

Toegepaste wiskunde
op een PC

Algorithmen in QBasic toepassingen

J.H. Kruizinga



Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

SUB KARAKTERISTIEKPOLYNOOM	26
De polynoomrelaties vormen een Sturmrij	27
De stelling van Greshgorin. Gebruik van Intervaldeling en of het Newton-algorithme voor het bepalen van de eigenwaarden	28
SUB EIVALI (SUB EIVALN op de schijf)	28
Eigenvectoren van een matrix als de eigenwaarden bekend zijn	29
Eigenvector-bepaling met inverse substitutie. SUB EIGVEC	29, 31
Het op Hessenbergvorm brengen van een volle matrix	32
De Housholdertransformatie. SUB HESEIVECI (op de schijf)	32
De basisprincipes van de QL- en QR-methoden	33
SUB HESTQL in HESTQL.BAS op de schijf.	
Data oefenvoorbeelden	33
Twee oplossingsmethoden voor het eigenwaardeprobleem $A.X = \mu.B$	34
Met uitsluitend de SUB HESTQL of met SUB CHOLESKYREDUCTIE	35
SUB CHOLESKYREDUCTIE in CHOLTQL.BAS op de schijf.	
Toelichting op het programma ALG343.BAS met de gebruikte Subs: EIGENP, SCALE, HESQR, REALVE en COMVE, voor bepalen van de reële of complexe eigenwaarden en eigenvectoren van een reële matrix	36
Het oplossen van een Hermitisch eigenwaardeprobleem met SUB HESTQL	38
Oefenmateriaal bij dit hoofdstuk	38
Referenties	40
De complete Matrixeigenwaarde en Eigenvectorprogramma's. EWAVECI.BAS, EWAVECN.BAS, TQLHES.BAS, CHOLTQL.BAS en ALG343.BAS zijn op de schijf aanwezig	
5. Kettingbreuken	42
Definities van een kettingbreuk, de partiële teller en noemer, de naderende-breuk. Recursierelaties	43
De determinant-eigenschap	43
Het verband tussen de n -de orde partiële som van een machtsreeks en de n -de orde benadering van de corresponderende kettingbreuk	43
Equivalentie van kettingbreuken	43
Contractie van kettingbreuken	44
Kettingbreuken die van een variabele z afhangen.	
De Hankel-determinant en de Jacobi-identiteit	45
Het Rutishauser Quotient-Difference-algorithme	46
De graad van de partiële teller $P_k(z)$ en noemer $Q_k(z)$	46
De fout in de kettingbreukbenadering bij afbreken na N termen	46

Samenhang tussen de verschillende relaties	47
Speciale kettingbreuken	47
Verwantschap kettingbreuk en asymptotische reeks	47
Het Stieltjes kettingbreuktype	48
Het Carlemann convergentie-criterium	48
Referenties	49
6. QD- en Naderende Breuk Algorithmen	50
QD-implementatie met behulp van dubbelarrays. Het daarop gebaseerde programma voor het uitprinten van de QD verbanden	50
SUB QDANDERS en SUB QD	51
SUB FWAARDE1 en SUB FWAARDE	52
SUB CQD voor het complexe variabelen-type	53
SUB CWAARDEKET.	54
QD-ALG.BAS programma, voor het via de machtreekscoëfficiënten berekenen van de kettingbreukcoëfficiënten (op de schijf)	
7. Benaderingen met Kettingbreuken van de Elementaire Functies	56
Kettingbreukbenaderingen van even en oneven machtreksen	56
Kettingbreukcoëfficiënten in formulevorm	57
De definitie en notatie voor de Gegeneraliseerde Hypergeometrische reeks. Enkele functies in deze voorstelling	57
Eigenschappen van de elementaire functies.	
Toepassing hiervan voor het gebruik op het interval $[2.E-38, 2.E+38]$	58
ELEMSUB.BAS (de in S-Basic gebruikte dubbellengte Subs) (op de schijf)	
8. Simultaan benaderen van de Wortels	60
Aangepaste recursierelaties voor het QD-algorithme en een schema voor het bepalen van de wortels van een n^{de} -graads polynoom	60
Limietstellingen voor het geval van enkelvoudige reële wortels en voor enkelvoudig toegevoegd-complexe wortels	61
Uitgewerkte voorbeelden	62
SUB PROGRSSIEFQD. (NULPUNT.BAS op de schijf)	63
Simultaan bepalen van de eigenwaarden via de karakteristieke vergelijking, uit de met differentietechnieken benaderde differentiaalvergelijking. KARAKTER.BAS op de schijf.	
Referenties	64

9. Enkele Voorbeelden uit de Getallentheorie	65
Het Euclides-algorithme. De kettingbreuk representaties met alle partiële tellers gelijk 1	65, 66
Oplossen van een diofantische vergelijking	66
Periodieke kettingbreuken. De kettingbreukontwikkeling van \sqrt{N} en een oplossing van de vergelijking van Pell	68
GETALQB.BAS op de schijf.	
Referenties	68
10. Enkele Speciale Functies	69
De integraalvoorstelling van de gammafunctie en enkele eigenschappen. FUNCTION FAC	69
De machtreeks $\log \Gamma(1 + z)$. De constante van Euler en de Riemann Zeta functie	69, 70
ZETAGAM.BAS (de Subs: ZETA, QDANDERSFUN, EXPALGORITME, QDANDERS en FUNCTIEWAARDE) op de schijf.	
Uitbreidingen van $\Gamma(1 + X)$, voor $X > 1$ en $X < 0$	70
FUNCTION GAMMA (GAMMA.BAS op de schijf)	71
Definitie van de functie PSI. SUB PSI.	72
Definities van de betafunctie. Het Hankelsymbool	72
Het Pochhammersymbool.	
De errorfunctie en de errorfunctie-complement	72, 73
Referenties	73
11. Algorithmen voor Diverse Besselfuncties	74
Machtreeksvoorstellingen van: $J_0(x)$, $I_0(x)$, $J_1(x)$, $I_1(x)$, $Y_0(x)$ en $Y_1(x)$	74, 75
De asymptotische voorstellingen $P(v, x)$ en $Q(v, x)$	76
SUB BESSELASYMPTOTIEK	77
Alternatieve asymptotische ontwikkelingen (met Modulus en argument). SUB ARCTANGENS en SUB ALTMODULUS	78, 79
Het bepalen van de kettingbreukcoëfficiënten met behulp van de programma's QDBESSEL.BAS en ASYM-ALT.BAS op de schijf.	
Machtreeks-implementaties van $K_0(z)$ en $K_1(z)$	81
KIZMACHT.BAS op de schijf.	
Het bepalen van de kettingbreukcoëfficiënten, met KNPSIFAC.BAS op de schijf	81
Nieuwe aysmptotische implementaties voor: $J_0(y)$, $J_1(y)$, $Y_0(x)$ en $Y_1(x)$ met behulp van $K_0(i.y)$ en $K_1(i.y)$	82
Referenties	83

Basis implementaties van: $I_0(z)$, $I_1(z)$, $K_0(z)$ en $K_1(z)$,
 voor z complex: van de: Subs I0Z, I1Z, K0Z en K1Z,
 in INKNCOMP.BAS op de schijf.

De standaard te gebruiken subs K0Z, K1Z, I0Z, I1Z,
 WAARDEKET, op de schijf in I0I1COMP.BAS.

SUB J0Y0 op de schijf in J0Y0SUB.BAS.

SUB J1Y1 op de schijf in J1Y1SUB.BAS

12. Eigenschappen van Laplace-Transformaties	84
Definities en eigenschappen	84, 85
De convolutiestelling. Asymptotische eigenschappen	85, 86
Tabel van functies en Laplace-getransformeerden	86
Gewone differentiaalvergelijking met rechterlid	86
Laplace-getransformeerde van $J_0(t)$	87, 88
Laplace-getransformeerden van het type Errorfunctie	88, 89
Laplace-getransformeerden van: $\sin(t)/t$, en $\log(t) - \gamma$	90
De lineaire diffusievergelijking op $0 \leq x \leq \infty$ en	90
analoog in cilindercoördinaten, op $0 \leq r \leq \infty$	91
analoog in cilindercoördinaten, op $0 \leq r \leq 1$	93
Referenties	94
13. Numerieke Laplace Transformatie	95
Definities. De inverse transformatie in reële voorstelling	95
De trapeziumregelbenadering en de Fourierreeksbenadering	96
Discretisatie en afbreekfouten	97, 98
Complexe machtreeks-voorstelling van de	
Laplace-getransformeerde	99
Numerieke inversie, via een complexe kettingbreukbenadering	99, 100
De Subs LAPLACEQD en WAARDELAP	100, 101
LAPLACE.BAS (op de schijf)	
Enkele Laplace-getransformeerde partiële diff. vergelijkingen	102, 103
ERR-PART.BAS, I0-PART.BAS, K0K1PART.BAS (op de schijf)	
Referenties	103
14. Enkele Partiële Differentiaalvergelijkingen	104
Typen lineaire partiële differentiaalvergelijkingen	
De één-dimensionale diffusievergelijking met lineaire	
homogene randcondities	104
Oplossen met de separatiemethode	105
De eigenwaarde-vergelijking	106
De eigenwaarden en eigenfuncties	106
Eigenschappen van Sturm-Liouville randwaardeproblemen	107, 108
Differentie-benaderingen	108

Oplossen van een gediscretiseerd randwaardeprobleem	108, 109
Referenties	110
15. Een Besselfunctie Randwaarde Probleem	112
Analytische uitwerking van een één-dimensionaal diffusieprobleem in cylinder coördinaten	112, 117
ANALYBES.BAS (op de schijf)	
Speciale oplossingen	119
16. Discretisatie van de Differentiaalvergelijking	121
Het vorige randwaardeprobleem met differentieaanpak	121
Als sommige coëfficiënten in de randvoorwaarden nul zijn	123
Formulering van de Ultracentrifuge-stoftransport-vergelijking	124
De separatie en discretisatieaanpak	125
Hoe de discretisatieaanpak kan worden gebruikt als de differen- tiaalvergelijking singulariteiten heeft in de randpunten	126
BESSELDF.BAS en ULTRA.BAS op de schijf.	
Referenties	128
17. Een Potentiaal probleem	129
De ladingsverdeling in een geaarde cilindrische tank.	
Een oplossingsmethode voor een Poisson-vergelijking.	129
De oneindig lange cylinder	133
Schattingen voor het aantal mee te nemen termen in de ontwikkeling	134
SUBs ENKELSOM, DUBBELSOM ONEINDIGECYLINDER	135
POT-FUN.BAS op de schijf.	
Referenties	137
Appendix A. Besselfuncties	138
Formuleverzameling van de gebruikte Besselfuncties	138, 141
Referenties	141
Appendix B. Een Alternatieve Eigenvector Bepaling	142
Richardson-Romberg extrapolatie	142
Runge-Kutta 4-de orde integratie-algorithme	143
ULTRICA.BAS en ULTRARUN.BAS op de schijf.	
Referenties	143
Appendix C. De Programma's en Algorithmen	144

Programma's, Subs en Functions ingedeeld in groepen	144, 146
Appendix D. De Functie van Green	147
De Greenfunctie-constructie. Referenties	147, 150
De Laplacetransformatie en Greenfunctie-constructie toe- gepast bij het oplossen van een één-dimensionale diffusie- vergelijking met niet homogene randcondities	150, 151
Uitwerking en implementatie van een numerieke integratie- methode voor de Greenfunctie	151, 157
Een speciale oplossing (om de nauwkeurigheid van de methode te kunnen testen)	150, 157
SUB GREENFUNCTIE	158
GREEN3.BAS op de schijf.	
Subs voor het lezen en schrijven van random files	160
De Green-functie constructie voor een cilindrisch probleem en de numerieke uitwerking	161, 164
SUB BESSELGREENFUNCTIE	164
GREENBES.BAS op de schijf.	
Uitwerking van een speciale oplossing	166, 167
Geïmplementeerde als SUB LAPOPLOSSING	167
Opgenomen als GREENEXACT.BAS op de schijf.	
Een Greenfunctie voor dezelfde differentiaalvergelijking maar andere randcondities en een andere rechterlid-functie	168, 170
De SUBs GREENBESSELFUNCTIE en SUB RECHTERLID, in GREENBES op de schijf.	
Uitwerking van een particuliere test-oplossing	169
SUB LAPOPLOSSING in GREXACT op de schijf.	

Op de schijf zijn de volgende programma's aanwezig:

HORNER	BAS	816	Horner algorithme
NEWTON	BAS	1425	Bepalen reële wortels polynoom
BAIRSTOW	BAS	2512	Bepalen alle wortel polynoom
INTERVAL	BAS	1280	Bepalen wortel (polynoom of functie)
INVERS	BAS	1864	Matrix inversie, oplossen stelsel vergelijkingen
MATRIXFU	BAS	2802	Operaties met machtreekscoëfficiënten
LAGUER	BAS	5942	Wortels polynoom met complexe coëfficiënten
BUHRMAN	BAS	1949	Coëfficiënten inverse machtreeks
EIVALI	SUB	1378	e.w. tri-diagonaal-matrix (intervaldeling)
EIVALN	SUB	1363	e.w. tri-diagonaal-matrix (Newton)
EIGENVEC	SUB	1751	e.v. tri-diagonaal-matrix (inverse substitutie)
EWAVECI	BAS	7341	e.w. & e.v. volle sym. matrix (interval)
EWAVECN	BAS	7282	e.w. & e.v. volle sym. matrix (newton)
HESTQL	BAS	5553	alle e.w. & e.v. volle sym. matrix
CHOLTQL	BAS	6214	alle e.w. & e.v. $A.X = \mu.B.X$
ALG343	BAS	17899	alle e.w. & e.v. volle reële matrix
QDARRAYS	BAS	1310	printen van de QD (kettingbreuk) relaties
QD-ALG	BAS	2961	kettingbreuk QD & functiewaarde bepaling
QDANDERS	BAS	2687	kettingbreuk QD & functiewaarde bepaling
ELEMSUB	BAS	7491	dubbellenge elementaire subs
NULPUNT	BAS	2373	simultaan wortels van polynoom (progr. QD)
KARAKTER	BAS	4815	als nulpunt.bas (gediscretiseerde dif. verg.)
GETALQB	BAS	7507	toepassingen getaltheorie
ZETAGAM	BAS	5452	met de coëff. zetafunctie coëff. gammafunctie
GAMMA	BAS	1808	gammafunctie
FAC-PSI	BAS	754	faculteit en de PSI-functie (N geheel)
QDBESSEL	BAS	4563	kett. coëff. $J_v(x), Y_v(x) \quad v = 1, 2$
ASYM-ALT	BAS	6734	als QDBESSEL nu met modulus en argument
KNPSIFAC	BAS	2694	kett. coëff. $I_v(x), K_v(x) \quad v = 1, 2$
KIZMACHT	BAS	5051	$I_v(z), K_v(z) \quad v = 1, 2. \quad z = x + i.y$
INKNCOMP	BAS	9151	basis principe subs $I_v(z), K_v(z) \quad v = 1, 2$
IO1K0K1	BAS	14907	gebruiksklare subs $I_v(x), K_v(x) \quad v = 1, 2$
JOY0SUB	BAS	2759	combinatie sub $J_0(x), Y_0(x)$
J1Y1SUB	BAS	2768	combinatie sub $J_1(x), Y_1(x)$
JOX	BAS	1979	functie $J_0(x)$
J1X	BAS	1997	functie $J_1(x)$
LAPLACE	BAS	9938	Numerieke Laplace-inversie (11 functies)
ERR-PART	BAS	4452	lineaire partiële diff. N.L.I
IO-PART	BAS	9391	cirkel symm. part. diff. $[r_0, 1]$ N.L.I
K0K1PART	BAS	10192	cirkel symm. part. diff. $[r_0, \infty)$ N.L.I
ANALYBES	BAS	11321	Analytische opl. cyl. symm. part. diff.
BESSELD	BAS	12992	discretisatie opl. cyl. symm. part. diff.
ULTRA	BAS	8009	ultracentrifge.opl. part. diff. (discret.)
POT-FUN	BAS	7579	potentiaal verdeling in geaarde tank
ULTRICHA	BAS	4401	e.w. ultracentr. (Richardson-extrpolatie)
ULTRARUN	BAS	3577	e.v. ultracentr. bepaald met Runge-Kutta
GREEN3	BAS	10021	Greenfunctie lin. part. diff.
GREEN2	SUB	2887	Greenfunctie met andere interpolatie aanpak
GREENBES	BAS	24968	Greenfunctie voor cirkel symm. part. diff.
GREXACT	BAS	20176	Speciale exacte test opl. zonder Greenfunctie
GREENBE	SUB	4493	Greenfun. cirkel symm. andere randcondities
GREXACT	SUB	1538	special test opl. voor ander randcondities

52 file(s)

303067 bytes

Inleiding en verantwoording

In een tijdsbestek van één decennium heeft de computer, die we nu op ons bureau hebben staan, een rekencapaciteit gekregen, die vergelijkbaar of groter is dan die van de meeste Mainframes van 10 jaar geleden. In die periode is er een formidabele hoeveelheid administratieve, bedrijfskundige en recreatieve software ontwikkeld. Een merkwaardig feit is echter, dat van de wiskundige algorithmen, aanwezig in de subroutine libraries van Mainframes, zo weinig voor PC gebruikers beschikbaar zijn gesteld.

Voor Fortran en Pascal gebruikers is er inmiddels "Numerical Recipes. The Art of Scientific computing"¹⁾, echter voor de veel grotere groep Basic gebruikers is er voor zover mij bekend niet iets dergelijks op de markt verschenen. Om hierin verbetering te brengen heb ik besloten de algorithmen die ik voor eigen gebruik heb geïmplementeerd algemeen beschikbaar te maken. De lineaire algebra had ik vroeger al geïmplementeerd in het Mbasic dialect van m'n eerste CP/M Kaypro machine, om thuis oefenmateriaal bij een college "Industriële wiskunde" te kunnen voorbereiden. Later heb ik deze verzameling sterk uitgebreid en is alles herschreven in QBasic 4.X.

Er is voor QBasic²⁾ gekozen omdat deze taal volwassen is geworden, over een goede compiler en library manager beschikt, zodat men naar wens de Subs en Functions, in een object library kan onderbrengen. Om te voorkomen dat de algorithmen alleen worden gebruikt als zwarte dozen, zijn steeds de wiskundige en numerieke achtergronden kort toegelicht. Bewijzen worden in 't algemeen niet gegeven, geïnteresseerde gebruikers die over voldoende kennis van algebra, analyse en differentiaalvergelijkingen beschikken, zullen via de toelichtingen en verwijzingen naar een beperkt aantal standaard leerboeken, deze gemakkelijk kunnen vinden.

In de eerste helft van het boek ligt de nadruk op het ontwikkelen van de algorithmen. In de latere hoofdstukken worden de algorithmen gebruikt, om er een aantal in wetenschap en techniek veel voorkomende partiële differentiaalvergelijkingen, mee op te lossen. Eerst op de klassieke manier en later met geheel nieuw ontwikkelde methoden, op basis van Numerieke-Laplace-inversie en Greenfunctie constructies. Het voorliggende is dus niet een standaard toegepaste wiskunde leerboek, maar heeft als doel de kennis van toegepaste wiskunde te vergroten en het gebruik er van op een PC te stimuleren.

Voor wat betreft de Numeriek wiskundige achtergronden, wordt uitsluitend verwezen naar de uitstekende goedkope studenten-uitgaven: *Einführung in die Numerische Mathematik* I, II van Stoer en Bulirsch en voor het gebruik van Speciale Functies naar *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, edited by Milton Abramowitz and Irene Stegun. Een recente Nederlandstalige inleiding is *Speciale functies in de Mathematische Fysica*, van N.M. Temme. Bij de uit Algol en Fortran bewerkte algorithmen zijn steeds de oorspronkelijke literatuurbronnen vermeld.

Omdat bij de implementaties veel gebruik wordt gemaakt van kettingbreuken (o.a. van het Rutishauser QD algoritme), worden de gebruikte ket-

tingbreuk eigenschappen eerst kort samengevat.

Bij de toepassingen in de latere hoofdstukken, zoals Numerieke-Laplace-transformaties, wordt verwezen naar de oorspronkelijke publicaties en bij de partiële differentiaalvergelijkingen en de Greenfuncties naar bekende leerboeken.

Het voorliggende bevat geen handleiding voor het gebruik van de Microsoft²⁾ QBasic-taal, er wordt verondersteld dat de gebruiker hiermee voldoende vertrouwd is.

Teneinde de omvang van het boek te beperken, zijn op verzoek van de uitgever, alle in de oorspronkelijk tekst opgenomen complete programma's en de meeste volumineuze algorithmen, verhuisd naar een aparte diskette. Hierbij moest een keuze worden gemaakt, óf deze programmatuur in één (300 kbyte) grote (ASCII) file opnemen, óf ieder programma in een eigen aparte file plaatsen. Het voordeel van één grote file is dat deze gemakkelijk kan worden uitgeprint en de informatie overzichtelijk (met pagina verwijzing), beschikbaar komt. Vanwege het grote nadeel dat alleen de bezitters van een geschikte tekst-editor hieruit de programma's voor direct gebruik op de computer kunnen selecteren, is er gekozen voor direct te gebruiken files.

De schijf kan rechtstreeks bij de auteur besteld worden; correspondentie-adres: Travertin 11; 8084 EG 't Harde.

Samenvatting van de inhoud van de diverse hoofdstukken

In het eerste hoofdstuk is een aantal bekende en minder bekende kleine programma's verzameld, welke later geregeld worden gebruikt, zoals: Het Horner algoritme, het Newton-Horner algoritme voor het bepalen van de reële wortels van polynomen, de Bairstow methode voor het bepalen van alle wortels (de complexe inbegrepen) van een polynoom met reële coëfficiënten, de intervalhalverings methode voor het bepalen van de nulpunten van polynomen en transcendente functies en een Gauss-Jordan algoritme voor het oplossen van een stelsel lineaire vergelijkingen en het bepalen van de inverse matrix.

In Hoofdstuk 2 worden de complexe-variabelen-eigenschappen, welke bij een groot aantal toepassingen worden gebruikt, eerst kort samengevat. Vervolgens worden de daarop gebaseerde elementaire: SUBS en FUNCTIONS, voor: optellen, aftrekken, vermenigvuldigen, delen, $\exp(z)$, $\log(z)$, $\sinh(z)$, $\cosh(z)$, $\tan(z)$, $\arctan(z)$, \sqrt{z} en $\text{cabs}(z)$ geïmplementeerd. Als eerste toepassing van deze Subs, is de zeer robuuste Laguerre methode voor het bepalen van de wortels van polynomen met complexe coëfficiënten geïmplementeerd.

In Hoofdstuk 3 worden de volgende operaties met machtreeksen, waarvan de coëfficiënten zijn gegeven, geïmplementeerd (ze worden later gebruikt bij deimplementatie's van zeta-, gamma- en enkele Besselfuncties):

- Vermenigvuldiging van twee machtreeksen.
- Delen van een machtreeks door een andere machtreeks.
- Een machtreeks als exponent in een e -macht.

Als een alternatief voor deze toepassingen, is ook een algoritme voor het genereren van de coëfficiënten in een Burhrmann-Lagrange reeksontwikkeling geïmplementeerd.

In Hoofdstuk 4 zijn diverse matrix eigenwaarde- en eigenfunctie-algorithmen behandeld, o.a. via het genereren van de coëfficiënten van het karakteristieke polynoom van een tri-diagonaalmatrix.

De eigenvector bepaling met behulp van inversie iteratie.

Het op Hessenberg vorm brengen van een volle reële matrix met de Housholder transformatie. Het TQL algoritme.

Oplossen van het matrixeigenwaarde probleem: $A.x = \mu.B.x$.

Het bepalen van de reële en complexe eigenwaarden en eigenvectoren van een niet symmetrisch reëel matrixeigenwaarde probleem.

Aanwijzingen hoe Hermitische eigenwaarde problemen kunnen worden opgelost met behulp van het TQL-algoritme.

Hoofdstuk 5 bevat een korte samenvatting van de meest belangrijke eigenschappen van Kettingbreuken, met als hoofddoel de behandeling van het door Rutishauser gevonden Quotient Difference algoritme. Het basis-materiaal hiervoor is gekozen uit: Applied and Computational Analysis, van Peter Henrici.

In Hoofdstuk 6 worden verschillende implementatie's van het QD-algoritme voor reële machtrekcoëfficiënten en de algoritmen voor het bepalen van de Functiewaarde van een kettingbreuk behandeld. Tenslotte één implementatie voor het complexe-variabelen-type, welke de basis is voor de latere Numerieke Laplace-inversie-methode.

In Hoofdstuk 7 wordt geïllustreerd hoe dubbellenge Subs van de elementaire functies met kettingbreuken kunnen worden benaderd (oorspronkelijk gebruikt in het Kaypro S-Basic dialect).

In Hoofdstuk 8 wordt getoond hoe met een aangepast QD-algoritme simultaan, de wortels van een N^{de} -graads polynoom zijn te benaderen. Verder wordt behandeld, hoe van een met differentie technieken benaderde differentiaalvergelijking (randwaardeprobleem), via het karakteristieke polynoom, simultaan de eigenwaarden kunnen worden benaderd.

In Hoofdstuk 9 zijn enkele klassieke kettingbreuk toepassingen, zoals het bepalen van de GGD, het oplossen van Diofantische vergelijkingen en de vergelijking van Pell, geïmplementeerd.

In Hoofdstuk 10 worden enkele eigenschappen van de Riemann Zetafunctie gebruikt, teneinde met behulp van kettingbreuken de constante van Euler en de Gammafunctie te kunnen implementeren.

Verder nog definities van de later te gebruiken functies: De ψ -functie, de

Betafunctie de Errorfunctie en de Pochhammer en Hankel symbolen.

In Hoofdstuk 11 zijn op basis van de in Appendix A gegeven formuleverzameling van Besselfuncties, bekende en nieuw ontwikkelde algorithmen van de Besselfuncties: $J_0(z)$, $J_1(z)$, $Y_0(z)$, $Y_1(z)$, $I_0(z)$, $I_1(z)$, $K_0(z)$, $K_1(z)$ voor zowel z reëel als complex, uitgewerkt en geïmplementeerd.

In Hoofdstuk 12 beginnen de uitgebreide toepassingen. Eerst worden enkele eigenschappen van Laplace-transformaties en het gebruik hiervan bij het oplossen van een aantal gewone en partiële differentiaalvergelijkingen besproken.

In Hoofdstuk 13 wordt de achtergrond van de Numerieke Laplace-inversietechniek, op basis van de complexe-kettingbreuken-methode behandeld. Met behulp van de geïmplementeerde algorithmen worden alle in Hoofdstuk 11 vermelde voorbeelden opgelost.

In Hoofdstuk 14 wordt in een korte samenvatting behandeld hoe met behulp van de separatie-methode, enkele typen Parabolische differentiaalvergelijkingen kunnen worden opgelost. Tevens worden de belangrijkste eigenschappen van Sturm-Liouville randwaarde-problemen vermeld. Met behulp van deze hulpmiddelen wordt, van een diffusieprobleem in een cirkel symmetrisch coördinaten systeem, de analytische oplossing voor verschillende typen homogene randcondities, met behulp van Besselfuncties de algemene oplossing uitgewerkt.

In Hoofdstuk 15 wordt toegelicht hoe een randwaarde-probleem met behulp van differentie benaderingen kan worden getransformeerd in een matrix eigenwaarde probleem en hoe de oplossing met de methoden besproken in Hoofdstuk 4, is te bepalen.

