4) (Inverse) extrapolatie (zie dit colloquiumverslag, hoofdstuk 8, of JOYCE [1971]);

5) Analyse van tijdreeksen en/of signalen (zie colloquiumverslag deel 1b, sectie 4.2, of Digital Signal Processing 1, 2, IEEE-press);

6) "Plaatjes" maken (zie plotpakketten);

7) Data reduceren;

8) Gemakkelijke evaluatie (zie ook colloquiumverslag deel 1b, sectie 4.1).

Aan de hand van de beschikbare programmatuur in de programmatheken

ACCULIB - versie 6
IMSL - editie 4
NAG - mark 5
NUMAL - versie 18

wordt aandacht geschonken aan 7) en 8).
In hoofdstuk 9.1 beschouwen wij benaderingen met polynomen. Deze technieken zijn nuttig als de data een rustig verloop hebben, zodat de graad van het benaderend polynoom laag gehouden kan worden.
In hoofdstuk 9.2 beschouwen wij benaderingen met spline-functies, die als verruiming van de klasse van polynomen opgevat kunnen worden en lokaal zijn.
In hoofdstuk 9.3 laten wij zien dat de lineaire technieken gebruikt kunnen worden bij een grote klasse van separabele niet-lineaire problemen.

Wij hopen dat het werk van SMITH [1969] en/of PAYNE [1970] zal resulteren in een makkelijk te gebruiken pakket, en dat daarmee dit werk overbodig is geworden.

Aan het eind van dit hoofdstuk is een inhoudsopgave en een index opgenomen, zodat een gebruiker het verlangde kan vinden zonder het geheel te lezen.

Als notatie voor een vector gebruiken wij een kleine letter, bijvoorbeeld: f, of ondergeïndiceerde kleine letters, gescheiden door komma's en omsloten door een haakjespaar, bijvoorbeeld: $(f_1,f_2,\ldots,f_n)$, of een verzamelingsnotatie, bijvoorbeeld: $\{f_k\}_{k=1}^{n} \in \mathbb{R}^n$. Een element van een vector geven wij aan door een ondergeïndiceerde kleine letter, bijvoorbeeld: f_k. Voor een matrix gebruiken wij hoofdletters. Het i-de element uit de j-de kolom van een matrix A geven wij aan met A_{ij}. Voor een functie gebruiken wij ook kleine letters. Een functiewaarde geven wij aan met de naam, met daarachter het argument (of meerdere) tussen haakjes, bijvoorbeeld: f(x). Een inproduct noteren wij met de "bra" en "kets", bijvoorbeeld: <f,g>. De klasse van polynomen van graad n duiden wij aan met Π_n.

9.1. Benadering met polynomen

9.1.1. Polynoomrepresentaties

In paragraaf 9.1.1.1 *geven we verschillende representatievormen. In paragraaf* 9.1.1.2 *geven we een maat voor de conditie van de representaties.*

In paragraaf 9.1.1.3 *geven we voorbeelden van bekende representaties.*

9.1.1.1. Representatievormen

Een polynoom van de graad n kunnen we voorstellen door:

(9.1.1) $f(a;x) = \sum_{k=0}^{n} a_k x^k$ (machtssom);

(9.1.2) $g(b;x) = \sum_{k=0}^{n} b_k T_k(x)$ (Chebyshevsom);

(9.1.3) $h(c;x) = \sum_{k=0}^{n} c_k \phi_k(x)$ (som van overige orthogonale polynomen);

(9.1.4) $\sum_{k=0}^{n} d_k \prod_{\ell=0}^{k-1} (x-x_\ell)$ (Newtonvorm);

(9.1.5) $\sum_{k=0}^{n} \ell_k \prod_{\substack{j=0 \\ j\neq k}}^{n} (x-x_j)/(x_k-x_j)$ (Lagrangesom).

9.1.1.2. Conditie

Als maat voor de conditie van een representatie definiëren wij de conditiefunctie als de 1-norm van de vector van de relatieve afgeleiden naar de parameters; bijvoorbeeld:

(9.1.6) $K(f(a;x)) ::= \sum_i \left| \frac{a_i}{f(a;x)} \partial_{a_i} f(a;x) \right|$, voor (9.1.1),

(9.1.7) $K(g(b;x)) ::= \sum_i \left| \frac{b_i}{g(b;x)} \partial_{b_i} g(b;x) \right|$, voor (9.1.2).

STELLING 9.1.1 (NEWBERY [1974]). *Als de machtssom (of Chebyshevsom) tekenvaste of strikt alternerende coëfficiënten heeft, dan geldt*

(9.1.8) $\max_{x\in[-1,1]} K(f(a;x)) = \max_{x\in[-1,1]} K(g(b;x))$.

Het bewijs volgt uit de waarden van de conditiefuncties voor $x = \pm 1$; zie Lemma in bijlage 1.

9.1.1.3. Voorbeelden

VOORBEELD 9.1.1 (Conditie van de representatie van een afgekapte Chebyshev-reeksbenadering van J_0 (CLENSHAW [1962]).

Als benadering van $J_0(x)$ geeft Clenshaw de vormen:

$$\sum_{k=0}^{12'} b_{2k}\, T_{2k}(x/8) = \sum_{k=0}^{12} a_{2k}(x/8)^{2k}.$$

De bijbehorende conditiegetallen zijn

$$K(f(a_0,a_2,\ldots,a_{24};\pm 8)) = 427,$$

$$K(g(b_0,b_2,\ldots,b_{24};\pm 8)) = 1.$$

VOORBEELD 9.1.2 (Conditie van machtssom versus Chebyshevsom, RUTISHAUSER [1968a]).

De polynoomrepresentaties

$$P_5(x) = \sum_{k=0}^{5} a_k x^k = 1 - 13.7x + 67.5x^2 - 153x^3 + 162x^4 - 64.8x^5$$

$$= \sum_{k=0}^{2} b_{2k+1}\, T^*_{2k+1}(x)$$

$$= -\,(0.522\, T^*_1(x) + .352\, T^*_3(x) + .126\, T^*_5(x)),$$

met $T^*_k(x)$ het k-de verschoven Chebyshevpolynoom: $T^*_k(x) = T_k(2x-1)$, hebben de conditiegetallen

$$K(f(a_0,\ldots,a_5;\pm 1)) = 462,$$

$$K(g(b_1,b_3,b_5;1)) = 1.$$

OPMERKINGEN.

1. In Voorbeeld 9.1.1 kunnen we een transformatie van de onafhankelijke variabele toepassen:

$$\sum_{k=0}^{12'} b_{2k}\, T_{2k}(x/8) = \sum_{k=0}^{12'} b_{2k}\, T_k(y(x)), \qquad y = x^2/32 - 1.$$

BEASLY [1965] heeft gewezen op de mogelijk gunstige eigenschappen van de machtsvorm in $T_2(x/8)$. Hij geeft de benadering van $J_0(x)$ in de representaties

$$\sum_{k=0}^{12'} b_{2k}\, T_{2k}(x/8) = \sum_{k=0}^{12} c_{2k}\,(x/8)^{2k} = \sum_{k=0}^{12} a_k\,(T_2(x/8))^k .$$

De bijbehorende conditiegetallen zijn:

$$K(g(b_0,\ldots,b_{24};\pm 8)) = 1,$$

$$K(f(c_0,\ldots,c_{24};\pm 8)) = 427,$$

$$K(f(a_0,\ldots,a_{12};\pm 8)) = 4.4.$$

Deze transformaties kan men verkrijgen via TRFORM (ACCULIB).

2. GAUTSCHI [1972] hanteert het conditiebegrip

$$\mathrm{cond}_\infty M_n = \|M_n\|_\infty \cdot \|M_n^{-1}\|_\infty ,$$

waarin

$$M_n : \mathbb{R}^n \longrightarrow P_{n-1}$$

$$(u_0,u_1,\ldots,u_{n-1}) \longmapsto \sum_{k=0}^{n-1} u_k\, f_k(x) \quad \text{(som van orthogonale polynomen)}.$$

Binnen de klasse van orthogonale polynomen heeft hij een maat om aan te geven naar welke polynomen we het best kunnen ontwikkelen wat betreft de conditie van de resulterende polynoomrepresentatie. Hij geeft:

$$\mathrm{cond}_\infty M_n \le \max_{0\le k\le n-1} \left(\frac{\mu_0}{h_k}\right)^{\frac{1}{2}} \max_{a\le x\le b} \sum_{k=0}^{n-1} |f_k(x)|,$$

met als voorbeelden:

Chebyshev $\quad f_k(x) = T_k(x)$: $\mathrm{cond}_\infty M_n \le 2^{\frac{1}{2}} n$

Legendre $\quad f_k(x) = P_k(x)$: $\mathrm{cond}_\infty M_n \le n\,(2n-1)^{\frac{1}{2}}$.

3. In HART c.s. [1968, p.66] wordt ook onderscheid gemaakt naar de conditie

van de probleemformulering ("conditioning") en hoe in een rekenkundig proces de afrondfouten zich voortplanten ("stability"). Het voorbeeld

$$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1$$

is slechter geconditioneerd in de expliciete vorm; de conditiegetallen zijn:

$$K(f(-1,...,32;1)) = 99,$$

$$K(g(1;1)) = 1.$$

4. GAUTSCHI [1973] geeft de conditie van de transformatie van een lineaire representatie naar de productvorm.

5. REIMER [1977] stelt: "*Within quite a general family of evaluation schemes for polynomials, Clenshaw's algorithm based on the Chebyshev polynomials of the second kind is of almost maximum numerical stability*".

6. NEWBERY [1975] stelt een transformatie van de onafhankelijke variabele voor, zodat de getransformeerde onafhankelijke variabele gedefinieerd is op een kleiner interval. Ook hiermee is nog geen universele goed-geconditioneerde polynoomrepresentatie gevonden.

7. Het voorbeeld gegeven door HAYES [1970, p.57, table 2] is misleidend, omdat de "ruistermen": $a_{11}\ T_{11}$, $a_{12}\ T_{12}$, negatieve coëfficiënten hebben; de equivalente machtssom is daardoor niet meer tekenvast, hetgeen op theoretische gronden onjuist is en geen goede vergelijking kan bieden, omdat het verschillende polynomen betreft.

9.1.2. Evaluatie van polynoomrepresentaties

In deze paragraaf geven we een overzicht van algoritmen voor de evaluatie van polynomen. De beschikbare implementaties zijn samengevat in Tabel 9.1.1.

9.1.2.1. Machtssom

Voor de evaluatie van de machtssom kennen we de "splitting"-algoritmen (TRAUB c.s. [1974]):

$$(9.1.9)\qquad \sum_{k=0}^{n} a_k x^k = (a_0+a_1x+\ldots+a_qx^q) + \\ + x^{q+1}((a_{q+1}+\ldots+a_{2q+1}x^q)+\ldots+x^{q+1}(a_{n-q}+\ldots+a_nx^q))\ldots),$$

met q+1 een deler van n+1.

Als speciaal geval, q = 0, hebben we het Hornerschema. De "splitting"-algoritmen zijn efficiënt als men behalve de polynoomwaarden ook de waarden van afgeleiden wil hebben.

Programmatuur, gebaseerd op het artikel van TRAUB c.s. [1974] is opgenomen in NUMAL (TAYPOL, DERPOL, NORDERPOL).

OPMERKINGEN.

1. Voor de evaluatie van een polynoom op meerdere punten verwijzen we naar KUNG [1973], of AHO c.s. [1974]; de algoritmen komen neer op generalisatie van de Hornerregel, geïnterpreteerd als polynoomdeling.

   VOORBEELD (Polynoomevaluatie voor meerdere punten; MOENCK c.s. [1972]).
   Zij $p(x) = x^3 - 3x + 5$ te evalueren voor x = -1, 1, 2, 3, dan geldt:

$$\begin{aligned} p(x) &= p_1^{(1)}(x) * (x^2-1) + p_0^{(2)}(x+1) + r_{-1}\ , \\ p(x) &= p_1^{(1)}(x) * (x^2-1) + p_0^{(2)}(x-1) + r_1\ , \\ p(x) &= p_1^{(2)}(x) * (x^2-5x+6) + p_0^{(2)}(x-2) + r_2\ , \\ p(x) &= p_1^{(2)}(x) * (x^2-5x+6) + p_0^{(2)}(x-3) + r_3\ , \end{aligned}$$

   met $r_k = p(k)$, k = -1,1,2,3. Hierbij moeten we opmerken dat we de factoren van de polynomen, waarmee we gaan delen, eerst moeten vormen en vervolgens de polynoomdeling uitvoeren. De productvorming kan efficiënt via FFT en staat bekend als "fast polynomial multiplication"; de deling kan ook efficiënt via FFT en staat bekend als "preconditioning".

2. Een foutenanalyse van de "splitting"-algoritmen is gegeven door WOZNIAKOWSKI [1974].

9.1.2.2. Chebyshevsom

Voor de evaluatie van de Chebyshevsom wordt algemeen de Clenshaw-

algoritme gebruikt. Deze algoritme maakt gebruik van de recurrente betrekking voor de Chebyshevpolynomen:

$$T_0 = 1, \quad T_1 = x, \quad T_{k+1} = 2x\,T_k - T_{k-1}, \qquad k = 1,2,\ldots \ .$$

De Clenshaw-algoritme is te verkrijgen als een gegeneraliseerde Hornerregel uit

$$\sum_{k=0}^{n} a_k\,T_k(x) = (1,x)\sum_{k=0}^{n}\begin{pmatrix}2x & 1\\ -1 & 0\end{pmatrix}^k\begin{pmatrix}a_k\\ 0\end{pmatrix}$$

(9.1.10)

$$= (1,x)\left[\begin{pmatrix}a_0\\ 0\end{pmatrix}+\begin{pmatrix}2x & 1\\ -1 & 0\end{pmatrix}\left[\begin{pmatrix}a_1\\ 0\end{pmatrix}+\ldots+\begin{pmatrix}2x & 1\\ -1 & 0\end{pmatrix}\begin{pmatrix}a_n\\ 0\end{pmatrix}\right]\ldots\right].$$

Programmatuur voor de evaluatie van een Chebyshevsom is opgenomen in NUMAL (CHEPOLSER). CHEPOLSER kan men efficiënt gebruiken voor de bijzondere Chebyshevsom

$$(9.1.11)\qquad \sum_{k=0}^{n} b_k\,T_{2k}(x) = \sum_{k=0}^{n} b_k\,T_k(y)\ ,$$

met de transformatie van de onafhankelijke variabele: $y = T_2(x)$. Voor de oneven-Chebyshevsom is er echter een aparte implementatie nodig; de algoritme van Clenshaw [1962] kan verkregen worden als gegeneraliseerde Hornerregel uit de representatie

$$(9.1.12)\qquad \sum_{k=0}^{n} c_k\,T_{2k+1}(x) = x(1,-1)\sum_{k=0}^{n}\begin{pmatrix}2T_2(x) & -1\\ 1 & 0\end{pmatrix}^k\begin{pmatrix}c_k\\ 0\end{pmatrix}.$$

De implementatie ODDCHEPOLSER is opgenomen in bijlage 3.

In sommige toepassingen kan men x "vrij" kiezen; als men deze vrijheid benut door ϕ, met $x = \cos\phi$, te kiezen dan verkrijgt men door deze transformatie van de onafhankelijke variabele

$$(9.1.13)\qquad \sum_{k=0}^{n} c_k\,T_k(x) = \sum_{k=0}^{n} c_k\cos k\phi.$$

Algoritmen en implementaties voor het rechterlid van (9.1.13) zijn gegeven in het colloquiumverslag, deel 1b, sectie 4.2; een samenvatting van de representatievormen van (9.1.13) is gegeven in bijlage 2.

OPMERKING. Als men de Chebyshevsom-representatie heeft en men wil meerdere afgeleiden berekenen, dan kan men, analoog aan de machtssom, een familie van "splitting"-algoritmen construeren.

9.1.2.3. Orthogonale som

De evaluatie van een som van orthogonale polynomen wordt veelal gedaan via een generalisatie van de Clenshaw-algoritme.

LEMMA 9.1.1 (GAUTSCHI [1975, p.38]; DEUFLHARD [1976]; LUKE [1969, vol.I,8.6]; *zij geven niet de representatie* (9.1.15)).

Zij $\{f_k(x)\}_{k=0}^{\infty}$ *een verzameling functies, gedefinieerd door*

$$f_0 \text{ gegeven,}$$

$$f_1 = (x-\beta_0)f_0, \tag{9.1.14}$$

$$f_{k+1} = (x-\beta_k)f_k - \gamma_k f_{k-1}, \qquad k = 0,1,\ldots, \quad (\gamma_0 = 0),$$

dan geldt

$$\sum_{k=0}^{n} a_k f_k(x) = (f_0,f_1) \sum_{k=0}^{n} \prod_{j=0}^{k-1} \begin{pmatrix} x-\beta_j & 1 \\ -\gamma_j & 0 \end{pmatrix} \begin{pmatrix} a_k \\ 0 \end{pmatrix}. \tag{9.1.15}$$

BEWIJS. Uit

$$f_k(x) = (f_k,f_{k-1}) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = (f_1,f_0) \prod_{j=1}^{k-1} \begin{pmatrix} x-\beta_j & 1 \\ -\gamma_j & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$= (f_0,f_1) \prod_{j=0}^{k-1} \begin{pmatrix} x-\beta_j & 1 \\ -\gamma_j & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

volgt

$$\sum_{k=0}^{n} a_k f_k(x) = (f_0,f_1) \sum_{k=0}^{n} \prod_{j=0}^{k-1} \begin{pmatrix} x-\beta_j & 1 \\ -\gamma_j & 0 \end{pmatrix} \begin{pmatrix} a_k \\ 0 \end{pmatrix}.$$

Implementaties gebaseeerd op de algoritme gegeven in Lemma 9.1.1 zijn ORTPOLSER (NUMAL), EVAPOL (ACCULIB;A60,FIV). RLOPDC(IMSL) rekent het inproduct uit; d.w.z. de orthogonale polynomen worden via de drie-terms homogene recursierelatie uitgerekend. Bovendien heet de evaluatie daar:

"Predict response using orthogonal polynomials". De Clenshaw-algoritme is wezenlijk efficiënter; de stabiliteit wordt in beide gevallen beheerst door de homogene recursie. In het algemeen is de bovengrens van de absolute evaluatiefout evenredig met n^2; voor een dalende reeks $a_k \sim 1/k^2$, is de bovengrens evenredig met n (ELLIOT [1968]; DEUFLHARD [1976]).

OPMERKINGEN.