Eerst wordt de implementatie van het in het vorige hoofdstuk analytisch opgeloste randwaarde-probleem behandeld en vervolgens wordt nog een meer gecompliceerd randwaarde-probleem besproken en geïmplementeerd.

In Hoofdstuk 16 wordt met behulp van de separatie-methode, van een potentiaalprobleem, in de vorm van een Poisson vergelijking, de oplossing besproken (verkapte Greenfunctie constructie) en geïmplementeerd.

Appendix A bevat de complete formule-verzameling van alle in dit boek gebruikte eigenschappen van Besselfuncties.

In Appendix B worden eerst de eigenwaarden van het in hoofdstuk 15 behandelde probleem met behulp van de Richardson extrapolatie-methode zeer nauwkeurig bepaald, waarna met behulp van een implementatie van een Runge-Kutta algoritme, de bijbehorende eigenvectoren worden berekend.

In Appendix C geeft een indeling van de behandelde complete programma's en van de geïmplementeerde algorithmen.

In Appendix D wordt de achtergrond van de 1-dimensionale Greenfunctie behandeld. Als voorbeeld wordt een partiële differentiaalvergelijking met behulp van een Laplace-transformatie eerst in de vorm van een niet-homogeen randwaarde-probleem gebracht. Vervolgens wordt een Greenfunctie geconstrueerd en een numerieke integratiemethode uitgewerkt. Uitwerking en implementatie, met behulp van Greenfuncties, van het in Hoofdstuk 12 gebruikte diffusieprobleem in een cirkel symmetrisch coördinaten systeem, voor verschillende randcondities.

Voor het testen van de nauwkeurigheid van de numerieke methode, zijn voor twee rechterlidfuncties, de analytisch oplossingen uitgewerkt en geïmplementeerd.

- 1) Numerical Recipes. The Art of Scientific Computing.
William H. Press Brian P. Flannery Saul A. Teukolosky William T. Vetterling.
- 2) Microsoft QuickBasic 4.0, 4.1, 4.5.
Voor professionele technische of wetenschappelijke toepassingen, is het veel krachtigere Microsoft Fortran 5.0, vanwege het complexe datatype en de statements voor matrix operaties, beter geschikt. (QB Objectfiles van de Subs, zijn aan te roepen in Fortan 5.0)

1. Wortels van polynomen en Lineaire vergelijkingen

In dit eerste hoofdstuk wordt een aantal deel-programma's ontwikkeld, die we later zullen gebruiken bij het oplossen van matrix-eigenwaarde-problemen. Hiertoe behoren o.a. de verschillende algorithmen voor het bepalen van de wortels van een N^{de} -graads polynoom en het oplossen van stelsels lineaire vergelijkingen.

We beginnen met het Horner algorithm. Het N^{de} -graads polynoom:

$$P_N(X) = a_0 \cdot X^N + a_1 \cdot X^{N-1} + \dots + a_{N-1} \cdot X + a_N ,$$

laat zich ook schrijven als:

$$P_N(X) = (\dots(((a_0 \cdot X + a_1) \cdot X + a_2) \cdot X + \dots + a_{N-1}) \cdot X + a_N) .$$

Door te stellen:

$$b_0 = a_0 \quad \text{en} \quad b_k = b_{k-1} \cdot X + a_k \quad (k = 1, 2, \dots, N) ,$$

wordt een formulering voor het Horner algorithm verkregen. Vormen we een nieuw polynoom met de juist gevonden coëfficiënten: b_k ($k = 0, 1, \dots, N$), dan laat zich het proces als volgt voortzetten:

$$c_0 = a_0 = b_0 \quad c_k = c_{k-1} \cdot X + b_k \quad (k = 1, 2, \dots, N) ,$$

geschreven als polynoom in de oorspronkelijke coëfficiënten a_k :

$$\begin{aligned} C_k(X) &= a_0 \cdot X^k + (a_0 \cdot X + a_1) \cdot X^{k-1} \\ &\quad + (a_0 \cdot X^2 + a_1 \cdot X + a_2) \cdot X^{k-2} \\ &\quad + \dots + a_k = (k+1) \cdot a_0 \cdot X^k + k \cdot a_1 \cdot X^{k-1} + \dots + a_k . \end{aligned}$$

Voor $k = N - 1$, vinden we hieruit: $C_{N-1}(X) = P'_N(X)$ de afgeleide van $P_N(X)$.

Ook op het polynoom $C_{N-1}(X)$ ($k = 0, 1, \dots, N$) laat dit proces zich weer toepassen en dus kunnen ook de hogere afgeleiden worden bepaald.

Het bovenstaande is eenvoudig te generaliseren met behulp van een dubbel array $A_{N,N}$, met de coëfficiënten van $P_N(X)$ in de nulde rij. Het Horner algorithm schema laat zich dan formuleren als:

$$\begin{aligned} A_{k+1,0} &= A_{0,0} \quad (k = 0, 1, \dots, N) \\ A_{k+1,j} &= A_{k+1,j-1} + A_{k,j} \quad (j = 0, 1, \dots, N) . \end{aligned}$$

Met behulp van een Taylorontwikkeling, kunnen we een polynoom $P_N(X)$

rangschikken naar machten van b.v. $X - H$. Het laat zich hiermee bewijzen, dat de nevendiagonaal elementen $A_{j+1,N-j}$ in de matrix A , de coëfficiënten

$$q_{N-j} = A_{j+1,N-j} = P^{(j)}(H)/j! \quad (j = 0, 1, \dots, N),$$

in het gerangschikte polynoom:

$$Q_N(X) = q_0 \cdot X^N + q_1 \cdot X^{N-1} + \dots + q_{N-1} \cdot X + q_N$$

voorstellen.

Hiermede is een efficiënte methode gevonden om de wortels van een polynoom $P_N(X)$ met een bepaald bedrag te vermeerderen of te verminderen.

Het bovenstaande algoritme is als volgt uitgewerkt:

```
DEFINT I - N: DEFDBL A - H, O - Z 'HORNER.BAS
INPUT "Graadpolynoom = "; N :DIM A(N + 1, N + 1), B(N + 1): CLS
INPUT "Vermeerder de wortels met H = "; H:PRINT
FOR K = 0 TO N: PRINT "A(0";K;"")=" ;; INPUT A(0, K): NEXT K
PRINT
FOR K = 0 TO N: PRINT A(0, K); : NEXT K: PRINT
FOR K = 0 TO N: A(K + 1, 0) = A(0, 0)
FOR J = 1 TO N: A(K + 1, J) = H * A(K + 1, J - 1) + A(K, J): NEXT J
NEXT K: B(0) = A(0, 0)
FOR J = 0 TO N - 1: B(N - J) = A(J + 1, N - J): NEXT J
FOR J = 0 TO N: PRINT B(J); : NEXT J: PRINT :END
```

Voorbeelden.

$$P_5(X) = 3X^5 + 5X^4 - 17X^3 + 0X^2 - 8X + 1 \text{ levert voor } H = 2:$$

$$\begin{array}{cccccc} 3 & 35 & 143 & 258 & 188 & 25 \end{array} .$$

$$P_5(X) = 1X^5 + 0X^4 + X^3 + 19X^2 + 3X + 31 \text{ levert voor } H = 2:$$

$$\begin{array}{cccccc} 1 & 10 & 41 & 105 & 171 & 153 \end{array} .$$

Dit algoritme verleent soms ook goede diensten bij het simultaan bepalen van alle wortels van een N^{de} -graads polynoom, met behulp van het later bij de kettingbreuken te behandelen "Progressive QD" algoritme. Dit algoritme is nl. alleen te gebruiken als in het polynoom alle coëfficiënten $\neq 0$ zijn. Indien er één of meer coëfficiënten nul zijn, vermeerder (of verminder) dan eerst de wortels met een geschikte gekozen waarde, zodat een polynoom met alle coëfficiënten $\neq 0$ ontstaat.

Een andere nuttige toepassing is het NEWTON-HORNER-algoritme voor het benaderen van de wortels van een hogere graads vergelijking¹⁾.

$$F(X) = A_0 \cdot X^n + A_1 \cdot X^{n-1} + \dots + A_n = 0 .$$

Deze methode berust op een Taylorontwikkeling van $F(X)$ in de omgeving van een wortel $X = X_0$:

$$F(X) = F(X_0) + (X - X_0) \cdot F'(X_0) + \dots + (X - X_0)^n \cdot F^{(n)}(X_0) = 0 .$$

Bij verwaarlozing van alle termen hoger dan van de eerste graad geldt:

$$0 = F(X_0) + (X - X_0) \cdot F'(X_0) .$$

Ligt X_0 voldoende dicht bij een wortel, dan volgt hieruit een nieuwe benadering:

$$X_1 = X_0 - F(X_0)/F'(X_0) .$$

Het iteratieproces is dus te formuleren als:

$$X_{i+1} = R(X_i) \quad \text{met} \quad R(X) = X - F(X)/F'(X) .$$

Dit iteratieproces is geïmplementeerd in de SUB NEWTON waarmee alle reële wortels van een polynoom met reële coëfficiënten zeer nauwkeurig kunnen worden bepaald.

```

DEFINT I - N: DEFDBL A - H, O - Z
SUB NEWTON (N, A(), C(), Z1)
DIM B(N + 1), D(N + 1)
DO: PRINT "Voer in een schatting voor de wortel = Z "; :
INPUT Z: Z1 = Z: C(0) = A(0): D(0) = B(0): J = 0: PRINT
DO: FOR L = 1 TO N: C(L) = Z1 * C(L - 1) + A(L): NEXT L
FOR L = 1 TO N - 1: D(L) = Z1 * D(L - 1) + B(L): NEXT L
Z2 = -C(N) / D(N - 1): Z1 = Z1 + Z2: J = J + 1
LOOP WHILE ABS(C(N)) > 5E - 16 AND J < 25
PRINT "Z1 = "; Z1, "F(Z1) = "; C(N): PRINT: PRINT
PRINT "De volgende wortel uit oorsponkelijk polynoom bepalen?";:
INPUT " J/N "; J1$: PRINT
LOOP WHILE UCASE$(J1$) = "J"
END SUB

```

Het complete programma NEWTON.BAS (aanwezig op de schijf) vraagt om de volgende gegevens.

Invoer: Orde polynoom N .
 Coëfficiënten $A(0), A(1), \dots, A(N)$. $A(0)$ coëff. hoogste graad.
 Schatting van een te benaderen wortel X_0 .

Uitvoer: Benaderde wortel.

Het programma vraagt vervolgens of de volgende wortel bepaald moet worden uit de oorspronkelijke vergelijking, of na het uitdelen van de gevonden wortel, uit de gereduceerde vergelijking.

Voorbeeld.

$$X^5 - 25X^4 + 200X^3 - 600X^2 + 600X - 120 = 0 \quad (5 \text{ pos. wortels}) .$$

Eventuele complexe wortels worden niet gevonden. Ook zijn goede schattingen van de wortels vaak niet bekend. Het is dus noodzakelijk om nog andere methoden beschikbaar te hebben.

Met de methode van BAIRSTOW, kunnen van een N^{de} - graads polynoom $F(X)$, met reële coëfficiënten, alle wortels (de toegevoegd complexe inbegrepen) worden bepaald. De werkwijze is als volgt:

Genereer een kwadratisch polynoom $G(X) = X^2 - P.X - Q = 0$ met nog nader te bepalen coëfficiënten P en Q , zodanig dat de wortels van $G(X) = 0$ tevens wortels van $F(X) = 0$ zijn. Voor willekeurige P en $Q \neq 0$ geldt dat $F(X)$ is te schrijven als:

$$F(X) = F1(X).(X^2 - P.X - Q) + A.X + B = 0 .$$

Hierin is $F1(X)$ een polynoom van de graad $N - 2$. In deze splitsing zijn de coëfficiënten van het polynoom $F1(X)$ en de coëfficiënten A en B in de lineaire restterm, functies van P en Q . Taylor ontwikkeling van $A = A(P, Q)$ en $B = B(P, Q)$ naar P en Q levert na linearisering een stelsel lineaire vergelijkingen op. Op iteratieve wijze worden hieruit een P - en een Q -waarde bepaald, welke $A(P, Q) = 0$ en $B(P, Q) = 0$ maken. Voor deze P en Q is de lineaire restterm $A.X + B = 0$. Dus een paar reële of een paar toegevoegd complexe wortels van $G(X) = 0$ zijn tevens wortels van het polynoom $F(X) = 0$.

Een volgende paar wortels van $F(X) = 0$ vinden we door de aanpak te gaan herhalen op het polynoom $F1(X) = 0$. Dit proces op de nieuwe $F1(X) = 0$ steeds herhalen tot alle wortels van $F(X) = 0$ zijn gevonden.

Het bovenstaande is uitgewerkt als SUB BAIRSTOW.

```
DEFINT I - N: DEFDBL A - H, O - Z
SUB BAIRSTOW (N, A(), WR(), WI(), P, Q, SUM)
DIM B(N + 1), C(N + 1)
M = 1: EPS = 5E - 15: B(0) = 1#: C(0) = 1#: L = 1
DO
  DO: B(1) = A(1) - P: C(1) = B(1) - P: FOR K = 2 TO N
    B(K) = A(K) - P * B(K - 1) - Q * B(K - 2)
    C(K) = B(K) - P * C(K - 1) - Q * C(K - 2): NEXT K: M = M + 1
```

```

CB = C(N - 1) - B(N - 1)
DE = C(N - 2) * C(N - 2) - CB * C(N - 3)
DP = (B(N - 1) * C(N - 2) - B(N) * C(N - 3)) / DE
DQ = (B(N) * C(N - 2) - B(N - 1) * CB) / DE
P = P + DP: Q = Q + DQ: SUM = ABS(DP) + ABS(DQ)
LOOP WHILE SUM > EPS AND M < 50
IF SUM > .05# THEN PRINT "Niet convergent de som ="; SUM: EXIT SUB
END IF
DO
PRINT "M="; M; "Som="; SUM
PRINT "P="; P; "Q="; Q: PRINT
D = P * P / 4# - Q: R = ABS(D): R = SQR(R)
IF D ≥ 0 THEN
WR(L) = -P / 2# + R: WR(L + 1) = -P / 2# - R
ELSE
WR(L) = -P / 2#: WI(L) = R
WR(L + 1) = -P / 2#: WI(L + 1) = -R
END IF
N = N - 2: L = L + 2
IF N > 2 THEN
FOR K = 1 TO N: A(K) = B(K): NEXT K
"Nieuw af te splitsen polynoom is: X^2 + P1 * X + Q1 = 0 ."
M = 1: INPUT "P1="; P1: INPUT "Q1="; Q1: P = P1: Q = Q1
EXIT DO
ELSEIF N = 1 THEN
WR(L) = -B(1): WI(L) = 0: EXIT SUB
ELSEIF N = 2 THEN
P = B(1): Q = B(2)
ELSEIF N ≤ 0 THEN EXIT SUB
END IF
LOOP UNTIL N ≤ 0
LOOP
END SUB

```

Het complete programma BAIRSTOW.BAS is aanwezig op de schijf.

Invoer: Orde polynoom N .
 Coëfficiënten $A(0), A(1), \dots, A(N)$.
 Schattingen voor P en Q .

Uitvoer: Als proces convergeert, benaderingen voor alle wortels.

Zijn niet alle wortels in de eerste stap gevonden, dan worden er nieuwe

schattingen voor P en Q gevraagd en wordt het proces herhaald.

Voorbeeld.

$$X^4 - 4X^3 + 8X^2 + 24X + 36 = 0$$

levert de wortels:

$$X(1) = -1 + i.1 : X(2) = -1 - i.1$$

$$X(3) = +3 + i.3 : X(4) = +3 - i.3 .$$

Wanneer men de nulpunten van gecompliceerde functies moet bepalen, is het vaak lastig om de afgeleide van de functie te implementeren. Met een intervalhalverings-methode, is het dan ten koste van wat langere rekentijd mogelijk, met alleen functiewaarde-bepalingen, alle reële nulpunten zeer nauwkeurig te bepalen. De aanpak is als volgt:

Bepaal van de continue functie $F(x)$ op het interval $[a, b]$ (a en b geen wortels), binnen $[a, b]$ een interval $x_1 \leq x \leq x_2$, zodanig dat $F(x_1)$ en $F(x_2)$ van teken verschillen. Bepaal $x_0 = (x_1 + x_2)/2$ (het midden van $[x_1, x_2]$) en vervolgens $F(x_0)$. Voor $|F(x_0)| > \text{eps}$ (eps de gekozen nauwkeurigheidsgrens), wordt er getest in welke van de beide deelintervallen $[x_1, x_0]$, $[x_0, x_2]$, de tekenwisseling van $F(x_0)$ plaats vindt. Op dit gevonden kleinere interval, wordt de procedure herhaald tot er een x_0 is gevonden met $|F(x_0)| < \text{eps}$. De methode is als volgt opgezet.

Invoer: Orde polynoom N .

Coëfficiënten $A(0), A(1), \dots, A(N)$.

Intervalgrenzen Z_1 en Z_2 waarvoor $F(Z)$ van teken gaat wisselen.

Uitvoer: Benaderde wortel Z_0 .

```

DEFINT I - N: DEFDBL A - H, O - Z
SUB INTERVAL (N, Z0, F00, A(), C())
  EPS = 5.E - 14
DO: PRINT "Schatting ondergrens wortel Z1=" ; : INPUT Z1
  PRINT "Schatting bovengrens wortel Z2=" ; : INPUT Z2
  F10 = POLYNOOM (N, Z1, A(), C())
  PRINT " Z1 ";Z1 ; " F(Z1)
  F20 = POLYNOOM (N, Z2, A(), C())
  PRINT " Z2 ";Z2 ; " F(Z2)
LOOP UNTIL SGN(F10) <> SGN(F20)
WHILE ABS(Z2 - Z1) > EPS
  Z0 = (Z1 + Z2) / 2#: F00 = POLYNOOM (N, Z0, A(0), C())
  IF SGN(F00) = SGN(F10) THEN Z1 = Z0 ELSE Z2 = Z0

```

WEND
END SUB

Het volgende voorbeeld illustreert dat het vinden van intervalgrenzen waarvoor de functiewaarde van teken wisselt, zelfs bij een derdegraads polynoom uiterst lastig kan zijn:

$$X^3 + 11X^2 - 102X + 181 = 0 .$$

Dat de negatieve wortel binnen $[-20, 0]$ ligt, is gemakkelijk te bepalen. Het vinden van intervallen voor de positieve wortels, is veel lastiger. Ze liggen nl. binnen het interval $[3.21\dots, 3.22]$.

Later, bij het bepalen van de eigenwaarden van een matrix, wordt daarom deze methode met een intervaldelings algoritme uitgebreid, dat is gebaseerd op de constructie van een rij Sturmfuncties.

Alle tot nu toe behandelde methoden zijn beperkt tot polynomen met reële coëfficiënten, de uitbreiding naar polynomen met complexe coëfficiënten moeten we uitstellen. Dit onderwerp wordt in het volgende hoofdstuk behandeld, omdat eerst de nodige hulpmiddelen voor het werken met complexe variabelen ontwikkeld moeten worden.

Oplossen Stelsels Lineaire vergelijkingen en Matrix inversie.

De Gauss eliminatie-methode voor het oplossen van stelsels lineaire vergelijkingen berust op de volgende bekende matrix-operaties:

Rijen verwisselen (pivot strategie: ook rij met grootste kop-element).

Vermenigvuldigen (delen) van een rij met (door) een constante $\neq 0$ en vervolgens optellen bij (aftrekken van) een andere rij. Deze methode wordt in alle leerboeken uitvoerig beschreven en behoeft daarom geen verdere toelichting. Met een kleine uitbreiding kan men met de Gauss eliminatie-methode gelijktijdig de inverse van de coëfficiënt-matrix berekenen. Dit is uitgewerkt in onderstaande SUB GAUSSJORDAN.

Het complete programma INVERS.BAS is op de schijf aanwezig.

Invoer: $N =$ Orde matrix.
Rij gewijze de Coëfficiënt-matrix A .
Coëfficiënten rechterlid-vector D .

Uitvoer: De Inverse matrix in A en de Oplossingsvector in D .

DEFINT $I - N$: DEFDBL $A - H, O - Z$

```

SUB GAUSSJORDAN (N, A(), D())
DIM B(N, 2 * N + 1)
FOR I = 1 TO N: FOR J = 1 TO N + 1: B(I, J + N) = 0
B(I, J) = A(I, J): NEXT J: B(I, N + I) = 1#: NEXT I
FOR I = 1 TO N: B(I, 2 * N + 1) = D(I): NEXT I
FOR K = 1 TO N
IF K <> N THEN
M = K: FOR I = K + 1 TO N
IF ABS(B(I, K)) > ABS(B(M, K)) THEN M = I: NEXT I
IF M <> K THEN
FOR J = K TO 2 * N + 1
B = B(K, J): B(K, J) = B(M, J): B(M, J) = B: NEXT J
END IF
FOR J = K + 1 TO 2 * N + 1
B(K, J) = B(K, J) / B(K, K): NEXT J
IF K <> 1 THEN
FOR I = 1 TO K - 1: FOR J = K + 1 TO 2 * N + 1
B(I, J) = B(I, J) - B(I, K) * B(K, J): NEXT J, I
IF K = N THEN GOTO A50
END IF
FOR I = K + 1 TO N: FOR J = K + 1 TO 2 * N + 1
B(I, J) = B(I, J) - B(I, K) * B(K, J)
NEXT J, I, K
A50: FOR I = 1 TO N: FOR J = 1 TO N
A(I, J) = B(I, J + N): NEXT J: NEXT I
FOR J = 1 TO N: D(J) = B(J, 2 * N + 1): NEXT J
END SUB

```

Voorbeeld. $AX = b$ met:

$$b^T = [23, 32, 33, 31], \quad X^T = [1, 1, 1, 1],$$

en

$$A = \begin{vmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{vmatrix} \quad A^{-1} = \begin{vmatrix} 68 & -41 & -17 & 10 \\ -41 & 25 & 10 & -6 \\ -17 & 10 & 5 & -3 \\ 10 & -6 & -3 & 2 \end{vmatrix}.$$

Referentie.

1) Josef Stoer. Einführung in die Numerische Mathematik.

2. Complexe Getallen

Een groot aantal latere toepassingen kan zonder het gebruik van complexe getallen niet worden opgelost. Omdat de QB-taal het complexe variabele datatype niet kent, moeten we eerst zelf de nodige hulpmiddelen daarvoor ontwikkelen en implementeren.

We veronderstellen de volgende eigenschappen van de complexe getallen, die we geregeld zullen gebruiken, als bekend.

$z = x + iy$ (x en y reëel en $i =$ de imaginaire grootheid, $i^2 = -1$).
 $x = \operatorname{Re} z$ (Reële deel van z) en $y = \operatorname{Im} z$ (Imaginaire deel van z).

Met $z_1 = x_1 + iy_1$ en $z_2 = x_2 + iy_2$ gelden de eigenschappen:

$$\begin{aligned} z_1 + z_2 &= (x_1 + x_2) + i(y_1 + y_2) \\ z_1 - z_2 &= (x_1 - x_2) + i(y_1 - y_2) \\ z_1 \cdot z_2 &= (x_1 \cdot x_2 - y_1 \cdot y_2) + i(x_1 \cdot y_2 + x_2 \cdot y_1) \\ z_1 / z_2 &= \{(x_1 \cdot x_2 + y_1 \cdot y_2) + i(-x_1 \cdot y_2 + x_2 \cdot y_1)\} / (x_2^2 + y_2^2) . \end{aligned}$$

Representatie door Modulus en argument.

$$\begin{aligned} z &= x + iy & x &= r \cdot \cos \alpha & y &= r \cdot \sin \alpha \\ \text{Modulus } z &= |z| = r = \sqrt{x^2 + y^2} \geq 0 & \alpha &= \arg z \\ z &= r \cdot (\cos \alpha + i \sin \alpha) = r \cdot e^{i\alpha} . \end{aligned}$$

De hoek α is op een veelvoud van 2π na bepaald.

$$\begin{aligned} e^z &= e^{x+iy} = e^x \cdot (\cos y + i \sin y) \\ 1/z &= 1/(x + iy) = (x - iy)/(x^2 + y^2) & \alpha &= -\arg z . \end{aligned}$$

Voor: $z_1 = r_1 \cdot \exp(i\alpha_1)$ en $z_2 = r_2 \cdot \exp(i\alpha_2)$ geldt:

$$\begin{aligned} z_1 \cdot z_2 &= r_1 \cdot e^{i\alpha_1} \cdot r_2 \cdot e^{i\alpha_2} = r_1 \cdot r_2 \cdot e^{i(\alpha_1 + \alpha_2)} \\ z_1 / z_2 &= (r_1 / r_2) \cdot e^{i(\alpha_1 - \alpha_2)} . \end{aligned}$$

We spreken van Hoofdwaarden, als voor $\arg z$ geldt:

$$-\pi < \arg z \leq \pi .$$

Door deze definitie is ook:

$$\log z = \log |z| + i \arg z \quad (z \neq 0)$$

bepaald voor alle waarden van z en volgt het argument uit:

$$\cos(\arg z) = \operatorname{Re} z/|z| \quad \text{en} \quad \sin(\arg z) = \operatorname{Im} z/|z| .$$

Met $\operatorname{Re} z = x$ en $\operatorname{Im} z = y$, gelden dus de relaties:

$$\begin{array}{ll} \arg z = \arctan(y/x) & \text{in rechter halfvlak} \\ \arg z = \pi/2 - \arctan(x/y) & \text{in bovenvlak} \\ \arg z = -\pi/2 - \arctan(x/y) & \text{in ondervlak} . \end{array}$$

Met een bekende stelling uit de vlakke meetkunde (de buitenhoek van een driehoek in een cirkel), kunnen we dit samenvatten in de formule:

$$\arg z = 2 \cdot \arctan \left(\frac{y}{x + \sqrt{x^2 + y^2}} \right) .$$

Met behulp van de hoofdwaaarde kunnen we als volgt ook gebroken machten van z definiëren. Voor een willekeurige reële of complexe α geldt:

$$z^\alpha = e^{\alpha \cdot \log z} = e^{\alpha \cdot \log |z| + i\alpha \arg z} .$$

Met behulp van deze relatie implementeren we later de in veel toepassingen belangrijke functies $z^{\pm \frac{1}{2}}$.

De arctangens functie $w = \arctan(z)$.

Deze functie wordt in de meeste leerboeken niet behandeld, daarom hier een korte afleiding.

Uit $w = \arctan(z)$ volgt:

$$z = \tan(w) = \frac{\sin(w)}{\cos(w)} = -i \cdot \frac{e^{iw} - e^{-iw}}{e^{iw} + e^{-iw}} ,$$

$$iz = \frac{e^{2iw} - 1}{e^{2iw} + 1} ,$$

$$1 + iz = (1 - iz) \cdot e^{2iw} \quad \text{en} \quad e^{2iw} = \frac{1 + iz}{1 - iz}$$

$$w = \arctan(z) = \frac{1}{2i} \cdot \log \frac{1 + iz}{1 - iz} .$$

Substitutie van $z = x + iy$ en verdere uitwerking geeft:

$$2wi = \log \frac{(1-y) + ix}{(1+y) - ix} = \log \frac{(1-x^2-y^2) + 2ix}{(1+y)^2 + x^2} .$$

Met behulp van de besproken eigenschappen, bepalen we hieruit het argument en de modulus:

$$\text{ARG} = \arctan \frac{2x}{(1-x^2-y^2) + \sqrt{(2x)^2 + (1-x^2-y^2)^2}}$$

$$\log |[...]| = \log \left[\frac{(1-y)^2 + x^2}{(1+y)^2 + x^2} \right] .$$

Hiermee hebben we gevonden dat:

$$w = \arctan(z) = k.\pi + \text{ARG} - \frac{i}{4} . \log |[...]| .$$

Deze relatie zullen we later in een Numerieke Laplace-inversie- toepassing en bij sommige Besselfuncties gaan gebruiken.