1. In geen enkele programmatheek worden de bijzondere orthogonale polynomen

$$f_{k+1} = x\, f_k - c_k\, f_{k-1}, \qquad k = 1,2,\ldots$$

$$f_0 = 1, \quad f_1 = x$$

apart genoemd.

2. Als testvoorbeeld voor de implementaties met betrekking tot evaluatie van een som van orthogonale polynomen, gebruiken we de relatie

$$x^n = \sum_{k=0}^{n} c_k^n\, f_k(x),$$

met

$$f_{k+1}(x) = (x-\beta_k) f_k - \gamma_k\, f_{k-1}, \qquad k = 1,2,\ldots$$

$$f_0 = 1, \quad f_1 = x - \beta_0 ,$$

en

$$c_n^n = 1, \quad c_{n-1}^n = \beta_{n-1} + c_{n-2}^{n-1}, \qquad n \geq 2$$

$$c_{k-1}^{n+1} = c_{k-2}^n + \beta_{k-1}\, c_{k-1}^n + \gamma_k\, c_k^n, \qquad k = 2,3,\ldots,n$$

$$c_0^n = \langle x^n, 1\rangle / \langle 1,1\rangle;$$

$\{\beta_k = 0\}$ is een speciaal geval.

(Voor de transformatie van de machtssom naar de orthogonale som hebben HAMMING [1962] en SALZER [1973] gewezen op het onnodig vormen van $\{c_k^n\}$; in paragraaf 9.1.3.1 gaan we er nader op in.)

3. De implementatie EVAGEN (ACCULIB;A60) evalueert

$$\phi(x) = \sum_{k=0}^{n} a_k\, f_k(\psi(x));$$

dit kan ook eenvoudig met een aanroep van EVAPOL:

```
EVAGEN(n,b,c,a,x,fie,psi) = fie(x) * EVAPOL(n,b,c,a,psi(x),der).
```

9.1.2.4. Newtonvorm

Programmatuur voor de evaluatie van polynomen in de Newtonvorm

$$P_n(x) = \sum_{k=0}^{n} c_k \prod_{j=0}^{k-1} (x-x_j)$$

is: NEWTON (NUMAL) en NIP (ACCULIB;A60). De algoritme is de Hornerachtige regel

$$P_n(x) = c_0 + (x-x_0)[c_1 + (x-x_1)[c_2 + \ldots + (x-x_{n-1})c_n] \ldots].$$

9.1.2.5. Samenvatting implementaties

Tabel 9.1.1.

Implementaties voor evaluatie van polynoomrepresentaties

Representatie	Implementatienaam (Programmatheek; taal)
Machtssom	POL (NUMAL)
Chebyshevsom	CHEPOLSER (NUMAL)
	ODDCHEPOLSER (zie bijlage 3)
	E02AEA/F (NAG)
Som van orthogonale polynomen	EVAPOL (ACCULIB;A60,FIV)
	EVAGEN (ACCULIB;A60)
	ORTPOLSER (NUMAL)
	RLOPDC (IMSL)
Newtonpolynoom	NEWTON (NUMAL)
	NIP (ACCULIB;A60)

9.1.3. Transformatie van polynoomrepresentaties

In deze paragraaf beschouwen we algoritmen en beschikbare implementaties voor de transformatie van representatie van een polynoom. De transformatie in 9.1.3.1 *wordt gebruikt voor de bepaling van de (benaderde) coëfficiënten van een Chebyshevreeks als de coëfficiënten van de machtreeks voorhanden zijn. De transformatie in* 9.1.3.2 *wordt inwendig gebruikt bij de bepaling van de minimaxbenadering.*

9.1.3.1. Machtssom naar Chebyshevsom

Zij

$$\sum_{k=0}^{n} a_k x^k = \sum_{k=0}^{n} b_k T_k(x),$$

dan kan het verband tussen de coëfficiënten $\{a_k\}$ en $\{b_k\}$ worden gegeven door

$$b = \left[\prod_{k=0}^{n-2} (I_k \oplus M_{n+1-k})\right] D_{n+1}\, a, \tag{9.1.16}$$

met:

I_k de k-de eenheidsmatrix;

M_k een k*k - bovendriehoeksmatrix met
$(M_k)_{ii} = 1$, $i = 1,2,3,\ldots,k$,
$(M_k)_{22} = 2$,
$(M_k)_{i,i+2} = 1$, $i = 1,2,\ldots,k-2$,
en overige elementen 0;

$\oplus$ de operatie: directe som met twee matrices als operanden:

$$A \oplus B = \begin{pmatrix} A & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & B \end{pmatrix},$$

D_{n+1} een diagonaalmatrix met diagonaalelementen
$(1, 2^{-1}, 2^{-2}, \ldots, 2^{-n+2}, 2^{-n+1}, 2^{-n+1})$.

De formule (9.1.16) is geïnspireerd op HAMMING [1962, p.225-257]; essentieel is dat de "tussenresultaten" van een Hornerschema worden onderkend als een polynoom dat wordt bijgehouden in de Chebyshevsomrepresentatie. Het "updaten" is gemakkelijk, omdat

$$2x\left(\sum_{k=0}^{\ell} c_k^{(\ell)}\ T_k(x)\right) = \sum_{k=0}^{\ell+1} c_k^{(\ell+1)}\ T_k(x),$$

met

$$\begin{aligned}
c_0^{(\ell+1)} &= c_1^{(\ell)} \quad , \\
c_1^{(\ell+1)} &= 2c_0^{(\ell)} + c_2^{(\ell)} \quad , \\
c_2^{(\ell+1)} &= c_1^{(\ell)} + c_3^{(\ell)} \quad , \\
&\vdots \\
c_{\ell-1}^{(\ell+1)} &= c_{\ell-2}^{(\ell)} + c_{\ell}^{(\ell)}, \quad c_{\ell}^{(\ell+1)} = c_{\ell-1}^{(\ell)}, \quad c_{\ell+1}^{(\ell+1)} = c_{\ell}^{(\ell)}.
\end{aligned}$$

ILLUSTRATIE 9.1.1 (Machtssom naar Chebyshevsom).

Zij het polynoom $a_0 + a_1x + a_2x^2$ gegeven; de coëfficiënten van de Chebyshevsom worden verkregen door toepassing van (9.1.16)

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = (I_0 \oplus M_3)\, D_3 \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 1 \\ & 2 & 0 \\ & & 1 \end{pmatrix} \begin{pmatrix} 1 & & \\ & \frac{1}{2} & \\ & & \frac{1}{2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} a_0 + a_2/2 \\ a_1 \\ a_2/2 \end{pmatrix};$$

ofwel

$$a_0 + x(a_1+a_2x) = 2x(\tfrac{1}{2}a_1+\tfrac{1}{2}a_2x) =$$

$$a_0 + 2x(\tfrac{1}{2}a_1T_0+\tfrac{1}{2}a_2T_1) = (a_0+\tfrac{1}{2}a_2)T_0 + a_1T_1 + \tfrac{1}{2}a_2T_2.$$

Uit formule (9.1.16) volgt de inverse-transformatie door voorvermenigvuldiging met de inverses van de $(I_k \oplus M_{n+1-k})$-matrices en de inverse van de diagonaalmatrix. De bekende algoritme, die gebruik maakt van de ontwikkeling van de machten van x in een Chebyshevsom, kan opgevat worden als een manier van uitwerking van formule (9.1.16), namelijk: eerst de matrices met elkaar vermenigvuldigen en dan de matrix-maal-vectoroperatie; THACHER [1964] en FRASER [1965] geven een formule voor het product van de matrices (zie ook het bewijs in bijlage 1).

Het nut van deze transformatie is, dat wij uitgaande van een machtssom de veelal beter gekonditioneerde Chebyshevsom kunnen verkrijgen. Als we bovendien nog enige termen weglaten, dan hebben we een korter polynoom verkregen ten koste van een wat grotere afbreekfout; wij hebben het polynoom wat ingekort. Als we de ingekorte vorm representeren in de machtssom dan spreekt men van economiseren of telescoperen van de oorspronkelijke machtssom.

OPMERKINGEN.

1. In bijlage 4 is een ALGOL 60 procedure, POLCHS, opgenomen die de transformatie: coëfficiënten van machtssom naar coëfficiënten van Chebyshevsom uitvoert; de transformatie naar coëfficiënten van andere orthogonale sommen (bijvoorbeeld "verschoven" Chebyshevsom) kan op analoge wijze geschieden.

2. SALZER [1973] heeft de algoritme van Hamming gegeneraliseerd voor de transformatie tussen orthogonale sommen.

3. In ACCULIB zijn de routines TRFORM (ALGOL 60) en TRFORD (ALGOL 60 en FORTRAN IV) beschikbaar, die een orthogonale som naar de machtssom transformeren; alhoewel de resulterende machtssom efficiënt te evalueren is verdient het aanbeveling de conditiefuncties van beide representaties te vergelijken.

4. Als test voor de transformatie (9.1.16) kan men gebruiken:

$$\sum_{k=0}^{n} a_k = \sum_{k=0}^{n} b_k,$$

$$\sum_{k=0}^{n} a_k (-1)^k = \sum_{k=0}^{n} b_k (-1)^k.$$

5. Economiseren of telescoperen wordt veelal "direct" gedaan (DEKKER [1967, p.227-278]); het volgende programmafragment expliceert dit.

```
...
co T_k(x) = Σ c_kl x^l co;
for k from n by -1 to 0 while real h = a[k]/2↑(k-1); h < eps
do for l from k-2 by -2 to 0
   do a[l]-:= h*c[k,l] od
od;
...
```

6. THACHER [1964] heeft de transformatie van machtreeks naar Chebyshev-reeks beschouwd.

7. In de NAG-programmatheek bepaalt men de benaderende polynomen in de Chebyshevsomrepresentatie; deze is goed geconditioneerd. Als de machtssom en Chebyshevsom gelijk conditiegetal hebben, dan levert men het uiteindelijk polynoom in de machtssom af via een transformatie van Chebyshevsom naar machtssom.

8. Als een machtssom slecht geconditioneerd is en men transformeert naar de Chebyshevsom, dan verkrijgt men weliswaar een beter geconditioneerde representatie, maar de slechte conditie van de machtssom als "tussenresultaat" laat zich gelden in onnodig onnauwkeurige coëfficiënten van de Chebyshevsom; men moet dan de machtssom als tussenresultaat vermijden.

9.1.3.2. Newtonvorm naar machtssom

Zij

$$\sum_{k=0}^{n} c_k \prod_{\ell=0}^{k-1} (x-x_\ell) = \sum_{k=0}^{n} a_k x^k ,$$

dan kan het verband tussen de coëfficiënten $\{c_k\}$ en $\{a_k\}$ worden gegeven door

$$a = \left[\prod_{k=0}^{n-1} (I_k \oplus B_{n+1-k})\right] c \qquad (9.1.17)$$

met

B_k een $k * k$ - bovenbidiagonaalmatrix met 1 op de diagonaal en $-x_{n+1-k}$ op de (boven)nevendiagonaal,

I_k de k-de eenheidsmatrix,

$\oplus$ de operatie: directe som met twee matrices als operanden:

$$A \oplus B = \begin{pmatrix} A & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & B \end{pmatrix} .$$

De transformatie (9.1.17) is veelal geïmplementeerd via vorming van de productmatrix, gevolgd door de matrix-maal-vectoroperatie; efficiënter is echter herhaald de matrix-maal-vectoroperatie uit te voeren en niet uit te

gaan van de productmatrix. De implementaties NEWGRN (NUMAL) en NEWHOR (ACCULIB;A60,FIV) doen dit laatste via de algoritme:

representeer $\prod_{\ell=0}^{k-1} (x-x_\ell)$ als machtssom;

vorm de lineaire combinatie van de machtssommen.

Inverse-algoritmen (machtssom $\longrightarrow$ Newtonvorm) kunnen verkregen worden door voorvermenigvuldiging van (9.1.17) met inverses van de factormatrices; de N_k^{-1} matrices hebben machten van x_{n+1-k} op de bovennevendiagonalen (zie b.v. NEWBERY [1974]).

OPMERKING. De N_k^{-1}-maal-vectoroperatie kan efficiënt via de FFT (AHO c.s. [1974, p.256, theorem 7.2], KUNG [1973, theorem 2.2]); een open vraag is echter of dit lonend is bij kleine matrices.

ILLUSTRATIE 9.1.2 (Newtonvorm naar machtssom).
Zij het polynoom $c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1)$ gegeven; de coëfficiënten van de machtssom worden verkregen door toepassing van (9.1.17)

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 & -x_0 & 0 \\ & 1 & -x_0 \\ & & 1 \end{pmatrix} \left(\begin{array}{ccc} 1 & \vdots & \\ \cdots & \cdots & \cdots \\ & \vdots\ 1 & -x_1 \\ & \vdots & 1 \end{array}\right) \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} =$$

$$= \begin{pmatrix} c_0 - x_0c_1 + x_0x_1c_2 \\ c_1 - (x_1+x_0)c_2 \\ c_2 \end{pmatrix} ;$$

ofwel

$$c_0 + (x-x_0)(c_1 + (x-x_1)c_2) =$$

$$= c_0 - x_0c_1 + x_0x_1c_2 + (c_1 - (x_1+x_0)c_2)x + c_2x^2.$$

9.1.3.3. Samenvatting van implementaties

Tabel 9.1.2.

Implementaties voor de transformatie van polynoomrepresentaties

Transformatie	Implementatie (Programmatheek; taal)
machtssom → Chebyshevsom	POLCHS (zie bijlage 4)
orthogonale som → machtssom	TRFORM (ACCULIB;A60) TRFORD (ACCULIB;A60,FIV) RLDOPM (IMSL)
Newtonvorm → machtssom	NEWGRN (NUMAL) NEWHOR (ACCULIB;A60,FIV)

OPMERKING. De (verbeterde) implementatie van NEWGRN (NUMAL) in de geest van Hamming is:

```
"CODE" 31050;
"PROCEDURE" NEWGRN(N,X,C);
"VALUE" N; "INTEGER" N; "ARRAY" X,C;
"BEGIN"
   "INTEGER" K;
   "PROCEDURE" ELMVEC(L,U,SHIFT,A,B,X); "CODE" 34020;
   "FOR" K:=N-1 "STEP" -1 "UNTIL" 0 "DO"
   ELMVEC(K,N-1,1,C,C,-X[K])
"END" NEWGRN;
```

Ter illustratie hebben we ook de oude versie opgenomen.

```
"CODE" 31050;
"PROCEDURE" NEWGRN(N,X,C);
"VALUE" N; "INTEGER" N; "ARRAY" X,C;
"BEGIN" "INTEGER" J,K,KM1; "REAL" XKM1,XXJ,XXJM1;
   "ARRAY" XX[0:N];
   "PROCEDURE" ELMVEC(L,U,SHIFT,A,B,X); "CODE" 34020;
   XX[0]:=1; KM1:=0;
   "FOR" K:=1 "STEP" 1 "UNTIL" N "DO"
   "BEGIN" XX[K]:=1; XKM1:=X[KM1];
      XXJ:=XX[KM1];
      "FOR" J:=KM1 "STEP" -1 "UNTIL" 1 "DO"
      "BEGIN" XXJM1:=XX[J-1];
         XX[J]:=XXJM1 - XXJ * XKM1;
         XXJ:=XXJM1
      "END";
      XX[0]:=-XX[0] * XKM1;
      ELMVEC(0,KM1,0,C,XX,C[K]);
      KM1:=K
   "END"
"END" NEWGRN;
```

9.1.4. Interpolatie met polynomen

Voor de achtergronden en de verschillen van de diverse algoritmen verwijs ik naar DEKKER [1967], ZEGELING [1976] en de daar gegeven referenties. Een overzicht van de beschikbare programmatuur staat samengevat in Tabel 9.1.3.

9.1.4.1. Samenvatting implementaties

Tabel 9.1.3.

Implementaties van interpolatie-algoritmen

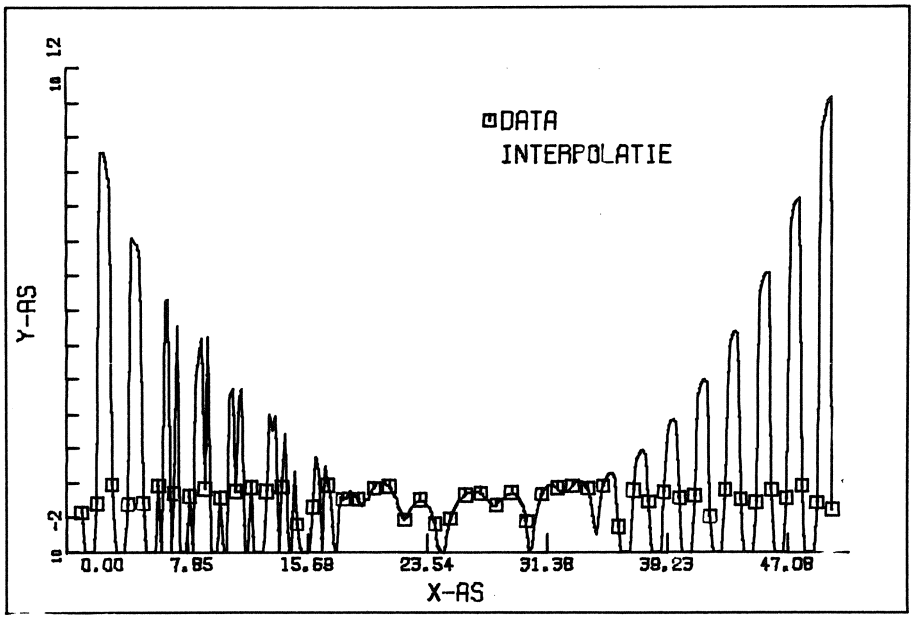
Interpolatie-algoritme	Implementatie (Programmatheek; <taal>)
Newton	NEWTON (NUMAL)
	NIP (ACCULIB;A60,FIV)
Aitken	E01AAA/F (NAG)
Everett	E01ABA/F (NAG)

OPMERKINGEN.

1. De Lagrange-interpolatie was "handig" bij interpolatie in tabellen; met de opkomst van de zakrekenmachines kan deze techniek weer gaan floreren. Voor enige voorbeelden zie ABRAMOWITZ c.s. [1964], DEKKER [1967] of SMITH [1975].