In de uitwerking van de betreffende formules, worden de complexe variabelen gesplitst in hun samenstellende Reële en Imaginaire delen. Om dit duidelijk te laten uitkomen, noteren we dit als volgt.

Achter het *Reële deel* van een variabele-naam plaatsen we de letter R van R(eël) of we laten de naam met X beginnen. Analoog plaatsen we achter het *Imaginaire deel* van de variabele, de letter I van I(maginair), of we laten de naam met een Y beginnen. In gecompliceerde toepassingen is de directe uitwerking van de complexe operaties niet alleen erg tijdrovend, maar zal ook zeer onoverzichtelijke programma's opleveren.

De operaties optellen, aftrekken en vermenigvuldigen laten zich eenvoudig implementeren. De operatie *delen* is b.v. al een stuk lastiger. Omdat de noemer nul kan worden, moeten de afrondingsfouten ook bij een kleine noemer, zo klein mogelijk worden gehouden.

We gaan daarom gebruiksklare Subs of Functions ontwikkelen voor de volgende operaties:

Optellen, aftrekken, vermenigvuldigen, delen, wortel(Z), $\log(Z)$, $\text{cabs}(Z)$, $\exp(Z)$, $\cosh(Z)$ en $\sinh(Z)$.

Voor sommige operaties geven we twee implementaties.

Optellen en aftrekken zijn eenvoudig en laten we hier achterwege (zie echter het programma LAGUER.BAS).

Vermenigvuldigen.

$$Z30 = X30 + i.Y30 = (X10 + i.Y10).(X20 + i.Y20)$$

SUB CMAAL (X10, Y10, X20, Y20, X30, Y30)

12

```
X30 = X10 * X20 - Y10 * Y20
Y30 = X10 * Y20 + X20 * Y10
END SUB
```

Delen.

$$Z30 = X30 + i.Y30 = (X10 + i.Y10)/(X20 + i.Y20)$$

```
SUB CDELEN (X10, Y10, X20, Y20, X30, Y30)
  X50 = X20^2 + Y20^2
  IF X50 = 0 THEN PRINT "Delen door nul." : EXIT SUB
  X30 = (X10 * X20 - Y10 * Y20)/X50
  Y30 = (-X10 * Y20 + X20 * Y10)/X50
END SUB
```

Delen.

$$C = CR + iCI = (AR + iAI)/(BR + iBI) .$$

(Met voorzieningen om de afrondingsfouten te beperken.)

```
SUB CDIV (AR, AI, BR, BI, CR, CI)
  IF BR = 0 AND BI = 0 THEN PRINT "Delen door nul.": EXIT SUB
  IF ABS(BR) ≥ ABS(BI) THEN
    R = BI/BR: DNR = BR + R * BI
    CR = (AR + AI * R)/DNR: CI = (AI - AR * R)/DNR
  ELSE
    R = BR/BI: DNR = BI + R * BR
    CR = (AR * R + AI)/DNR: CI = (AI * R - AR)/DNR
  END IF
END SUB
```

De wortel.

$$Z20 = \sqrt{Z10}$$
$$Z10 = X10 + i.Y10 : Z20 = X20 + i.Y20 = \sqrt{(X10 + i.Y10)}$$

```
SUB CWORTEL (X10, Y10, X20, Y20)
  IF X10 = 0 AND Y10 = 0 THEN X20 = 0 : Y20 = 0 : EXIT SUB
  X50 = SQR(X10 ^ 2 + Y10 ^ 2): X30 = X10 + X50: EPS = .5E-15
  IF ABS(X30) < EPS THEN
    IF ABS(Y10) < EPS AND X10 < 0 THEN ARG = 2# * ATN(1#)
  ELSE
```

```

    ARG = ATN(Y10 / X30)
  END IF
  X50 = SQR(X50):X30 = X50 * COS(ARG): Y30 = X50 * SIN(ARG)
END SUB

```

De wortel.

$$BR + iBI = \sqrt{(AR + iAI)}.$$

De volgende numeriek meer efficiënte implementatie, maakt niet expliciet gebruik van de arctanfunctie om het argument te bepalen. Impliciet wordt gewerkt met de cyclometrische versies van de formules:

$$\cos(\alpha/2) = \sqrt{(1 + \cos(\alpha))/2} \quad \text{en} \quad \sin(\alpha/2) = +\sqrt{(1 - \cos(\alpha))/2}$$

```

SUB CSQR (AR, AI, BR, BI)
  X = ABS(AR): Y = ABS(AI): EPS = .5E - 15
  IF X < EPS AND Y < EPS THEN BR = 0: BI = 0
  ELSE
    IF X > Y THEN
      W = X * SQR(1# + (Y / X) ^ 2)
    ELSE
      W = Y * SQR(1# + (X / Y) ^ 2)
    END IF
    qq = W + X
    IF AR > 0 THEN
      BR = SQR(.5# * qq): BI = Y * SQR(.5# / qq)
    ELSE
      BR = Y * SQR(.5# / qq): BI = SQR(.5# * qq)
    END IF
    IF AI < 0 THEN BI = -BI
  END IF
END SUB

```

De Absolute waarde van $Z = X10 + i.Y10$

```

FUNCTION CABS (X10, Y10)
  X = ABS(X10): Y = ABS(Y10)
  IF X = 0 THEN CABS = Y: EXIT FUNCTION
  IF Y = 0 THEN CABS = X: EXIT FUNCTION
  IF X > Y THEN
    CABS = X * SQR(1# + (Y / X) ^ 2)

```

```

        ELSE
            CABS = Y * SQR(1# + (X / Y) ^ 2)
        END IF
    END FUNCTION

```

De logarithme.

$\text{LOG}(X10 + i.Y10) = \text{LOG}(|X10 + i.Y10|) + i.\text{Argument} = \text{AMP} + i.\text{ARG}$

```

SUB CLOG (X10, Y10, AMP, ARG)
    EPS = .5E-15
    IF ABS(X10) < EPS AND ABS(Y10) < EPS THEN
        PRINT "LOG oneindig":exit sub
    END IF
    X50 = SQR(X10 ^ 2 + Y10 ^ 2): X30 = X10 + X50
    IF ABS(X30) < EPS THEN
        IF ABS(X10) < EPS AND ABS(Y10) < EPS THEN
            ARG = 4# * ATN(1#)
        ELSE
            ARG = 2# * ATN(Y10 / X30)
        END IF
        AMP = LOG(X50)
    END SUB

```

De functies:

$\cosh(Z)$ en $\sinh(Z)$ met $Z = X10 + i.Y10$.

Leveren:

$\cosh(Z)$ in $Z20 = X20 + i.Y20$ en $\sinh(Z)$ in $Z30 = X30 + i.Y30$.

```

SUB CSINHCOSH (X10, Y10, X20, Y20, X30, Y30)
    E50=EXP(X10):CH=(E50+1#/E50) * .5#
    IF X10 < .2# THEN 'Correctie sinh(Z) voor X10 < 0.2
        SH = X10 * (1# + X10^2/6# + X10^4/120# + X10^6/5040#)
    ELSE
        SH=(E50-1#/E50) * .5#
    END IF
    C10 = COS(Y10): S10 = SIN(Y10)
    X20 = SH * C10: Y20 = CH * S10
    X30 = CH * C10: Y30 = SH * S10
END SUB

```

Een nadeel van deze wijze van implementeren is, dat steeds met 6 parameters moeten worden gewerkt. Dit is echter eenvoudig tot 3 te reduceren, als we de gemaakte notatie-afspraken loslaten en b.v. de volgende arrays definiëren: DIM $P(2)$, $Q(2)$, $R(2)$ in de betekenis van:

$$\begin{aligned} P(1) &= X10: P(2) = Y10: Q(1) = X20 \\ Q(2) &= Y20: R(1) = X30: R(2) = Y30 . \end{aligned}$$

In het volgende voorbeeld gaan we systematisch deze werkwijze toepassen bij het berekenen van de wortels van een Nde-graads polynoom met complexe coëfficiënten volgens de methode van Laguerre¹.

Voor het vinden van de wortels wordt een iteratieproces gebruikt, dat berust op de volgende voor polynomen geldende relaties:

Een polynoom: $P_n(z) = a_0 + a_1 z + \dots + a_n z^n$ met de wortels z_1, \dots, z_n , kan worden geschreven als:

$$P_n(z) = (z - z_1) \cdot (z - z_2) \dots (z - z_n) .$$

Differentiatie naar z levert de volgende relaties op:

$$\begin{aligned} F1 &= \frac{d \log \{|P_n(z)|\}}{dz} = \frac{P'_n(z)}{P_n(z)} = \frac{1}{(z - z_1)} + \frac{1}{(z - z_2)} + \dots + \frac{1}{(z - z_n)} \\ F2 &= \frac{d^2 \log \{|P_n(z)|\}}{dz^2} = \frac{P'_n(z)}{P_n(z)} - \frac{P''_n(z)}{P_n(z)} \\ F2 &= \frac{1}{(z - z_1)^2} + \frac{1}{(z - z_2)^2} + \dots + \frac{1}{(z - z_n)^2} . \end{aligned}$$

Stellen we de afstand van de gezochte wortel $z - z_1 = a$ en de afstand tot de overige wortels $z - z_i = b$ ($i = 2, 3, \dots, n$), dan laten de relaties voor $F1$ en $F2$ zich schrijven als:

$$F1 = \frac{1}{a} + \frac{n-1}{b} \quad \text{en} \quad F2 = \frac{1}{a^2} + \frac{n-1}{b^2} .$$

Elimineren we uit $F1$ en $F2$ de onbekende b , dan resulteert dit in de volgende vierkantsvergelijking in $1/a$:

$$(n-1) \cdot (1/a)^2 + \{F1^2 - 2 \cdot F1 \cdot (1/a) + (1/a)^2\} - (n-1) \cdot F2 = 0$$

met als oplossing de waarden:

$$a = \frac{n}{\{F1 \pm \sqrt{[F1^2 - n \cdot (F1^2 - (n-1) \cdot F2)]}\}} .$$

Voor een gekozen z -waarde, kunnen met de Horner-Newton-techniek de functiewaarden, $P_n(z)$, $P'_n(z)$, $P''_n(z)$ en dus $F1$ en $F2$ worden bepaald.

In de uitdrukking a , kan de factor $[F1^2 - n \cdot (F1^2 - (n-1) \cdot F2)]$ negatief worden en dus zal $\sqrt{[\dots]}$ dan complexe waarden aannemen. Het gehele proces moet daarom in complexe variabelen worden uitgewerkt. Het teken van de wortel moet zo worden gekozen, dat de absolute waarde van de noemer maximaal wordt.

In de volgende stap van het iteratieproces wordt $z = a$ genomen. Het blijkt dat dit proces, onafhankelijk van de gekozen startwaarde, altijd naar een wortel convergeert.

Voor een bewijs van deze stelling van Laguerre, zie Henrici¹⁾, Vol. 1 pag. 466-483, waar tevens de achtergronden van het convergentieproces worden behandeld met behulp van Möbius-transformaties. Voor het bepalen van telkens één wortel is het bovenstaande uitgewerkt in de SUB LAGUER.

```

SUB LAGUER (ar(), ai(), m, x(), eps)
  CONST e3 = 1E-15, mexit = 100
  DIM sq(2), h(2), gp(2), gm(2), g2(2), g(2), b(2), d(2), dx(2)
  DIM f(2), x1(2), du(2), tu(2): eps2 = eps * eps
  FOR ITER = 1 TO mexit
    b(1) = ar(m + 1): b(2) = ai(m + 1): bo = CABS(b())
    d(1) = 0: d(2) = 0: f(1) = 0: f(2) = 0
    FOR J = m TO 1 STEP -1
      CALL CMAAL(f(), x(), tu()): CALL CPLUS(tu(), d(), f())
      CALL CMAAL(d(), x(), tu()): CALL CPLUS(tu(), b(), d())
      CALL CMAAL(b(), x(), tu())
      b(1) = tu(1) + ar(J): b(2) = tu(2) + ai(J)
    NEXT J: bs = CABS(b())
    IF bs < e3 THEN dx(1) = 0: dx(2) = 0: GOTO 20
    IF CABS(d()) < e3 AND CABS(f()) < e3 THEN
      dx(1) = EXP((1# / m) * ln(bs / bo)): dx(2) = 0: GOTO 20
    END IF
    CALL CDIV(d(), b(), g())
    g2(1) = g(1) ^ 2 - g(2) ^ 2: g2(2) = 2# * g(1) * g(2)
    CALL CDIV(f(), b(), du())
    h(1) = g2(1) - 2# * du(1): h(2) = g2(2) - 2# * du(2)
    du(1) = CDBL(m - 1) * (m * h(1) - g2(1))
    du(2) = CDBL(m - 1) * (m * h(2) - g2(2))
    CALL CSQR(du(), sq())
    CALL CPLUS(g(), sq(), gp()): CALL CMIN(g(), sq(), gm())
    IF CABS(gp()) < CABS(gm()) THEN
      gp(1) = gm(1): gp(2) = gm(2)
    END IF
    du(1) = CDBL(m): du(2) = 0: CALL CDIV(du(), gp(), dx())
20:   x1(1) = x(1) - dx(1): x1(2) = x(2) - dx(2)

```

```

        IF (( $x(1) = x1(1)$ ) AND ( $x(2) = x1(2)$ )) THEN EXIT SUB
         $x(1) = x1(1)$ :  $x(2) = x1(2)$ 
    NEXT ITER
    PRINT "Stop routine LAGUER - te veel iteraties": PRINT
END SUB

```

Het stuurprogramma SUB NULPUNTEN zorgt er voor dat alle wortels worden gevonden.

```

SUB NULPUNTEN ( $ar()$ ,  $ai()$ ,  $m$ ,  $wortelr()$ ,  $worteli()$ )
    CONST eps = .00000002#
    DIM  $adr(m+2)$ ,  $adi(m+2)$ ,  $b(2)$ ,  $c(2)$ ,  $x(2)$ ,  $tu(2)$ 
    FOR  $J = 1$  TO ( $m+1$ ):  $adr(J) = ar(J)$ :  $adi(J) = ai(J)$ : NEXT  $J$ 
    FOR  $J = m$  TO 1 STEP -1:  $x(1) = 0$ :  $x(2) = 0$ 
        CALL LAGUER( $adr()$ ,  $adi()$ ,  $J$ ,  $x()$ , eps)
        IF ( $ABS(x(2)) \leq (2\# * eps * ABS(x(1)))$ ) THEN  $x(2) = 0$ 
         $wortelr(J) = x(1)$ :  $worteli(J) = x(2)$ 
         $b(1) = adr(J+1)$ :  $b(2) = adi(J+1)$ 
        FOR  $jj = J$  TO 1 STEP -1
             $c(1) = adr(jj)$ :  $c(2) = adi(jj)$ 
             $adr(jj) = b(1)$ :  $adi(jj) = b(2)$ 
            CALL CMAAL( $b()$ ,  $x()$ ,  $tu()$ ): CALL CPLUS( $tu()$ ,  $c()$ ,  $b()$ )
        NEXT  $jj$ 
    NEXT  $J$ 
END SUB

```

Alle andere hierbij gebruikte subs zijn te vinden in het programma LAGUER.BAS op de schijf.

In de SUB VOORBEELD, is een polynoom van de graad 3 geïmplementeerd. Deze kan via het menu worden gekozen.

De wortels zijn: $z_1 = -3 - i.3$: $z_2 = -3 + i.3$: $z_3 = 5 + i.5$.

De methode, kan natuurlijk ook worden gebruikt om de wortels van een polynoom met reële coëfficiënten te bepalen. Uiteraard moet steeds bij elke coëfficiënt ook het imaginaire deel = 0 worden ingevoerd.

Pas later bij het implementeren van de gemodificeerde Besselfuncties en Numerieke Laplace-inversie toepassingen, worden de voor complexe variabelen geschikte SUBs en FUNCTIONs, met grote frequentie gebruikt.

Referentie.

- 1) Peter Henrici. Applied and Computational complex analysis. Vol. 1.

3. Operaties met Machtreksen

Bij enkele later te behandelen onderwerpen, zoals o.a. genereren van machtreksen voor: $\log(\Gamma(1+x))$, $\Gamma(1+x)$ en het bepalen van alternatieve asymptotische machtreksen van de Besselfuncties $J_n(x)$ en $Y_n(x)$ ($n = 0, 1, 2, 3$) zijn enkele algorithmen voor operaties met machtreksen ontwikkeld welke ook hun nut hebben bij andere toepassingen.

Gegeven: Twee machtreksen met reële coëfficiënten $\neq 0$:

$$\begin{aligned} f_1(z) &= a_0 + a_1.z + a_2.z^2 + \dots + a_n.z^n + \dots \\ f_2(z) &= b_0 + b_1.z + b_2.z^2 + \dots + b_n.z^n + \dots \end{aligned}$$

Gevraagd wordt om de coëfficiënten van een machtreeks

$$f_3(z) = c_0 + c_1.z + c_2.z^2 + \dots + c_n.z^n + \dots$$

te bepalen, onder elk van de volgende bewerkingen:

- $f_3(z) = f_1(z).f_2(z)$
- $f_3(z) = f_1(z)/f_2(z)$
- $f_3(z) = \exp(f_1(z) - a_0)$.

Om de coëfficiënten-relaties te vinden, voegen we aan elke machtreeks een oneindige dimensionale semicirculaire coëfficiënt- matrix¹⁾ toe.

Zo wordt $f_1(z)$ voorgesteld door de semicirculaire matrix:

$$A = \begin{vmatrix} a_0 & a_1 & a_2 & a_3 & \dots \\ & a_0 & a_1 & a_2 & \dots \\ & & a_0 & a_1 & \dots \\ & & & a_0 & \dots \\ & & & & \dots \\ & & & & \dots \\ & & & & \dots \end{vmatrix}$$

waarin $a_{ij} = a_{j-i}$ voor $j \geq i$, $a_{ij} = 0$ voor $j < i$.

Analoog voegen we aan $f_2(z)$ de matrix B toe en aan $f_3(z)$ de matrix C . Van de oneindig dimensionale semicirculaire matrix A beschouwen we een submatrix A_N , bestaande uit de eerste N rijen en de eerste N kolommen. Het product van twee semicirculaire submatrices $A_N.B_N = (A.B)_N$ is weer een semicirculaire matrix. De gewenste operaties met machtreksen laten zich nu vertalen als operaties met semicirculaire submatrices. Op deze wijze kunnen we voor de operaties a), b) en c), relaties voor het genereren van de coëfficiënten afleiden.

Geval a).

Vermenigvuldigen:

$$f_3(z) = f_1(z) \cdot f_2(z)$$

$$C_n = (A \cdot B)_n = A_n \cdot B_n \quad c_{ij} = c_{j-i}$$

$$c_0 = a_0 \cdot b_0$$

$$c_1 = a_1 \cdot b_0 + a_0 \cdot b_1$$

$$c_2 = a_0 \cdot b_2 + a_1 \cdot b_1 + a_2 \cdot b_0$$

$$\text{---} = \text{---} \quad \text{---} \quad \text{---}$$

$$c_n = a_0 \cdot b_n + a_1 \cdot b_{n-1} + \dots + a_n \cdot b_0 .$$

Een implementatie is als volgt:

```
FOR K = 0 to M : D(K) = 0
  FOR I = K to 0 STEP -1
    D(K) = D(K) + A(I) * B(K - I)
  NEXT I : C(K) = D(K)
NEXT K
```

Geval b).

Delen:

$$f_3(z) = f_1(z) / f_2(z) \quad (B \cdot C)_n = A_n$$

$$c_0 = a_0 \cdot b_0$$

$$c_1 = a_1 \cdot b_0 - a_0 \cdot b_1$$

$$c_2 = a_2 \cdot b_0 - (c_0 \cdot b_2 + b_1 \cdot c_1)$$

$$\text{---} = \text{---} \quad \text{---} \quad \text{---}$$

$$c_n = a_n \cdot b_0 - (c_0 \cdot b_n + c_i \cdot b_{n-i} + \dots + c_{n-1} \cdot b_1) .$$

Met de implementatie:

```
C(0) = A(0) : C(1) = A(0) * B(1) - A(1) * B(0)
FOR K = 2 to M : D(K) = 0
  FOR I = K - 1 to 0 STEP -1
    D(K) = D(K) + C(I) * B(K - I)
  NEXT I : C(K) = A(K) * B(0) - D(K)
NEXT K
```

Een interessant speciaal geval volgt voor: $f_1(z) = 1$: Dan is:

$$f_3(z) = 1/f_2(z) = f_2(z)^{-1} .$$

A is nu de eenheidsmatrix en we genereren de coëfficiënten in de matrix:

$$C_n = B_n^{-1} \quad \text{voor } n = 1, 2, \dots .$$

Dit zelfde resultaat is ook te verkrijgen door op de klassieke manier de inverse van B_N^{-1} te bepalen (B_N rechts aanvullen met een eenheidsmatrix, door slim vegen in B_N , op de plaats van de eenheidsmatrix, de inverse matrix genereren).

Voor de elementen van de inverse matrix volgt:

$$c_0 = 1/b_0 ; \quad c = -b_0/b_1^2$$

$$c_2 = 1/b_0^3 \begin{vmatrix} b_1 & b_2 \\ b_0 & b_1 \end{vmatrix} = (b_1 \cdot b_1 - b_0 \cdot b_2)/b_0^3 .$$

Algemeen:

$$c_n = (-)^n / b_0^{n-1} \cdot \begin{vmatrix} b_1 & b_2 \dots & b_n \\ b_0 & b_1 \dots & b_{n-1} \\ & b_0 \dots & b_{n-2} \\ & & \dots \\ & & & b_0 & b_1 \end{vmatrix} .$$

Geval c).

$$f_3(z) = \exp(f_1(z) - a_0) .$$

Stel $h_1(z) = (f_1(z) - a_0)$ dan volgt uit de Taylor-ontwikkeling:

$$f_3(z) = \exp(f_1(z) - a_0) = 1 + h_1(z)/1! + \dots + h_n(z)/n! .$$

Construeer met methode a) stapsgewijze de hulpfuncties:

$$h_n(z) = h_{n-1}(z) \cdot h_1(z), n = 2, 3, \dots .$$

De coëfficiënten laten zich eenvoudig genereren en worden in het volgende programma opgeslagen in een dubbelarray $E_{m,m}$.

```
FOR I = 1 TO M: E(1,I) = A(I): NEXT I
  FOR L = 1 TO M
    FOR K = L + 1 TO M: C(K) = 0
      FOR I = K - 1 TO L STEP -1
        C(K) = C(K) + E(L, I) * A(K - I)
```

```

NEXT I: E(L + 1, K) = C(K)
NEXT K: FOR J = 0 TO M: C(J) = 0: NEXT J
NEXT L

```

Uit $B_{m,m}$ de functie $f_3(z)$ opbouwen en de factoren $1/(n!)$ genereren:

```

Z = 1#: FOR K = 1 TO M: Z = Z * CDBL(K)
FOR I = 1 TO M: C(I) = C(I) + E(K, I)/Z: NEXT I
NEXT K

```

Op soortgelijke wijze kan methode c) ook worden gebruikt om: $f_3(z) = \exp(f_1(z) - a_0)$ te bepalen (veronderstel $a_0 = 1$).

Stel:

$$h_1(z) = b_1 \cdot z + b_2 \cdot z^2 + \dots + b_n \cdot z^n + \dots, \quad \text{met } f_1 = 1 + h_1$$

ontwikkel $f_4(z) = 1/(1 + h_1(z))$ in de meetkundige reeks:

$$f_4(z) = 1 - h_1(z) + h_1(z)^2 + \dots + (-)^n \cdot h_1(z)^n + \dots$$

De coëfficiënten van de opvolgende machten $(h_1(z))^k$, $k = 1, 2, \dots$ worden weer opgebouwd in een array $E_{m,m}$.

Vervolgens wordt uit $E_{m,m}$ de functie $f_4(z)$ opgebouwd. Het verschil met het voorgaande is, dat nu de factoren $(-)^n$ in plaats $1/(n!)$ moeten worden gegenereerd. Dit kan als volgt:

```

Z = -1#: FOR K = 1 TO M: Z = -Z
FOR I = 1 TO M: Q(I) = Q(I) + E(K, I) * Z: NEXT I
NEXT K: Q(0) = 1#

```

Tenslotte wordt met methode a): $f_3 = f_1(z) \cdot f_4(z)$ bepaald.

In het programma MATRIXFU.BAS (aanwezig op de schijf) worden de twee manieren van delen geïllustreerd aan de voorbeelden: $\exp(-x) = 1/\exp(x)$ en $\tan(x)/x = \{\sin(x)/x\} / \cos(x)$ (waarin $x^2 = z$ is gesteld).

De inverse machtreeks van een analytische functie

Als in $z = z_0$ een functie $f(z)$ analytisch is en $f'(z_0) \neq 0$, dan bestaat de inverse functie $z = g(w)$ en is analytisch in de omgeving van $w_0 = f(z_0)$. De coëfficiënten van deze inverse machtreeks, de Buhrmann-Lagrange-reeks²⁾, kunnen dan eenvoudig als volgt worden bepaald:

Beschouw voor $z_0 = 0$ en $w_0 = 0$ de machtreeks:

$$w = \sum_{j=1}^{\infty} a_j \cdot z^j \quad a_1 \neq 0$$

en de inverse machtreeks

$$z = \sum_{n=0}^{\infty} b_n \cdot w^n .$$

Directe substitutie van de machtreeks w in de inverse reeks z geeft:

$$z = b_1 \cdot \left(\sum_{j=1}^{\infty} a_j z^j \right) + b_2 \cdot \left(\sum_{j=1}^{\infty} a_j \cdot z^j \right)^2 + \dots .$$

Uitwerken levert de relaties:

$$\begin{aligned} a_1 \cdot b_1 &= 1 \\ a_2 \cdot b_1 + a_1^2 \cdot b_2 &= 0 \\ a_3 \cdot b_1 + 2 \cdot a_1 \cdot a_2 \cdot b_2 + a_1^3 \cdot b_3 &= 0 \\ &\dots \end{aligned}$$

Hieruit kunnen we wel systematisch maar slechts moeizaam de coëfficiënten: b_1, b_2, b_3, \dots berekenen.

Voor een computer-implementatie gaan we als volgt te werk: We voeren een hulpreeks:

$$z_j = \sum_{n=j}^{\infty} c_{j,n} \cdot w^n$$

in, waarmee we de volgende algemene relaties af leiden:

$$\begin{aligned} c_{1,n} &= b_n , \quad (n = 1, 2, \dots) \\ b_1 &= c_{1,1} = 1/a_1 \end{aligned}$$

en voor $n > 1$

$$a_1 \cdot c_{1,n} = - \sum_{j=2}^n a_j \cdot c_{j,n}$$

waarin

$$c_{j,n} = \sum_{k=1}^{n+1-j} b_k \cdot c_{j-1, n-k} , \quad (2 \leq j \leq n) .$$

Uit de reeds berekende coëfficiënten b_1, b_2, \dots, b_{j-1} , kan dus telkens een opvol-

gende b_j worden bepaald, tot tenslotte b_n is gevonden.

Het bovenstaande uitgewerkt in de volgende sub.

```

DEFINT I - N: DEFDBL A - H, O - Z
SUB BUHRMAN (M, A(), B())
DIM C(M + 1, M + 1)
C(1, 1) = 1# / A(1): B(1) = C(1, 1)
  FOR N = 2 TO M: FOR J = N TO 2 STEP -1: C(J, N) = 0
    FOR K = 1 TO N + 1 - J
      C(J, N) = C(J, N) + B(K) * C(J - 1, N - K):
    NEXT K: NEXT J: C(1, N) = 0
    FOR L = 2 TO N: C(1, N) = C(1, N) + A(L) * C(L, N): NEXT L
    C(1, N) = -C(1, N) / A(1): B(N) = C(1, N)
  NEXT N: B(0) = 0: A = A(0)
  FOR I = 1 TO M: B(0) = B(0) - B(I) * A: A = A * A(0): NEXT I
  FOR I = 1 TO M: B(I) = C(1, I): NEXT I
END SUB

```

Het complete programma BUHRMAN.BAS is op de schijf aanwezig.

Invoer: Aantal termen N .

Machtrees-coëfficiënten $A(k)$ $k = 0, 1, \dots, N$

Uitvoer: Machtrees-coëfficiënten $B(k)$ $k = 0, 1, \dots, N$

Hier volgen enkele toepassingen.

Bepaal de machtrees-coëfficiënten van $\tan(x)$, uitgaande van de coëfficiënten van

$$\arctan(x) = \sum_{n=0}^{\infty} (-x)^{2n+1} / (2n+1) .$$

Deze kunnen we als volgt genereren:

```

Z1 = 1#: A(0) = 0
FOR I = 0 TO M\2
  A(2 * I + 1) = Z / CDBL(2 * I + 1): Z = -Z
NEXT I

```

Vergelijk de zo verkregen machtrees coëfficiënten van $\tan(x)$, met die verkregen met de $\{\sin(x)/x\}/\{\cos(x)\}$ aanpak gebruikt in MATRIXFU.BAS.

Bepaal de coëfficiënten in de machtrees van $\text{Log}(1+x)$, met behulp van de

machtreeks-coëfficiënten in de inverse functie:

$$\exp(x) - 1 = \sum_{n=0}^{\infty} (x)^n / n! .$$

Deze laatste genereren we met behulp:

```
Z = 1#: A(0) = 0
FOR I = 1 TO M:
  Z = Z * CDBL(I): A(I) = 1#/Z
NEXT I
```

Referenties.

- 1, 2) Peter Henrici. Applied and computational complex analysis. Vol I.
- 2) M. Abramowitz and Irene Stegun. Handbook of Mathematical Functions.

4. Bepalen van Eigenwaarden en Eigenvectoren van Matrices

Methoden voor het bepalen van de eigenwaarden (e.w.) en de eigenvectoren (e.v.) van matrices met reële elementen.

We beginnen onze beschouwing met reële tri-diagonale matrices. Dit is geen beperking, omdat elke volle reële symmetrische matrix met behulp van een gelijkvormigheids-transformatie is om te zetten in een symmetrische tri-diagonale matrix.

We gaan uit van de volgende standaard tri-diagonale (niet noodzakelijk symmetrische) matrixeigenwaarde formulering:

$$\begin{vmatrix} A_{1,1} - X & A_{1,2} & & & \\ A_{2,1} & A_{2,2} - X & A_{2,3} & & \\ & A_{3,2} & A_{3,3} - X & A_{3,4} & \\ & & & \text{---} & \\ & & & & A_{M,M-1} \\ & & & & A_{M,M-1} & A_{M,M} - X \end{vmatrix} = 0 . \quad (1)$$

Deze vorm wordt gevonden, als we een niet zelfgeadjungeerde 2-de orde randwaarde probleem met behulp van differentie-technieken benaderen met een stelsel differentie-vergelijkingen.

Deze asymmetrie vormt geen probleem voor het bepalen van de e.w. omdat in de voor de oplossing te gebruiken algorithmen de co-diagonaalelementen uitsluitend worden gebruikt in de combinatie $A_{k,k+1} * A_{k+1,k}$.

Een symmetrie-getransformeerd stelsel met: co-diagonaalelementen:

$$b_{k+1} = \sqrt{(A_{k,k+1} * A_{k+1,k})}, \quad k = 1, \dots, M \quad (2)$$

en gelijke diagonaal-elementen $a_k = A_{k,k}$, heeft dus de zelfde e.w. We noteren een symmetrisch tri-diagonaal e.w. systeem algemeen als:

$$\begin{vmatrix} a_1 - X & b_2 & & & \\ b_2 & a_2 - X & b_3 & & \\ & b_3 & a_3 - X & b_4 & \\ & & & \text{---} & \\ & & & & b_N \\ & & & & b_N & a_N - X \end{vmatrix} = 0 \quad (3)$$

Alle a_k en b_k zijn reëel; b_1 wordt niet gebruikt.

De e.v. van een asymmetrisch systeem worden gevonden door op de e.v. van het symmetrie gecorrigeerde systeem een terug-transformatie:

$$(b_1 = 1 : G_1 = 1 : G_k = G_{k-1} \cdot \sqrt{(b_k/b_{k-1})} \quad k = 2, \dots, M) \quad (4)$$

toe te passen.

Vormen we uit matrix (3) de determinanten van alle deelmatrices met k rijen en k kolommen ($k = 1, 2, \dots, N$), die in de linker bovenhoek beginnen, dan gelden voor de zo gegenereerde polynomen in X :

$$P_k(X) = \det \left\{ \begin{array}{cccc} a_1 - X & b_2 & & \\ b_2 & a_2 - X & b_3 & \\ & & \dots & \dots \\ & & & b_k \\ & & & b_k & a_k - X \end{array} \right\}$$

de recursierelaties:

$$P_0(X) = 1 : P_1(X) = (a_1 - X) \cdot P_0(X) \quad (5)$$

$$P_k(X) = (a_k - X) \cdot P_{k-1}(X) - b_{k-1}^2 \cdot P_{k-2}(X)$$

$$\text{met voor } k = N \text{ de karakteristieke vergelijking } P_N(X) = 0 . \quad (6)$$

De polynoom-relaties (5) stellen ons in staat om expliciet de coëfficiënten van het karakteristieke polynoom (6) te genereren. In alle volgende berekeningen wordt niet de volle tri-diagonaalmatrix als invoer gebruikt, maar worden alleen de diagonaal- en co-diagonaalelementen ten opgeslagen in drie vectorarrays:

De boven-diagonaal-elementen in: $U()$
 De diagonaal-elementen in: $D()$
 De onder-diagonaal-elementen in: $V()$

Het volgende sub-programma, levert de coëfficiënten van het Karakteristieke Polynoom af in array $C()$ (de coëfficiënt van de hoogste graad heeft de index 1).

```
DEFINT I - N: DEFDBL A - H, O - Z
SUB KARAKTERISTIEKPOLYNOOM (N, V(), D(), U(), C())
DIM B(N + 1), P(N + 1, N + 1) 'Hulp arrays B(), P()
B(1) = 0: FOR K = 2 TO N + 1: B(K - 1) = V(K - 1) * U(K - 2): NEXT K
P(0,0) = 1#: P(1,0) = D(1): P(1,1) = -1#: K = 2
WHILE K ≤ N
  P(K,0) = D(K) * P(K - 1,0) - B(K) * P(K - 2,0)
  FOR J = 1 TO K - 2
    P(K,J) = D(K) * P(K - 1,J) - P(K - 1,J - 1) - B(K) * P(K - 2,J)
  NEXT J
  P(K,K - 1) = D(K) * P(K - 1,K - 1) - P(K - 1,K - 2)
  P(K,K) = -P(K - 1,K - 1): K = K + 1
```



```

WEND
FOR J = 0 TO N: C(J) = P(N, N - J): NEXT J
END SUB

```

Dit sub-programma wordt later gebruikt in het programma KARAKTER.-BAS (zie kettingbreuk-toepassingen) om van enkele met differentiemethoden benaderde randwaarde-problemen met de SUB KARAKTERISTIEKEPOLYNOOM eerst de coëfficiënten en vervolgens met het kettingbreuk programma SUB QDPROGRESIEF simultaan de nulpunten (e.w.) te bepalen.

Het besprokene geeft nu de mogelijkheid om een korte samenvatting te geven van een aantal door Rutishauser⁶⁾, Wilkinson en anderen¹⁻¹⁰⁾ ontwikkelde zeer perfecte ALGOL algorithmen, voor het bepalen van de e.w. en e.v. van reële symmetrische tri-diagonaal-matrices.

De basisprincipes van de eigenwaarde-bepaling, welke worden gebruikt in een aantal van deze algorithmen berusten op de volgende eigenschappen:

De met de relaties (5) gevormde rij polynomen

$$P_0(x), P_1(x), \dots, P_n(x)$$

vormen een Sturm rij.

De stelling van Sturm is als volgt te formuleren:

Het aantal eigenwaarden groter dan x_1 wordt (mits alle $b_k \neq 0$) gegeven door het aantal malen dat twee opvolgende elementen in de rij:

$$P_0(x_1), P_1(x_1), \dots, P_n(x_1)$$

hetzelfde teken hebben.

Ook geldt dat:

$$\operatorname{sgn} P_k(x) = \begin{cases} 1 & \text{voor } x \text{ naar } -\infty \\ (-)^k & \text{voor } x \text{ naar } \infty \end{cases}$$

Met behulp van deze eigenschap kan een zeer effectieve methode voor het scheiden van de eigenwaarden worden geïmplementeerd.

De stelling van Greshgorin:

Van een matrix A met complexe elementen (a_{ij}) , liggen alle eigenwaarden van A in de vereniging van de cirkels:

$$|z - a_{ij}| \leq R_i \quad (i = 1, 2, \dots, n) \quad \text{met straal } R_i = \sum_{j=1, i \neq j}^n |a_{ij}|.$$

Deze eigenschap wordt gebruikt om de intervalgrenzen, waarbinnen een eigenwaarde ligt, telkens te schatten.

De Polynoom recursie-relaties, Greshgorin, Sturm, gecombineerd met een Interval-halverings-methode (traag maar zeer robuust) of de Newton-methode (snel, maar iets minder nauwkeurig bij meervoudige e.w.) vormen een solide basis voor twee algorithmen, waarmee met grote nauwkeurigheid een gewenst aantal eigenwaarden van een symmetrische tri-diagonaalmatrix kan worden bepaald. De methode van Newton is mogelijk, omdat de differentiatie van de relaties (5) de volgende betrekkingen voor de afgeleiden oplevert:

$$\begin{aligned} P'_0(X) &= 0 : P'_1(X) = -1 \\ P'_k(X) &= -P_{k-1}(X) + (a_k - X) \cdot P'_{k-1}(X) - b_{k-1}^2 \cdot P_{k-2}(X) \\ P'_N(X) &= P'_N(X) . \end{aligned} \quad (7)$$

Wilkinson¹⁾ heeft bewezen dat het voordelen heeft, om niet de polynomen $P_k(x)$ (5) zelf, maar de hieruit afgeleide relaties:

$$Q_k(X) = P_k(X)/P_{k-1}(X) ,$$

als volgt te gebruiken:

$$\begin{aligned} Q_1(x) &= a_1 - X \\ Q_k(x) &= (a_k - X) - b_{k-1}/Q_{k-1}(X) \quad (k = 2, \dots, n) . \end{aligned} \quad (8)$$

Op basis van het bovenstaande zijn twee Algol programma's¹⁾ bewerkt:

```
SUB EIVALI (N2%,K2%,DIA#(),CD#(),ZLA#())      (Interval- aanpak)
SUB EIVALN (N2%,K2%,DIA#(),CD#(),ZLA#())      (Newton- aanpak).
```

De betekenis van de parameters is voor beide gelijk:

Invoer: Orde matrix is $N2$.
 Gewenste aantal eigenwaarden $K2$.
 Gesymmetriseerde co-diagonaal elementen in array $CD()$.
 Diagonaal elementen in array $DIA()$.

Uitvoer: De e.w.: X_1, \dots, X_{k2} in array $ZLA()$.

```
DEFINT I - N: DEFDBL A - H, O - Z '(intervaldeling)
SUB EIVALI (N2, K2, DIA(), CD(), ZLA())
DIM XO(N2 + 1) : N1 = 1: K1 = 1:
XMIN = DIA(N2): XMAX = XMIN
IF N1 < N2 THEN
  H2 = ABS(CD(N2)): XMIN = XMIN - H2: XMAX = XMAX + H2
```

```

END IF
FOR I = N2 - 1 TO N1 STEP -1
  IF I = N1 THEN
    H2 = ABS(CD(I + 1))
  ELSE
    H2 = ABS(CD(I + 1)) + ABS(CD(I))
  END IF
  DIAGI = DIA(I)
  IF DIAGI - H2 < XMIN THEN XMIN = DIAGI - H2
  IF DIAGI + H2 > XMAX THEN XMAX = DIAGI + H2
NEXT I: ZME = 5E-14
IF XMAX + XMIN > 0 THEN E1 = ZME * XMAX
  ELSE E1 = -ZME * XMIN
IF E ≤ 0 THEN E = E1
FOR I = K1 TO K2: ZLA(I) = XMAX: XO(I) = XMIN: NEXT I
XU = XMAX: FOR K = K2 TO K1 STEP -1
  IF XU > ZLA(K) THEN XU = ZLA(K)
  I = K: WHILE I ≥ K1 OR XL < XMIN: XL = XO(I): I = I - 1: WEND
  WHILE (XU - XL) > (2# * ZME * (ABS(XL) + ABS(XU)) + E)
    ZLAK = XL + (XU - XL) / 2#: Q = DIA(N1) - ZLAK
    IF Q < 0 THEN K4 = 1 ELSE K4 = 0
    FOR I = N1 + 1 TO N2: IF Q = 0 THEN Q = ZME
      Q = DIA(I) - ZLAK - CD(I) * CD(I) / Q
      IF Q < 0 THEN K4 = K4 + 1
    NEXT I: K4 = K4 - 1 + N1
    IF K4 < K THEN
      IF K4 < K1 THEN
        XO(K1) = ZLAK: XL = ZLAK
      ELSE
        XL = ZLAK: XO(K4 + 1) = ZLAK
        IF ZLA(K4) > ZLAK THEN ZLA(K4) = ZLAK
      END IF
    ELSE XU = ZLAK
  END IF
WEND: ZLA(K) = XL + (XU - XL) / 2#: NEXT K
END SUB

```

De Subs EIVALI en EIVALN (Newton-aanpak) zijn aanwezig op de schijf als EIVALI.SUB en EIVALN.SUB.

Eigenvectoren van een matrix, als de eigenwaarden bekend zijn.

Wanneer van een matrix A van de orde N , een eigenwaarde X met voldoende nauwkeurigheid bekend is, kan een iteratieproces worden gebruikt om de bij-

behorende eigenvector U te bepalen. Stel we willen bij de eigenwaarde Z_j , waarvoor een goede benadering X_j is gevonden, de eigenvector U_j bepalen ($j = 1, \dots, N$). Veronderstel alle e.w. enkelvoudig en voldoende in grootte verschillend, dan volgt uit het volgende iteratieproces, toegepast op de vector b_i :

$$(A - X_j \cdot I) \cdot b_{i+1} = b_i ,$$

(I de eenheids-matrix), de inverse iteratie-relatie:

$$b_{i+1} = (A - X_j \cdot I)^{-1} \cdot b_i \quad (i = 0, 1, \dots) . \quad (9)$$

De inverse matrix $(A - X_j \cdot I)^{-1}$ bestaat en heeft eigenwaarden $1/(Z_k - X_j)$.

We spreken van een goede benadering van de eigenwaarde X_j , als het verschil $(Z_j - X_j)$, klein is t.o.v. $(Z_k - X_j)$ voor alle $j \neq k$.

Een begin-iteratie-vector b_0 is (mits niet alle $r_k = 0$ zijn) altijd te schrijven als:

$$b_0 = r_1 \cdot U_1 + r_2 \cdot U_2 + \dots + r_n \cdot U_n .$$

Dus volgt na een voldoende groot aantal stappen i in het iteratieproces, dat alle termen in de som:

$$b_i = \left[r_1 \cdot U_j + \sum_{k=2}^n r_k \cdot U_k \cdot \left\{ \frac{(Z_j - X_j)}{(Z_k - X_j)} \right\}^i \right]$$

naar nul gaan.

Dus zal de vector b_i naderen naar de eigenvector U_j .

Op basis van het bovenstaande is de volgende bewerking van het door Peters en Wilkinson²⁾ ontwikkelde Algol-programma gemaakt:

```
SUB EIGVEC (N2%,DIA#(),CD#(),V#(),D#(),U#(),XLA#,NF%)
```

```
Invoer: Orde matrix is N2.
         Eigenwaarde XLA.
         Gesymmetriseerde buitendiagonaal elementen in array CD().
         Diagonaal elementen in array DIA().
         Hulp arrays D(), U().
```

```
Uitvoer: Convergentie-parameter NF "Niet Falen = -1"
         Genormeerde eigenvector in array V().
```

```
DEFINT I - N: DEFDBL A - H, O - Z
SUB EIGENVEC (N2, DIA(), CD(), V(), D(), U(), XLA, NF)
DIM P(40), F(40), INTE(40)
```

```

N1 = 1: XME = 5E - 14: K = N1: NF = -1: XNM = ABS(DIA(N1))
FOR I = N1 + 1 TO N2: XNM = XNM + ABS(DIA(I)) + ABS(CD(I))
NEXT I
E2 = XNM * .001#: E3 = XME * XNM: E4 = E3 * CDBL(N2)
U = E4 / SQR(CDBL(N2))
FOR I = N1 TO N2: V(I) = U: NEXT I: U = DIA(N1) - XLA
IF N1 = N2 THEN W = 0 ELSE W = CD(N1 + 1)
FOR I = N1 + 1 TO N2
IF CD(I) = 0 THEN CDI = E3 ELSE CDI = CD(I)
IF ABS(CDI) ≥ ABS(U) THEN
XU = U / CDI: P(I) = XU: D(I - 1) = CDI:
U(I - 1) = DIA(I) - XLA
IF I = N2 THEN F(I - 1) = 0 ELSE F(I - 1) = CD(I + 1)
U = W - XU * U(I - 1): W = -XU * F(I - 1): INTE(I) = -1
ELSE
XU = CDI / U: P(I) = XU: D(I - 1) = U:
U(I - 1) = W: F(I - 1) = 0
U = DIA(I) - XLA - XU * W: IF I < N2 THEN W = CD(I + 1)
INTE(I) = 0
END IF
NEXT I
IF U = 0 THEN D(N2) = E3 ELSE D(N2) = U
U(N2) = 0: F(N2) = 0: ITS = 1: NC = -1
WHILE NC = -1 AND NF = -1
FOR I = N2 TO N1 STEP -1
V(I) = (V(I) - U * U(I) - W * F(I)) / D(I)
W = U: U = V(I): NEXT I: XNM = 0
FOR I = N1 TO N2: XNM = XNM + ABS(V(I)): NEXT I
IF XNM < 1# THEN
IF XNM = 0 THEN
V(K) = E4: IF K < N2 THEN K = K + 1 ELSE K = N1
ELSE
XU = E4 / XNM:
FOR I = N1 TO N2: V(I) = V(I) * XU: NEXT I
END IF
FOR I = N1 + 1 TO N2
IF INTE(I) = -1 THEN
U = V(I - 1): V(I - 1) = V(I): V(I) = U - P(I) * V(I)
ELSE
V(I) = V(I) - P(I) * V(I - 1)
END IF
NEXT I: ITS = ITS + 1
IF ITS > 5 THEN NF = 0
ELSE
NC = 0

```

```

      END IF
    WEND
  END SUB

```

De hierboven behandelde methoden, voor het bepalen van de eigenwaarden en eigenvectoren, vormen de basis voor het oplossen van met differentie-technieken benaderde differentiaalvergelijkingen. Deze methode wordt later gebruikt bij het oplossen van randwaarde-problemen.

Veel andere toepassingen leiden tot vol bezette matrices.

Met behulp van gelijkvormigheidstransformaties kan echter elke matrix, met determinantwaarde $\neq 0$, op een Hessenberg-vorm worden gebracht.

Een reële symmetrische matrix, wordt in een symmetrische tridiagonaalmatrix getransformeerd.

Een matrix A heeft de Hessenberg-vorm, als de elementen $a_{ij} = 0$ zijn voor $i > j + 1$ ($i, j = 1, 2, \dots, n$).

Een reële matrix wordt op een Hessenberg-vorm gebracht, met de volgende door Housholder ontwikkelde transformatie-methode, met behulp van de volgende rij transformaties:

$$A_k = P_k \cdot A_{k-1} \quad \text{met} \quad A_0 = A \quad \text{en} \quad P_k = I - 2 \cdot V_k \cdot V_k^T .$$

Hierin is P_k een orthogonale transformatie-matrix, I de eenheidsmatrix, de vector V_k (met een lengte gelijk 1) wordt zo gekozen dat de component a_{ij} in de k -de stap voor $j - k \geq 2$ gelijk nul wordt. (Zie Stoer en Bulirsch¹¹). Martin en Wilkinson^{8,9}) hebben op basis hiervan, de Algol-programma's: ORTHES en ORTHBAK ontwikkeld.

Voor het bepalen van de eigenwaarden en eigenvectoren van een vol bezette symmetrische matrix, is de SUB HESEIVECI($N2\%$, $K2\%$, $A\#()$, $ZLA()$, $ZV()$) ontwikkeld, hierin zijn de volgende programma-delen verwerkt:

Een bewerking van ORTHES (op Hessenberg-vorm brengen van de matrix).

De hierboven behandelde subs EIVALI en EIVEC voor het bepalen van de e.w. en e.v. van de tri-diagonaal-matrix.

Een bewerking van ORTHBAK voor de terug-transformatie van de e.v.

Deze zeer grote SUB HESEIVECI is opgenomen in het gelijknamige programma HESEIVECI.BAS op de schijf.

Het programma is geschikt om van een vol bezette symmetrische matrix *alle* of *een gewenst beperkt aantal* eigenwaarden en eigenvectoren te bepalen (het is een trage maar robuuste methode).

```

Invoer:  Orde matrix           :   N2.
         Gewenste aantal e.w   :   K2.
         De matrix coëfficiënten :   A() (per rij).

Uitvoer: Eigenwaarden         :   ZLA().

```

Genormeerde eigenvectoren: $ZV()$.

Een andere veel gebruikte methode om alle eigenwaarden en eigenvectoren van een symmetrische matrix A te bepalen, is de QL-methode. Deze methode werd ontwikkeld door Rutishauser⁸⁾. Hieruit is later door Francis⁷⁾, de veel betere QR-methode afgeleid.

Het basisprincipe van de QR methode berust op het volgende: Uit $A = Q.R$ en $B = R.Q$ waarin Q een unitaire en R een boven-driehoeks-matrix is, volgt dat: $B = R.Q = Q.H.A.Q$. Voor de rij matrices $A = A_0, A_1, A_2, \dots$ geconstrueerd met behulp van dit algoritme geldt, dat A_N gaat naderen naar een boven-driehoeks- vorm, met de eigenwaarden als diagonaal-elementen.

Wilkinson⁸⁾ en medewerkers hebben een mengsel van de QL en QR methoden verwerkt tot een zeer nauwkeurig snel werkend algoritme.

Ze gaan uit van het gemodificeerde algoritme:

$$A_k = Q_k.L_k \quad A_{k+1} = L_k.Q_k = Q_k^H.A_k.Q_k$$

met Q unitaire en L een onder-driehoeks-matrix.

De convergentie wordt nog verbeterd, als in het rekenproces de matrix A_k wordt vervangen door $A_k - g.I$, waarin de parameter-waarde g (I de eenheids-matrix) geschikt wordt gekozen.

Voor A symmetrisch en Q orthogonaal, zijn de A_k 's reëel en symmetrisch.

Op basis van het bovenstaande is de SUB HESTQL($N2\%$, $D\#()$, $A\#()$, $Z\#()$) ontwikkeld. Het is een bewerking van het algol programma TQL⁸⁾ en is samengevoegd met de Housholder-transformatie. Het complete programma HESTQL.BAS, waarmee *alle* e.w. en e.v. worden bepaald, werkt snel en zeer nauwkeurig. Het is te vinden op de schijf.

Invoer: Orde matrix $N2$.

Rijgewijze de matrix-elementen in $A()$.

Uitvoer: Eigenwaarden in $D()$.

Eigenvectoren $Z()$.

Hieronder volgen enkele oefenvoorbeelden.

Voorbeeld 1.

$$\begin{vmatrix} 5 & 4 & 1 & 1 \\ 4 & 5 & 1 & 1 \\ 1 & 1 & 4 & 2 \\ 1 & 1 & 2 & 4 \end{vmatrix} \quad \text{Eigenwaarden: } 1, 2, 5, 10.$$

Eigenvectoren:

$$\begin{array}{cccc}
 -.7071068 & 0 & .3162278 & -.6324555 \\
 .7071068 & -3.2711026E - 18 & .3162278 & -.6324555 \\
 -2.149029E - 18 & -.701068 & -.6324555 & -.3162278 \\
 -2.149029E - 18 & .701068 & -.6324555 & -.3162278
 \end{array}$$

Voorbeeld 2.

$$\left| \begin{array}{cccc}
 6 & 4 & 4 & 1 \\
 4 & 6 & 1 & 4 \\
 4 & 1 & 6 & 4 \\
 1 & 4 & 4 & 6
 \end{array} \right| \quad \text{Eigenwaarden: } -1, 5, 5, 15.$$

Eigenvectoren:

$$\begin{array}{cccc}
 -.5 & .0 & -.7071068 & -.5 \\
 .5 & -.7071068 & -5.642755E - 15 & -.5 \\
 .5 & .7071068 & -5.642755E - 15 & -.5 \\
 -.5 & -1.734724E - 17 & .7071068 & -.5
 \end{array}$$

Veel toepassingen in de klassieke mechanica en in de quantenmechanica worden geformuleerd in de vorm: $A.X = \mu.B.X$, waarin A en B reële symmetrische matrices zijn en B positief definit is.

Dit probleem kan op verschillende manieren worden opgelost.

Een aanpak met de boven behandelde methoden is als volgt:

De positief definitie matrix B , heeft reële postieve eigenwaarden en kan dus worden getransformeerd in de vorm: $B = Y.D^2.Y^T$. Hierin is Y de e.v. matrix, de matrix Y^T de getransponeerde van Y en een matrix D^2 met de eigenwaarden in de hoofd-diagonaal.

Vorm met behulp hiervan: $(D^{-1}.Y^T.A.Y.D^{-1}).(D.Y^T.X) = \mu.(D.Y^T.X)$.

Stellen we hierin: $(D^{-1}.Y^T.A.Y.D^{-1}) = P$ en $(D.Y^T.X) = Z$, dan is weer een tri-diagonaal e.w. probleem: $P.Z = \mu.Z$ met $X = (Y.D^{-1}.Z)$ verkregen.

Als gevolg van de vele matrix-operaties en het tweemaal moeten aanroepen van HESTQL is dit een zeer trage en geheugen-verslindende methode.

Een betere (verwante) aanpak is, om de matrix B met behulp van LU decompositie⁵⁾: $U = LT$ en $B = L.L^T$ terug te brengen tot een standaard Tri-diagonaal-eigenwaarde probleem.