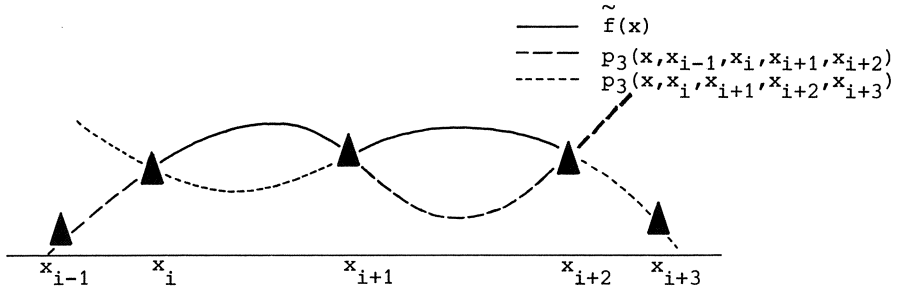
2. Bij equidistante interpolatie heeft het interpolerend polynoom de neiging bij de randpunten te gaan slingeren met een grote amplitude. BULIRSCH & RUTISHAUER (in: SAUER c.s. [1968]) geven een afschrikwekkend voorbeeld van equidistante interpolatie van de ruisfunctie: $\{x_i\}_{i=0}^{49} = \{i\}_{i=0}^{49}$, $\{f_i\}_{i=0}^{49}$ met $f_i < 1$, voor alle i. (De 5*i tot aan 5*(i+1) decimalen van ¶ vormen de cijfers achter de komma; dus $f_0 = .14159$, $f_1 = .26535$ etc.) Het resulterende pathologische gedrag van het interpolatiepolynoom is weergegeven in Illustratie 9.1.3. Met behulp van deze illustratie kunnen we aanvoelen waarom er bij interpolatie evenveel steunpunten links als rechts van het interpolatiepunt moeten liggen. Bovendien is de waarde van het polynoom voor x buiten het interval in absolute waarde groot; dit is in overeenstemming met de vuistregel: in het algemeen is interpolatie nauwkeuriger dan extrapolatie.

ILLUSTRATIE 9.1.3 (Opslingerend gedrag van de logaritme van het naar onderen afgeknotte interpolatiepolynoom).



3. Stuksgewijze interpolatie houdt de graad van het interpolerende polynoom laag, maar geeft voor elk interval de coëfficiënten van een polynoom; in het volgende voorbeeld is geïnterpoleerd met derdegraads polynomen.

ILLUSTRATIE 9.1.4 (Geregen interpolatie met derdegraads polynomen).



Wij hebben als benaderende functie:

$$\tilde{f}(x) = \begin{cases} p(x;x_0,x_1,x_2,x_3) & \text{voor } x_1 \leq x \leq x_2 \\ \ldots & \\ p(x;x_{i-1},x_i,x_{i+1},x_{i+2}) & \text{voor } x_i \leq x \leq x_{i+1} \\ \ldots & \\ p(x;x_{n-3},x_{n-2},x_{n-1},x_n) & \text{voor } x_{n-2} \leq x \leq x_{n-1}. \end{cases}$$

We zien dat de benaderende functie op de steunpunten i.h.a. knikken vertoont; in hoofdstuk 9.2 introduceren we geregen derdegraads polynomen die ook continu zijn in de eerste afgeleide (Quasihermite-interpolatie; Akima) en in het algemeen splinefuncties, die continu zijn tot in de (graad-1)-ste afgeleide.

4. De resultaten van KUNG [1973] met betrekking tot "fast interpolation" zijn voor de data-aanpassing niet interessant, omdat we de graad van het interpolerend polynoom laag houden vanwege gladheidseisen.

5. De interpolatie op [-1,1] met steunpunten $\{\cos k\pi/n\}_{k=0}^{n-1}$ heet Chebyshevinterpolatie.

9.1.5. Kleinste-kwadratenbenadering met polynomen

In paragraaf 9.1.5.1 *geven wij een algemene probleemformulering. In paragraaf* 9.1.5.2 *gaan we in op de bepaling van de ontwikkeling van een functie naar een orthogonale basis; het algemene discrete kleinste-kwadratenprobleem wordt besproken in paragraaf* 9.1.5.3. *In paragraaf* 9.1.5.4 *gaan we in op de conditie van de discrete probleemformulering. In Tabel* 9.1.4 *geven wij een overzicht van de implementaties.*

9.1.5.1. Probleemformulering

Deze approximatievorm kan men zien als de ontwikkeling van een functie, f, naar een (onafhankelijke) klasse van (eenvoudiger) functies, bijvoorbeeld: (orthogonale) polynomen. Zij $\{\phi_k\}$ een basis van een Hilbertruimte, dan luidt de opgave:

$$(9.1.18) \qquad \min_{\{c_k\}} \| f - \sum_{k=0}^{n} c_k \phi_k \|_2 .$$

De noodzakelijke en voldoende voorwaarde voor een minimum leidt tot de normaalvergelijkingen

(9.1.19) $Ac = b$,

met de zogenaamde Gramm-matrix

(9.1.20) $A_{ij} = \langle \phi_i, \phi_j \rangle$

en momenten

(9.1.21) $b_i = \langle f, \phi_i \rangle$.

9.1.5.2. Orthogonale basis

Bij een orthogonale basis luidt de oplossing van (9.1.18)

(9.1.22) $c_k = \langle f, \phi_k \rangle / \langle \phi_k, \phi_k \rangle$.

Als de functie discreet gegeven is, dan kunnen we

- óf een continue $\tilde{f}$ construeren uit $\{f_k\}$,
- óf, uitgaande van een discreet inproduct, polynomen construeren, orthogonaal over de puntverzameling $\{x_k\}$.

Bij de eerste aanpak kan men globale informatie omtrent de (achterliggende) f benutten; men kan bijvoorbeeld een spline-benadering $\tilde{f}$ uit $\{f_k\}$ construeren. Hierbij kan men zoeken naar methodes, analoog aan de techniek van de verzwakkingsfactoren (zie colloquiumverslag deel 1b, sectie 4.2), zodat alle coëfficiënten simultaan efficiënt bepaald kunnen worden.

Bij de tweede aanpak moeten we eerst de discrete orthogonale polynomen construeren en vervolgens de inproducten (9.1.22) evalueren; deze methode staat bekend als de Forsythe-methode (FORSYTHE [1957]; zie b.v. SHAMPINE [1975]): zij

(9.1.23)
$$\phi_0 = 1, \quad \phi_1 = x - \beta_0,$$
$$\phi_{k+1} = (x-\beta_k)\phi_k - \gamma_k \phi_{k-1}, \qquad k = 1,2,\ldots,$$

dan geldt, met $\{\phi_k\}$ orthogonaal, voor de benodigde coëfficiënten

$$\beta_k = \langle x\phi_k,\phi_k\rangle / ((\prod_{j=1}^{k} \gamma_j)\ \langle 1,1\rangle), \qquad k = 0,1,\ldots$$

$$\gamma_k = \langle \phi_k,\phi_k\rangle / ((\prod_{j=1}^{k-1} \gamma_j)\ \langle 1,1\rangle), \qquad k = 1,2,\ldots \tag{9.1.24}$$

$$c_k = \langle f - \sum_{\ell=0}^{k-1} c_\ell\ \phi_\ell,\phi_k\rangle / ((\prod_{j=1}^{k} \gamma_j)\ \langle 1,1\rangle), \qquad k = 0,1,\ldots,$$

met als discreet inproduct

$$\langle f,g\rangle = \sum_{i=1}^{m} w_i^2\ f(x_i)g(x_i).$$

De (ongewijzigde) Forsythe-methode is geïmplementeerd als:

RLFOTH (IMSL); RLFOTW (IMSL); E02AFA/F (NAG);
E02ADA/F (NAG); GENFIT (ACCU;A60); GENFIW (ACCU;A60);
POLFIW (ACCU;FIV).

OPMERKINGEN.

1. SHAMPINE [1975] stelt dat numeriek-betere resultaten worden verkregen door evaluatie van $\{c_k\}$ via (9.1.24) in plaats van

$$c_k = \sum_{j=1}^{m} w_j^2\ f(x_j)\phi_k(x_j) / ((\prod_{i=1}^{k} \gamma_i)\ \langle 1,1\rangle);$$

   (9.1.24) kan men zien als het gemodificeerde Gramm-Schmidt proces.

2. HAYES [1970, p.49-50] heeft erop gewezen dat we nog randcondities kunnen opleggen aan de benaderende functie bij de Forsythe-algoritme.

3. In NAG worden de (discrete) orthogonale polynomen, die via de Forsythe-algoritme verkregen zijn, afgeleverd in de Chebyshevsomrepresentatie. Hiermede is een parameterreductie uitgevoerd. CADWELL c.s. [1961] leveren de machtssom af; dit is te verkiezen als de condities van de machtssom en Chebyshevsom gelijk zijn.

4. THACHER [1966] en SCRATON [1970] stellen transformaties van de onafhankelijke variabele voor, zodat de resulterende Chebyshevreeks sneller convergeert. In de NAG programmatheek gebruikt men deze techniek.

9.1.5.3. Niet-orthogonale basis

Bij een discreet gegeven $\{f_k\}$ luidt opgave (9.1.18) in matrixnotatie

$$(9.1.25)\qquad \Phi c \simeq f \Longleftrightarrow \min_{\{c_k\}} \left\| \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix} - \begin{pmatrix} \phi_0(x_1) & \cdots & \phi_n(x_1) \\ \vdots & & \vdots \\ \phi_0(x_m) & \cdots & \phi_n(x_m) \end{pmatrix} \begin{pmatrix} c_0^{(m)} \\ \vdots \\ c_n^{(m)} \end{pmatrix} \right\|_2 .$$

LAWSON c.s. [1974] gaan van (9.1.25) uit. De oplossing met minimale lengte kan men representeren als

$$c = \Phi^+ f$$

met Φ^+ de pseudo-inverse van Φ. De pseudo-inverse maal f kan men verkrijgen uit (PETERS c.s. [1970]):

1. Singuliere-waardenontbinding (niet noodzakelijk volle rang). Zij

   $$\Phi = U\Sigma V^T,$$

   dan geldt

   $$\Phi^+ = V\Sigma^+ U^T, \quad \Phi^+ f = V\Sigma^+ U^T f,$$

   met

   $$\Sigma_{ii}^+ = \begin{cases} 1/\Sigma_{ii}, & \Sigma_{ii} \neq 0 \\ 0, & \Sigma_{ii} = 0. \end{cases}$$

2. QR-ontbinding (volle rang). Zij

   $$\Phi = QR,$$

   dan geldt

   $$\Phi^+ f = R\backslash Q^T f.$$

Programmatuur voor de QR- en singuliere-waardenontbinding is gegeven in sectie 1.1 van het colloquiumverslag 1a (ACCULIB bevat bovendien CSVD voor een complexe matrix).

OPMERKINGEN.

1. Voor een grote en ijle matrix zie GENTLEMAN [1972] en/of PAIGE c.s. [1973].

2. Bij toevoeging of weglating van rijen van de matrix kan men de factorisatie "updaten". Voor de QR-ontbinding zie LAWSON c.s. [1974, ch.27]; voor de SVD zie BUNCH c.s. [te verschijnen].

3. Voor lineaire (on-)gelijkheidsvoorwaarden zie LAWSON c.s. [1974]. Voor kwadratische ongelijkheidsvoorwaarden zie GOLUB c.s. [1969].

4. Een zinnige niet-minimale lengteoplossing van het kk-probleem kan men verkrijgen door een regularisatieterm toe te voegen, d.w.z. vind:

$$\min_{\{c_k\}} \left\{ \| f(x_i) - \sum_{k=0}^{n} c_k \, \phi_k(x_i) \|_2 + \lambda \| c_k \|_2 \right\}.$$

De matrixformulering luidt:

$$\begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_n(x_1) \\ \vdots & \vdots & & \vdots \\ \phi_0(x_m) & \phi_1(x_m) & \dots & \phi_n(x_m) \\ \lambda & & & 0 \\ & \ddots & & \\ 0 & & & \lambda \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} \simeq \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

waarin men de "λ-knop" kan variëren. Voor generalisaties zie LAWSON c.s. [1974, par.25.4]: ridge regression, damped least squares. De oplossing luidt (RUTISHAUSER [1968b]), $c = VD'U^T f$, met $\Phi = UDV^T$, $D'_{ii} = D_{ii}/(D_{ii}^2+\lambda^2)$.

9.1.5.4. Conditie van het discrete kleinste-kwadratenprobleem

Inzicht in de conditie van het discrete kleinste-kwadratenprobleem

$$\Phi c \simeq f$$

wordt gegeven door de bereikbare ongelijkheid

$$\|c-\tilde{c}\| \leq \varepsilon\{\kappa^2(\Phi)\|r\|_2 + \kappa(\Phi)(\|c\|_2+1)\} + O(\varepsilon^2), \tag{9.1.26}$$

met

$\kappa(\Phi)$ het conditiegetal:
grootste singuliere waarde gedeeld door kleinste singuliere waarde,

r $= \Phi c - f$,

εE verstoring van Φ,

εe verstoring in f,

$\|\phi\|_2 = \|f\|_2 = \|E\|_2 = \|e\|_2 = 1$;

(VAN DER SLUIS [1975], LAWSON c.s. [1974, ch.9]).

GEVOLGEN.

Bij een slecht model, $\|r\|_2$ is niet "klein", heeft de probleemformulering als versterkingsfactor de conditie van de matrix in het kwadraat; voor zinnige antwoorden moeten we derhalve proberen een goed model te kiezen ($\|r\|$ klein) met de nevenconditie dat de resulterende matrix een klein conditiegetal heeft. VAN DER SLUIS [1969] stelt in het algemeen een transformatie van de onafhankelijke vector x voor, met een diagonaalmatrix D, zodat de matrix AD^{-1} kolommen van gelijke lengte heeft; het kleinste conditiegetal, dat bereikbaar is met kolomschaling, wordt dan benaderd op een factor $n^{\frac{1}{2}}$ (n = aantal kolommen). In LAWSON c.s. [1974, ch.25] wordt nader ingegaan op transformaties als men a priori informatie over de covariantiematrix heeft. GOLUB c.s. [1974] stellen een kolom- en rijschaling voor, zodat het conditiegetal van de resulterende matrix minimaal is; zij stellen echter: "*If the matrix arises from a least-squares problem it is clear that scaling on the left in effect merely changes the norm and thus converts it to a different least-squares problem*".

EFFECT VAN DE REKENKUNDIGE PROCESSEN

De grote kracht van de achterwaartse foutenanalyse is de ontkoppeling van de perturbatieaspecten en de afrondfouten. Formule (9.1.26) geeft het effect van perturbaties in Φ en f aan, zonder nog een nadere uitspraak te doen over de oorzaak van de verstoringen. Voor de practijk is, bij "goedaardige" processen (BAUER [1974]; STOER [1972]), het effect van meetfouten het grootst en die van afrondfouten ondergeschikt; men kan ook zeggen dat "goedaardige" processen juist door deze relatie gedefinieerd worden. De

geïnteresseerde lezer wordt verwezen naar LAWSON c.s. [1974] voor het effect van de "teruggeworpen" afrondfouten van de QR-ontbinding.

OPMERKINGEN.

1. Bij de ontwikkeling naar een discreet orthogonaal stelsel is de corresponderende matrix orthogonaal met conditiegetal

$$\kappa(\Phi) = \sqrt{\max_k h_k / \min_\ell h_\ell}, \quad k,\ell \in \{0,1,\ldots,n-1\},$$

waarin

$$h_k = (\prod_{i=1}^{k} \gamma_i) \ <1,1>.$$

Als n groter wordt, dan neemt r af.

2. Als we een goed-geconditioneerd kleinste-kwadratenprobleem hebben, dan kan men, als het een goed model is, de techniek van oplossing via de normaalvergelijkingen overwegen in verband met (betere; ca. factor 2) efficiëntie. Voor een vergelijking van de verschillende algoritmen zie BJÖRCK [1977]. Voor programmatuur voor oplossing van deze stelsels zie colloquiumverslag 1a Tabel 2, onder symmetrische niet-positief definiete matrix.

9.1.5.5. Samenvatting implementaties

Tabel 9.1.4.

Implementaties voor kleinste-kwadratenbenaderingen

Algoritme	Implementatie (Programmatheek; taal)
Forsythe	RLFOTH (IMSL) E02AFA/F (NAG) E02ADA/F (NAG) GENFIT (ACCU;A60)
Gewogen Forsythe	RLFOTW (IMSL) E02ADA/F (NAG) GENFIW (ACCU;A60) POLFIW (ACCU;FIV)
Singuliere-waarden ontbinding (niet noodzakelijk volle rang)	LSVALR (IMSL) F01BGA/F (NAG) F01BHA/F (NAG) SOLOVR (NUMAL) CSVD (ACCU; complexe matrix FIV)
QR-ontbinding (volle rang)	LLSQAR (IMSL) F04ANA/F (NAG) LSQORTDECSOL (NUMAL) LEASTSQ (ACCU;A60) DCMPOS,SOLVLQ (ACCU;FIV)

9.1.6. Minimaxbenadering met polynomen

Deze benaderingsvorm is van belang bij het benaderen van speciale functies; speciaal in die zin dat ze vaak voorkomen. Het is dan efficiënt deze functies te vervangen door een zo nauwkeurig mogelijke benadering, $\tilde{f}$, *die goedkoop is in evaluatie en geheugengebruik. In paragraaf* 9.1.6.1 *gaan we in op karakterisering; in Tabel* 9.1.5 *zijn de implementaties samengevat.*

9.1.6.1. Karakterisering en algoritmen

De naam, minimaxbenadering, is te begrijpen uit de eis: vind $\tilde{f}$ uit een functieklasse zo, dat

$$\min_{\tilde{f}} \|f-\tilde{f}\|_\infty \ ;$$

men noemt $\|f\|_\infty = \sup_{x\in X} |f(x)|$, de Chebyshevnorm, de uniforme norm of de L_∞-norm. Voor de benadering, $\tilde{f}$, kunnen we elementen uit een eindige lineaire ruimte, R, beschouwen:

$$\tilde{f} = \sum_{k=1}^{n} d_k \, \phi_k(x),$$

met $\{\phi_k\}_{k=1}^{n}$ een basis.

De benaderingsfunctie is veelal een polynoom (NAG-filosofie voor approximatie van functies) of een rationale functie (IMSL-filosofie); deze laatste klasse is niet lineair in de parameters. Het voordeel van rationale benaderingen boven polynomen is dat men in principe op een oneindig interval een functie door een enkele rationale functie kan benaderen en dat het aantal vermenigvuldigingen of delingen, nodig voor de evaluatie, kan worden verkleind door deze te schrijven als een eindige kettingbreuk.

Voor existentie- en eenduidigheidsstellingen van een benadering van een continue functie met polynomen en rationale functies zie RICE [1964, 1969], CHENEY [1966] of RIVLIN [1969].