Uit

$$A.X = \mu.L.L^T.X \quad \text{met} \quad (L.L^T)^{-1} = B^{-1} \quad \text{en} \quad L^{-T}.L^T = 1$$

volgt:

$$(L^{-1}.A.L^{-T}).L^T.X = \mu.L^{-1}.L.L^T.X = \mu.L^T.X .$$

Stel:

$$L^{-1}.A.L^{-T} = P \text{ met } L^T.X = Z \text{ (} X = L^{-T}.Z \text{) en } (L^T.X)^T = Z^T$$

dan volgt weer het standaard e.w.-probleem:

$$P.Z = \mu.Z .$$

Nu we een procedure hebben om $P = L^{-1}.A.L^{-T}$ te bepalen, kunnen we de eigenwaarden μ_k en eigenvectoren Z_k , $k = 1, \dots, N$ berekenen. Tenslotte levert de terug-transformatie $X_k = L^{-T}.Z_k$ de gezochte eigenvectoren van het oorspronkelijke probleem $A.X = \mu.B.X$.

Alle geschetste operaties, de LU decompositie van B en de transformaties $P = L^{-1}.A.L^{-T}$ en $X_k = L^{-T}.Z_k$ zijn geïntegreerd in het sub-programma: SUB CHOLESKYREDUCTIE.

Het e.w.-systeem $P.Z = \mu.Z$ wordt opgelost met SUB HESTQL.

De gebruikte parameters van het programma CHOLTQL.BAS (op de schijf te vinden) zijn de volgende:

Invoer: Orde matrix N .

Rijgewijze de matrixelementen in: $P()$.

Rijgewijze de matrixelementen in: $B()$ (postief definit).

Uitvoer: Eigenwaarden in $D()$ en Eigenvectoren in $Z()$.

Hulp array: $A()$.

Hier volgt een uitgewerkt voorbeeld:

$$A = \begin{vmatrix} 1 & 1 & 5 \\ 1 & 1 & .25 \\ .5 & .25 & 2 \end{vmatrix} \quad B = \begin{vmatrix} 2 & 2 & 2 \\ 2 & 5 & 5 \\ 2 & 5 & 11 \end{vmatrix}$$

$$P = \begin{vmatrix} .5 & 0 & -.1443376 \\ 0 & 0 & -5.892557E-02 \\ -.1443376 & -5.892557E-02 & .4166667 \end{vmatrix}$$

Eigenwaarden:

$-9.024314062434369E - 03$; $.3150468298896966$; $.6106441508394045$

Eigenvectoren $P.z = \mu.z$:

$-4.288618E - 02$	$-.6086472$	$.7922811$
$-.9875658$	$.1458736$	$5.860625E - 02$
$-6.174487E - 02$	$-.3183995$	$-.2479453$

Eigenvectoren van $Ax = \mu.Bx$:

$.5398463$	$-.5145987$	$.526391$
$-.5084265$	$.4026197$	$.2817799$
$-6.174487E - 02$	$-.3183995$	$-.2479435$

De oplossing van een reëel niet-symmetrisch matrix-eigenwaarde-probleem, is meer gecompliceerd, omdat er complexe eigenwaarden en eigenvectoren kunnen optreden. De te gebruiken algoritmen moeten dus worden aangepast om te kunnen rekenen met complexe variabelen. Het aantal bewerkingen is een veelvoud van die bij symmetrische e.w.-algorithmen. Om optredende afrondingsfouten binnen de perken te houden vindt er eerst een schaling van de matrix plaats. De Housholder methode wordt gebruikt voor het op Hessenberg-vorm brengen. De eigenwaarden worden berekend met een dubbelstap QR-proces, de eigenvectoren worden bepaald met het inverse iteratie-proces. Er is een compleet nieuwe bewerking gemaakt van een door J. Grab and M.A. Brebner¹⁰⁾ (in enkellengte Fortran) geschreven programma.

Ofschoon in het nieuwe programma er meer dan 100 goto's zijn weggewerkt, draagt het nog vele sporen van het Fortran verleden en is het geen voorbeeld voor gestructureerd programmeren geworden.

In het complete programma ALG343.BAS (op de schijf), zijn ook de later in de literatuur voorgestelde verbeteringen aangebracht. Het omvat de volgende 5 subs:

Het stuurprogramma:

```
SUB EIGENP(N%,A#(),EVR#(),EVI#(),VECR#(),VECI#(),INDIC%() )
regelt de aanroep van alle andere SUBs:
```

```
SCALE, HESQR, REALVE, COMPVE.
```

SUB SCALE, verzorgt de schaling van A.

SUB HESQR, berekent de reële en complexe eigenwaarden van A.

SUB REALVE, berekent de bijbehorende reële eigenvectoren.

SUB COMVE, berekent de complexe eigenvectoren.

De betekenis van de parameters is als volgt:

Invoer: N = orde en de volle reële matrix A.

Uitvoer: EVR() reële deel e.w. EVI() imaginaire deel e.w.

VECR() reële deel e.v. VECI() imaginaire deel e.w.
 INDIC() index array, geeft aan welke e.w. zijn gevonden.

Voorbeeld.

Matrix:

$$A = \begin{vmatrix} 4 & -5 & 0 & 3 \\ 0 & 4 & -3 & -5 \\ 5 & -3 & 4 & 0 \\ 3 & 0 & 5 & 4 \end{vmatrix}$$

Eigenwaarden:

Reële deel e.w.	Imaginaire deel e.w.
12	0
1	5
1	-5
2	0

Eigenvectoren:

Reële deel e.v.	Imaginaire deel e.v.
$\begin{vmatrix} -.5 & 1 & 1 & .5 \\ .5 & 0 & 0 & .5 \\ -.5 & 0 & 0 & -.5 \\ -.5 & -1 & -1 & .5 \end{vmatrix}$	$\begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}$

Een algorithm voor het oplossen van het matrix eigenwaarde probleem met complexe elementen is niet opgenomen.

Hermitische eigenwaarde-matrices kunnen echter als volgt worden opgelost.

Beschouw het hermitische eigenwaardeprobleem: $C.w = \mu.w$, waarin: $C = A + iB$ (A, B symmetrische reële matrices, i de imaginaire eenheid), de eigenvector $w = u + iv$ (u, v reëel), μ de reële eigenwaarden van de hermitische matrix.

Het eigenwaarde-probleem:

$$(A + iB)(u + iv) = \mu(u + iv)$$

is equivalent met het symmetrische reële e.w.-probleem van de orde $2n$:

$$\begin{vmatrix} A & -B \\ B & A \end{vmatrix} \begin{vmatrix} u \\ v \end{vmatrix} = \mu \begin{vmatrix} u \\ v \end{vmatrix}.$$

Voor hermitisch matrices geldt:

$$A^T = A \quad \text{en} \quad B^T = -B.$$

Alle eigenwaarden zijn dubbel. Bij de eigenwaarde μ hebben we dus $\begin{vmatrix} u \\ v \end{vmatrix}$ en

ook $\begin{vmatrix} -v \\ u \end{vmatrix}$ als eigenvector.

Als voorbeeld kiezen we de matrix:

$$C = \begin{vmatrix} 2 & 3 - 3i \\ 3 + 3i & 5 \end{vmatrix}.$$

De eigenwaarden vinden we uit:

$$\begin{vmatrix} 2 - \mu & 3 - 3i \\ 3 + 3i & 5 - \mu \end{vmatrix} = \mu^2 - \mu - 8 = (\mu + 1)(\mu - 8) = 0.$$

Voor $\mu = -1$ volgt uit de lineaire vergelijkingen:

$$\begin{matrix} 3 & u + (3 - 3i) & v = 0 \\ (3 + 3i) & u + 6 & v = 0 \end{matrix} \left\{ \begin{array}{l} \text{de eigenvector :} \\ \begin{vmatrix} u \\ v \end{vmatrix} = \begin{vmatrix} 1 - i \\ -1 \end{vmatrix} \end{array} \right.$$

Voor $\mu = 8$ volgt analoog uit:

$$\begin{matrix} -6 & u + (3 - 3i) & v = 0 \\ (3 + 3i) & u - 3 & v = 0 \end{matrix} \left\{ \begin{array}{l} \text{de eigenvector :} \\ \begin{vmatrix} u \\ v \end{vmatrix} = \begin{vmatrix} 1 \\ 1 + i \end{vmatrix} \end{array} \right.$$

De equivalente reële matrix:

$$\begin{vmatrix} 2 & 3 & 0 & 3 \\ 3 & 5 & -3 & 0 \\ 0 & -3 & 2 & 3 \\ 3 & 0 & 3 & 5 \end{vmatrix}$$

opgelost met HESTQL.BAS levert direct eigenwaarden: $-1, -1, 8, 8$ en de bijbehorende genormeerde eigenvectoren:

.816497	.000000	-.577350	.000000
-.408248	.408248	-.577350	-.577350
.000000	.816497	.000000	.577350
-.408248	-.408248	-.577350	.577350

Alle hierna volgende programma's, zijn menugestuurd en geschikt voor direct gebruik. In de meeste worden keuzevoorbeelden met diverse data-invoermogelijkheden geïllustreerd: klassieke subroutines, datastatements, toetsenbord-invoer en datafile-invoer.

Hieronder volgt aanvullend oefenmateriaal van het datafile type. De ge-

bruikte nomenclatuur is als volgt:

Alle files hebben de uitgang .DAT. Filenamen die met een S beginnen zijn van het symmetrische type en dus geschikt voor alle besproken algoritme-typen. De files beginnende met een A(symmetrisch), kunnen alleen worden opgelost met Sub EIGENP in ALG343.BAS. De matrices $S4 \times 4$.DAT en $S4 \times 4$ -1.DAT, $S5 \times 5$ -1.DAT en $S5 \times 5$ -2.DAT, zijn allemaal positief definit en kunnen dus worden gebruikt als B matrix in de SUB CHOLESKYREDUCTIE. $A6 \times 6$.DAT is een typisch parasitair testvoorbeeld, geschikt om de nauwkeurigheid van het algoritme EIGENP te onderzoeken. Het heeft nl. een 6-voudige eigenwaarde $\mu = 10$. Als gevolg van afrondingsfouten treden er echter kleine imaginaire componenten op. De nauwkeurigheid is echter zeer goed en veel beter dan vermeld door Grab¹⁰). Hieronder volgen de betreffende matrices.

$S4 \times 4$.DAT:	$S4 \times 4$ -2.DAT:	$S5 \times 5$.DAT
5#, 4#, 1#, 1#	6#, 4#, 4#, 1#	10#, 1#, 2#, 3#, 4#
4#, 5#, 1#, 1#	4#, 6#, 1#, 4#	1#, 9#, -1#, 2#, -3#
1#, 1#, 4#, 2#	4#, 1#, 6#, 4#	2#, -1#, 7#, 3#, -5#
1#, 1#, 2#, 4#	1#, 4#, 4#, 6#	3#, 2#, 3#, 12#, -1#
		4#, -3#, -5#, -1#, 15#

$S5 \times 5$ -1.DAT	$S5 \times 5$ -2.DAT
12#, 1#, -1#, 2#, 1#	10#, 2#, 3#, 1#, 1#
1#, 14#, 1#, -1#, 1#	2#, 12#, 1#, 2#, 1#
-1#, 1#, 16#, -1#, 1#	3#, 1#, 11#, 1#, -1#
2#, -1#, -1#, 12#, -1#	1#, 2#, 1#, 9#, 1#
1#, 1#, 1#, -1#, 11#	1#, 1#, -1#, 1#, 15#

$S11 \times 11$.DAT

5#, 2#, 1#, 1#, 0, 0, 0, 0, 0, 0, 0
2#, 6#, 3#, 1#, 1#, 0, 0, 0, 0, 0, 0
1#, 3#, 6#, 3#, 1#, 1#, 0, 0, 0, 0, 0
1#, 1#, 3#, 6#, 3#, 1#, 1#, 0, 0, 0, 0
0, 1#, 1#, 3#, 6#, 3#, 1#, 1#, 0, 0, 0
0, 0, 1#, 1#, 3#, 6#, 3#, 1#, 1#, 0, 0
0, 0, 0, 1#, 1#, 3#, 6#, 3#, 1#, 1#, 0
0, 0, 0, 0, 1#, 1#, 3#, 6#, 3#, 1#, 1#
0, 0, 0, 0, 0, 1#, 1#, 3#, 6#, 3#, 1#
0, 0, 0, 0, 0, 0, 1#, 1#, 3#, 6#, 2#
0, 0, 0, 0, 0, 0, 0, 1#, 1#, 2#, 5#

S8×8.DAT

611#	196#	-192#	407#	-8#	-52#	-49#	29#
196#	899#	113#	-192#	-71#	-43#	-8#	-44#
-192#	113#	899#	196#	61#	49#	8#	52#
407#	-192#	196#	611#	8#	44#	59#	-23#
-8#	-71#	61#	8#	411#	-599#	208#	208#
-52#	-43#	49#	44#	-599#	411#	208#	208#
-49#	-8#	8#	59#	208#	208#	99#	-911#
29#	-44#	52#	-23#	208#	208#	-911#	99#

AS4×4.DAT

-2#	1#	1#	1#
-7#	-5#	-2#	-4#
0#	-1#	-3#	-2#
-1#	0	-1#	0

AS6×6.DAT

-41#	55#	4#	3#	2#	51#
-2#	10#	55#	4#	3#	2#
-3#	0#	10#	55#	4#	3#
-4#	0	0	10#	55#	4#
-55#	0	0	0	10#	55#
-51#	55#	4#	3#	2#	61#

Referenties.

- 1) Barth, Martin en Wilkinson. Numer. Math. 9, 386–393, 1967.
- 2) Peters en Wilkinson: Contribution 11/18. Wilkinson & Reinsch. Handbook for Automatic Computation Vol. II.
- 3) Martin en Wilkinson: Numer. Math. 12, 349–368 (1968).
- 4) C. Reinsch and F.L. Bauer: Numer. Math. II, 264–272 (1968). Rational QR transformation with Newton shift for Tri-diagonal Matrices.
- 5) R.S. Martin and J.H. Wilkinson: Reduction of Symmetric Eigenproblem $Ax = \mu.Bx$ and Related Problems to Standard Form. Numer. Math. 11, 99–110 (1968).
- 6) Rutishauser, H: Solution of eigenvalue problems with the LR-transform. Nat. Bur. Standards Appl. Math. Ser: 49, 47–81 (1958).
- 7) Francis, J.G.F: The QR transformation, Parts 1 and 2, hebben verschillende type matrices-Algoprogramma's ontwikkeld. Comp. J. 4, 265- -271, 332–345 (1961, 1962).
- 8) H. Bowdler, R.S. Martin, C. Reinsch and J.H. Wilkinson: The QD and QL Algorithms for Symmetric Matrices. Numer. Math. 11, 293- -306 (1968).
- 9) R.S. Martin and J.H. Wilkinson: Similarity Reduction of a General Matrix to Hessenberg Form. Numer. Math. 12, 349–368 (1968).

- 10) J. Grad and M.A. Brebner: Eigenvalues and eigenvectors of a real general matrix. Reed 12 October 1967. Computer services University of Birmingham, England. Collected algorithms from CACM.
- 11) Stoer und Bulirsch: Einführung in die Numerische Mathematik. Vol. 1 (242–245) Vol. 2 (23–29).

5. Kettingbreuken

De grondslagen van de Kettingbreuk-theorie, behoorde vroeger tot het standaard wiskunde-programma. De beginselen zijn uitstekend beschreven in het klassieke leerboek: *Beknopte Hoogere Algebra* van Dr. F. Schuh³⁾. Voor een moderne behandeling zie Henrici¹⁾, of voor een korte samenvatting CWI Tract 10⁵⁾, pag. 82–102.

Reeds Christaan Huygens maakte gebruik van kettingbreuken om langzaam convergerende rekenprocessen te versnellen. Na de komst van de computer is dit gebruik eerst afgenomen, echter na de publicatie in 1954 van het Quotient Difference algoritme door Rutishauser²⁾, is een belangrijk nieuw toepassingsgebied ontstaan. Met behulp van het QD-algoritme, kan men nu op een standaardmanier, uit de coëfficiënten van de machtreeks, de kettingbreukcoëfficiënten bepalen.

Omdat deze latere uitbreidingen voornamelijk in vaktijdschriften zijn te vinden, geven we eerst een korte samenvatting van de belangrijkste eigenschappen. De meeste eigenschappen zijn eenvoudig met behulp van volledige inductie te bewijzen, waar dat niet het geval is, zijn ze in de vermelde literatuur^{1,2)} te vinden. Vervolgens worden verschillende implementaties van het QD algoritme en functiewaarde-bepalingsalgorithmen besproken. Deze algorithmen worden daarna in bijna alle volgende hoofdstukken gebruikt bij het oplossen van zeer diverse toepassingen.

Enkele eigenschappen van kettingbreuken.

Een kettingbreuk wordt voorgesteld door het schema:

$$(B_0) + \frac{A_1}{B_1 + \frac{A_2}{B_2 + \frac{A_3}{B_3 + \frac{A_4}{B_4 + \dots}}}} \quad (1)$$

Hierin zijn de grootheden A_n, B_n ($n = 1, 2, \dots$) reële of complexe getallen. In sommige literatuur wordt nog een term B_0 toegevoegd, men spreekt dan van een onzuivere kettingbreuk.

Dit schema wordt alleen nog in oudere literatuur gebruikt. Tegenwoordig wordt dit als volgt afgekort:

$$\frac{A_1}{B_1} + \frac{A_2}{B_2} + \frac{A_3}{B_3} + \dots \quad (2)$$

Sommige auteurs laten zelfs de scheidingsstrepen weg en noteren dan:

$$\frac{A_1}{B_1+} \frac{A_2}{B_2+} \frac{A_3}{B_3+} \dots \quad (3)$$

In recente literatuur gebruikt men in analogie met het sommatie symbool Σ voor machtreeksen, als volgt het Φ symbool voor kettingbreuken:

$$\frac{n}{\Phi} \frac{A_k}{B_k} \quad (n \text{ eventueel oneindig}) \quad (4)$$

A_k noemen we de partiële teller.

B_k noemen we de partiële noemer.

W_k noemen we de k de naderende breuk.

Voor het bepalen van de opvolgende naderende breuken W_k , definiëren we grootheden P_k en Q_k als volgt:

$$\begin{aligned} W_1 = P_1/Q_1 &= \frac{A_1}{B_1} & W_2 = P_2/Q_2 &= \frac{A_1}{B_1} + \frac{A_2}{B_2} \\ W_m = P_m/Q_m &= \frac{A_1}{B_1} + \dots + \frac{A_m}{B_m} \end{aligned} \quad (5)$$

Uitgaande van de beginvoorwaarden:

$$P_0 = 0 \quad P_1 = A_1 \quad Q_0 = 1 \quad Q_1 = B_1 \quad (6)$$

gelden de volgende recursieformules voor P_m en Q_m :

$$\begin{aligned} P_m &= A_m \cdot P_{m-2} + B_m \cdot P_{m-1} & (m = 2, 3, \dots) \\ Q_m &= A_m \cdot Q_{m-2} + B_m \cdot Q_{m-1} & (m = 2, 3, \dots) \end{aligned} \quad (7)$$

Hieruit volgt direct de belangrijke determinant-formule:

$$P_{m-1} \cdot Q_m - P_m \cdot Q_{m-1} = (-)^m \cdot A_1 \cdot A_2 \dots A_m \quad (8)$$

Deze betrekking kunnen we ook herleiden tot:

$$\frac{P_m}{Q_m} = \frac{A_1}{(Q_0 \cdot Q_1)} - \frac{A_1 \cdot A_2}{(Q_1 \cdot Q_2)} + \dots + (-)^m \cdot \frac{A_1 \cdot A_2 \dots A_m}{(Q_m \cdot Q_{m-1})} \quad (9)$$

Met deze relatie is het verband gelegd tussen een kettingbreuk C en een machtreeks S , in die zin, dat de n^{de} benadering W_n van de kettingbreuk C correspondeert met de n^{de} partiële som S_n van S ($C \approx S$).

We spreken van de equivalentie (\approx) van een kettingbreuk en een machtreeks indien voor alle $Q_k \neq 0$ ($k = 1, 2, \dots$) geldt:

$$\sum_{m=1}^{\infty} \frac{A_m}{B_m} \approx \sum_{m=1}^{\infty} (-)^m A_1 \cdot A_2 \dots A_m / (Q_{m-1} \cdot Q_m). \quad (10)$$

Een kettingbreuk

$$C = \sum_{m=0}^{\infty} \frac{1}{\beta_m} \quad \text{met } \beta_m > 0 \quad (11)$$

convergeert als de reeks $S = \sum_{m=0}^{\infty} \beta_m$ divergeert.

Equivalentente kettingbreuken.

Uit de definitie-vergelijking en recursierelaties volgt direct, dat de waarde van een kettingbreuk niet verandert als we de partiële tellers en noemers met geschikt gekozen getallen $R_k \neq 0$ vermenigvuldigen.

$$\frac{A_1 \cdot R_1}{B_1 \cdot R_1} + \frac{A_2 \cdot R_1 \cdot R_2}{B_2 \cdot R_2} + \frac{A_3 \cdot R_2 \cdot R_3}{B_3 \cdot R_3} + \dots + \frac{A_n \cdot R_{n-1} \cdot R_n}{B_n \cdot R_n}. \quad (12)$$

Kiezen we de getallen: R_1, R_2, \dots, R_n als volgt:

$$\begin{array}{ll} B_1 \cdot R_1 = 1 & R_1 = 1/B_1 \\ B_2 \cdot R_2 = 1 & R_2 = 1/B_2 \\ \dots & \dots \\ B_n \cdot R_n = 1 & R_n = 1/B_n \end{array} \quad (13)$$

Dan krijgen we een kettingbreuk met alle partiële noemers gelijk 1.

Op analoge wijze kunnen andere equivalentente vormen worden afgeleid b.v. met partiële tellers gelijk 1 (zie de getallentheorie-toepassingen).

In de hierna komende beschouwingen veronderstellen we, in verband met het hierna te bespreken QD-algorithme, dat alle kettingbreuken op een vorm met partiële noemers gelijk 1 zijn gebracht.

Contractie van kettingbreuken.

Onder het Even-deel (Oneven-deel) van de kettingbreuk

$$C = \sum_{k=1}^{\infty} \frac{A_k}{1} \quad (14)$$

verstaan we de getransformeerde kettingbreuk welke alleen benaderingen van Even (Oneven) orde bevat:

$$C_{\text{even}} = \frac{A_1}{1 + A_2} - \frac{A_2 \cdot A_3}{1 + A_3 + A_4} - \dots - \frac{A_{2k-2} \cdot A_{2k}}{1 + A_{2k-1} + A_{2k}} \quad (15)$$

$$C_{\text{oneven}} = A_1 - \frac{A_1 \cdot A_2}{|1 + A_2 + A_3|} - \dots - \frac{A_{2k-1} \cdot A_{2k}}{|1 + A_{2k} + A_{2k+1}|} . \quad (16)$$

(Aanwijzing voor een afleiding zonder gebruik van functietheorie: Bij een even contractie, gebruik $P_k(z)$ voor $k = n, n-1$ en $n-2$, elimineer P_{k-1} en P_{k-3} . Substitueer $n = 2m$. Analoog voor de Q's. Bij de oneven contractie: gebruik P_k voor $k = n-1, n-2, n-3$, elimineer de even indices, etc).

We beschouwen vervolgens kettingbreuken welke als volgt van een variabele (parameter) z afhangen:

$$C(z) = \frac{\prod_{i=1}^{\infty} A_i \cdot z}{|1|} . \quad (17)$$

De kettingbreuk $C(z)$ hangt op eenduidige manier samen met de machtreeks

$$R(z) = C_0 + C_1 \cdot z + C_2 \cdot z^2 + \dots + C_n \cdot z^n + \dots . \quad (18)$$

$C(z) \approx R(z)$ als alle Hankeldeterminanten $H_{k,n} \neq 0$ zijn en de onderstaande vorm hebben en bovendien de begincondities:

$H_{k,0} = 1, H_{k,1} = C_k$ gelden:

$$H_{k,n} = \begin{vmatrix} C_n & \cdot & \cdot & C_{n+k-1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ C_{n+k-1} & \cdot & \cdot & C_{n+2k-2} \end{vmatrix} \neq 0 \quad (19)$$

voor $n = 1, 2, \dots, k = 1, 2, 3, \dots$

Voor Hankeldeterminanten geldt de volgende identiteit van Jacobi:

$$(H_{k,n})^2 - H_{k,n-1} \cdot H_{k,n+1} + H_{k+1,n-1} \cdot H_{k-1,n+1} = 0 . \quad (20)$$

Voor de naderende breuken $W_k = P_k/Q_k$ gelden nu de gewijzigde begincondities en recursierelaties.

$$P_0 = 0 \quad P_1 = A_1 \quad Q_0 = 1 \quad (21)$$

$$P_k = z \cdot A_k \cdot P_{k-2} + P_{k-1} \quad Q_k = z \cdot A_k \cdot Q_{k-2} + Q_{k-1} . \quad (22)$$

Om uit de coëfficiënten van de machtreeks $R(z)$, systematisch de kettingbreukcoëfficiënten voor $W_n(z)$ te kunnen bepalen, noteren we:

$$C_n(z) = \frac{\prod_{i=1}^{\infty} A_i \cdot z}{|1|} \quad (23)$$

ook als volgt:

$$C(z) = \frac{C_0 |}{| 1} - \frac{q_{1,0} \cdot z |}{| 1} - \frac{e_{1,0} \cdot z |}{| 1} - \frac{q_{2,0} \cdot z |}{| 1} - \frac{e_{2,0} \cdot z |}{| 1} .$$

De coëfficiënten $q_{k,0}$ (quotiënten) en $e_{k,0}$ (differenties) zijn eenduidig met het door Rutishauser afgeleide QD (Quotient Difference) algorithm, als volgt uit (18) de machtreekscoëfficiënten te bepalen als we uitgaan van de begincondities

$$e_{0,n} = 0, \quad n = 1, 2, \dots \quad q_{1,n} = C_{n+1}/C_n, \quad n = 0, 1, 2, \dots \quad (25)$$

en onderstaande recursierelaties toepassen:

$$\begin{aligned} e_{m,n} &= \{q_{m,n+1} - q_{m,n}\} + e_{m-1,n+1} , \\ q_{m,n+1} &= (e_{m,n+1}/e_{m,n}) \cdot q_{m,n+1} , \end{aligned} \quad (26)$$

met $m = 1, 2, \dots, n = 0, 1, \dots$. Deze relaties kunnen we als volgt in een schema noteren:

$$\begin{array}{cccccccc} & q_{1,0} & & & & & & & \\ 0 & & e_{1,0} & & & & & & \\ & q_{1,1} & & q_{2,0} & & & & & \\ 0 & & e_{1,1} & & e_{2,0} & & & & \\ & q_{1,2} & & q_{2,1} & & q_{3,0} & & & \\ 0 & & e_{1,2} & & e_{2,1} & & e_{3,0} & & \\ & q_{1,3} & & q_{2,2} & & q_{3,1} & & q_{4,0} & \\ 0 & & e_{1,3} & & e_{2,2} & & e_{3,1} & & \\ & q_{1,4} & & & & & & & \\ 0 & & & & & & & & \end{array} \quad (27)$$

(Te bewijzen met behulp van Hankeldeterminanten.)

De partiële tellers $P_k(z)$ en de partiële noemers $Q_k(z)$ van de naderende breuk W_k zijn nu functies van z . De teller $P_k(z)$ is van de graad $[(k-1)/2]$ en de noemer $Q_k(z)$ heeft de graad $[k/2]$.