Voor de constructieve bepaling van de minimaxbenadering worden algemeen uitwisselingsalgoritmen, vernoemd naar Remes of Maehly, gebruikt (MEINARDUS [1964], CHENEY [1966]).

OPMERKINGEN.

1. In ZEGELING [1976, Tabel IIIb] wordt een voorbeeld gegeven van de minimaxbenadering van een even en oneven functie (cosinus respectievelijk sinus). MURNAGHAM & WRENCH [1959] hebben bewezen dat als de te benaderen functie even of oneven is, de minimaxbenadering dit ook is; op grond daarvan en op grond van reductie van het interval (vanwege symmetrie en periodiciteit) zouden wij de functies

   $$\sin(\pi/2\ y)/y, \quad y \in [0,.5]$$
   $$\cos(\pi/2\ y) \ , \quad y \in [0,.5],$$

   met $y = \sqrt{x}$, in de ∞-norm benaderen. Voor meer details en ALGOL 60 implementaties over benaderingen van deze en verwante functies, zie HEMKER c.s. [1973]. CODY en KUKI in RICE [1971] zijn ook zeer lezenswaardig. Handboeken zijn LYUSTERNIK c.s. [1967], en HART c.s. [1968].

2. Als stopkriterium voor de minimaxalgoritmen kan men gebruik maken van de monotonie van de fout, $u^{(i)}$, in de steunpunten bij opeenvolgende

Remes-iteratieslagen (zie MEINARDUS [1964, par.7]; het voorstel van FRASER [1965, p.300,302] en dat gebruikt in de procedure STIEFEL (DEKKER [1967]):

$$u^{(i)} < .99E^{(i)},$$

met

$u^{(i)}$: alternance-constante in i-de iteratieslag,

$E^{(i)}$: maximale fout in overige punten (of gebied) van het interval in i-de iteratieslag,

heeft het nadeel dat de iteratie kan "flip-floppen".

3. Meestal heeft men een "kort" polynoom als minimaxbenadering; het is de vraag of de algoritmen voor snelle evaluatie van polynomen op veel punten voordeel bieden (BORODIN c.s. [1971]) en, of de (a priori) schattingen van ELLIOT c.s. [1973] voor de grootte van de benaderingsfout, E_n, van toepassing zijn. Anderzijds worden schattingen gegeven door de stellingen van Jackson; zie CHENEY [1966].

4. FRASER [1965] definieert de bijna-minimaxbenaderingen als volgt. Zij $P_n(x)$ de minimaxbenadering van $f(x)$ met $E_n = \|f(x)-P_x(x)\|_\infty$. Elk polynoom $Q_n(x)$, waarvoor $\varepsilon_n = \max_{a\le x\le b}|P_n(x)-Q_n(x)|$ voldoende klein is, wordt een bijna-minimaxbenadering genoemd. Practisch stelt hij: $\varepsilon_n/E_n < .1$. Voor $f \in C[-1,1]$ geldt

$$1 \le S_n(f)/E_n(f) \le 1+\lambda_n,$$

met

λ_n monotoon toenemend, $\lambda_1 = 1.436$, $\lambda_{1000} = 4.07$,

$S_n(f) = \|s_n(f;x)-f(x)\|_\infty$,

$S_n(f;x)$: de eerste n termen van de Chebyshevreeks.

Een overzicht van de relatie tussen E_n en S_n is gegeven door GAUTSCHI [1975; 1.2.2].

9.1.6.2. Samenvatting implementaties

Tabel 9.1.5.

Implementaties voor de bepaling van minimaxbenaderingen

Algoritme	Implementatie (Programmatheek; taal)
1e Remesalgoritme:	
Chebyshevsom	E02ACA/F (NAG)
Newtonvorm	POLCHEB1 (ACCU;A60)
Andere basis-functies	UNICHEB (ACCU;A60)
2e Remesalgoritme:	
machtsvorm	MINIMAXPOL (NUMAL)

OPMERKINGEN

1. De eerste Remesalgoritme onderscheidt zich van de tweede door het aantal uitwisselingen per iteratieslag, n.l. één (1^e Remes) of meerdere (2^e Remes).
2. GOLUB, G.H. & L.B. SMITH (Collected Algorithms 414 CACM) hebben een ALGOL 60 implementatie gegeven voor de bepaling van de minimaxbenadering van een continue functie door een Chebyshevsysteem van functies.
3. IMSL bevat een routine, IRATCU, die de minimaxbenadering van een continue functie door een gewogen rationale functie bepaalt.
4. Men kan dit probleem ook formuleren als een lineair programmeringsprobleem; BARRODALE c.s. [1975] hebben een FORTRAN IV implementatie gegeven. Het voordeel van deze aanpak is dat de basisfuncties niet een Chebyshevsysteem hoeven te vormen en niet aan de Haarconditie hoeven te voldoen.

9.2. Benadering met splines

Spline-functies zijn op te vatten als een generalisatie van polynomen; zij zijn o.a. oplossingen van minimalisatieproblemen. De laatste jaren is het benaderen met splines nogal in beweging. In dit hoofdstuk willen wij, geleid door de beschikbare programmatuur, een samenhangende inleiding geven; het mini-manual van NAG, mark 5, is zeer lezenswaardig. Handboeken zijn AHLBERG c.s. [1967], BÖHMER [1974] en SCHULTZ [1974]. Als proceedings zijn o.a. verschenen GREVILLE [1969a], SCHOENBERG [1969], BÖHMER c.s. [1975]. Programmatuur, naast die beschikbaar in de programmatheken, is verschenen in BÖHMER [1974], COX c.s. [1976; opgenomen in NAG], DE BOOR [1977], SPÄTH [1973]; programmatuur verschenen in de tijdschriften kan men vinden in de collected algorithms van de ACM onder "interpolation", "curve and surface fitting" en "smoothing". Uit de literatuur blijkt dat er, uitgaande van de historische strooklat, een spectrum van spline-soorten is ontstaan; wij zullen ons beperken tot: (deficiënte) (polynoom) splines, natuurlijke splines, periodieke splines, kubische splines, B-splines, orthonormale splines en kardinaal splines.

DEFINITIES. De klasse $S_{n,k}$ van (polynoom) splines (SCHUMAKER [1969, p.96]):

$$S_{n,k} = \{\, s(t) \mid \text{er bestaan } a = x_0 < x_1 < \ldots < x_{r+1} = b \text{ en natuurlijke getallen (multipliciteit) } m_1,\ldots,m_r \text{ met } 1 \le m_i \le n+1 \text{ en } \textstyle\sum_{i=1}^{r} m_i = k, \text{ zodat } s(t) \in \Pi_n \text{ in ieder interval } I_i = (x_{i-1},x_i) \text{ en } s(t) \in C^{n-m_i} \text{ in een open omgeving van } x_i \,\}. \tag{9.2.1}$$

Men spreekt van n-de graads splines op k knooppunten (met multipliciteit $\le n+1$) in [a,b]. Bij enkelvoudige knooppunten heet de spline "simpel", bij samenvallende knooppunten - d.w.z. discontinuïteit in de lagere afgeleiden - spreekt men van een deficiënte spline (AHLBERG c.s. [1969, p.7]); bij equidistante steunpunten spreekt men van een kardinaal spline.
Bij een spline van oneven graad (2m-1) spreken we van een natuurlijke spline (notatie van de klasse: NS) als op $(-\infty,a)$ en (b,∞) de spline een polynoom is van graad m-1 (GREVILLE [1969b, p.2]); $s^{(m)}(a) = s^{m}(b) = 0$.
We spreken van een periodieke spline als $s^{(\ell)}(a+) = s^{(\ell)}(b-)$, $\ell = 0,1,\ldots,n-1$.
Als we met een derdegraads polynoom op ieder deelinterval te doen hebben spreken we van een kubische spline. De ruimte van de spline-functies is een Hilbertruimte (AHLBERG c.s. [1969, p.3]); als basis kunnen we nemen de

B-splines met compacte drager (DE BOOR [1977,1975], GREVILLE [1969b], COX [1976]); anderzijds kunnen wij een orthonormale basis invoeren (AHLBERG c.s. [1967], NITSCHE [1969], SCHOENBERG [1975]). In het vervolg zullen wij vooral de (kubische) B-spline gebruiken;
notatie:

$$M_{nj}(x),$$

met

n = orde (graad +1; n = 4 voor kubische splines),

j geeft de steunpunten aan waarbinnen de spline ongelijk aan nul is, $M_{nj}(x) \geq 0$, $x \in [x_{j-n}, x_j]$ (soms doen wij dit expliciet door bijvoorbeeld te geven $M_{44}(x; x_0=x_1, x_2, x_3=x_4)$, waarmee we bedoelen dat de steunpunten $\{x_k\}_{k=0}^{4}$ zijn en dat x_0 en x_1 evenals x_3 en x_4 samenvallen).

Bovendien zetten wij ons af tegen representaties die gebruik maken van "afgeknotte polynomen" met notatie:

$$(x-x_k)_+^j = \begin{cases} (x-x_k)^j, & x > x_k \\ 0 & \text{elders.} \end{cases}$$

9.2.1. Splines als oplossing van minimalisatieproblemen

In deze paragraaf zetten wij enige minimalisatieproblemen op een rij waarvoor spline-functies oplossingen zijn; dit geeft een indicatie van het nut van splines.

9.2.1.1. Minimalisatie van de benaderde kromming of buigingsenergie (HOLLADAY (AHLBERG c.s. [1967, p.3]))

(9.2.2) Probleem: Bepaal $f \in C^2[a,b]$ zodanig, dat geldt

$$\int_a^b (f''(x))^2 \, dx \quad \text{is minimaal,}$$

met interpolatieconditie

$$f(x_i) = y_i, \qquad i = 1,2,\ldots,n$$

en vrije voortzetting ($a = x_1$, $b = x_n$)

$$f''(a) = f''(b) = 0.$$

Oplossing: f is een natuurlijke spline met knooppunten $\{x_i\}_{i=1}^{n}$. (Notatie: $f \in NS_3(x_1,x_2,\ldots,x_n)$). In sectie 9.2.5 gaan wij hier nader op in.

OPMERKING. Voor het algemenere probleem met hogere graads (polynoom) splines als oplossing, zie AHLBERG c.s. [1967, p.156].

9.2.1.2. Minimalisatie van spreiding en "kromming" (REINSCH [1967,1971])

(9.2.3) Probleem: Bepaal $f \in C^2[a,b]$ zodanig, dat geldt

$$J(f) = \int_a^b (f''(x))^2 \, dx \quad \text{is minimaal,}$$

met approximatieconditie:

$$E(f) = \sum_{i=1}^{n} \left(\frac{f(x_i)-y_i}{\delta y_i} \right)^2$$

is begrensd, $\delta y_i > 0$.

Oplossing: $f \in NS_3(a = x_1,x_2,\ldots,b = x_n)$. In sectie 9.2.7 gaan we hier nader op in.

OPMERKING. Nauw gecorreleerd met probleem (9.2.3) zijn de probleemstellingen: gegeven $w \geq 0$, minimaliseer $J(f) + wE(f)$ (regularisatie); gegeven S^*, minimaliseer $E(f)$ met $J(f) \leq S^*$ (kleinste-kwadraten met ongelijkheidsvoorwaarden).

9.2.1.3. Kleinste-kwadratenbenadering met splines met vaste knooppunten (SCHUMAKER [1969]).

(9.2.4) Probleem: Bepaal de spline $s \in S'_{n,k}$ zodanig, dat geldt

$$\|f-s\|_2 \quad \text{is minimaal.}$$

Oplossing:

$$s(x) = \sum_{i=1}^{n+k+1} \langle f,\psi_i \rangle \, \psi_i(t),$$

met $\{\psi_i(t)\}_{i=1}^{n+k+1}$ een orthonormale basis voor $S'_{n,k}$ $(x_1,\ldots,x_n)$. In sectie

9.2.6.1 geven wij verdere literatuurverwijzingen.

9.2.1.4. Kleinste-kwadratenbenadering met splines met variabele knooppunten

(9.2.5) Probleem: Bepaal de spline met variabele knooppunten, $s \in \tilde{S}_{n,k}$, zodanig dat geldt

$$\|f-s\|_2 \quad \text{is minimaal.}$$

Hiervoor is in het algemeen geen eenduidige oplossing te geven (SCHUMAKER [1969, ex.4.5]); het bepalen van een oplossing kan men zien als een separabel niet-lineair kleinste-kwadratenprobleem, met de knooppunten als niet-lineaire parameters en de coëfficiënten van de spline-representatie als lineaire parameters. In sectie 9.2.6.2 gaan wij hier nader op in.

9.2.2. Kubische spline-representaties

De representaties van elementen uit $S_{4,k}$ kunnen we onderverdelen in twee groepen:

1. Derde-graads polynoom op ieder knooppunt-interval.

VOORBEELD (representatie van splines in IMSL)

$$(9.2.6) \qquad s(x) = c_{j3}x^3 + c_{j2}x^2 + c_{j1}x + c_{j0} \quad \text{voor } x \in I_j.$$

2. Via basisfuncties van de Hilbertruimte.

VOORBEELDEN.

1. Representatie in B-splines $\{M_{4j}(x)\}$,

$$(9.2.7) \qquad s(x) = \sum_{j=1}^{k+4} c_j M_{4j}(x);$$

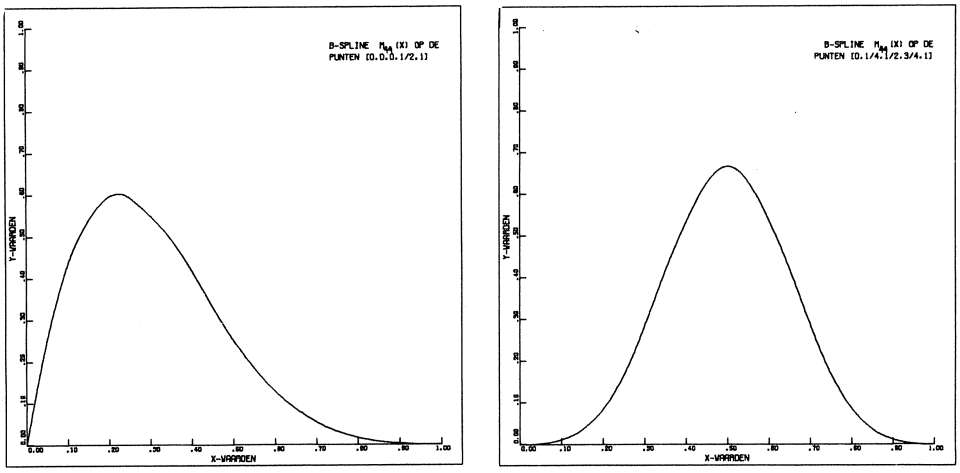
dit is de representatie van splines waarvoor NAG gekozen heeft.

2. Representatie in afgeknotte polynomen,

$$(9.2.8) \qquad s(x) = \sum_{j=0}^{3} b_j x^j + \sum_{j=1}^{k} c_j (x-x_j)_+^3 .$$

Illustratie 9.2.1.

B-splines: $M_{44}(x;0,0,0,\frac{1}{2},1)$, $M_{44}(x;0,\frac{1}{4},\frac{1}{2},\frac{3}{4},1)$



OPMERKINGEN.

1. Varianten voor (9.2.7) en (9.2.8) voor natuurlijke splines worden gegeven in GREVILLE [1969, p.27 en p.3].

2. BÖHM [1976] geeft de Beziervariant van de polynoomrepresentatie per interval:

$$s(\lambda) = b_{3j}(1-\lambda)^3 + 3b_{3j+1}(1-\lambda)^2\lambda + 3b_{3j+2}(1-\lambda)\lambda^2 + b_{3j+3}\lambda^3 ,$$

met

$$\lambda = (x-x_{j-1})/(x_j-x_{j-1}), \qquad x \in I_j.$$

De parameters van deze spline-representatie zijn de "Bezierpunten"

$$\{b_0,\{d_j \mid d_j = 2b_{3j+1} - b_{3j+2},\ j=0,1,\ldots,m\},b_{3m}\}.$$

Deze representatie is handig bij het wijzigen van een spline via een wijziging van d_k en vindt toepassing bij "Computer Aided Design".

3. De representatie (9.2.6) bevat meer parameters dan die van (9.2.7) of (9.2.8). De representatie (9.2.7) raden wij aan voor numerieke toepassingen; representatie (9.2.8) is alleen van theoretisch nut gebleken.

9.2.3. Conditie van de kubische spline-representatie

Als speciaal geval van sectie 9.1.1.2 verkrijgen we de conditiefuncties:

$$
\begin{aligned}
&K(s(c;x)) ::= \sum_{j=1}^{k+4} |c_j\, M_{4j}(x)| \;/\; |s(c;x)|, \\
&K(s(b;x)) ::= \left\{ \sum_{j=0}^{3} |b_j x^j| + \sum_{j=1}^{k} |c_j (x-x_j)_+^3| \right\} \;/\; s(b;x).
\end{aligned}
\tag{9.2.9}
$$

VOORBEELD (Conditie: B-spline representatie, afgeknotte polynoomrepresentatie).

Zij

$$M_{ni}(x) = \sum_{r=i-n}^{n} \frac{(x_r-x)_+^{n-1}}{\omega'_{ni}(x_r)},$$

dan geldt voor de conditiefuncties van het linker- en rechterlid de ongelijkheid

$$|M_{ni}(x)| \le \sum_{r=i-n}^{n} \left| \frac{(x_r-x)_+^{n-1}}{\omega'_{ni}(x_r)} \right|.$$

Illustratie 9.2.2 (condities van $M_{44}(x)$ als B-spline en zijn afgeknotte polynoomrepresentatie).

Zij

$$M_{44}(x) = \sum_{r=0}^{4} \frac{(x_r-x)_+^{3}}{\omega'_{44}(x_r)},$$

dan geldt, op grond van de compacte drager,

$$|M_{44}(x_0)| = 0,$$

terwijl voor equidistante $\{x_k \mid x_{j+1} = x_j+h,\ j = 0,1,2,3\}$ geldt

$$\sum_{r=0}^{4} \left| \frac{(x_r-x_0)_+^{3}}{\omega'_{44}(x_r)} \right| = \frac{28}{3h}.$$

9.2.4. Evaluatie van kubische splines

Als de spline gerepresenteerd wordt door polynoompjes op ieder deelinterval, dan moeten we voor de evaluatie eerst het betreffende interval op-

zoeken en vervolgens het polynoom evalueren. Programmatuur hiervoor is: ICSEVU (IMSL); SPLINE, SMUVAL (ACCULIB). Als de spline gerepresenteerd is in basis-splines dan kunnen wij eerst de basis evalueren en vervolgens de som evalueren; anderzijds kunnen wij de som evalueren zonder expliciet de basisfuncties te evalueren. Voor het speciale geval van B-splines, kunnen wij eerst vaststellen welke M_{4i} een bijdrage leveren en vervolgens door herhaalde convexe combinaties de som-spline evalueren; de algoritme is gegeven door DE BOOR [1977] en COX [1972,1976]. Programmatuur voor de evaluatie van een spline in de B-spline-representatie is E02BBA/F (NAG).