Algemeen geldt:

$$R(z) \cdot Q_n(z) - P_n(z) = A_1 \cdot A_2 \cdot A_3 \dots A_{n+1} \cdot (-z)^n + \dots \quad (28)$$

Delen we linker- en rechterlid door $Q_n(z)$, dan is de eerste term in het rechterlid

$$A_1 \cdot A_2 \cdot A_3 \dots A_{n+1} \cdot (-z)^n / Q_n(z) \quad (29)$$

een goede maat voor de afbreekfout die men maakt als de kettingbreuk-ontwikkeling na n termen wordt afgebroken.

De kettingbreukcoëfficiënten A_m hangen als volgt samen, met de grootheden $e_{m,0}$ en $q_{m,0}$ in het QD-schema en de Hankeldeterminanten:

$$A_1 = C_0 \quad (30)$$

$$A_{2m+1} = -e_{m,0} = \frac{-H_{m+1,0} \cdot H_{m-1,1}}{H_{m,0} \cdot H_{m,1}} \quad \{m = 0, 1, 2, \dots\} \quad (31)$$

$$A_{2m+2} = -q_{m,0} = \frac{-H_{m+1,1} \cdot H_{m,0}}{H_{m,1} \cdot H_{m+1,0}} \quad \{m = 0, 1, 2, \dots\} . \quad (32)$$

Speciale Kettingbreuken: De relatie Kettingbreuk en Asymptotische reeks.

De asymptotische reeks

$$f(z) = \sum_{k=0}^{\infty} d_k / z^{k+1} \quad (33)$$

correspondeert formeel met de kettingbreuk:

$$C(z) = \frac{\infty}{\Phi} \frac{A_k \cdot z^{-1} |}{| 1} . \quad (34)$$

Met behulp van de equivalentietransformatie (12), waarin we:

$$R_1 = R_3 = R_5 = \dots = z \quad R_0 = R_2 = R_4 = \dots = 1 \quad (35)$$

stellen, laat zich dit in de in de volgende vorm brengen:

$$C(z) = \frac{A_1 \cdot z |}{| z} + \frac{A_2 |}{| 1} + \frac{A_3 |}{| z} + \frac{A_4 |}{| 1} + \frac{A_5 |}{| z1} + \dots . \quad (36)$$

Dit type speciale kettingbreuken, noemt men Stieltjes- Kettingbreuken.

Convergentie-eigenschap: Als de Stieltjes- kettingbreuk $C(z)$ convergeert voor $z = 1$, dan is de convergentie uniform voor alle z in het opengesneden z -vlak mits $|\arg(z)| < \pi$.

Door de even en de oneven contractie van dit type $C(z)$ te bepalen (beide overeenkomend met de benadering $d_{2m} \cdot z^{-2m}$ van $f(z)$), is door Rutishauser²⁾ het QD-algorithme gevonden.

De Stieltjes-transformatie.

Laat $\Omega(t)$ een begrensde niet dalende functie zijn op het interval $[0, \infty)$, dan is de Stieltjes-integraal:

$$f(z) = \int_0^{\infty} d\Omega(t)/(z+t)$$

in het langs de positieve x -as open gesneden complexe z -vlak een analytische functie.

Laat verder de reële getallen μ_0, μ_1, \dots (genoemd de momenten), bepaald zijn met behulp van deze functie $\Omega(t)$ volgens:

$$\mu_n = \int_0^{\infty} t^n d\Omega(t) \quad (n = 0, 1, 2, \dots) . \quad (37)$$

Dan is de asymptotische machtreeks $P(z) = \mu_0/z + \mu_1/z^2 + \mu_2/z^3 + \dots$ verkregen met behulp van de Stieltjes-transformatie:

$$f(z) = \int_0^{\infty} d\Omega(t)/(z+t) \quad (n = 0, 1, 2, \dots)$$

equivalent met de kettingbreuk

$$C(z) = \frac{\mu_0}{z} + \frac{A_2}{1} + \frac{A_3}{z} + \frac{A_4}{1} + \dots \quad (C(z) \approx P(z)) . \quad (38)$$

Een nodig en voldoende voorwaarde voor convergentie van de kettingbreuk $C(z)$ is, dat de asymptotische reeks

$$\sum_{k=0}^{\infty} \mu_k^{-1/2k} = \infty \quad (39)$$

divergeert.

Dit staat bekend als het Carleman criterium.

De voorgaande eigenschappen stellen ons in staat om hulpmiddelen te ontwikkelen om de in de inleiding genoemde applicaties, met behulp van kettingbreuken te kunnen oplossen.

In afzonderlijke hoofdstukken behandelen we drie verschillende implementaties van het QD-algoritme en twee voor de Functiewaardebepaling voor het reële variabele type. Vervolgens wordt een QD- en een Functiewaarde-algoritme voor het complexe-variabelen-type geïmplementeerd. De laatste twee vormen het hart van alle Numerieke Laplace-inversie-toepassingen op basis van kettingbreuken. Enkele klassieke voorbeelden uit de getallentheorie, welke met de besproken eigenschappen zijn te behandelen zoals:

Oplossen diofantische vergelijkingen.

De vergelijking van Pell.

Gelijktijdig benaderen van alle wortels van een N -de graads Polynoom.
Kettingbreuk-benaderingen van de machtreksen (zie Elementaire- functies).

Kort na de aanschaf van mijn eerste PC, een KAYPRO-2 CP/M computer, heb ik kettingbreuk-ontwikkelingen gebruikt, bij het maken van dubbellenge elementaire functie-procedures.

De KAYPRO werd nl. geleverd met een S-BASIC compiler. Deze kende wel het dubbellenge variabelen type, maar beschikte niet over interne dubbellenge library routines. Deels als goede oefenstof, deels als curiositeit zijn deze routines in het hoofdstuk "Elementaire-functie benaderingen met kettingbreuken" opgenomen.

Kettingbreukontwikkelingen van speciale functies als o.a:

De Zetafunctie en de Gammafunctie.

Nieuwe ontwikkelde Besselfunctie-implementaties van: $J_0(x)$, $Y_0(x)$, $J_1(x)$, $Y_1(x)$, $K_0(z)$, $K_1(x)$.

Referenties.

- 1) Peter Henrici. Applied and computational complex analysis. Vol. I, pag. 553–661 en Vol. II pag. 473–640.
- 2) Heinz Rutishauser. Der Quotienten-Differenzen- Algorithmus. Z. Angew. Math. Physik, Vol. V, 233–251, 1954.
- 3) F. Schuh. Beknopte Hoogere Algebra. Pag. 617–665.
- 4) P. Wijdenes. Middel-Algebra. Deel II. Pag. 293–345.
- 5) C.G. van der Laan, N.M. Temme. CWI Tract 10. Calculation of special functions; the gamma function, the exponential integrals and errorlike functions.

6. QD- en Naderende Breuk Algorithmen

In dit hoofdstuk implementeren we de in het vorige hoofdstuk besproken QD-algorithmen (25) en (26) en algorithmen voor het bepalen van de functiewaarde $W_K = P_K(z)/Q_K(z)$ van een kettingbreuk.

In alle hierna te bespreken implementaties, gebruiken we de volgende standaard variabelendefinities: DEFINT $I - N$: DEFDBL $A - H, O - Z$. Verder nemen we aan, dat alle globale arrays die in de subs worden gebruikt, in het hoofdprogramma zijn gedefinieerd.

Als eerste bespreken we een QD-implementatie met twee dubbelarrays. Dit is niet de meest efficiënte aanpak. Dubbelarrays gebruiken nl. veel ruimte en ook blijft alle overvloedige tusseninformatie opgeslagen. Deze methode is echter wel uitermate geschikt om de structuur van het algoritme te illustreren. De volgende grootheden worden in het programma gebruikt:

Machtreescoëfficiënten	A_k	$k = 0, \dots, 2M$
Dubbel arrays $Q_{n,k}, E_{n,k}$	$n = 1, \dots, M$	$k = 0, \dots, 2M$
Kettingbreukcoëfficiënten	D_k	$k = 0, \dots, 2M$

```

D(0) = A(0): FOR I = 0 TO 2 * M - 1: Q(1, K) = A(K + 1)/A(K)
NEXT K: D(1) = -Q(1, 0)
FOR K = 0 TO 2 * M - 2: E(1, K) = Q(1, K + 1) - Q(1, K)
NEXT K: D(2) = -E(1, 0)
FOR N = 2 TO M 'Q, E relaties
  FOR K = 0 TO 2 * (M - N) + 1
    Q(N, K) = Q(N - 1, K + 1) * E(N - 1, K + 1)/E(N - 1, K)
  NEXT K: D(2 * N - 1) = -Q(N, 0)
  FOR K = 0 TO 2 * (M - N)
    E(N, K) = Q(N, K + 1) - E(N, K + 1) + E(N - 1, K)
  NEXT K: D(2 * N) = -E(N, 0)
NEXT N

```

Op basis van het bovenstaande is het programma: QDARRAYS.BAS (op de schijf) gemaakt waarmee het QD-schema kan worden uitgeprint.

Invoer: Het aantal termen in het schema: M .

Uitvoer: Het $Q_{N,K}$ en $E_{N,K}$ relatie-schema.

Het volgende programma is een modificatie van het bovenstaande. De nieuw berekende informatie wordt echter stelselmatig over de niet meer relevante gegevens heen geschreven. In plaats van de twee dubbelarrays zijn er nu slechts twee 1-dimensionale arrays nodig.

De volgende grootheden worden gebruikt.

Aantal termen van de reeks $2M$
 Machtreekscoëfficiënten $A_k \quad k = 0, \dots, 2M$
 Hulparrays $Q_k, E_k \quad k = 0, \dots, 2M$
 Kettingbreukcoëfficiënten $D_k \quad k = 0, \dots, 2M$

```

SUB QDANDERS(M ,A(), D() )
  DIM Q(2 * M + 1), E(2 * M + 1) :D(0) = A(0) 'Ket. coëff. in D()
  'Opbouw begincondities in kolom 1 en 2 v/d arrays Q en E
  FOR K = 0 TO 2 * M - 1: Q(K) = A(K + 1)/A(K)
  NEXT K: D(1) = -Q(0)
  FOR K = 0 TO 2 * M - 2: E(K) = Q(K + 1) - Q(K)
  NEXT K: D(2) = -E(0)
  FOR N = 2 TO M 'recursierelaties
    FOR K = 0 TO 2 * (M - N) + 1: Q(K) = Q(K + 1) * E(K + 1)/E(K)
    NEXT K: D(2 * N - 1) = -Q(0)
    FOR K = 0 TO 2 * (M - N): E(K) = Q(K + 1) - E(K + 1) + E(K)
    NEXT K: D(2 * N) = -E(0)
  NEXT N
END SUB

```

In het sub-programma QD worden slechts drie arrays en drie hulpvariabelen gebruikt. Ook het aantal rekenkundige operaties is minimaal, in latere toepassingen zullen we daarom meestal dit algoritme gebruiken.

De gebruikte variabelen zijn:

Machtreekscoëfficiënten $A_k \quad k = 0, \dots, N$
 Hulparraycoëfficiënten $B_k \quad k = 0, \dots, N$
 Kettingbreukcoëfficiënten $D_k \quad k = 0, \dots, N$
 Hulpvariabelen H_0, H_1, H_2

```

SUB QD (N , A(), D() )
  DIM B(N + 1)
  I = 0: WHILE I ≤ N: B(I) = 0: I = I + 1: WEND: D(0)=A(0):K = 1
  WHILE K ≤ N: I = 2: H0 = 0: H1 = B(1): B(1) = A(K)/A(K - 1)
    WHILE I ≤ K
      IF I MOD 2 = 0 THEN H2 =B(I - 1) - H1 + H0
      H0 = B(I): B(I) = H2
      IF I MOD 2 <> 0 THEN H2 = B(I - 1) * H1/H0
      H1 = B(I): B(I) = H2
      I = I + 1
    WEND: D(K) = -B(K): K = K + 1
  WEND
END SUB

```

Teneinde optredende afrondingsfouten te beperken, zullen we in latere toe-

passingen, als de machtrekscoefficienten a_k in formulevorm beschikbaar zijn, direct de verhouding a_k/a_{k-1} in het algoritme invoeren.

De Functiewaardebepaling van een Kettingbreuk.

Ook dit algoritme kan weer op diverse manieren worden geïmplementeerd. We bespreken twee vormen. De meest eenvoudige opzet is met twee hulpparrays en volgt weer exact de recursierelaties.

De gebruikte grootheden zijn:

Kettingbreukcoëff.	D_k	$k = 0, \dots, 2M$ met $2M = N$
Hulpparrays	P_k, Q_k	$k = 0, \dots, 2M$
Variabelen	Z	Argumentwaarde.
	WN	Functiewaarde:
	$FOUT$	een variabele die kan worden gebruikt, om de fout in de functiewaarde bij afbreken van de kettingbreuk na N termen te schatten.

```

SUB FWAARDE1 (M ,D(), Z, WN, FOUT)
  DIM P(2 * M + 1), Q(2 * M + 1) 'Begincondities
  P(0)=0: P(1)=D(0): Q(0)=1: Q(1)=1#: FOUT=D(0):K=2
  WHILE K ≤ 2 * M
    P(K) = P(K - 1) + Z * D(K - 1) * P(K - 2)
    Q(K) = Q(K - 1) + Z * D(K - 1) * Q(K - 2)
    FOUT = FOUT * D(K - 1) * Z: K = K + 1
  WEND: WN = P(2 * M)/Q(2 * M): FOUT = FOUT/Q(2 * M)
END SUB

```

In de volgende implementatie gebruiken we geen hulpparrays. Met behulp van slechts vijf hulpvariabelen wordt het algoritme uitgewerkt.

Kettingbreukcoëff.	$D()$	$k = 0, \dots, 2M$
Hulpvariabelen	$P0, P1, Q0, Q1, H1$	
Argumentwaarde Z , Functiewaarde WN , Afbreekfout $FOUT$.		

```

SUB FWAARDE(M, D(), Z, WN, FOUT)
  P0 = 0: P1 = D(0): Q0 = 1#: Q1 = 1#: FOUT = D(0): K = 1
  WHILE K ≤ 2 * M
    H1 = P1 + Z * D(K) * P0: P0 = P1: P1 = H1
    H1 = Q1 + Z * D(K) * Q0: Q0 = Q1: Q1 = H1
    FOUT = FOUT * D(K) * Z: K = K + 1
  WEND: WN = P1/Q1: FOUT = FOUT/Q1
END SUB

```

Uit de structuur van het QD-algoritme volgt, dat machtreksen die we in een

kettingbreuk willen ontwikkelen, altijd met een constante term moet beginnen en dat er geen coëfficiënten met de waarde nul mogen optreden.

In machtreksen met uitsluitend *even machten* lossen we dit op, door in de reeks: $f(X) = A_0 + A_2 \cdot X^2 + \dots + A_{2N} \cdot X^{2N}$, $X^2 = Z$ te stellen en vervolgens de machttreks $F(Z) = A_0 + A_2 \cdot Z + \dots + A_{2N} \cdot Z^N$ te converteren.

Machtreksen met uitsluitend oneven machten, van de vorm: $f(X) = A_1 \cdot X + A_3 \cdot X^3 + \dots + A_{2N+1} \cdot X^{2N+1}$, delen we eerst door X en substitueren vervolgens in de verkregen reeks weer $X^2 = Z$. We transformeren dus de machttreks $F(Z) = f(X^2)/X$.

Bij het berekenen van de functiewaarden kan gemakkelijk rekening worden gehouden met deze transformaties.

De boven geïmplementeerde algoritmen stellen ons in staat, een groot aantal praktijkproblemen van het reële variabelentype op te lossen. Voor de later te behandelen Numerieke Laplace-inversie-toepassingen hebben we echter deze algoritmen nodig gebaseerd op het gebruik van complexe getallen. Omdat de behandelde kettingbreukeigenschappen ook gelden voor complexe getallen, laten deze zich op de hierna beschreven wijze implementeren.

Bij de implementatie van het algoritme in complexe variabelen, gaan we uit van het reële QD sub-programma en vervangen de gebruikte variabelen door de volgende:

Machtreekscoëfficiënten: $A_k = AR_k + i AI_k \quad k = 0, \dots, N$
 Hulparraycoëfficiënten: $B_k = BR_k + i BI_k$
 Kettingbreukcoëfficiënten: $C_k = CR_k + i CI_k$ (Uitvoer)
 Hulpvariabelen: $Z1 = X1 + i Y1$ en XN .

$(H0 = H0R + i H0I: H1 = H1R + i H1I: H2 = H2R + i H2I)$

Substitutie en uitwerking met behulp van de complexe-getallen-eigenschappen levert het sub-programma CQD.

```
SUB CQD (N ,AR() ,AI() ,CR() ,CI() )
DIM BR(N + 1), BI(N + 1): CR(0) = AR(0): CI(0) = 0: I = 0
WHILE I ≤ N: BR(I) = 0: BI(I) = 0: WEND: K = 1
WHILE K ≤ N: I = 2: H0R = 0: H0I = 0: H1R = BR(1): H1I = BI(1)
  XN = AR(K - 1)^2 + AI(K - 1)^2
  BR(1) = (AR(K) * AR(K - 1) + AI(K) * AI(K - 1))/XN
  BI(1) = (AI(K) * AR(K - 1) - AR(K) * AI(K - 1))/XN
  WHILE I ≤ K
    IF I MOD 2 = 0 THEN
      H2R = BR(I - 1) - H1R + H0R: H2I = BI(I - 1) - H1I + H0I
      H0R = BR(I): H0I = BI(I): BR(I) = H2R: BI(I) = H2I
    ELSE
```

```

      X1 = BR(I - 1) * H1R - BI(I - 1) * H1I:
      Y1 = BR(I - 1) * H1I + BI(I - 1) * H1R
      H2R = (X1 * H0R + Y1 * H0I)/(H0R^2 + H0I^2)
      H2I = (Y1 * H0R - X1 * H0I)/(H0R^2 + H0I^2)
      H1R = BR(I): H1I = BI(I): BR(I) = H2R: BI(I) = H2I
    END IF
    I = I + 1
  WEND: CR(K) = -BR(K): CI(K) = -BI(K): K = K + 1
WEND
END SUB

```

Op analoge wijze laat zich een functiewaarde- bepaling implementeren. De foutschatting is achterwege gelaten in dit programma.

De betekenis van de variabelen is als volgt:

Hulpvariabelen: $P0 = P0R + i P0I$ $Q0 = Q0R + i Q0I$
 $P1 = P1R + i P1I$ $Q1 = Q1R + i Q1I$
 $H0 = H0R + i H0I$ $Z1 = X1 + i Y1$

Invoer: Graad N : Argument $Z = X + i Y$
 Kettingbreukcoëff.: $C_k = CR_k + i CI_k$ ($k = 0, 1, \dots, N$)

Uitvoer: Functiewaarde $W = WR + i WI$.

```

SUB CWAARDEKET( N, CR(), CI(), X, Y, WR, WI)
  P0R = 0: P0I = 0: P1R = CR(0): P1I = CI(0) 'Begincondities
  Q0R = 1#: Q0I = 0: Q1R = 1#: Q1I = 0 'Begincondities
  K = 1
  WHILE K ≤ N
    X1 = CR(K) * P0R - CI(K) * P0I
    Y1 = CR(K) * P0I + CI(K) * P0R
    H0R = P1R + X * X1 - Y * Y1: H0I = P1I + Y * X1 + X * Y1
    P0R = P1R: P0I = P1I: P1R = H0R: P1I = H0I
    X1 = CR(K) * Q0R - CI(K) * Q0I
    Y1 = CR(K) * Q0I + CI(K) * Q0R
    H0R = Q1R + X * X1 - Y * Y1: H0I = Q1R + Y * X1 + X * Y1
    Q0R = Q1R: Q0I = Q1I: Q1R = H0R: Q1I = H0I: K = K + 1
  WEND
  QN = (Q1R^2 + Q1I^2)
  IF QN = 0 THEN PRINT "Delen door nul ":EXIT SUB
  WR = (P1R * Q1R + P1I * Q1I)/QN
  WI = (P1I * Q1I - P1R * Q1I)/QN
END SUB

```

Het gebruik van bovenstaande algoritmen is sterk van het type toepassing afhankelijk. Bij een complexe machtreeks met reële coëfficiënten gebruiken we het reële QD-algorithme om de reële kettingbreuk-coëfficiënten te bepalen. De functiewaardeberekening voor de complexe argumentwaarden moet echter met behulp van SUB CWAARDEKET worden bepaald. In de latere hoofdstukken Besselfuncties en Numerieke Laplace-inversie-toepassingen zullen we uitvoerig gebruik maken van deze programma's.

In het op de schijf aanwezige programma QD-ALG.BAS is geïllustreerd hoe de berekening kan worden aangepast voor machtreeksen met uitsluitend:

even machten, als b.v. $\cos(X)$, $J_0(X)$ etc.

oneven machten, b.v. $\sin(X)$, $\tan(X)$, $\log(1 + X)$, $J_1(X)$ etc.

normale machtreeksen (alle machten aanwezig), b.v. $\exp(X)$.

Invoer gegevens:

N aantal in te voeren termen machtreeks.

$A(k)$ $k = 0, 1, \dots, N$, de machtreekscoëfficiënten naar keuze via toetsenbord of subroutine.

X de argumentwaarde waarvoor de functiewaarde F wordt bepaald.

Let op het convergentie interval!

Programma uitvoer:

$C(k)$ $k = 0, 1, \dots, N$ kettingbreukcoëfficiënten.

X argumentwaarde.

F functiewaarde.

7. Benaderingen met Kettingbreuken van de Elementaire Functies

Het is bekend dat de machtreeks:

$$\log(1 + X) = 1 - X/1 + X^2/2 - X^3/3 + X^4/4 + \dots$$

op het interval $[0, 1]$ zeer langzaam convergeert en er meer dan 10000 termen nodig zijn om een nauwkeurigheid van 14 cijfers te garanderen. Kettingbreukontwikkelingen zijn uitermate geschikt om het convergentieproces te versnellen, voor $\log(1 + X)$ zijn maximaal 30 wijzergetallen nodig voor een nauwkeurigheid van 15 cijfers.

De elementaire functieroutines in computertalen worden daarom meestal geïmplementeerd als kettingbreuk (of Chebyshev polynoom) ontwikkelingen. Een machtreeks-voorstelling van een functie is alleen in een Kettingbreuk te ontwikkelen als de reeks begint met een constante term $\neq 0$ en er ook verder geen termen in de reeks ontbreken.

In de ontwikkeling van $\cos(X)$ ontbreken alle oneven machten:

$$\cos(X) = 1 - X^2/2! + X^4/4! - X^6/6! + X^8/8! - \dots$$

Teneinde een reeks te krijgen die wel aan de gestelde eisen voldoet, maken we de transformatie $Z = X^2$ en bepalen de kettingbreuk van:

$$F(Z) = 1 - Z/2! + Z^2/4! - Z^3/6! + \dots$$

Bij het bepalen van de functiewaarde met naderende breuken substitueren we nu X^2 in plaats van X immers $F(X^2) = \cos(X)$.

Een voorbeeld van een machtreeks met alleen oneven machten is:

$$\sin(X) = X - X^3/3! + X^5/5! - X^7/7! + \dots$$

Dit type machtreeksen delen we eerst door X , voordat we in het rechterlid $X^2 = Z$ stellen. We bepalen dus de kettingbreukontwikkeling van:

$$F(Z) = 1 - Z/3! + Z^2/5! - \dots$$

en beschouwen

$$X.F(X^2) = \sin(X)$$

$\exp(X)$ is een machtreeks waarin geen machten ontbreken:

$$\exp(X) = 1 + X/1! + X^2/2! + X^3/3! + X^4/4! + \dots$$

Het QD-algorithme is direct toepasbaar.

Voor $\exp(X)$ en vele andere functies, die te formuleren zijn als bepaalde types hypergeometrische reeksen, heeft reeds Gauss de kettingbreukcoëfficiënten in formulevorm afgeleid. Dit heeft het grote voordeel dat het QD-algoritme niet wordt gebruikt en afrondingsfouten grotendeels kunnen worden vermeden. In de hierna te behandelen subroutines is daarom steeds gebruik gemaakt van in de literatuur bekende formules voor de kettingbreukcoëfficiënten.

Voor $\exp(X)$ is $D_0 = 1$, $D_1 = -1$ en voor $k = 1, 2, \dots$ gelden de formules:

$$D_{2k+1} = k/\{(2k-1).(2k)\} \quad \text{en} \quad D_{2k+2} = (-1)^k/\{(2k).(2k+1)\}$$

dit levert:

$$\begin{aligned} D(0) = 1: D(1) = -1: D(2) = 1/2: D(3) = -1/6: D(4) = 1/6 \\ D(5) = -1/10: D(6) = 1/10: D(7) = -1/14: D(8) = 1/14: \\ D(9) = -1/18: D(10) = 1/18: D(11) = -1/22: D(12) = 1/22 \end{aligned}$$

Omdat in de literatuur veelvuldig gebruik wordt gemaakt van Gegeneraliseerde Hypergeometrische reeksen en wij hier ook regelmatig gebruik van gaan maken, geven we de volgende definitie.

Laat p en q twee gehele getallen en a_1, a_2, \dots, a_p en b_1, b_2, \dots, b_q reële of complexe getallen zijn {met $(y)_n = y.(y+1)...(y+n-1)$, $n = 1, 2, \dots$ } dan wordt, mits $(b_i)_n \neq 0$, $i = 1, 2, \dots, q$, de formule van de machtreeks:

$$F(X) = \sum_{n=0}^N \frac{(a_1)_n.(a_2)_n \dots (a_p)_n X^n}{(b_1)_n.(b_2)_n \dots (b_q)_n.n!} \quad (\text{resp. } N = \infty)$$

een gegeneraliseerde hypergeometrische reeks genoemd.

In de wiskunde wordt voor $F(X)$ de volgende verkorte notatie gebruikt:

$$F(X) = {}_pF_q(a_1, a_2, \dots, a_p; b_1, b_2, \dots, b_q; X) .$$

(Let op: komma's scheiden de parameters in de teller, puntkomma's scheiden teller en noemer en argument).

In de meest interessante toepassingen zijn $p = 0, 1$ of 2 en $q = 1$ of 2 .

In deze notatie geldt voor de veel gebruikte functies:

- | | | | |
|----|----------------|--------------------------------|------------------------|
| 1) | $\sin(X)/X$ | $= {}_0F_1(3/2; -X^2/4)$ | $-\infty < X < \infty$ |
| 2) | $\cos(X)$ | $= {}_0F_1(1/2; -X^2/4)$ | $-\infty < X < \infty$ |
| 3) | $\exp(X)$ | $= {}_1F_1(1; 1; X)$ | $-\infty < X < \infty$ |
| 4) | $\log(1+X)/X$ | $= {}_2F_1(1, 1; 2; -X)$ | $0 \leq X \leq 1$ |
| 5) | $\arctan(X)/X$ | $= {}_2F_1(1/2, 1; 3/2; -X^2)$ | $-1 \leq X \leq 1$ |
| 6) | $(1+X)^{-k}$ | $= {}_2F_1(k, 1; 1; -X)$ | $-1 < X < 1$ |

Van het hypergeometrische type ${}_0F_1(p; z)$ in 1) en 2) zijn geen expliciete formules beschikbaar om de kettingbreukcoëfficiënten te genereren. De van machtreeken afgeleide kettingbreukontwikkelingen convergeren op een beperkt interval meestal zeer goed. In de meeste gevallen zijn minder dan 20 termen ruim voldoende voor een nauwkerigheid van 15 cijfers.