Tabel 9.2.1.

Samenvatting implementaties voor evaluatie van splines

Representatie	Programmatuur
Polynoom op I_j	ICSEVU (IMSL) SPLINE (ACCULIB,FIV) SMUVAL (ACCULIB,FIV)
B-spline	E02BBA/F (NAG)

OPMERKINGEN.

1. E02BBA/F evalueert een spline waarvan de knooppunten mogelijk samenvallen. Bij de keuze van de B-spline basis kunnen wij de "extra" knooppunten aan de randen laten samenvallen. Integratie van een dergelijke spline kan eenvoudig (COX [1975]):

$$\int_a^b s(x)dx = \int_a^b \sum_{i=1}^{m} c_i^* M_{4i}(x)dx = \tfrac{1}{4} \sum_{i=1}^{m} c_i^* .$$

GAFFNEY [1976] is ingegaan op de onbepaalde integraal van een B-spline.

9.2.5. Interpolatie met kubische splines

In deze paragraaf memoreren wij drie interpolatieprocessen:

- *interpolatie in een tabel via splines;*
- *bepaling natuurlijke spline als oplossing van* (9.2.2);
- *bepaling van de coëfficiënten als we interpoleren met B-splines.*

9.2.5.1. Interpolatie in een tabel met behulp van kubische splines

Bij dit proces is het de bedoeling uit een aantal getabelleerde waarden een niet-getabelleerde waarde te verkrijgen. Programmatuur voor ééndimensionale tabelinterpolatie is E01ADA/F (NAG); programmatuur voor tweedimensionale tabelinterpolatie is E01ACA/F (NAG), IBCIEU (IMSL). ACCULIB bevat een implementatie, gebaseerd op de Akima-interpolatie (CACM,1974); deze heet ITPLBV.

9.2.5.2. Bepaling van natuurlijke spline als oplossing van (9.2.2)

Bij kubische splines kan men een stelsel vergelijkingen opstellen voor de tweede afgeleiden in de steunpunten (STOER [1972, p.80 e.v.]; AHLBERG c.s. [1967, p.10 e.v.]; BÖHMER [1974, p.20 e.v.]; GREVILLE [1969b, p.31,32]). De matrix is trigiagonaal; bij periodieke randcondities zijn bovendien de overige hoekelementen ongelijk aan nul. Deze (dominante) tridiagonale stelsels worden opgelost via in principe een "LR"-ontbinding, gevolgd door een heen- en terugsubstitutie. Voor de periodieke spline ligt het iets lastiger (men kan deze matrix als een rang-1 gecorrigeerde tridiagonale matrix opvatten); voor nadere details zie BJÖRCK c.s. [1977]. Programmatuur voor de bepaling van de niet-periodieke natuurlijke spline is ICSICU (IMSL), SPLCON (ACCULIB,FIV), SPLCON2 (ACCULIB,A60). Een ALGOL 60 implementatie voor bepaling van de periodieke, natuurlijke spline van REINSCH: period koeffspline, is gepubliceerd in SAUER c.s. [1968,II,§4,p.270]. Wil men interpoleren met hogere graads splines, dan is programmatuur gepubliceerd in CACM 480 (SPLINECOEFF, A60 (LYCHE & SCHUMAKER)) en CACM 472 (NATSPLINEEQ, A60 (HERRIOT & REINSCH)).

OPMERKINGEN.

1. In ICSICU kan men via een optie de randcondities nader specificeren; SPLCON gaat er van uit dat de tweede afgeleide in de randpunten nul is, terwijl SPLCON2 een vastlegging van de eerste afgeleide in de randpunten verwacht.
2. Programmatuur voor bepaling van coëfficiënten van een twee-dimensionale bikubische natuurlijke spline is IBCICU (IMSL), ITPLBV (ACCULIB,FIV).
3. Convergentieuitspraken over de interpolerende spline kan men vinden in STOER [1972, p.86], BÖHMER [1974, p.28 e.v.], AHLBERG c.s. [1967, p.22 e.v.].
4. COX [1977] heeft de bepaling van de natuurlijke spline verruimd door deze te bepalen als B-spline met randcondities.

9.2.5.3. Interpolatie met B-splines (SCHUMAKER [1969])

De probleemstelling is: bepaal de coëfficiënten $\{c_j\}$ uit het stelsel vergelijkingen

$$(9.2.10) \qquad \sum_{j=1}^{k+4} c_j M_{4j}(x_i) = f_i, \qquad i = 1,2,\ldots,k+4.$$

Men kan dit opvatten als het grensgeval van een overbepaald stelsel - probleem (9.2.4), waarbij men de spline als B-spline representeert -; de programmatuur voor (9.2.10), E02BBA/F (NAG), is ook geschikt voor het overbepaalde probleem. De conditie van probleem (9.2.10) is exponentieel in de orde van de spline (BÖHMER [1974, p.188]); DE BOOR c.s. [1977] hebben een terugwaartse foutenanalyse gegeven. Als wij (9.2.8) als basis zouden nemen, dan is zelfs voor lage orde het bijbehorende stelsel slecht geconditioneerd (SCHUMAKER [1969]). De knooppunten hoeven niet samen te vallen met de meetpunten; zij moeten echter wel voldoen aan de Schoenberg-Whitney condities.

9.2.5.4. Samenvatting implementaties

Tabel 9.2.2.

Implementaties voor interpolatie met (kubische) splines

Algoritme/Probleem	Implementatie
Tabelinterpolatie	
1-dimensionaal	E01ADA/F (NAG)
2-dimensionaal	E01ACA/F (NAG) IBCIEU (IMSL) ITPLBV (ACCULIB;FIV)
Natuurlijke spline	ICSICU (IMSL) SPLCON (ACCULIB;FIV;A60) SPLCON2 (ACCULIB;A60)
B-spline interpolatie	E02BBA/F (NAG)

9.2.6. Kleinste-kwadratenbenadering met kubische splines

In deze paragraaf gaan wij nader in op het minimalisatieprobleem (9.2.4); *de nadruk ligt op de programmatuur waarbij de oplossing gere-*

presenteerd wordt in B-splines.

9.2.6.1. Orthogonale basis

Practische programmatuur is nog niet verschenen. Theoretische bijdragen zijn te vinden in AHLBERG c.s. [1967], NITSCHE [1969], SCHOENBERG [1975].

9.2.6.2. Niet-orthogonale basis (discrete data)

Probleem (9.2.4) luidt

$$\min_{\{c_k\}} \|f - \phi c\|_2 \,. \tag{9.2.11}$$

Bij een keuze van B-splines als basis heeft de matrix ϕ een blokbandstructuur; de breedte is gelijk aan de orde van de spline.

Probleem (9.2.5) luidt

$$\min_{\{x_k, c_k\}} \|f - \phi c\|_2 \,, \tag{9.2.12}$$

met x_k de plaats van de knooppunten. In Illustratie 9.3.2 geven wij aan hoe de coëfficiënten $\{c_k\}$ en de knooppunten $\{x_k\}$ gescheiden kunnen worden bepaald.

De uitwerking is een aardige programmeeropgave, omdat bij variatie van de knooppunten de lengte van de blokken van ϕ zich onderling kunnen gaan wijzigen. Programmatuur voor (9.2.11) is E02BBA/F (NAG; Fast-Givensrotaties en mogelijk samenvallende knooppunten) en ICSFKU (IMSL). LAWSON c.s. [1974] gebruiken aangepaste Householder transformaties (BSEQHT). Programmatuur voor (9.2.12) is ICSVKU (IMSL; een herhaald oplossen van een lineair kleinste-kwadratenprobleem).

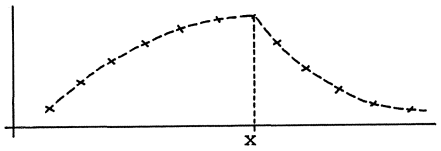
De knooppunten hoeven niet samen te vallen met de meetpunten; zij moeten echter wel voldoen aan de Schoenberg-Whitney condities. Een optimale keuze van het aantal en de ligging is (nog) niet bekend. VARAH [1977] is ingegaan op onder andere de conditie van de benadering met B-splines via het spectrum van de Gramm-matrix.

HAYES [1974] geeft over de keuze van de knooppunten de volgende toelichting. Het aantal knooppunten en hun posities kiest men op basis van: "vallen en opstaan", ervaring, en een algemeen beeld van de verlangde kromme; meer knooppunten zijn nodig in gebieden waar de functie sterk

varieert en minder waar deze slechts langzaam verandert. In de laatste gebieden kunnen te veel knooppunten aanleiding geven tot ongewenste fluctuaties; de exacte posities zijn niet vaak kritiek. De Schoenberg-Whitney condities behoeden voor rangdeficiëntie; ruwweg gesproken treedt dit op als, in een bepaald gebied van de metingen, relatief te veel knooppunten zijn gekozen t.o.v. de datapunten.

OPMERKINGEN.

1. Met behulp van de NAG-programmatuur kan men bijvoorbeeld de plaats, x, van een knik (meervoudig knooppunt) opsporen, bijvoorbeeld in:



2. BÖHMER [1974], DE BOOR [1977] en SPÄTH [1973] bevatten implementaties met betrekking tot splines; de NPL-programmatheek bevat de implementatie L2SPLINE, een generalisatie van E02BBA voor splines van hogere orde.

3. Voor aanpassing van meerdimensionale functies verwijzen wij naar HAYES [1974] en het overzichtsartikel van SCHUMAKER [1976]. Voor plotprogrammatuur wordt in Groningen McLAIN [1974] gebruikt.

4. Algoritmen voor de bepaling van de oplossing van een overbepaald stelstel met lineaire (on)gelijkheidsvoorwaarden zijn gegeven in LAWSON c.s. [1974]; zij maken echter geen gebruik van de speciale structuur van de matrix bij gebruik van B-splines. Voor het probleem met gelijkheidsvoorwaarden bevelen wij formule "20.10" in LAWSON c.s. [1974] aan; programmatuur voor pseudo-inverse-maal-vector is LLSQAR (IMSL; hier wordt de pseudo-inverse bepaald en vervolgens het inproduct uitgerekend), en F01BGA/F (NAG; als $A = U\Sigma V^T$ dan wordt afgeleverd U^TB,Σ,V). Zie ook COX [1977].

5. Bij keuze van de afgeknotte polynomen als basis is het kleinste-kwadratenprobleem slecht geconditioneerd.

9.2.6.3. Samenvatting implementaties

Tabel 9.2.3.

Implementaties voor kleinste-kwadratenbenaderingen met splines

Algoritme	Implementatie
Resultaat in representatie (9.2.7): vaste (mogelijk samenvallende) knooppunten	E02BAA/F (NAG)
Resultaat in representatie (9.2.6): a. vaste knooppunten (niet samenvallend)	ICSFKU (IMSL)
b. variabele knooppunten (niet samenvallend)	ICSVKU (IMSL)

OPMERKING. De genoemde implementaties gebruiken de B-spline-representatie.

9.2.7. Smoothing

Deze categorie kan men opvatten als restgroep; met de algemeen geaccepteerde technieken verkrijgt men geen bevredigende resultaten en gaat dan "smoothen", dat wil zeggen: doet iets anders om tevreden te geraken. In deze categorie kan men als belangrijke deelgroep de regularisatiemethoden onderscheiden (zie voor literatuur over regularisatie: hoofdstuk 7 van dit colloquiumverslag). Probleem (9.2.3) behoort tot deze categorie.

HAYES [1974a] heeft een samenvatting gegeven van de verschenen programmatuur. De algoritme SMOOTH (A60) van REINSCH [1967,1970] is opgenomen in IMSL (ICSSCU) en ACCULIB (SMUTH1, A60 & FIV).

Programmatuur voor minimalisatie van $\int_{x_1}^{x_m} (f^{(k)}(x))^2 \, dx$ onder de conditie $\sum_{r=1}^{m} (f(x_r)-f_r)^2 \leq S$ is verschenen in WOODFORD [1971] en LYCHE c.s. [1973; CACM 480].

Tabel 9.2.4.

Implementaties van smoothing-programmatuur

Algoritme	Implementatie
POWELL, in: HAYES [1970]	E02AAA/F (NAG)
REINSCH [1967,1970]	ICSSCU (IMSL) SMUTH1 (ACCULIB;A60,FIV)

OPMERKINGEN.

1. E02AAA/F zal verdwijnen in mark 6 van NAG.

2. Illustraties uit AIRD c.s. [1976], gebaseerd op de Reinsch-implementaties, zijn opgenomen in bijlage 5; door keuze van S en $\{\delta y_i\}$ kan men de benadering sturen.

Een andere data-reductie-techniek is voortgekomen uit het ontwerpen van digitale filters. Uitgebreide literatuur is te vinden in RABINER c.s. [1972, 1976] en WILSKY [1977].

Als speciaal geval kennen wij het "all-pole"-model, - lineaire predictie (voor een tutorial review zie MAKHOUL in RABINER c.s. [1976]) -, waarbij wij p en $\{a_k\}_{k=1}^{p}$ proberen te bepalen, zo dat voor de metingen $\{s_k\}$ geldt:

$$\sum_n \left\{ s_n - \sum_{k=1}^{p} a_k \, s_{n-k} \right\}^2$$

is minimaal. Als we deze sommatie over een eindig aantal (N) uitstrekken hebben we de covariantiemethode van het autoregressieve (AR) model; in matrix-notatie is dit het kleinste-kwadratenprobleem

$$\begin{pmatrix} s_0 & s_{-1} & \cdots & s_{-p+1} \\ s_1 & s_0 & \cdots & s_{-p+2} \\ \vdots & \vdots & & \\ s_{N-1} & s_{N-2} & \cdots & s_{N-p} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} \simeq \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{pmatrix} .$$

Bij sommatie over een oneindig aantal spreekt men van de autocorrelatiemethode; de matrix van de normaalvergelijkingen is symmetrisch en Toeplitz. Een algoritme dat hiervan gebruik maakt is gegeven door TRENCH (ZOHAR [1969]).

9.3. Separabele kleinste-kwadratenprobleem

In deze paragraaf geven wij aan dat kleinste-kwadratenproblemen met lineaire en niet-lineaire parameters terug te voeren zijn op kleinste-kwadratenproblemen met alleen niet-lineaire parameters; als voorbeelden noemen wij een exponentiële aanpassing en splines met variabele knooppunten.

Zij $\{a_k\}$ en $\{\alpha_k\}$ te bepalen uit

$$(9.3.1) \qquad \min_{\{a_k,\alpha_k\}} \|y_i - \sum_{j=1}^{n} a_j \phi_j(\alpha,t_i)\|_2^2 = \min_{\{a_k,\alpha_k\}} \|y - \Phi(\alpha)a\|_2^2 .$$

Men kan nu voor elke α de $a(\alpha)$ bepalen, zo dat (9.3.1) minimaal is; substitutie van deze $a(\alpha)$ geeft (voor nadere details zie GOLUB c.s. [1973])

$$(9.3.2) \qquad \min_{\alpha} \|(I - \Phi(\alpha)\Phi^{+}(\alpha))y\|_2^2 = \min_{\alpha} \|P^{\perp}_{\Phi(\alpha)}y\|_2^2 = \min_{\alpha} \sum_{i=r+1}^{m} (U^T(\alpha)y)_i^2,$$

met de m*n-matrix

$$\Phi = U\Sigma V^T$$

en $r(\alpha)$ de rang van $\Phi(\alpha)$; in het minimum worden de lineaire parameters gegeven door

$$a = V\Sigma^{+}U^Ty.$$

De operator $P^{\perp}_{\Phi(\alpha)} = I - \Phi(\alpha)\Phi^{+}(\alpha)$ is een projector op het orthogonale complement van de kolommen van Φ; men spreekt van variabele-projectiemethoden (GOLUB c.s. [1973], KROGH [1974], KAUFMAN [1975], RUHE c.s. [1974]). Het minimum van (9.3.2) kunnen we bepalen met programmatuur zoals genoemd in sectie 5.2.3 van het colloquiumverslag deel 1a.

De Fréchet-afgeleide van de residuevector $P^{\perp}_{\Phi(\alpha)}y$ luidt

$$(9.3.3) \qquad D(P^{\perp}_{\Phi(\alpha)}y) = -P^{\perp}_{\Phi(\alpha)}\, D(\Phi(\alpha))\Phi^{+}y - (P^{\perp}_{\Phi(\alpha)}\, D(\Phi(\alpha))\Phi^{+})^Ty$$

(GOLUB c.s. [1973]); deze is nodig bij gradiëntmethoden.

OPMERKINGEN.

1. Inzicht in de niet-lineaire problematiek verkrijgen wij door de benadering (POWELL, zie MURRAY [1972])

$$(9.3.4) \qquad \min_{\alpha} \|P^{\perp}_{\Phi(\alpha)}y\|_2^2 \sim \min_{\alpha} \sum_{k} [\ell_k^{(0)}(\alpha)]^2,$$

met

$$\ell_k^{(0)}(\alpha) = (P^{\perp}_{\Phi(\alpha^{(0)})}y)_k + \sum_j J_{jk}(\alpha^{(0)})(\alpha-\alpha^{(0)})_j,$$

waarin

$$J = D(P^{\perp}_{\Phi(\alpha^{(0)})}y),$$

m.a.w.

$$J_{jk}(\alpha^{(0)}) = \frac{\partial}{\partial \alpha_j} (P^{\perp}_{\Phi(\alpha^{(0)})} y)_k \Big|_{\alpha=\alpha^{(0)}}.$$

De minimale oplossing (Gauss- Newton) van het rechterlid van (9.3.4) wordt gegeven door

$$\alpha - \alpha^0 = J^+ P^{\perp}_{\Phi(\alpha^{(0)})} y. \tag{9.3.5}$$

De Marquardt-techniek kunnen we opvatten als regularisatie van het lineaire probleem, d.w.z.

$$\min_{\alpha} \{ \sum_k [\ell_k(\alpha)]^2 + \lambda \sum_k (\alpha-\alpha^{(0)})_k^2 \}, \tag{9.3.6}$$

met als oplossing

$$\alpha - \alpha^{(0)} = \left(\frac{J}{\lambda I} \right)^+ P^{\perp}_{\Phi(\alpha^0)} y; \tag{9.3.7}$$

anderzijds kunnen wij een grote $\alpha - \alpha^{(0)}$ dempen door te stellen

$$\tilde{\alpha} = \alpha^{(0)} + \gamma(\alpha-\alpha^{(0)}), \tag{9.3.8}$$

waarbij de dempingsparameter γ zo bepaald moet worden, dat het niet-lineaire probleem (9.3.4) minimaal is op deze lijn (lijnminimalisatie). Dit zoekproces, via oplossen van herhaalde linearisaties, is een gradiëntmethode; voor andere methoden zie BUS (deel 1b, hoofdstuk 5.2 van dit colloquium) en MURRAY [1972]. Een overzicht van algoritmen voor de oplossing van niet-lineaire kleinste kwadratenproblemen is gegeven door GILL c.s. [1976].