In de oudere talen o.a. GW-Basic waren de elementaire functies: $\sin x$, $\cos x$, $\arctan x$, $\log(1+x)$ en $\exp(x)$ gedefinieerd op het interval $[-2.E-38, 2.E+38]$. Door slim gebruik te maken van de respectieve functie-eigenschappen, is het mogelijk programma's te ontwikkelen welke aan deze eis voldoen.

De noodzakelijke transformaties veroorzaken vooral bij de zeer grote en de zeer kleine argumentwaarden, door accumulatie van afron dingsfouten bij sommige functies, wel enig verlies aan nauwkeurigheid.

We gebruiken de functie-eigenschappen van:

a. $\sin(x)$:

Voor $x < 0$ gebruiken we $\sin(x) = -\sin(|x|)$.

Beschouw verder x positief. We gebruiken de periodiciteitseigenschap $\sin(x + k.\pi) = (-1)^k \cdot \sin(x)$ om het argument x te transformeren.

Eerst naar het interval $[0, \pi]$ en vervolgens naar $[0, \pi/2]$.

b. $\cos(x)$:

Via de transformatie $X = x - \pi/2$, bepalen we $\cos(x)$ als $\sin(X)$.

c. $\exp(x)$:

Test weer op $x < 0$, gebruik voor $x < 0$ $\exp(-|x|) = 1/\exp(|x|)$.

Beschouw verder weer $x \geq 0$.

Splits het argument $x \geq 0$ in een geheel deel $k = \text{int}(|x|)$ plus de fractie $X = |x| - k$ op $[0, 1]$, dus $\exp(|x|) = \exp(k) * \exp(X)$.

Stellen we $\text{EXP}K = 1\#$ en $e = 2.71828182845904523$, dan is $\text{EXP}K = \exp(k)$ te genereren als: FOR $I = 1$ to K : $\text{EXP}K = \text{EXP}K * e$: NEXT K .

d. $\arctan(x)$:

Test op $x < 0$: Voor $x < 0$ is $\arctan(x) = -\arctan(|x|)$ voor $|x| > 1$ geldt:

$$\arctan(x) + \arctan(1/x) = \begin{cases} \pi/2 & \text{voor } x \text{ positief} \\ -\pi/2 & \text{voor } x \text{ negatief} \end{cases}$$

e. $\log(X)$:

$\log(X) = \log_e(X)$ voor $X > 0$.

Test eerst of $X < 1$ is, voor $X < 1$ gebruiken we de eigenschap:

$\log(X) = -\log(1/X)$ en brengen zo dit geval terug naar $X > 1$.

Voor $X > 1$ stellen we: $X = a.10^k$, k geheel en $1 \leq a < 2$.

Stel $x = a - 1$ met $0 < x \leq 1$, dus is $\log(a) = \log(1+x)$.

Bepaal k door X te delen door 10 en een test op $X/10 < 2$.

Met behulp van $\log(10) = 2.30258509299404568$ bepalen we $\text{Log}(X)$ als:

$$\log(X) = \text{sign}(X) * \{\log(a) + k.\log(10)\}.$$

Op basis van bovenvermelde eigenschappen en kettingbreuk-ontwikkelingen zijn de subs in het onderstaande programma, als S-Basic dubbellenget-procedures ontwikkeld, geïmplementeerd en gebruikt.

Alle huidige computertalen beschikken nu over een goede dubbellenget-functie-subroutine-bibliotheek. Deze subs dienen dus uitsluitend nog als illustratie voor het gebruik van kettingbreukbenaderingen.

In het op de schijf aanwezige programma ELEMSUB.BAS kunnen via een menu de verschillende functies worden gekozen. Voor elke gekozen argumentwaarde wordt de volgende Output geleverd:

Argument Interne-functiewaarde Berekende-functiewaarde.

Door het aantal te gebruiken kettingbreukencoëfficiënten N in de subs te wijzigen kan worden nagegaan hoe de nauwkeurigheid van de functiewaarde hierdoor wordt beïnvloed. Met behulp van het in het vorige hoofdstuk behandelde programma QD-ALG.BAS, kan men inzicht krijgen hoe groot de afrondingsfouten zijn als het QD-algoritme wordt gebruikt.

8. Simultaan Benaderen van de Wortels van Hogeregraadsvergelijkingen

Het QD-algorithme is in principe geschikt voor het simultaan benaderen van alle enkelvoudige wortels van hogeregraads polynomen (en ook voor het bepalen van de kleinste wortels van een gehele functie).

Het algorithme in de besproken vorm blijkt numeriek instabiel te zijn. Een modificatie, bekend als de "Progressive form"¹⁾ is minder instabiel en heeft voor het gestelde doel een beter aangepaste vorm. Dit algorithme werkt volgens een rijgewijze, in plaats van de besproken kolomsgewijze opbouw van de recursierelaties.

In het geval dat alle Hankeldeterminanten $\neq 0$ zijn, bestaat het QD-schema (dit is altijd het geval, als van een polynoom $p(z)$ alle coëfficiënten $\neq 0$ zijn). Indien er coëfficiënten in het polynoom nul zijn, vermeerder (verminder) dan de wortels van $p(z)$ met een geschikt gekozen getal a , door invoering van een nieuwe variabele $z^* = z - a$. Bepaal de nieuwe coëfficiënten van $p^*(z^*) = p(a + z^*)$ met behulp van een Taylorreeksontwikkeling of met het vroeger behandelde Horner-algorithme.

Voor een N -de-graads polynoom:

$$A_0.Z^N + A_1.Z^{N-1} + \dots + A_N = 0 \quad (\text{alle } A_n \neq 0, n = 0, 1, \dots, N)$$

gelden voor het rijgewijze opbouwschema, de recursierelaties:

$$q_{n+1,k} = (e_{n,k} - e_{n+1,k-1}) + q_{n,k}$$

$$e_{n+1,k} = e_{n,k} \cdot q_{n,k+1} / q_{n+1,k}$$

met de beginvoorwaarden

$$q_{0,1} = -A_1/A_0, \quad q_{1-k,k} = 0, \quad k = 2, 3, \dots, m$$

$$e_{1-k,k} = A_{k+1}/A_k, \quad k = 1, 2, \dots, m-1$$

en de zijcondities:

$$e_{n,0} = e_{n,m} = 0, \quad n = 1, 2, \dots$$

Het QD-schema volgens deze rijgewijze opbouw, illustreren we aan een polynoom van de vierde graad:

$$A_0.Z^4 + A_1.Z^3 + A_2.Z^2 + A_3.Z + A_4 = 0 \quad (A_k \neq 0, k = 1, \dots, 4)$$

$$\begin{array}{cccccc}
& -A_1/A_0 & & 0 & & 0 & & 0 \\
0 & & A_2/A_1 & & A_3/A_2 & & A_4/A_3 & \\
& q_{1,1} & & q_{0,2} & & q_{-1,3} & & \\
0 & & e_{1,1} & & e_{0,2} & & e_{-1,3} & \\
& q_{2,1} & & q_{1,2} & & q_{0,3} & & q_{-1,4} \\
0 & & e_{2,1} & & e_{1,2} & & e_{0,3} &
\end{array}$$

Dit schema wordt eenvoudiger als we op de lege plaatsen in de 1-ste rij de niet-nul-elementen van de 2-de rij er tussen schuiven. Analoog voor de volgende rijen-paren (in 3-de rij de 4-de invoegen etc). Het zo verkregen schema heeft een geschikte vorm voor computer-implementatie en ziet er als volgt uit:

$$\begin{array}{cccccccc}
0 & -A_1/A_0 & A_2/A_1 & 0 & A_3/A_2 & 0 & A_4/A_3 & 0 & 0 \\
0 & q_{1,1} & e_{1,1} & q_{0,2} & e_{0,2} & q_{-1,3} & e_{-1,3} & q_{-2,4} & 0 \\
0 & q_{2,1} & e_{2,1} & q_{1,2} & e_{1,2} & q_{0,3} & e_{0,3} & q_{-1,4} & 0 \\
0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\
0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots &
\end{array}$$

Voor een polynoom met alleen enkelvoudige wortels gelden de volgende belangrijke limietstellingen¹⁾:

$$\lim_{n \rightarrow \infty} q_{n,k} = Z_k \quad \text{en} \quad \lim_{n \rightarrow \infty} e_{n,k} = 0 .$$

De betekenis hiervan is dat alle $q_{k,n}$ kolommen convergeren naar de respectieve wortels Z_k , $k = 1, \dots, n$, terwijl alle $e_{k,n}$ kolommen naar nul gaan. In het geval dat we het polynoom laten beginnen met de constante term convergeren de $q_{k,n}$ kolommen naar de wortel $1/Z_k$, terwijl de $e_{k,n}$ kolommen weer naar nul gaan. Deze vorm is geschikt om de in absolute waarde kleinste wortels van een gehele functie te benaderen.

Heeft $p(z)$ echter wortels met gelijke modulus (toegevoegd complexe wortels) dan convergeren de $e_{k,n}$ kolommen niet meer naar nul. Met een procedure, die berust op de constructie van Hadamard polynomen¹⁾, zijn uit de laatste twee stappen van het iteratieschema ook de toegevoegd-complexe wortels te bepalen. Voor het meest voorkomende geval van enkelvoudige toegevoegd-complexe wortels, gelden dan de limietstellingen¹⁾:

$$\lim_{n \rightarrow \infty} (q_{n,k+1} + q_{n,k+2}) = G_k \quad \text{en} \quad \lim_{n \rightarrow \infty} q_{n+1,k+1} \cdot q_{n,k+2} = E_k .$$

Met behulp van de parameters G_k en E_k construeren we de vierkantsvergelijking:

$$Z^2 - G_k \cdot Z + E_k = 0$$

waaruit de toegevoegd-complexe wortels kunnen worden bepaald.

We illustreren het geval van enkelvoudige wortels aan de derdegraads vergelijking:

$$Z^3 - 6.Z^2 + 11.Z - 6 = (Z - 3).(Z - 2).(Z - 1) = 0 .$$

Vanwege de zijcondities, zijn de 0^{de} en 2^m^{de} kolom in het schema altijd nul en worden in de volgende voorbeelden weg gelaten.

De oneven kolommen in het iteratie schema zijn q waarden en convergeren naar de wortels. De even kolommen zijn e waarden en gaan naar nul.

$q_{n,1}$	$e_{n,1}$	$q_{n,2}$	$e_{n,2}$	$q_{n,3}$
6.000000	-1.833333	0.000000	-0.545546	.000000
4.166666	-.566666	1.287879	-.231016	.545454
3.600000	-.255555	1.623529	-.110486	.776471
3.344444	-.135142	1.768599	-5.5408e-2	.886956
3.202930	-7.7832e-2	1.848332	-2.2850e-2	.942365
3.00004	-7.78e-5	1.999734	-4.767e-7	.999999

De laatste rij correspondeert met $n = 20$.

De vergelijking: $Z^3 - 7.Z^2 + 24.Z - 18 = 0$ gebruiken we om te illustreren, hoe het proces verloopt als er reële en complexe wortels zijn.

$q_{n,1}$	$e_{n,1}$	$q_{n,2}$	$e_{n,2}$	$q_{n,3}$
7.0000	-3.428572	0.00000	-0.75	0.0000
3.571429	-2.571429	2.678572	-0.21	0.75
0.999999	-12.96	5.04	-0.05	0.96
-11.96	-19.46167	17.96	-2.227e-3	1.0
7.50167	-3.90158	-1.50389	-1.484e-3	1.000222
3.6	-2.6	2.4	5.32e-19	1.000
1.0	-13.00	5.00	1.07e-19	1.00

De laatste rijen corresponderen met n , respectievelijk $n+1$. In het iteratieproces gaat $e_{n,2}$ weer naar nul, dus $q_{n,1} = 1$ is de reële wortel $z = 1$.

De kolom $e_{n,1}$ gaat niet naar nul, de kolommen $q_{n,1}$ en $q_{n,2}$ aan weerszijden hiervan in opvolgende n^{de} en $(n+1)^{\text{de}}$ rijen leveren de coëfficiënten:

$$G_1 = q_{n+1,1} + q_{n+1,2} = 1.00 + 5.00 = 6$$

$$E_1 = q_{n,1} * q_{n+1,2} = 5 * 3.60 = 18 ,$$

in de vierkantsvergelijking:

$$U^2 - 6.U + 18 = 0 ,$$

met de wortels:

$$U_1 = 3 + 3i \quad \text{en} \quad U_2 = 3 - 3i .$$

Uit het voorafgaande is de SUB PROGRESSIEFQD ontwikkeld.

```

DEFINT I - N: DEFDBL A - H, O - Z
SUB PROGRESSIEFQD (K, A(), B(), C(), D())
  FOR L = 2 TO K: A(2 * L - 2) = C(L) / C(L - 1)
  A(2 * L - 1) = 0: NEXT L
  A(0) = 0: A(1) = -C(1) / C(0): A(2 * K) = 0
  PRINT "Voer aantal stappen QD-algorithme in: "; :
  INPUT "N="; N
  CLS: PRINT "Resultaat QD-algorithme na "; N; " stappen."
  FOR M = 1 TO N: B(0) = 0
    FOR L = 1 TO 2 * K - 1 STEP 2
      B(L) = A(L + 1) - A(L - 1) + A(L): NEXT L
    FOR L = 2 TO 2 * K STEP 2
      B(L) = A(L) * B(L + 1) / B(L - 1): NEXT L
    B(2 * K) = 0
    FOR L = 0 TO 2 * K
      IF M = N - 1 THEN D(L) = B(L)
      A(L) = B(L): B(L) = 0: NEXT L
    FOR L = 1 TO 2 * K - 1
      IF M = N - 1 OR M = N THEN PRINT A(L); : ELSE GOTO 10
    NEXT L: PRINT : PRINT : PRINT
10 NEXT M
  M=1: PRINT "De nulpunten zijn: ": PRINT
  FOR L = 2 TO 2 * K STEP 2
    IF ABS(A(L)) < .001 AND L ≤ 2 * K THEN
      B(M) = A(L - 1): C(M) = 0: GOTO 20
    ELSE
      A = A(L - 1) + A(L + 1): B = D(L - 1) * A(L + 1)
      'Toegevoegd-complexe wortels volgen uit: U2 - A.U + B = 0
      E = SQR(ABS(A * A / 4 - B)): B(M) = A / 2#: C(M) = E
      B(M + 1) = A / 2#: C(M + 1) = -E: M = M + 1: L = L + 2:
    END IF
  20 M = M + 1
  NEXT L
END SUB

```

Het complete programma NULPUNT.BAS is aanwezig op de schijf. Het is geschikt om alle enkelvoudige wortels (de complexe inbegrepen) van hogere-

graads polynomen met reële coëfficiënten te bepalen. De in absolute waarde kleinste wortels zijn het meest nauwkeurig.

In het hoofdstuk Randwaarde-problemen wordt behandeld, hoe met behulp van differentietechnieken, de coëfficiënten van het bijbehorende Karakteristieke polynoom, uit een differentie-benadering in N roosterpunten van de lineaire 2-de orde differentiaalvergelijking met bijbehorende randcondities, kan worden bepaald. Dit is uitgewerkt in het sub-programma:

KARAKTERISTIEKPOLYNOOM ($N\%$, $B\#()$, $D\#()$, $P\#()$, $C\#()$).

Het sub-programma PROGRESSIEQD is vereenvoudigd, omdat symmetrische tridiagonaal-matrixeigenwaarde-problemen alleen reële-eigenwaarden bezitten. Deze subs zijn opgenomen in het programma KARAKTER.BAS op de schijf.

Het programma bevat 4-lineaire differentiaalvergelijking-subroutines voor o.a.: $\cos(x)$, Besselfunctie $J_0(x)$ en twee van het confluent hypergeometrische-type (zie oplossen randwaardeproblemen).

Teneinde de nauwkeurigheid van deze methode te testen, is ook de Besselfunctie $J_0(x)$ als machtreeks geïmplementeerd en kunnen de verschillende aanpakken worden vergeleken.

Referenties.

- 1) Peter Henrici. Applied and computational complex analysis. Vol. I.
- 2) Heinz Rutishauser. Der Quotienten-Differenzen-Algorithmus. Z. Angew. Math. Physik, 233-251 Vol. V, 1954.

9. Enkele Voorbeelden uit de Getallentheorie

Het meest klassieke voorbeeld uit de theorie der kettingbreuken is het bepalen van de Grootste Gemene Deler (GGD) G , van twee gehele getallen K_0 en K_1 ($K_0 > K_1$). Als K_0 en K_1 onderling ondeelbaar zijn is $G = 1$.

In de rekenkunde bepalen we G , door het toepassen van het rekenschema van Euclides.

We beschouwen het voorbeeld : 224/99

$$\begin{array}{r}
 99 \) \ 224 \quad (\underline{2} \\
 \underline{198} \\
 26 \) \ 99 \quad (\underline{3} \\
 \underline{78} \\
 21 \) \ 26 \quad (\underline{1} \\
 \underline{21} \\
 5 \) \ 21 \quad (\underline{4} \\
 \underline{20} \\
 1 \) \ 5 \quad (\underline{5} \\
 \underline{5} \\
 0
 \end{array}
 \quad G = 1 \text{ ondeelbaar.}$$

De onderstreepte cijfers (2, 3, 1, 4, 5, 1) zijn de wijzergetallen van de noemers in de kettingbreukontwikkeling.

Formeel is dit proces als volgt te formuleren: Bepaal twee rijen gehele getallen:

$$K_2, K_3, \dots, K_m, \dots \quad \text{en} \quad B_0, B_1, \dots, B_m, \dots$$

zodanig dat voldaan is aan de relaties:

$$\begin{array}{lll}
 K_0 = B_0 \cdot K_1 + K_2 & \text{resp.} & K_0/K_1 = B_0 + K_2/K_1 & (K_2 < K_1) \\
 K_1 = B_1 \cdot K_2 + K_3 & \text{resp.} & K_1/K_2 = B_1 + K_3/K_2 & (K_3 < K_2) \\
 K_2 = B_2 \cdot K_3 + K_4 & \text{resp.} & K_2/K_3 = B_2 + K_4/K_3 & (K_4 < K_3)
 \end{array}$$

en in het algemeen

$$K_m = B_m \cdot K_{m+1} + K_{m+2}, \quad K_m/K_{m+1} = B_m + K_{m+2}/K_{m+1} \quad (K_{m+2} < K_{m+1})$$

en tenslotte $K_n B_n \cdot K_{n+1} = B_n \cdot G$.

Uit deze betrekkingen volgt direct de kettingbreuk:

$$K_0/K_1 = B_0 + \frac{1}{B_1 + \frac{1}{B_2 + \frac{1}{\ddots + \frac{1}{B_{n-2} + \frac{1}{B_{n-1} + 1/B_n}}}}}$$

$$K_0/K_1 = B_0 + \frac{1}{|B_1|} + \frac{1}{|B_2|} + \dots + \frac{1}{|B_n|} .$$

Het proces breekt af. Dit is karakteristiek voor een rationale breuk.

Omdat we zijn uitgegaan van een onechte breuk K_0/K_1 , resulteert dit in een onzuivere kettingbreuk.

Uit het bovenstaande blijkt, dat met behulp van het Euclides-schema een speciale kettingbreuk wordt gegenereerd, met alle partiële tellers 1. Noteren we in het bovenstaande rekenschema voor $m = 0, 1, \dots, n$, de rij rationale getallen $R_0 = K_0/K_1$, $R_m = K_m/K_{m+1}$, dan volgt dat:

$$R_0 = B_0 + 1/R_1, \quad R_1 = B_1 + 1/R_2, \quad R_m = B_m + 1/R_{m+1} \quad \text{met} \quad R_n = B_n .$$

Hieruit blijkt dat de wijzergetallen ook uit de volgende relaties kunnen worden bepaald:

$$R_0 = K_0/K_1, \quad B_m = [R_m], \quad 1/R_{m+1} = R_m - [R_m] .$$

Met $[R_m]$ wordt bedoeld het aantal gehelen uit R_m .

Deze betrekkingen zullen we later gebruiken.

Met behulp van de vroeger gegeven definities, kunnen we uit de kettingbreuk-coëfficiënten, de opvolgende naderende breuken bepalen:

$$W_1 = P_1/Q_1 = A_1/B_1 \quad W_2 = P_2/Q_2 \dots$$

en dus $K_0/K_1 = P_n/Q_n$ berekenen.

Alle partiële tellers zijn 1, dus geldt voor de determinantformule:

$$P_m \cdot Q_{m-1} - P_{m-1} \cdot Q_m = (-1)^m, \quad (m = 1, 2, \dots, n) .$$

Deze betrekking kan worden gebruikt voor het oplossen van de diofantische vergelijking:

$$K \cdot x - L \cdot y = 1 .$$

Hierin zijn K en L twee priemgetallen ($K > L$). We ontwikkelen K/L in een eindige kettingbreuk en bepalen de naderende breuk $P_n/Q_n = K/L$.

Stel $P_n = K$ en $Q_n = L$, substitutie in de determinantvergelijking: $P_n \cdot Q_{n-1} - P_{n-1} \cdot Q_n = (-)^n$, levert de betrekking: $K \cdot Q_{n-1} - L \cdot P_{n-1} = (-)^n$. Voor $n =$ even volgt hieruit de basisoplossing: $x_0 = Q_{n-1}$, $y_0 = P_{n-1}$ en voor $n =$ oneven: $x_0 = L - Q_{n-1}$ en $y_0 = K - P_{n-1}$.

Omdat K en L wederkerig priem zijn, volgt uit de basis-oplossing x_0, y_0 , direct de algemene oplossing:

$$x = x_0 + K \cdot m \quad y = y_0 + L \cdot m \quad (m \text{ willekeurig geheel}) .$$

Een andere toepassing is het ontwikkelen van een irrationaal getal, in een oneindig voortlopende kettingbreuk.

De periodieke of repeterende kettingbreuken, vormen hiervan een speciale ondergroep.

Een oneindig voortlopende kettingbreuk heet periodiek of repeterend, als van af een zeker rangnummer, een groep wijzergetallen zich periodiek gaat herhalen.

$$G = B_0 + \frac{1}{|B_1|} + \frac{1}{|B_2|} + \dots + \frac{1}{|B_n|} + \frac{1}{|2 \cdot B_0|} + \frac{1}{|B_1|} + \dots .$$

De groep wijzergetallen $B_1, B_2, \dots, B_n, 2 \cdot B_0$, herhaalt zich periodiek, de index n noemt men de periode.

Ieder getal Z van de vorm $D \pm \sqrt{X}$ (D en X geheel en X niet het kwadraat van een geheel getal), is te schrijven als wortel van een vierkantsvergelijking met gehele coëfficiënten $(Z - D)^2 = X$, en kan naar een periodieke kettingbreuk worden ontwikkeld.

Beschouwen we het speciale geval dat $D = [\sqrt{X}]$, dan hebben we de mogelijkheid om \sqrt{X} naar een periodieke kettingbreuk te ontwikkelen.

De wijzergetallen B_m ($n = 0, 1, \dots, n$), kunnen worden gevonden door telkens oplossen van een vierkantsvergelijking, maar nog veel eenvoudiger met het eerder behandelde gemodificeerde rekenschema van Euclides.

Dit verloopt als volgt:

We starten met de beginconditie:

$$R_0 = \sqrt{X}$$

en de vervolrelaties:

$$B_m = [R_m] \quad 1/R_{m+1} = R_m - [R_m] \quad m = 0, 1, \dots, n .$$

Als voorbeeld nemen we $\sqrt{6}$:

$$R_0 = \sqrt{6} = 2.442 \quad B_0 = [\sqrt{6}] = 2, \quad R_1 = 1/(\sqrt{6} - 2) = (\sqrt{6} + 2)/2$$

$$B_1 = [R_1] = 2 \quad R_2 = 2/(\sqrt{6} - 2) = \sqrt{6} + 2 \quad B_2 = 4$$

$$R_3 = 1/(\sqrt{6} - 2) = R_1 \quad \text{en dus} \quad R_4 = R_2 .$$

Continue voortzetting levert algemeen:

$$B_1 = B_3 = B_5 = \dots = 2, \quad B_2 = B_4 = B_6 = \dots = 4.$$

Op een analoge wijze als bij de diofantische vergelijkingen kunnen ook oplossingen van de vergelijking van Pell:

$$X^2 - L.Y^2 = 1 \quad (\text{voor } L \text{ geheel en geen kwadraat}) \quad (\text{A})$$

worden gevonden.

Aanwijzing: Bepaal de kettingbreuk $\sqrt{L} = P_n/Q_n$ en leidt de volgende gemodificeerde determinantrelatie af:

$$P_n^2 - L.Q_n^2 = (-1)^{n-1} \quad (\text{waarin de index } n \text{ de periode is}).$$

Hieruit volgt dat:

Voor $n = \text{oneven}$: $X = P_n$, $Y = Q_n$ een oplossing is.

Voor $n = \text{even}$ substitueren we $n = 2m + 1$ in (A), dan volgt uit: $P_{2m+1}^2 - L.Q_{2m+1}^2 = 1$, de oplossing: $X = P_{2m+1}$ $Y = Q_{2m+1}$.

Als laatste voorbeeld ontwikkelen we de wortels uit de vierkantsvergelijking: $X^2 - B.X - A = 0$ in een kettingbreuk (A en B geheel).

De wortels X_1 , X_2 voldoen aan de symmetrierelaties:

$$X_1 + X_2 = B \quad \text{en} \quad X_1 \cdot X_2 = -A,$$

eliminatie van X_2 leidt direct tot:

$$X_1 = -A/(B - X_1) = -\frac{A}{|B} + \frac{A}{|B} + \dots + \frac{A}{|B} + \dots$$

$$X_2 = B - X_1.$$

In het programma GETALQB.BAS (op de schijf), zijn de besproken onderwerpen geïmplementeerd. Om ook diofantische vergelijkingen te kunnen oplossen voor grote priemgetallen, is een algoritme opgenomen om priemgetallen te bepalen. De wijzergetallen van \sqrt{N} zijn te bepalen met de vierkantsvergelijkingsaanpak SUB WIJZERGETALLEN.

In het menu zijn de verschillende keuzemogelijkheden aangegeven.

Referenties.

Peter Henrici. Applied an computational complex analysis. Vol. II.

F. Schuh. Beknopte Hoogere Algebra.

P. Wijdenes. Middel-Algebra II.

10. Enkele Speciale Functies

De Riemann Zetafunctie en de Gammafunctie.

In vele toepassingen, o.a. bij de Besselfuncties in het volgende hoofdstuk, spelen de gammafunctie en de daaruit afgeleide ψ -functie een belangrijke rol. De gammafunctie kan op verschillende manieren worden gedefinieerd.

Een bekende integraal-voorstelling is:

$$\Gamma(z) = \int_0^{\infty} e^{-t} \cdot t^{z-1} \cdot dt \quad \text{Re } z > 0 .$$

Enkele eigenschappen van $\Gamma(z)$ zijn:

$$\Gamma(1+z) = z \cdot \Gamma(z) \quad \Gamma(1) = 1 \quad \Gamma(1/2) = \sqrt{\pi}$$

$$\Gamma(n+1) = 1 \cdot 2 \cdot 3 \dots n = n!$$

$$\Gamma(2z) = (2\pi)^{1/2} \cdot 2^{2z-1} \cdot \Gamma(z) \Gamma(z+1/2)$$

$$\Gamma(z) \Gamma(1-z) = \pi / \sin \pi z .$$

$\Gamma(n+1) = n!$ laat zich als volgt implementeren:

```

DECLARE FUNCTION FAC# (MM%)
FUNCTION FAC (MM)
  IF MM > 1 AND MM < 40 THEN
    FAC = CDBL(MM) * FAC (MM-1)
  ELSEIF MM=0 OR MM=1 THEN
    FAC= 1#
  END IF
END FUNCTION

```

Een nadeel van recursieve programmering is, dat er veel stackruimte nodig is (gebruik voor MM > 40 b.v. CLEAR , , 10000.).