2. Als het berekenen van de gradiënt moeilijk is, dan kan men deze gradiënt benaderen door differentiequotiënten; men kan dit opvatten als een manier van lineariseren van het niet-lineaire probleem (9.3.4).

3. Uit (9.3.2) blijkt dat we voor de berekening van het minimum van $\| P^{\perp}_{\Phi(\alpha)} y \|_2^2$ een niet-lineaire kleinste-kwadratenoplosser, een singuliere-waardenontbinder en een getransponeerdematrix-maal-vector operator nodig hebben; deze zijn aanwezig in IMSL, NAG en NUMAL.

4. Vergelijkingen tussen implementaties gebaseerd op de separatie-techniek en implementaties gebaseerd op het volle probleem zijn gepubliceerd door:

a. GOLUB c.s. [1973].
Zij onderzochten de problemen: scheiding van exponenten, scheiding van Gaussianen, pieken in het Mössbauer-spectrum. Als methoden hanteerden zij: directe zoekmethode (PRAXIS van BRENT), gradiëntmethoden (gedempte Gauss-Newton en Marquardt), een variabele metriek methode; bij de directe zoekmethoden en de gradiëntmethoden pasten zij ook de variabele projectietechniek toe. Bij de scheiding van exponenten waren de Gauss-Newton- en Marquardtalgoritme met variabele projectie de meest efficiënte.

b. KAUFMAN [1975].
Zij wijzigde de Fréchet-afgeleide van de residufunctie en verkreeg daarmee een efficiëntere implementatie, terwijl het aantal iteraties niet toenam.

c. RUHE c.s. [1974].
Zij onderzochten het effect van verschillende benaderingen van de Fréchet-afgeleide. Bovendien verruimden zij de separatie tot parameters die eenvoudig te bepalen zijn en parameters die moeilijker te bepalen zijn.

5. Een verruiming van het conditiebegrip, voor niet-lineaire problemen, is gegeven door STOL [1975].

Illustratie 9.3.1 (Exponentiële aanpassing).

Bepaal $\{a,b\}$ uit

$$\min_{\{a,b\}} \sum_{k=0}^{1} (a\, e^{bk} - e^{2k})^2. \tag{9.3.9}$$

Separatie geeft

$$r(b) = \min_{b} \left\| \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - 1/(1+e^{2b}) \begin{pmatrix} 1 \\ e^{b} \end{pmatrix} (1, e^{b}) \right] \begin{pmatrix} 1 \\ e^{2} \end{pmatrix} \right\|_2 . \tag{9.3.10}$$

Hiervoor zouden we MININ (NUMAL) of ZXFIB (IMSL) kunnen gebruiken.
Scanning geeft

b	r(b)	a	
1	$e(e-1)/\sqrt{1+e^2}$	(nog niet relevant)	globaal
3	$e^2(e-1)/\sqrt{1+e^6}$	(nog niet relevant)	
2	0	1	locaal (bisectie)

Toelichting:

b = 1

$$\min_a \left\| \begin{pmatrix} 1 \\ e \end{pmatrix}(a) - \begin{pmatrix} 1 \\ e^2 \end{pmatrix} \right\|_2 =$$

$$\min_a \left\| \frac{1}{\sqrt{1+e^2}} \begin{pmatrix} 1 & e \\ -e & 1 \end{pmatrix} \left[\begin{pmatrix} 1 \\ e \end{pmatrix}(a) - \begin{pmatrix} 1 \\ e^2 \end{pmatrix} \right] \right\|_2 =$$

$$\min_a \left\| \frac{1}{\sqrt{1+e^2}} \left[\begin{pmatrix} 1+e^2 \\ 0 \end{pmatrix}(a) - \begin{pmatrix} 1+e^3 \\ e(e-1) \end{pmatrix} \right] \right\|_2 =$$

$$\left\| \begin{matrix} 0 \\ e(e-1)/\sqrt{1-e^2} \end{matrix} \right\|_2 = e(e-1)/\sqrt{(1-e^2)} \ .$$

b = 3

$$\min_a \left\| \begin{pmatrix} 1 \\ e^3 \end{pmatrix}(a) - \begin{pmatrix} 1 \\ e^2 \end{pmatrix} \right\|_2 =$$

$$\min_a \left\| \begin{pmatrix} \sqrt{1+e^6} \\ 0 \end{pmatrix}(a) - \begin{pmatrix} (1+e^5)/\sqrt{1+e^6} \\ e^2(1-e)/\sqrt{1+e^6} \end{pmatrix} \right\|_2 = e^2(e-1)/\sqrt{1+e^6} \ .$$

b = 2 (bisectie) $\min_a \left\| \begin{pmatrix} 1 \\ e^2 \end{pmatrix}(a) - \begin{pmatrix} 1 \\ e^2 \end{pmatrix} \right\|_2 = 0, \quad a = 1.$

OPMERKINGEN.

1. In plaats van het minimum van (9.3.10)- of algemener (9.3.2) - te zoeken, zouden wij ook het nulpunt van de gradiënt kunnen zoeken; dit is een stationair punt van het minimalisatieprobleem (BARRODALE c.s. [1970], BRENT [1973]). Een implementatie voor het speciale geval in illustratie 9.3.1 is, SPATH [CACM 295]; over het gebruik van de regel van Kramer hierin heb ik de volgende gedachten:
   1. Het 2×2-stelsel (normaal) vergelijkingen is symmetrisch; hiervan wordt geen gebruik gemaakt.
   2. Symmetrisch semi-positieve stelsels kunnen zich singulier gedragen ten gevolge van afrondfouten; hierop wordt niet getest. (Overigens is programmatuur verschenen voor symmetrisch indefiniete stelsels als LEQ1S (IMSL) en in de handboekserie (BUNCH c.s. [1976]).)
   3. MOLER [1974; Purdue II] en BAUER [1974] hebben laten zien dat zelfs voor deze kleine stelsels via de regel van Kramer een te groot residu wordt verkregen.

2. Men kan het probleem lineariseren door de logaritme te nemen van het linker- en rechter lid van elke vergelijking; wij zoeken dan een (rechte lijn) aanpassing

$$\min_{\{\ell na,b\}} \left\| \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \ell na \\ b \end{pmatrix} - \begin{pmatrix} 0 \\ 2 \end{pmatrix} \right\|_2 .$$

COX c.s. [1974] memoreren dat dan gelet moet worden op schalingsaspecten.

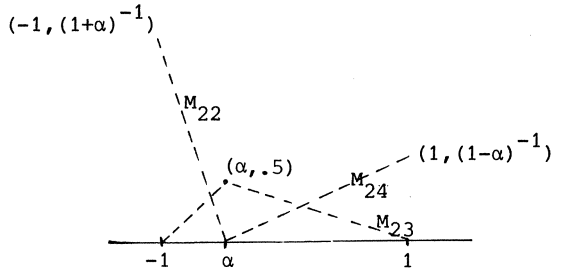
3. Programmatuur voor speciale niet-lineaire kleinste-kwadratenaanpassing is RSMITZ (IMSL; model $\alpha+\beta\gamma^x$); overige specifieke programmatuur kan men vinden in de Index to the Collected Algorithms van de ACM onder "curve and surface fitting". Literatuur voor deze en verwante problemen is BARRODALE c.s. [1970] en OSBORNE [1975].

Illustratie 9.3.2 (Spline-aanpassing met variabele knooppunten)

Als basisfuncties nemen wij de splines

$$M_{2j}(x_0=x_1,\alpha,x_3=x_4), \quad j = 2,3,4.$$

Op $[-1,1]$ zien de basisfuncties eruit als



$$M_{22} = (-x+\alpha)/(1+\alpha)^2 \quad \Big\} \quad x \in [-1,\alpha)$$

$$M_{23} = \begin{cases} (x+1)/(2(\alpha+1)) \\ (1-x)/(2(1-\alpha)) \end{cases} \quad \Big\} \quad x \in [\alpha,1)$$

$$M_{24} = (x-\alpha)/(1-\alpha)^2 \quad \Big\}$$

Benadering van $f(x) = x$ door $\sum_{j=2}^{4} c_j M_{2j}(x)$ geeft voor de coëfficiënten

$$c_2 = -(1+\alpha), \quad c_3 = 2\alpha, \quad c_4 = 1-\alpha,$$

met als residu

$$r(\alpha) = x - [-(1+\alpha)M_{22}(x) + 2\alpha M_{23}(x) + (1-\alpha)M_{24}(x)] \equiv 0, \ \forall\alpha \in [-1,1].$$

(uit de normaalvergelijkingen, bestaande uit de Gramm-matrix en de momenten, zouden wij de coëfficiënten kunnen berekenen.)

Anderzijds hebben wij veelal de functie discreet gegeven; b.v. de punten $(-1,-1)$, $(-.5,-.5)$, $(.5,.5)$, $(1,1)$. Het discrete kleinste-kwadratenprobleem luidt

$$\Phi c \simeq f \iff \begin{pmatrix} 1/(1+\alpha) & 0 & 0 \\ (.5+\alpha)/(1+\alpha)^2 & 1/(4(1+\alpha)) & 0 \\ 0 & 1/(4(1-\alpha)) & (.5-\alpha)/(1-\alpha)^2 \\ 0 & 0 & 1/(1-\alpha) \end{pmatrix} \begin{pmatrix} c_2 \\ c_3 \\ c_4 \end{pmatrix} \simeq \begin{pmatrix} -1 \\ -.5 \\ .5 \\ 1 \end{pmatrix}.$$

In plaats van bepaling van de singuliere-waardenontbinding van deze bidiagonale matrix, voor scheiding van de variabelen, kunnen wij door voor-vermenigvuldiging met orthogonale matrices als residu verkrijgen

$$r(\alpha) = \overset{(0,0,0,1)}{\begin{pmatrix} 1 & 0 & & \\ 0 & 1 & & \\ & & c'' & s'' \\ & & -s'' & c'' \end{pmatrix}} \begin{pmatrix} 1 & & & \\ & c' & s' & \\ & -s' & c' & \\ & & & 1 \end{pmatrix} \begin{pmatrix} c & s & & \\ -s & c & & \\ & & 1 & 0 \\ & & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ -.5 \\ .5 \\ 1 \end{pmatrix} \equiv 0;$$

de coëfficiënten zijn wederom

$$c_2 = -(1+\alpha), \quad c_3 = 2\alpha, \quad c_4 = 1-\alpha.$$

OPMERKINGEN.

1. In het algemeen moet men het minimum van het residu nog opzoeken.

2. In beide gevallen is er geen eenduidige oplossing voor α.

Slotwoord

Ik dank hiermede de werkgroep Approximatie van Functies en de studiegroep "Lawson & Hanson" voor de stimulerende context waarbinnen veel materiaal aan het licht is gebracht. De gebruikers van het Rekencentrum van de RUG dank ik voor de subtiliteit waarmee zij "op hol geslagen" numerici wijzen op de realiteit. Mijn collega's wil ik bedanken voor de verhelderende discussies; Mevrouw Voorintholt, in het bijzonder, voor het beschikbaar maken van enige ALGOL 60 procedures en het vervaardigen van enige grafieken. Jaap Hollenberg wil ik bovendien bedanken voor de verhelderende illustratie 9.1.4.1.

LITERATUUR

Bij het samenstellen van deze bijdrage aan het colloquium zijn wij uitgegaan van:

- bibliografieën: CHENEY [1974b], LAWSON [1968], EINARSSON [1976];
- overzichtswerken op het gebied van de approximatie: CHENEY [1966,1974a], FRASER [1965], GAUTSCHI [1975], HAYES [1970], LAWSON [1968], LAWSON c.s. [1974], CODY [1970];
- overzichtswerken op het gebied van approximeren met splines: AHLBERG c.s. [1967], BÖHMER [1976], GREVILLE [1969b];
- de inhoud van de programmatheken;
- software literatuur: EVANS [1974, ed.], RICE [1971], RICE [1977];
- recente (verspreide) literatuur.

ABRAMOWITZ, M. & I. STEGUN (eds.)[1964], *Handbook of Mathematical functions with formulas, graphs and mathematical tables*, Appl. Math. Ser. 55.

AHLBERG, J.H., E.N. NILSON & J.L. WALSH [1967], *The theory of splines and their approximations*, Academic Press.

AIRD, T.J. & D.G. KAINER [1976], *Approximation, interpolation and smoothing by cubic spline functions*, Technical Note IMSL.

AHO, V.A., J.E. HOPCROFT & J.D. ULLMAN [1974], *The design and analysis of computer algorithms*, Addison-Wesley.

ANDERSSEN, R.S. & P. BLOOMFIELD [1974], *A time series approach to numerical differentiation*, Technometrics 16, 69-75.

BARRODALE, I., F.D.K. ROBERTS & C.R. HUNT [1970], *Computing best approximations by functions nonlinear in one parameter*, Computer J. 13, 382-386.

BARRODALE, I. & C. PHILIPS [1975], *Solution of an overdetermined system of linear equations in the Chebyshev norm, Algorithm 495*, TOMS 1, 264-270.

BAUER, F.L. [1974], *Computational graphs and rounding error*, SIAM J. Numer. Anal. 11, 87-96.

BEASLEY, J.D. [1965], *A note on the rearrangement of Chebyshev series*, Computer J. 8, 278-279.

BISH, D.R.B. & J.R.A. COOPER [1976], *Guide to the NPL Algorithms Library*, NPL-report, NAC 64.

BJÖRCK, A. & G.H. GOLUB [1977], *Eigen problems for matrices associated with periodic boundary conditions*, SIAM Rev. 19, 5-16.

BJÖRCK, A. [1977], *Comment on the iterative refinement of least squares solutions*, LITH-MAT-R-1976-4 (Revized 1977-02-08).

BÖHM, W. [1976], *Parametric representation of cubic and bicubic splines*, Computing 17, 87-92.

BÖHMER, K. [1974], *Spline Funktionen, Theorie und Anwendungen*, Teubner.

BORODIN A. & I. MUNRO [1971], *Evaluating polynomials at many points*, IPL, 66-68.

BRENT, R.P. [1973], *Algorithms for minimization without derivatives*, Prentice Hall.

BUNCH, R.J. & C.P. NIELSEN [te verschijnen], *Modifying singular value and least squares problems*.

CADWELL, J.H. & D.E. WILLIAMS [1961], *Some orthogonal methods of curve and surface fitting*, Computer J. 4, 260-264.

CHENEY, E.W. [1966], *Introduction to approximation theory*, McGraw-Hill.

CHENEY, E.W. [1974a], *A survey of recent progress in approximation theory*, CNA-91, University of Texas, Austin.

CHENEY, E.W. [1974b], *A bibliography for approximation theory,* CNA-94, University of Texas, Austin.

CLENSHAW, C.W. [1962], *Chebyshev series for mathematical functions,* NPL - Math. tables, 5, HSO.

CODY, W.J. [1970], *A survey of practical rational and polynomial approximation of functions,* SIAM Rev. 12, 400-423.

CODY, W.J., W. FRASER & J.F. HART [1969], *Rational Chebyshev approximation using linear equations,* Numer. Math. 12, 242-251.

COX, M.G. [1972], *The numerical evaluation of B-splines,* JIMA 10, 134-149.

COX, M.G. & J.G. HAYES [1974], *Curve fitting: a guide and suite of algorithms for the non-specialist user,* NPL-report, NAC 26.

COX, M.G. [1975], *An algorithm for spline interpolation,* JIMA 15, 95-108.

COX, M.G. [1976], *The numerical evaluation of a spline from its B-spline representation,* NPL-report, NAC 68.

COX, M.G. [1977], *The incorporation of boundary conditions in spline approximation problems,* Proceedings Dundee Biennial Conference, NPL-report NAC 80.

CULLUM, J. [1971], *Numerical differentiation and regularization,* SIAM J. Numer. Anal. 8, 254-265.

DE BOOR, C. [1976], *Splines as linear combinations of B-splines - A survey,* in: LORENTZ, G.G., c.s. (eds.).

DE BOOR, C. [1977], *Package for calculating with B-splines,* SIAM J. Numer. Anal. 14, 441-472.

DE BOOR, C. & A. PINKUS [1977], *Backward error analysis for totally positive linear systems,* Numer. Math. 27, 485-490.

DAVIS, P.J. & P. RABINOWITZ [1975], *Methods of numerical integration,* Academic Press.

DEKKER, T.J. [1967], *Cursus Wetenschappelijk Rekenen A,* Numerieke Wiskunde, 3 delen, Mathematisch Centrum.

DEUFLHARD, P. [1976], *On algorithms for the summation of certain special functions,* Computing 17, 37-48.

ELLIOT, D. [1968], *Error analysis of an algorithm for summing certain finite series,* J. Austral. Math. Soc. 8, 213-221.

ELLIOT, D., BINH LAM [1973], *An estimate of $E_n(f)$ for large n,* SIAM J. Numer. Anal. 10, 1091-1102.

EINARSSON, B. [1976], *Bibliography on numerical software,* FOA. National Defence Research Institute, BOX 98, S-147 00 Tumba.

EVANS, D.J. (ed.) [1974], *Software for numerical mathematics,* Academic Press.

FORSYTHE, G.E. [1957], *Generation and use of orthogonal polynomials in data fitting with a digital computer,* J. Soc. Indust. Appl. Math. 5, 74-88.

FRASER, W. [1965], *A survey of methods of computing minimax and near-minimax polynomial approximation for functions of a single independent variable,* J. ACM 12, 295-314.

GAFFNEY, P.W. [1976], *The calculation of indefinite integrals of B-splines,* JIMA 17, 37-41.

GAUTSCHI, W. [1975], *Computational methods in special functions - A survey,* in: ASKEY, R. (ed.), *Theory and application of special functions,* Proceedings of an advanced seminar at the university of Wisconsin.

GAUTSCHI, W. [1972], *The condition of orthogonal polynomials,* Math. Comp. 26, 923-924.

GAUTSCHI, W. [1973], *On the condition of algebraic equations,* Numer. Math. 21, 405-424.

GENTLEMAN, W.M. [1972], *Basic procedures for large, sparse or weighted linear least squares problems,* Report CSRR - 2068.

GILL, P.E. & W. MURRAY [1976], *Algorithms for the solution of the non-linear least-squares problem,* NPL-report, NAC 71.

GOLUB, G.H. & M.A. SAUNDERS [1969], *Linear least squares and quadratic programming,* CS-134, CSD Stanford.

GOLUB, G.H. & V. PEREYRA [1973], *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate,* SIAM J. Numer. Anal. 10, 413-432.

GOLUB, G.H. & L.B. SMITH [1971], *Chebyshev approximation of continuous functions by a Chebyshev system of functions,* Comm. ACM 14, 737-746.

GOLUB, G.H. & J.M. VARAH [1974], *On a characterization of the best ℓ_2-scaling of a matrix,* SIAM J. Numer. Anal. 11, 472-479.

GREVILLE, T.N.E. (ed.) [1969a], *Theory and applications of spline functions,* Publ. 22 of the Math. Research Center, U.S. Army, Univ. of Wisconsin.

GREVILLE, T.N.E. [1969b], *Introduction to spline functions,* in: GREVILLE, T.N.E. (ed.) [1969a], 1-35.

HAMMING, R.W. [1962], *Numerical methods for scientists and engineers,* McGraw-Hill.

HART, J.F., E.W. CHENEY, C.L. LAWSON, H.J. MAEHLY, C.K. MESTENYI, J.R. RICE, H.C. THACHER & C. WITZGALL [1968], *Computer approximations,* SIAM Series in applied mathematics, Wiley.

HAYES, J.G. (ed.) [1970], *Numerical approximation to functions and data,* Athlone Press.

HAYES, J.G. [1974a], *Algorithms for curve and surface fitting,* in: EVANS, D.J. [1974], 219-233.

HAYES, J.G. [1974b], *Numerical methods for curve and surface fitting,* NPL-report NAC 50.

HERRIOT, J.G. & C.H. REINSCH [1973], *Procedures for natural spline interpolation,* ALGORITHM 472, Comm. ACM, 763-768.

HEMKER, P.W., W. HOFFMAN, S.P.N. VAN KAMPEN, H.L. OUDSHOORN & D.T. WINTER [1973], *Single and double-length computation of elementary functions,* Report NW 7/73, Mathematisch Centrum, Amsterdam.

JOYCE, D.C. [1971], *Survey of extrapolation processes in numerical analysis,* SIAM Rev. 13, 435-490.

KAUFMAN, L. [1975], *A variable projection method for solving separable nonlinear least squares problems,* BIT 15, 49-75.

KROGH, F.T. [1974], *Efficient implementations of a variable projection algorithm for nonlinear least squares problems,* Comm. ACM 17, 167-169.

KUNG, H.T. [1973], *Fast evaluation and interpolation,* Report Carnegie-Mellon University.

LAWSON, C.L. [1968], *Survey of computer methods for fitting curves to discrete data or approximating continuous functions,* SICNUM 3,3.

LAWSON, C.L. [1968], *Bibliography, Recent publications in approximation theory with emphasis on computer approximation,* Comp. Rev. 11, 691-699.

LAWSON, C.L. & R.J. HANSON [1974], *Solving least squares problems,* Prentice Hall.

LUKE, Y.L. [1969], *The special functions and their approximations,* Academic Press.

LUKE, Y.L. [1975], *Mathematical functions and their approximations,* Academic Press.

LORENTZ, G.G., C.K. CHUI & L.L. SCHUMAKER (eds.) [1976], *Approximation theory II.*

LYCHE, T. & L.L. SCHUMAKER [1974], *Procedures for computing smoothing and interpolating natural splines,* Algorithm 480, Comm. ACM 17.

LYUSTERNIK, L.A., O.A. CHERVONENKIS & A.R. YANPOLSKII [1965], *Handbook for computing elementary functions,* Pergamon.

Mc. LAIN, D.H. [1974], *Drawing contours from arbitrary data points,* Computer J. 17, 318-328.

MEINARDUS, G. [1964], *Approximation von Funktionen und ihre numerische Behandlung,* Springer tracts in natural philosophy, 4.

MOENCK, R. & A. BORODIN [1972], *Fast modular transforms via division,* Proceedings of the 13th Symposium on Switching and automata theory, IEEE Computer Society, 90-96.

MOLER, C.B. [1974], *Cramer's rule on 2-by-2 systems,* Conference on Math. software, Purdue (II).

MURNAGHAN, F.D. & J.W. WRENCH, Jr. [1959], *The determination of the Chebyshev approximating polynomial for a differentiable function,* MTAC 13, 185-193.

MURRAY, W. [1972], *Numerical methods for unconstrained optimization,* Academic Press.

NEWBERY, A.C.R. [1974], *Error analysis for polynomial evaluation,* Math. Comp. 28, 789-793.

NEWBERY, A.C.R. [1975], *Polynomial evaluation schemes,* Math. Comp. 29, 1046-1050.

NITSCHE, J. [1969], *Orthogonal Reihenentwicklung nach linearen Spline-Funktionen,* J. Approx. theory 2, 66-78.

OSBORNE, M.R. [1975], *Some special nonlinear least squares problems,* SIAM J. Numer. Anal. 12, 571-592.

PAIGE, C.C. & M.A. SAUNDERS [1973], *Solution of sparse indefinite systems of equations and least squares problems,* STAN-CS-73-399.

PAYNE, J.A. [1970], *An automatic curve-fitting package,* in: HAYES [1970].

PETERS, G. & J.H. WILKINSON [1970], *The least squares problem and pseudo-inverses,* Computer J. 13, 309-316.

RABINER, L.R. & C.M. RADER (eds.) [1972,1976], *Digital Signal Processing I,II,* IEEE, Wiley.

REIMER, M. [1977], *Auswertungsalgorithmen fast-optimaler numerischer Stabilität für Polynome,* Computing 17, 289-296.

REINSCH, C.H. [1967,1971], *Smoothing by spline functions I, II,* Numer. Math. 10, 177-183; Numer. Math. 16, 451-454.

RICE, J.R. [1964,1969], *The approximation of functions, Vol. I: Linear theory, Vol. II: Advanced topics,* Addison-Wesley.

RICE, J.R. (ed.) [1971], *Mathematical software.* Proc. Purdue I, Acad. Press.

RICE, J.R. [1977], *Software for numerical computation,* CSD-TR 214, Purdue University.

RIVLIN, T.J. [1969], *An introduction to the approximation of functions,* Blaisdell.

RIVLIN, T.J. [1974], *The Chebyshev polynomials,* Wiley.

RUHE, A. & P.A. WEDIN [1974], *Algorithms for separable nonlinear least squares problems,* Report Uminf-47.74.

RUTISHAUSER, H.[1968a], *Zum Problematik der Nulstellenbestimmung bei Polynomen,* in: DEJON, W. (ed.), *The fundamental theorem of algebra.*

RUTISHAUSER, H.[1968b], *Once again: The least squares problem,* Linear Algebra and its applications 1, 479-488.

SALZER, H.E. [1973], *A recurrence scheme for converting from one orthogonal expansion into another,* Comm. ACM 16, 705-707.

SAUER, R. & I. SZABÓ (eds.) [1968], *Mathematische Hilfsmittel des Ingenieurs, III,* Springer.

SCHOENBERG, I.J. [1969], *Approximation with special emphasis on spline functions,* Academic Press.

SCHOENBERG, I.J. [1975], *Notes on spline functions V, Orthogonal or Legendre splines,* J. Approx. Theory 13, 84-104.

SCHULTZ, M.H. [1973], *Spline analysis,* Prentice Hall.

SCHUMAKER, L.L. [1969], *Approximation by splines,* 65-86;
Some algorithms for the computation of interpolating and approximating spline functions, 87-102, in: GREVILLE, T.N.E. (ed.) [1969].

SCHUMAKER, L.L. [1976], *Fitting surfaces to scattered data,* in: LORENTZ, G.G. c.s. (eds.) [1976].

SCRATON, R.E. [1970], *A method for improving the convergence of Chebyshev series,* Computer J. 13, 202-203.

SHAMPINE, L.F. [1975], *Discrete least squares polynomial fits,* Comm. ACM 18, 179-180.

SMITH, J.M. [1975], *Scientific analysis on the pocket calculator,* John Wiley.

SMITH, L.B. [1969], *The use of man-machine interaction in data-fitting problems,* CS 131, CSD Stanford.

SPÄTH, H. [1967], *Exponential curve fit,* ALGORITHM 295, Comm. ACM. 10, 2.

SPÄTH, H. [1973], *Algorithmen zur Konstruktion glatter Kurven und Flächen,* Oldenburg.

STOER, J. [1972], *Einführung in die numerische Mathematik I,* Heidelberger Taschenbücher 105.

STOL, PH. TH. [1975], *A contribution to theory and practice of nonlinear parameter optimization,* Pudoc, Wageningen.

THACHER, H.C. [1964], *Conversion of a power to a series of Chebyshev polynomials,* Comm. ACM. 7, 181-182.

THACHER, H.C. [1966], *Independent variable transformations in approximation,* Proc. IFIP Congres 1965, 576-577.

TRAUB, J.F. & M. SHAW [1974], *On the number of multiplications for the evaluation of a polynomial and some of its derivatives,* J. ACM 21, 161-167.

VAN DER SLUIS, A. [1975], *Stability of solutions of linear least squares problems,* Numer. Math. 23, 241-254.

VARAH, J.M. [1977], *On the condition number of local bases for piecewise cubic polyncmials,* Math. Comp. 3, 37-44.

WILSKY, A.S. [1977], *Digital signal processing and control and estimation theory - Points of tangency, area of intersection, and parallel directions,* ESL-R-712, M.I.T.

WOODFORD, C.H. [1970], *An algorithm for data smoothing using spline functions,* BIT 10, 501-510.

WOZNIAKOWSKI, H. [1974], *Rounding error analysis for the evaluation of a polynomial and some of its derivatives,* SIAM J. Numer. Anal. 11, 780-787.

ZEGELING, H. [1976], *Numerieke methoden behandeld voor computergebruik. Lineaire approximatie,* ACCU - Reeks 7.

ZOHAR SHALHAV [1969], *Toeplitz matrix inversion; the algorithm of W.F. Trench,* J. ACM. 16, 592-601.

Bijlagen

Bijlage 1 (Relaties tussen coëfficiënten van machtssom- en Chebyshevsom-representaties)

LEMMA 1. *Zij*

$$(1.1)\qquad \sum_{k=0}^{n} a_k x^k = \sum_{k=0}^{n} b_k T_k(x), \qquad |x| \le 1$$

dan geldt:

$$(1.2)\qquad \sum_{k=0}^{n} a_k = \sum_{k=0}^{n} b_k;$$

$$(1.3)\qquad \sum_{k=0}^{n} (-1)^k a_k = \sum_{k=0}^{n} (-1)^k b_k;$$

$$(1.4)\qquad b_k = \begin{cases} \sum_{\ell=k}^{n} a_\ell \, \theta_{\ell k} \\ \\ \sum_{\ell=0}^{n}{}' a_\ell \, \theta_{\ell 0} \end{cases},$$

met

$$\theta_{\ell k} = \begin{cases} 2^{1-\ell} \binom{\ell}{\frac{\ell-k}{2}}, & \ell-k = \textit{even} \\ \\ 0, & \ell-k = \textit{oneven;} \end{cases}$$

$$(1.5)\qquad \begin{aligned} &\{a_k\} \textit{ strikt alternerend} \leftrightarrow \{b_k\} \textit{ strikt alternerend;} \\ &\{a_k\} \textit{ tekenvast} \qquad\qquad \leftrightarrow \{b_k\} \textit{ tekenvast.} \end{aligned}$$

BEWIJS VAN LEMMA 1.

Ad (1.2) en (1.3): Substitueer x = 1 respectievelijk x = -1 in (1.1).

Ad (1.4): Uit

$$\sum_{k=0}^{n} a_k x^k = \sum_{k=0}^{n} b_k T_k(x)$$

volgt

$$b_k = \sum_{\ell=k}^{n} a_\ell \langle x^\ell, T_k(x)\rangle \;/\; \langle T_k(x), T_k(x)\rangle$$

$$= \sum_{\ell=k}^{n} a_\ell \langle \sum_{q=0}^{\ell}{}' \theta_{\ell q}\, T_q(x), T_k(x)\rangle \;/\; \langle T_k(x), T_k(x)\rangle$$

$$= \begin{cases} \sum_{\ell=k}^{n} a_\ell\, \theta_{\ell k}, & k > 0 \\ \sum_{\ell=0}^{n}{}' a_\ell\, \theta_{\ell 0} & \end{cases} .$$

THACHER [1964] geeft voor $\theta_{\ell k}$ de formule

$$\theta_{\ell k} = \begin{cases} 2^{1-\ell} \binom{\ell}{\frac{\ell-k}{2}}, & \ell-k = \text{even} \\ 0, & \ell-k = \text{oneven}. \end{cases}$$

Ad (1.5): Volgt direct uit (1.4).

Bijlage 2 (Chebyshevsomrepresentaties)

De Chebyshevreeks

$$f_n(x) = \sum_{k=0}^{n} a_k\, T_k(x)$$

met

$$T_0 = 1, \quad T_1 = x$$

$$T_{k+1}(x) = 2x\, T_k(x) - T_{k-1}(x), \qquad k = 1,2,\ldots$$

kunnen wij representeren als:

(2.3) $$f_n(x) = (1,x) \sum_{k=0}^{n} \begin{pmatrix} 2x & 1 \\ -1 & 0 \end{pmatrix}^k \begin{pmatrix} a_k \\ 0 \end{pmatrix};$$

(2.4) $$f_n(x) = g_n^{(0)}(\lambda) = (1, -\lambda/2) \sum_{k=0}^{n} \begin{pmatrix} \lambda+1 & \lambda \\ 1 & 1 \end{pmatrix}^k \begin{pmatrix} a_k \\ 0 \end{pmatrix}, \quad \lambda = -4\sin^2(\arccos x)/2;$$

$$(2.5)\qquad f_n(x) = g_n^{(\pi)}(\lambda) = (1,-\lambda/2)\sum_{k=0}^{n}\begin{pmatrix}\lambda-1 & -\lambda\\ 1 & -1\end{pmatrix}^k\begin{pmatrix}a_k\\ 0\end{pmatrix},\quad \lambda = 4\cos^2(\arccos x)/2.$$

BEWIJS. Uit

$$T_k(x) = (T_k(x),T_{k-1}(x))\begin{pmatrix}1\\0\end{pmatrix} = (T_1(x),T_0(x))\begin{pmatrix}2x & 1\\ -1 & 0\end{pmatrix}^{k-1}\begin{pmatrix}1\\0\end{pmatrix}$$

$$= (T_0,T_1(x))\begin{pmatrix}2x & 1\\ -1 & 0\end{pmatrix}^{k}\begin{pmatrix}1\\0\end{pmatrix}$$

verkrijgen wij

$$f_n(x) = \sum_{k=0}^{n} a_k\, T_k(x) = (1,x)\sum_{k=0}^{n}\begin{pmatrix}2x & 1\\ -1 & 0\end{pmatrix}^k\begin{pmatrix}a_k\\0\end{pmatrix}.$$

Uitgaande van de factorisatie

$$\begin{pmatrix}2\cos\theta & 1\\ -1 & 0\end{pmatrix} = \begin{pmatrix}1 & 1\\ 0 & -1\end{pmatrix}\begin{pmatrix}\lambda+1 & \lambda\\ 1 & 1\end{pmatrix}\begin{pmatrix}1 & 1\\ 0 & -1\end{pmatrix},\quad \lambda = -4\sin^2\frac{\theta}{2}$$

verkrijgen wij

$$f_n(x) = g_n^{(0)}(\lambda) = (1,-\lambda/2)\sum_{k=0}^{n}\begin{pmatrix}\lambda+1 & \lambda\\ 1 & 1\end{pmatrix}^k\begin{pmatrix}a_k\\0\end{pmatrix}.$$

Anderzijds, uitgaande van de factorisatie

$$\begin{pmatrix}2\cos\theta & 1\\ -1 & 0\end{pmatrix} = \begin{pmatrix}1 & -1\\ 0 & -1\end{pmatrix}\begin{pmatrix}\lambda-1 & -\lambda\\ 1 & -1\end{pmatrix}\begin{pmatrix}1 & -1\\ 0 & -1\end{pmatrix},\quad \lambda = 4\cos^2\frac{\theta}{2}$$

verkrijgen wij

$$f_n(x) = g_n^{(\pi)}(\lambda) = (1,-\lambda/2)\sum_{k=0}^{n}\begin{pmatrix}\lambda-1 & -\lambda\\ 1 & -1\end{pmatrix}^k\begin{pmatrix}a_k\\0\end{pmatrix}.$$

OPMERKINGEN.

1. We hebben $g_n(\lambda)$ bovengeïndiceerd wegens de begrensdheid van

$$(2.6)\qquad \frac{\partial g_n^{(0)}(\lambda)}{\partial\lambda}\Delta\lambda = \varepsilon_\lambda \operatorname{tg}\frac{\theta}{2}\sum_{k=0}^{n} k a_k \sin k\theta \quad \text{(STOER [1972,p.65])},$$

(2.7) $$\frac{\partial g_n^{(\pi)}(\lambda)}{\partial \lambda}\,\Delta\lambda = \varepsilon_\lambda \operatorname{ctg}\frac{\theta}{2}\sum_{k=0}^{n} k a_k \sin k\theta$$ (STOER [1972,p.65])

voor de geïndiceerde waarden

$$0 \sim \theta \leftrightarrow x \sim 1 \quad (2.6)$$
$$\pi \sim \theta \leftrightarrow x \sim -1 \quad (2.7).$$

Representaties (2.4) en (2.5) staan bekend als de modificaties van Reinsch. Uit (2.6) en (2.7) zien we dat λ bepaald moet worden met een kleine relatieve fout, ε_λ; bepaling van λ bij de Chebyshevsom als x gegeven is in plaats van θ kan niet, in het algemeen, met een kleine relatieve fout. De implementatie van COX c.s. [1974] is derhalve geen verbetering voor willekeurige x.

2. Algemenere representaties kan men verkrijgen uit de factorisatie van Hollenberg

$$\begin{pmatrix} 2x & 1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 2x-a & a^2-2xa+1 \\ -1 & a \end{pmatrix}\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}.$$

Het nut van de eerder genoemde speciale gevallen ligt in de transformatie van de onafhankelijke variabele met kleine relatieve fout, ε_λ klein.

Bijlage 3 (ALGOL 60 procedure ODDCHEPOLSER)

```
"REAL" "PROCEDURE" ODDCHEPOLSER(N,X,A);
"VALUE" N,X; "INTEGER" N; "REAL" X; "ARRAY" A;
"COMMENT" ODDCHEPOLSER:=A[0]T[1](X)+A[1]T[3](X)+....+A[N]T[2N+1](X);
"IF" N<0 "THEN" ODDCHEPOLSER:=0.0 "ELSE"
"IF" N=0 "THEN" ODDCHEPOLSER:=X*A[0] "ELSE"
"IF" N=1 "THEN" ODDCHEPOLSER:=X*(A[0]+A[1]*(4*X*X-3)) "ELSE"
"BEGIN"
   "INTEGER" K;
   "REAL" H,R,S,Y;
   Y:=4*X*X-2;
   R:=A[N];
   H:=A[N-1]+R*Y;
   "FOR" K:=N-2 "STEP" -1 "UNTIL" 0 "DO"
   "BEGIN"
     S:=R;
     R:=H;
     H:=A[K]+R*Y-S;
   "END" K;
   ODDCHEPOLSER:=X*(H-R);
"END" ODDCHEPOLSER;
```

Purpose : Calculation of $\sum_{k=0}^{n} a_k \, T_{2k+1}(x)$.

Data : integer $n \geq 0$;

real $x \in [-1,1]$;

array a[0:n]: coefficients of the odd Chebyshev series.

Results : value of the series.

Algorithm: see (9.1.12).

$$\sum_{k=0}^{n} a_k \, T_{2k+1}(x) = x(1,-1)\left(\begin{pmatrix} a_0 \\ 0 \end{pmatrix} + \begin{pmatrix} 2T_2(x) & -1 \\ 1 & 0 \end{pmatrix}\left(\begin{pmatrix} a_1 \\ 0 \end{pmatrix} + \ldots + \right.\right.$$

$$\left.\left. + \begin{pmatrix} 2T_2(x) & -1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_n \\ 0 \end{pmatrix}\right) \ldots \right).$$

Author : C.G. van der Laan.

Bijlage 4 (ALGOL 60 procedure POLCHS)

```
"PROCEDURE" POLCHS(A,N);
"VALUE" N; "INTEGER" N; "ARRAY" A;
"COMMENT"    N                                  N
             SUM A[K]X**K IS TRANSFORMED INTO SUM B[K]T[K](X)
             K=0                              K=0
                (LITERATURE: HAMMING(1962, P. 256));
"IF" N>1 "THEN"
"BEGIN"
   "COMMENT"  SCALING;
   "INTEGER" K,L,TWOPOW;
   TWOPOW:=2;
   "FOR" K:=1 "STEP" 1 "UNTIL" N-2 "DO"
   "BEGIN"
      A[K]:=A[K]/TWOPOW;
      TWOPOW:=TWOPOW*2;
   "END";
   A[N-1]:=2*A[N-1]/TWOPOW;
   A[N]:=A[N]/TWOPOW;
   A[N-2]:=A[N-2]+A[N];
   "COMMENT" N<=2 READY;
   "FOR" K:=N-2 "STEP" -1 "UNTIL" 1 "DO"
   "BEGIN"
      A[K-1]:=A[K-1]+A[K+1];
      A[K]:=A[K]*2+A[K+2];
      "FOR" L:=K+1 "STEP" 1 "UNTIL" N-2 "DO"
      A[L]:=A[L]+A[L+2];
   "END";
"END" POLCHS;
```

Purpose : Transformation coefficients power sum representation into coefficients Chebyshev sum representation.

Data : integer $n \geq 0$;
array a[0:n] which contains the coefficients of the powers of the independent variable.

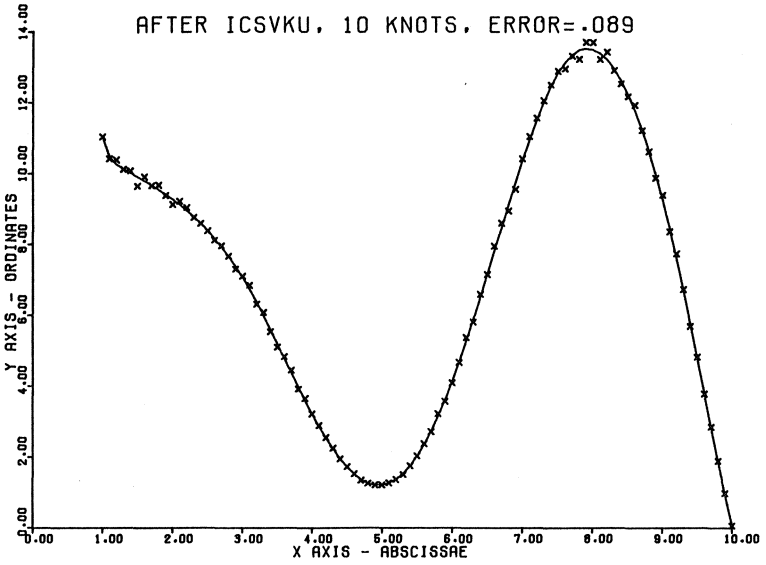
Results : the array a[0:n] is overwritten by the coefficients of the equivalent Chebyshev sum.

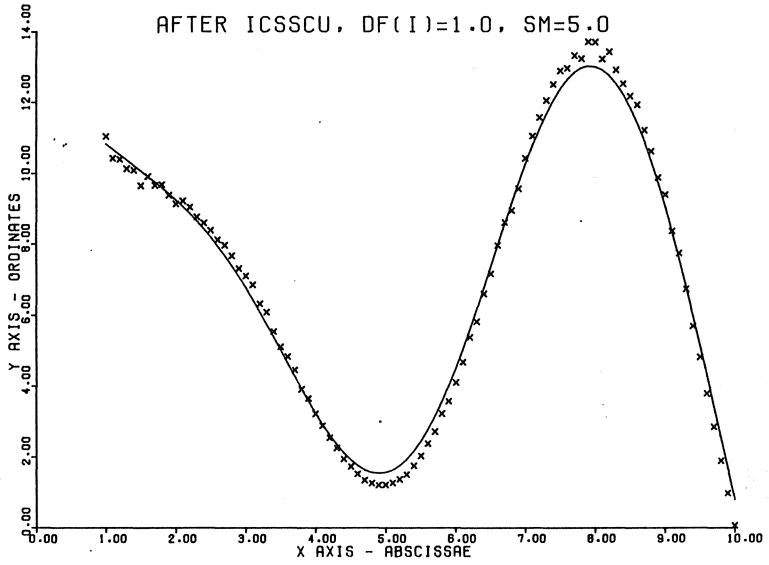
Algorithm: see (9.1.16).

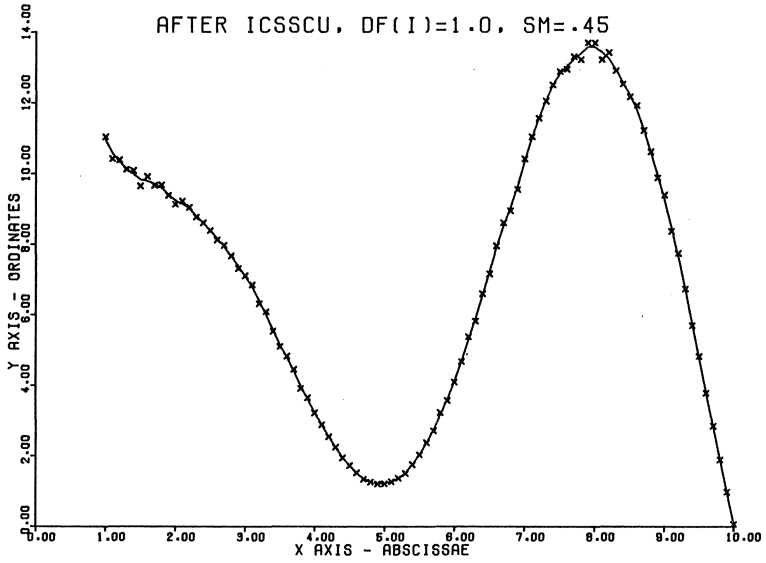
Author : C.G. van der Laan.

Bijlage 5 (Illustraties van kubische spline approximaties met programmatuur uit IMSL)

De onderstaande illustraties zijn ontleend aan AIRD c.s. [1976].






AFTER ICSSCU, DF(I)=1.0, SM=.45
Y AXIS - ORDINATES
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
0.00
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.00
X AXIS - ABSCISSAE

INDEX bij hoofdstuk 9, Approximatie van functies en data

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. GÖBEL & J. VAN DE LUNE, *Leergang Besliskunde, deel 1: Wiskundige basiskennis*, 1965. ISBN 90 6196 014 2.

MCS 1.2 J. HEMELRIJK & J. KRIENS, *Leergang Besliskunde, deel 2: Kansberekening*, 1965. ISBN 90 6196 015 0.

MCS 1.3 J. HEMELRIJK & J. KRIENS, *Leergang Besliskunde, deel 3: Statistiek*, 1966. ISBN 90 6196 016 9.

MCS 1.4 G. DE LEVE & W. MOLENAAR, *Leergang Besliskunde, deel 4: Markovketens en wachttijden*, 1966. ISBN 90 6196 017 7.

MCS 1.5 J. KRIENS & G. DE LEVE, *Leergang Besliskunde, deel 5: Inleiding tot de mathematische besliskunde*, 1966. ISBN 90 6196 018 5.

MCS 1.6a B. DORHOUT & J. KRIENS, *Leergang Besliskunde, deel 6a: Wiskundige programmering 1*, 1968. ISBN 90 6196 032 0.

MCS 1.6b B. DORHOUT, J. KRIENS & J.TH. VAN LIESHOUT, *Leergang Besliskunde, deel 6b: Wiskundige programmering 2*, 1977. ISBN 90 6196 150 5.

MCS 1.7a G. DE LEVE, *Leergang Besliskunde, deel 7a: Dynamische programmering 1*, 1968. ISBN 90 6196 033 9.

MCS 1.7b G. DE LEVE & H.C. TIJMS, *Leergang Besliskunde, deel 7b: Dynamische programmering 2*, 1970. ISBN 90 6196 055 X.

MCS 1.7c G. DE LEVE & H.C. TIJMS, *Leergang Besliskunde, deel 7c: Dynamische programmering 3*, 1971. ISBN 90 6196 066 5.

MCS 1.8 J. KRIENS, F. GÖBEL & W. MOLENAAR, *Leergang Besliskunde, deel 8: Minimaxmethode, netwerkplanning, simulatie*, 1968. ISBN 90 6196 034 7.

MCS 2.1 G.J.R. FÖRCH, P.J. VAN DER HOUWEN & R.P. VAN DE RIET, *Colloquium Stabiliteit van differentieschema's, deel 1*, 1967. ISBN 90 6196 023 1.

MCS 2.2 L. DEKKER, T.J. DEKKER, P.J. VAN DER HOUWEN & M.N. SPIJKER, *Colloquium Stabiliteit van differentieschema's, deel 2*, 1968. ISBN 90 6196 035 5.

MCS 3.1 H.A. LAUWERIER, *Randwaardeproblemen, deel 1*, 1967. ISBN 90 6196 024 X.

MCS 3.2 H.A. LAUWERIER, *Randwaardeproblemen, deel 2*, 1968. ISBN 90 6196 036 3.

MCS 3.3 H.A. LAUWERIER, *Randwaardeproblemen, deel 3*, 1968. ISBN 90 6196 043 6.

MCS 4 H.A. LAUWERIER, *Representaties van groepen*, 1968. ISBN 90 6196 037 1.

MCS 5 J.H. VAN LINT, J.J. SEIDEL & P.C. BAAYEN, *Colloquium Discrete wiskunde*, 1968.
ISBN 90 6196 044 4.

MCS 6 K.K. KOKSMA, *Cursus ALGOL 60*, 1969. ISBN 90 6196 045 2.

MCS 7.1 *Colloquium Moderne rekenmachines, deel 1*, 1969. ISBN 90 6196 046 0.

MCS 7.2 *Colloquium Moderne rekenmachines, deel 2*, 1969. ISBN 90 6196 047 9.

MCS 8 H. BAVINCK & J. GRASMAN, *Relaxatietrillingen*, 1969.
ISBN 90 6196 056 8.

MCS 9.1 T.M.T. COOLEN, G.J.R. FÖRCH, E.M. DE JAGER & H.G.J. PIJLS, *Elliptische differentiaalvergelijkingen, deel 1*, 1970.
ISBN 90 6196 048 7.

MCS 9.2 W.P. VAN DEN 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.
ISBN 90 6196 049 5.

MCS 10 J. FABIUS & W.R. VAN ZWET, *Grondbegrippen van de waarschijnlijkheidsrekening*, 1970. ISBN 90 6196 057 6.

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.
ISBN 90 6196 067 3.

MCS 12 T.J. DEKKER, *Numerieke algebra*, 1971. ISBN 90 6196 068 1.

MCS 13 F.E.J. KRUSEMAN ARETZ, *Programmeren voor rekenautomaten; De MC ALGOL 60 vertaler voor de EL X8*, 1971. ISBN 90 6196 069 X.

MCS 14 H. BAVINCK, W. GAUTSCHI & G.M. WILLEMS, *Colloquium Approximatietheorie*, 1971. ISBN 90 6196 070 3.

MCS 15.1 T.J. DEKKER, P.W. HEMKER & P.J. VAN DER HOUWEN, *Colloquium Stijve differentiaalvergelijkingen, deel 1*, 1972. ISBN 90 6196 078 9.

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. ISBN 90 6196 079 7.

MCS 15.3 P.A. BEENTJES, K. DEKKER, P.W. HEMKER & M. VAN VELDHUIZEN, *Colloquium Stijve differentiaalvergelijkingen, deel 3*, 1975.
ISBN 90 6196 118 1.

MCS 16.1 L. GEURTS, *Cursus Programmeren, deel 1: De elementen van het programmeren*, 1973. ISBN 90 6196 080 0.

MCS 16.2 L. GEURTS, *Cursus Programmeren, deel 2: De programmeertaal ALGOL 60*, 1973. ISBN 90 6196 087 8.

MCS 17.1 P.S. STOBBE, *Lineaire algebra, deel 1*, 1974. ISBN 90 6196 090 8.

MCS 17.2 P.S. STOBBE, *Lineaire algebra, deel 2*, 1974. ISBN 90 6196 091 6.

MCS 17.3 N.M. TEMME, *Lineaire algebra , deel 3*, 1976. ISBN 90 6196 123 8.

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. ISBN 90 6196 092 4.

MCS 19 A. HORDIJK, R. POTHARST & J.Th. RUNNENBURG, *Optimaal stoppen van Markovketens*, 1974. ISBN 90 6196 093 2.

MCS 20 T.M.T. COOLEN, P.W. HEMKER, P.J. VAN DER HOUWEN & E. SLAGT, *ALGOL 60 procedures voor begin- en randwaardeproblemen*, 1976. ISBN 90 6196 094 0.

MCS 21 J.W. DE BAKKER (red.), *Colloquium Programmacorrectheid*, 1975. ISBN 90 6196 103 3.

MCS 22 R. HELMERS, F.H. RUYMGAART, M.C.A. VAN ZUYLEN & J. OOSTERHOFF, *Asymptotische methoden in de toetsingstheorie; Toepassingen van naburigheid*, 1976. ISBN 90 6196 104 1.

MCS 23.1 J.W. DE ROEVER (red.), *Colloquium Onderwerpen uit de biomathematica, deel 1*, 1976. ISBN 90 6196 105 X.

MCS 23.2 J.W. DE ROEVER (red.), *Colloquium Onderwerpen uit de biomathematica, deel 2*, 1976. ISBN 90 6196 115 7.

MCS 24.1 P.J. VAN DER HOUWEN, *Numerieke integratie van differentiaalvergelijkingen, deel 1: Eenstapsmethoden*, 1974. ISBN 90 6196 106 8.

MCS 25 *Colloquium Structuur van programmeertalen*, 1976. ISBN 90 6196 116 5.

MCS 26.1 N.M. TEMME (ed.), *Nonlinear analysis, volume 1*, 1976. ISBN 90 6196 117 3.

MCS 26.2 N.M. TEMME (ed.), *Nonlinear analysis, volume 2*, 1976. ISBN 90 6196 121 1.

MCS 27 M. BAKKER, P.W. HEMKER, P.J. VAN DER HOUWEN, S.J. POLAK & M. VAN VELDHUIZEN, *Colloquium Discretiseringsmethoden*, 1976. ISBN 90 6196 124 6.

MCS 28 O. DIEKMANN, N.M. TEMME (EDS), *Nonlinear Diffusion Problems*, 1976. ISBN 90 6196 126 2.

MCS 29.1 J.C.P. BUS (red.), *Colloquium Numerieke programmatuur, deel 1A, deel 1B*, 1976. ISBN 90 6196 128 9.

MCS 29.2 H.J.J. TE RIELE (red.), *Colloquium Numerieke programmatuur, deel 2*, 1976. ISBN 144 0.

* MCS 30 P. GROENEBOOM, R. HELMERS, J. OOSTERHOFF & R. POTHARST, *Efficiency begrippen in de statistiek*, 1977. ISBN 90 6196 149 1.

MCS 31 J.H. VAN LINT (red.), *Inleiding in de coderingstheorie*, 1976. ISBN 90 6196 136 X.

MCS 32 L. GEURTS (red.), *Colloquium Bedrijfssystemen*, 1976. ISBN 90 6196 137 8.

MCS 33 P.J. VAN DER HOUWEN, *Differentieschema's voor de berekening van waterstanden in zeeën en rivieren*, ISBN 90 6196 138 6.

MCS 34 J. HEMELRIJK, *Oriënterende cursus mathematische statistiek*, ISBN 90 6196 139 4.

* MCS 35 P.J.W. TEN HAGEN (red.), *Colloquium Computer Graphics*, 1977. ISBN 90 6196 142 4.

MCS 36 J.M. AARTS, J. DE VRIES, *Colloquium Topologische Dynamische Systemen*, 1977. ISBN 90 6196 143 2.

De met een * gemerkte uitgaven moeten nog verschijnen.