De volgende machtreeks ontwikkeling is uitgangspunt voor de verdere beschouwingen:

$$\log \Gamma(1+z) = -\gamma \cdot z + \sum_{m=2}^{\infty} (-z)^m \cdot \frac{\zeta(m)}{m} \quad |z| < 1 \quad (1)$$

met $\zeta(m)$ de Riemann zetafunctie en γ , de constante van Euler:

$$\gamma = \lim_{m \rightarrow \infty} [1 + 1/2 + 1/3 + \dots + 1/m - \log m] = 0.5772156649015329 . \quad (2)$$

Uit de reeks $\log(\Gamma(1+z))$ volgt voor $z = 1$ een andere definitie voor de

constante van Euler:

$$\gamma = \sum_{m=2}^{\infty} (-)^m \cdot \frac{\zeta(m)}{m} . \quad (3)$$

De Riemann zetafunctie is gedefinieerd als:

$$\zeta(m) = \sum_{k=1}^{\infty} 1/k^m .$$

Deze reeks convergeert zeer slecht. Breken we de reeks af na N termen dan is de fout $O(1/N)$, dus ongeschikt voor numerieke berekening.

De volgende alternerende reeks is echter zeer bruikbaar:

$$\zeta(m) = \frac{1}{(1 - 2^{1-m})} \cdot \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^m} . \quad (4)$$

In onderstaand programma zijn met behulp van een kettingbreuk-ontwikkeling van reeks (4) de coëfficiënten $\zeta(m)$, $m = 2, \dots, M$ bepaald. Bij elke exponent m , is het maximaal aantal termen mee te nemen in de sommatie, zo bepaald dat overflowgrens niet wordt overschreden.

Alle resultaten zijn in overeenstemming met de tabel in het Handbook¹). Met de gevonden waarden $(-)^m \cdot \zeta(m)/m$ wordt eerst met behulp van (3) de constante van Euler γ berekend. De coëfficiënten $(-)^m \cdot \zeta(m)/m$ en gevonden waarde γ worden vervolgens gebruikt om een kettingbreukontwikkeling van $\log(\Gamma(1+z))$ te maken. Met behulp van de in het Hoofdstuk "Operaties met machtreksen" ontwikkelde SUB EXPALGORITHE ($M, A(), Q()$), genereren we de machtrekcoëfficiënten van:

$$\Gamma(1+z) = \exp \left\{ -\gamma \cdot z + \sum_{m=0}^{\infty} (-z)^m \frac{\zeta(m)}{m} \right\} \quad |z| < 1 \quad (5)$$

en de daarmee corresponderende kettingbreukontwikkelingen.

Dit is uitgewerkt in het programma ZETAGAM.BAS (op de schijf).

De gevonden ontwikkelingen gelden voor $|X| < 1$.

Als volgt maken we een uitbreiding die geldig is voor alle reële X waarden (uitgezonderd in de polen $X \neq 0, -1, -2, \dots$).

Voor $X > 1$ stellen we: $X = N + z$, N geheel en $0 \leq z \leq 1$ en verder gebruiken we de relatie:

$$\Gamma(1+X) = (N+z)(N-1+z) \dots (1+z)\Gamma(1+z) . \quad (6)$$

Voor $X < 0$ stellen we: $X = z - N$, N geheel en $0 \leq z \leq 1$ en gebruiken we de relatie:

$$\Gamma(1 + X) = \frac{\Gamma(1 + z)}{z \cdot (z - 1)(z - 2) \dots (z - N)} . \quad (7)$$

Met behulp van de, met het programma ZETAGAM.BAS berekende kettingbreukcoëfficiënten en de formules (6) en (7), is de FUNCTION GAMMA(XX#) ontwikkeld.

```

DEFINT I - N: DEFDBL A - H, O - Z
FUNCTION GAMMA (XX)
  IF XX < 0 AND (ABS(XX) = INT(ABS(XX))) THEN
    PRINT XX; " is een pool. "
  EXIT FUNCTION
  END IF
  DIM A(21): A(0) = 1#: A(1) = .5772156649015295#
  A(2) = 1.136279057208445#: A(3) = -1.200319367226335#
  A(4) = .2149579499151987#: A(5) = .1614583915942033#
  A(6) = .5911392604443578#: A(7) = -.5532593470364774#
  A(8) = 7.997817361106836D-02: A(9) = 1.414157638007961D-02
  A(10) = 3.547081631244121#: A(11) = -3.359539814439355#
  A(12) = 7.976279983888335D-03: A(13) = -8.828671172640859D-02
  A(14) = -9.531927116455007D-02: A(15) = .5407926440413647#
  A(16) = -.2282416475894038#: A(17) = -5.830726580843101D-02
  A(18) = 3.446504235718311D-02: A(19) = -.2238607056469128#
  A(20) = -4.872472653713942D-02: A(21) = .251607358459197#
  X = XX - 1#: B = 1#
  IF ABS(X) > 100# THEN PRINT "Over- of Underflow": EXIT FUNCTION
  IF X > 1# THEN
    N = INT(X): T = X - CDBL(N): B = X
    FOR I = 1 TO N - 1: B = B * (X - CDBL(I)): NEXT I: X = T
  END IF
  IF X < 0 THEN
    N = INT(X): T = X - CDBL(N): M = ABS(N): B1 = T
    FOR I = 1 TO M - 1: B1 = B1 * (T - CDBL(I)): NEXT I
    B = 1# / B1: X = T
  END IF
  P0 = 0: P1 = A(0): Q0 = 1#: Q1 = 1#: K = 1
  WHILE K ≤ 18
    H1 = P1 + A(K) * X * P0: P0 = P1: P1 = H1
    H1 = Q1 + A(K) * X * Q0: Q0 = Q1: Q1 = H1: K = K + 1
  WEND: GAMMA = B * P1 / Q1
END FUNCTION

```

Deze functie is opgenomen in het programma GAMMA.BAS op de schijf.
 Enkele aan $\Gamma(z)$ verwante, later te gebruiken functies.
 De ψ -functie is gedefinieerd als:

$$\psi(z) = \frac{d \log(\Gamma(z))}{dz} = \frac{\Gamma'(z)}{\Gamma(z)}. \quad (8)$$

Enkele eigenschappen zijn:

$$\psi(1+z) = \psi(z) + 1/z \quad (9)$$

$$\psi(1-z) = \psi(z) + \pi \cot \pi z \quad (10)$$

$$\psi(1) = -\gamma : \psi(1/2) = -\gamma - 2 \log 2 \quad (11)$$

$$\psi(n) = -\gamma + \sum_{k=1}^{n-1} \frac{1}{k}. \quad (12)$$

Hier volgt een implementatie van de ψ -functie voor MM geheel (wordt gebruikt bij de Besselfuncties $K_v(x)$ en $Y_v(x)$ $v = 1, 2$).

```
DEFDBL A - H, O - Z : DEFINT I - N
SUB FUNCTION PSI (MM)
  W1 = 0
  IF MM = 1 THEN
    PSI = -.5772156649015329#
  ELSE
    FOR k = 1 TO MM - 1: W1 = W1 + 1# / CDBL(k): NEXT k
    PSI = W1 - .5772156649015329#
  END IF
END FUNCTION
```

De betafunctie $B(x, y)$ is gedefiniëerd door de integraal:

$$B(x, y) = \int_0^1 t^{x-1} \cdot (1-t)^{y-1} \cdot dt \quad \text{voor } \operatorname{Re} x > 0, \operatorname{Re} y > 0. \quad (13)$$

Samenhang beta- en gammafunctie:

$$B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}. \quad (14)$$

Hier volgen nog twee functiesymbolen welke later bij de implementatie van enkele Besselfuncties worden gebruikt.

Het Hankelsymbool:

$$(v, n) = 2^{-2n} \{(4n^2 - 1)(4n^2 - 3^2) \dots [4n^2 - (2n - 1)^2]\} / n! = \frac{\Gamma(\frac{1}{2} + v + n)}{n! \Gamma(\frac{1}{2} + v - n)}.$$

Het Pochhammersymbool:

$$(a)_n = a.(a+1)(a+2)\dots(a+n-1) = \frac{\Gamma(a+n)}{\Gamma(a)} .$$

Bij het oplossen van partiële differentiaalvergelijkingen in het hoofdstuk Laplace transformaties, treden in enkele voorbeelden de volgende functies op:
De Errorfunctie

$$\operatorname{erf} x = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-y^2).dy$$

met de eigenschappen:

$$\operatorname{erf} \infty = 1 \quad \operatorname{erf}(-x) = -\operatorname{erf} x$$

en de hieruit afgeleide functie Errorfunctiecomplement:

$$\operatorname{erfc} x = 1 - \operatorname{erf} x = \frac{2}{\sqrt{\pi}} \int_x^\infty \exp(-y^2).dy .$$

Referenties.

- 1) Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables. Edited by Milton Abramowitz and Irene Stegun.
- 2) Speciale Functies in de Mathematische Fysica. N.M. Temme. Epsilon Uitgaven, Utrecht

11. Algorithmen voor Diverse Besselfuncties

Met behulp van de formules, vermeld in de Appendix A. Besselfuncties (App. Bes.), ontwikkelen we algorithmen voor de volgende Besselfuncties:

$$J_v(z), Y_v(z), K_v(z), I_v(z), \quad \text{voor de ordes } v = 0 \text{ en } v = 1 .$$

Deze functies worden in tal van wetenschappelijke en technische toepassingen (en in alle latere hoofdstukken) gebruikt.

De goed convergerende machtreeksvoorstellungen van de functies: $J_v(z)$ en $I_v(z)$ $v = 0, 1$, hebben allemaal reële machtreëscoefficiënten. Deze functies laten zich weer naar kettingbreuken ontwikkelen als we $J_1(z)$ en $I_1(z)$ delen door z en in de onderstaande vormen $z^2 = x$ stellen.

$$J_0(z) = \sum_{r=0}^{\infty} \frac{(-)^r \cdot z^{2r}}{2^{2r} \cdot r! \cdot r!} \quad (1)$$

$$I_0(z) = \sum_{r=0}^{\infty} \frac{z^{2r}}{2^{2r} \cdot r! \cdot r!} \quad (2)$$

$$J_1(z)/z = \sum_{r=0}^{\infty} \frac{(-)^r z^{2r}}{2^{2r+1} \cdot r! \cdot (r+1)!} \quad (3)$$

$$I_1(z)/z = \sum_{r=0}^{\infty} \frac{z^{2r}}{2^{2r+1} \cdot r! \cdot (r+1)!} \quad (4)$$

Deze vormen lenen zich in het bijzonder om direct de opvolgende coëfficiëntenverhoudingen a_{k+1}/a_k , in het QD-algorithme in te voeren. De verkregen ontwikkelingen zijn bruikbaar voor zowel reële als complexe argumentwaarden. Voorlopig beperken we ons tot reële waarden.

Bij de functie's $Y_v(z)$ en $K_v(z)$ $v = 0, 1$, kan, vanwege de factor $\log(z/2)$ in de formules, niet met één kettingbreukontwikkeling worden volstaan. Voor $Y_0(z)$ gaan we uit van de voorstelling:

$$\begin{aligned} \frac{\pi}{2} \cdot Y_0(z) = & \{ \log(z/2) + \gamma \} \cdot J_0(z) + (z/2)^2 - \frac{(1+1/2)}{(2!)^2} \cdot (z/2)^4 + \\ & + (-)^n \frac{(1+1/2+1/3+\dots+1/n)(z/2)^{2n}}{(n!)^2} + \dots \end{aligned} \quad (5)$$

met $\gamma = 0.5772156649\dots$, de constante van Euler.

We schrijven (5) als:

$$Y_0(z) = \frac{2}{\pi} \cdot \log(z/2) \cdot J_0(x) + S_0(x) , \quad (6)$$

waarin

$$S0(z) = \gamma \cdot J_0(z) + (z/2)^2 - \frac{(1 + 1/2) \cdot (z/2)^4}{(2!)^2} + \dots + \\ + (-)^n \cdot \frac{(1 + 1/2 + \dots + 1/n)}{(n!)^2} \cdot (z/2)^{2n} \quad (7)$$

weer een machtreeks is met alleen even machten. Vervangen we $J_0(z)$ in (7) door de machtreeks (1) en wordt weer $z^2 = x$ gesteld, dan kan weer direct het QD-algorithme worden gebruikt voor het bepalen van de kettingbreukcoëfficiënten.

Differentiatie van de betrekking (6) levert de formules:

$$-Y_1(z) = Y_0'(z) = \frac{d}{dz} \left\{ \frac{2}{\pi} \cdot \log(z/2) \cdot J_0(z) + S0(z) \right\} \quad (7a) \\ -Y_1(z) = -\frac{2}{\pi} \cdot \log(z/2) \cdot J_1(z) + \frac{2}{\pi} \cdot J_0(z)/z + \frac{d}{dz} S0(z)$$

Vermenigvuldigen we (7) met z , dan volgt de vorm:

$$z \cdot Y_1(z) = \frac{2}{\pi} \cdot z \cdot \log(z/2) \cdot J_1(z) + S1(z), \quad (8)$$

waarin

$$z \cdot J_1(z) \quad \text{en} \quad S1(z) = -\left\{ \frac{2}{\pi} \cdot J_0(z) + z \cdot \frac{d}{dz} S0(z) \right\}, \quad (8a)$$

weer machtreeksen in z^2 zijn.

Na substitutie $z^2 = x$, weer geschikt voor aanpak met het QD-algorithme.

Alle hierna te bespreken complete programma's zijn op de schijf te vinden.

In het programma QDBESSEL.BAS worden eerst de machtreekscoëfficiënten van de functies: $J_0(z)$, $J_1(z)$, $S0(z)$ en $S1(z)$ gegenereerd. Vervolgens wordt $z^2 = x$ gesteld en het QD-algorithme aangeroepen.

De operatie $z \cdot \frac{dS0(z)}{dz}$ in (8a), is als volgt te implementeren:

Vermenigvuldig de coëfficiënten van $S0(z)$, opgeslagen in de array A , met de lopende index, dus b.v.: $i \cdot A(i)$.

De kettingbreukontwikkelingen van: $J_0(z)$, $J_1(z)$, $S0(z)$ en $S1(x)$ worden vervolgens gebruikt voor opbouw van: $Y_0(z)$ en $Y_1(z)$ overeenkomstig de formules (6) en (8).

Om binnen een gekozen interval met bovengrens ZI , de te berekenen functiewaarden met een gewenste nauwkeurigheid te kunnen bepalen, moet eerst het benodigde aantal kettingbreukcoëfficiënten N worden bepaald. Dit kan eenvoudig worden gedaan door telkens voor een gekozen N , de functiewaarde te bepalen en te kijken of de nauwkeurigheid hiervan in overeenstemming is met de waarden in de Functietabellen van het Handbook¹⁾.

Voor het buitengebied $|z| > ZI$ ontwikkelen we algorithmen met behulp van de in (Appendix A, Besselfuncties) gegeven asymptotische formules.

De meeste in de literatuur beschreven methoden, om de asymptotische ontwikkelingen van $J_\nu(z)$ en $Y_\nu(z)$ $\nu = 1, 2$ te genereren, berusten op de relaties (17),..., (23) (App. Bes.).

$$J_\nu(z) = \sqrt{2/(\pi z)} \cdot \{P(\nu, z) \cdot \cos(Z) - Q(\nu, z) \cdot \sin(Z)\} \quad |\arg z| < \pi \quad (9)$$

$$Y_\nu(z) = \sqrt{2/(\pi z)} \cdot \{P(\nu, z) \cdot \sin(Z) + Q(\nu, z) \cdot \cos(Z)\} \quad |\arg z| < \pi \quad (10)$$

Hierin is de betekenis van $P(\nu, x)$, $Q(\nu, x)$ en (ν, k) als volgt:

$$P(\nu, z) \sim \sum_{k=0}^{\infty} \frac{(-)^k (\nu, 2k)}{(2z)^{2k}} = 1 - \frac{(\mu - 1)(\mu - 3^2)}{2! \cdot (8z)^2} + \dots \quad (11)$$

$$Q(\nu, z) \sim \sum_{k=0}^{\infty} \frac{(-)^k (\nu, 2k + 1)}{(2z)^{2k+1}} = \frac{(\mu - 1)}{1! \cdot (8z)} - \frac{(\mu - 1)(\mu - 3^2)(\mu - 5^2)}{3! \cdot (8z)^3} + \dots \quad (12)$$

waarin:

$$Z = z - (\nu/2 + 1/4)\pi \quad \text{en} \quad \mu = 4 \cdot \nu^2 \quad (13)$$

In het programma modificeren we de functies $Q(\nu, z)$ en $P(\nu, z)$, door hierin de parameter ZI (de ondergrens van asymptotische ontwikkelingen) als schaalfactor, als volgt te verwerken:

$$P^*(\nu, z) \sim \sum_{k=0}^{\infty} (-)^k \frac{(\nu, 2k)(ZI/z)^{2k}}{(2 \cdot ZI)^{2k}} \quad (14)$$

$$Q^*(\nu, z) \sim \sum_{k=0}^{\infty} (-)^k \frac{(\nu, 2k + 1) \cdot (ZI/z)^{2k}}{(2 \cdot ZI)^{2k+1}} \quad (15)$$

De Hankel-symbolen $(\nu, 2k)$ en $(\nu, 2k + 1)$ worden uitgewerkt met behulp van de Pochhammer-relatie:

$$\begin{aligned} (\nu, m) &= \frac{(-)^m (1/2 - \nu)_m \cdot (1/2 + \nu)_m}{m!} = \\ &= \frac{(-)^m (\mu - 1)(\mu - 3^2) \dots (\mu - (2m - 1)^2)}{\{m! \cdot (2)^{2m}\}} \end{aligned} \quad (16)$$

De formules (13),..., (16) zijn geïmplementeerd in het subprogramma:
SUB BESSELASYMPTOTIEK

De betekenis van de gebruikte parameters:

- M het aantal termen in de machtreeksontwikkeling.
 v de orde.
 $W = 4.v^2$ (de factor $\mu = 4.v^2$ genoemd in (15)).
 ZI de ondergrens voor argument z .
 $S = (W-1)/(8z)$ deze parametergroep wordt gebruikt om de oneven functie $Q^*(v, z)$ te transformeren in de even functie $Q^*(v, z)/S$.
 In de even functies $P^*(v, z)$ en $Q^*(v, z)/S$ stellen we weer $X = (ZI/z)^2$.
 $A()$ coëfficiënten van de functie $P^*(v, z)$
 $B()$ coëfficiënten van de functie $Q^*(v, z)/S$
 $C()$ coëfficiënten van functie $\{Q^*(v, X)/S\}/P^*(v, X)$, worden bepaald met het algoritme besproken bij Machtreeksoperaties.

```

DEFINT I - N:DEFDBL A - H,O - Z
SUB BESSELASYMPTOTIEK(M,W,ZI,A(),B(),C(),S )
  'P*(v, X) in Array B() met behulp van (v, 2k)
  S=(W-1#)/(8# * ZI):R=1#:T=-1#:L=2
  P=(8# * ZI)^2:B(0)=1#:N=1
  FOR K=2 TO 2 * M STEP 2
    R=R * (W-CDBL((L-1)^2)) * (W-CDBL(L+1)^2)/
      (P * CDBL((K-1) * K))
    B(N)=R * T :N=N+1:T=-T::L=L+4:NEXT K
  'Q*(V, X)/S in Array A() bepaald met behulp van (v, 2k + 1)
  R=1#:A(0)=R:T=-1#:L=4:N=1
  FOR K=3 TO 2 * M + 1 STEP 2
    R=R * (W-CDBL((L-1)^2)) * (W-CDBL(L+1)^2)/
      (P * CDBL((K-1) * K))
    A(N)=R * T :N=N+1:T=-T:L=L+4
  NEXT K '{Q * (v, X)/S}/P * (v, X) in C()}.
  C(0)=A(0):C(1)=A(1)-B(1)
  FOR K=1 TO M:D(K)=0:FOR I=K-1 TO 0 STEP -1
    D(K)=D(K)+C(I) * B(K-I):NEXT I
    C(K)=A(K) * B(0)-D(K):NEXT K
END SUB
  
```

De met SUB BESSELASYMPTOTIEK te genereren gegevens, opgeslagen in array's $A()$ en $B()$, maken het mogelijk om de formules (9) en (10) te implementeren volgens de klassieke aanpak.

Hieraan voorafgaande bespreken eerst nog een alternatieve methode die ook berust op de zelfde relaties 9, ..., 16.

Volgens de Hankelfunctie-definitie (App. Bes.), geldt voor de modules:

$$M_v(x) = |H_v(x)| = |J_v(x) + i.Y_v(x)| = \sqrt{\{J_v(x)^2 + Y_v(x)^2\}} \quad v = 0, 1 \quad (17)$$

en voor het argument:

$$\Phi_v = \arg H_v(x) = \arctan \{Y_v(x)/J_v(x)\} \quad (18)$$

respectievelijk:

$$\tan(\Phi_v) = Y_v(x)/J_v(x) . \quad (18a)$$

Hiermee definiëren we:

$$J_v(x) = M_v(x) \cos(\Phi_v) \text{ en } Y_v(x) = M_v(x) \sin(\Phi_v) , \quad v = 0, 1 . \quad (19)$$

Substitutie van de asymptotische reeksen (9) en (10) in (18a) levert:

$$\tan(\Phi_v) \approx \frac{\{P(v, x) \sin(Z) + Q(v, x) \cos(Z)\}}{\{P(v, x) \cos(Z) - Q(v, x) \sin(Z)\}} .$$

Voor $P(v, x) \cdot \cos(Z) \neq 0$ volgt hieruit dat:

$$\tan(\Phi_v) = \frac{\{\tan(Z) + Q(v, x)/P(v, x)\}}{\{1 - \tan(Z) \cdot Q(v, x)/P(v, x)\}} . \quad (20)$$

Stellen we in (20),

$$Q(v, x)/P(v, x) = \tan(\Phi) \quad (21)$$

dan volgt voor:

$$\tan(\Phi_v) = \{\tan(Z) + \tan(\Phi)\}/\{1 - \tan(Z) \cdot \tan(\Phi)\} = \tan(Z + \Phi) \quad (22)$$

en voor

$$\Phi_v = Z + \Phi = x - \pi(v/2 + 1/4) + \arctan\{Q(v, x)/P(v, x)\} . \quad (23)$$

In het programma ASYM-ALT.BAS worden de alternatieve en de klassieke asymptotische aanpakken als volgt uitgewerkt.

Met behulp van SUB BESSELASYMPTOTIEK worden de coëfficiënten van $G(X) = \{Q^*(v, X)/S\}/P^*(v, X)$ bepaald in de array $C()$. Uitgaande van deze coëfficiënten wordt met behulp van de: SUB ARCTANGENS (Operaties met machtreeksen), alle machtreekscoëfficiënten van $G(X)^k$ voor $k = 1, 2, \dots, N$, in de dubbel-array $E()$ opbouwd. Met behulp hiervan worden de coëfficiënten van $\arctan\{Q^*(v, x)/P^*(v, x)\}$ gegenereerd in de array $Q()$. $\{D()$ en $QN()$ zijn hulp-arrays}.

SUB ARCTANGENS ($M, C(), D(), Q(), QN(), S$)

DIM $E(35, 35)$

'Genereren machten van $C(X)$ voor opbouw $\arctan\{C(X)\}$ '

```

FOR K = 0 TO M: D(K) = 0: FOR I = K TO 0 STEP -1
  D(K) = D(K) + C(I) * C(K - I): NEXT I: Q(K) = D(K)
NEXT K ' C(X)^2
FOR K = 0 TO M: E(1, K) = C(K): QN(K) = C(K): NEXT K
P = S ^ 2: Y = P: T = -1#
'Genereer arctan((Q/S)/P) + opschuiven termen
FOR L = 2 TO M: Z = CDBL(2 * (L - 1) + 1):
  FOR K = 0 TO M: C(K) = 0
  FOR I = K TO 0 STEP -1
    C(K) = C(K) + QN(I) * Q(K - I): NEXT I
  E(L, K + L - 1) = T * C(K) * Y / Z: NEXT K: Y = Y * P: T = -T
FOR J = 0 TO M: QN(J) = C(J): NEXT J: NEXT L
FOR K = 0 TO M: Q(K) = 0: NEXT K 'Sommatie
FOR K = 1 TO M: FOR I = 0 TO M
  Q(I) = Q(I) + E(K, I): NEXT I: NEXT K
END SUB

```

Rest nog om $M_v(x)^2 = (J_v(x)^2 + Y_v(x)^2)$ te bepalen volgens (17).
 Substitutie van de asymptotische reeksen (9) en (10) levert na uitwerking:

$$M_v(x)^2 \approx \frac{2}{\pi x} \cdot \{P(v, x)^2 + Q(v, x)^2\} .$$

$M_v(x)$ kan worden berekend met de reeds beschikbare gegevens in:

SUB BESASYMPTREEKSEN(M, W, ZI, A(), B(), C(), S) .

Eenvoudiger is het echter om hiervoor de volgende formule te gebruiken (Watson⁵)
 pag. 449) te gebruiken:

$$M_v(x)^2 = (J_v(x)^2 + Y_v(x)^2) \approx \frac{2}{\pi x} \cdot \sum_{m=0}^{\infty} \frac{[1.3.5...(2m-1)].(v, m)}{(2x)^{2m}} . \quad (24)$$

Dit is uitgewerkt in het volgende sub-programma. De coëfficiënten worden opgeslagen in de array R():

```

SUB ALTMODULES (M, W, ZI, Z, R()) ' Modules alternatief
R(0) = 1#: Z = 1#: X = 1# / (2 * ZI) ^ 2: V = 1#: Y = X
FOR K = 1 TO M: Z = Z * CDBL(2 * K - 1) / CDBL(2 * K)
  V = V * (W - CDBL(2 * K - 1) ^ 2)
  R(K) = Z * X * V: X = X * Y
NEXT K
END SUB

```

In het programma ASYM-ALT.BAS (op de schijf) zijn met behulp van de besproken subs, de kettingbreukontwikkelingen en ook de functiewaarden van: $J_v(x)$ en $Y_v(x)$ ($v = 0, 1, 2, \dots$) voor $x \geq ZI$ te bepalen.

Het programma biedt de mogelijkheid, dit te doen volgens zowel de klassieke als de alternatieve aanpak.

De volgende parameters zijn gebruikt:

- W orde Besselfunctie.
- ZI ondergrens asymptotische ontwikkeling.
- M aantal termen.
- Z argumentwaarde.

De functiewaarden van $J_v(x)$ en $Y_v(x)$ ($v = 0, 1, 2, \dots$) worden berekend voor de waarden $x = ZI + k$, $k = 0, 1, \dots, 6$.

In het Handbook¹⁾ zijn op de pagina's 364–365 (9.2.28 en 9.2.29), zonder bewijs, drie termen in de ontwikkelingen van: $\arctan\{Q(v, x)/P(v, x)\}$ en van $M_v(x)^2$ gegeven. Met de boven gegeven implementaties kunnen we van beide ontwikkelingen, elk gewenst aantal termen in de ontwikkeling zelf bepalen, inclusief de kettingbreukcoëfficiënten.

Om voor argument waarden $|x| \geq ZI$ de functiewaarden met een gewenste nauwkeurigheid te kunnen berekenen, kunnen we weer als volgt het benodigde aantal termen N in de ontwikkelingen gaan bepalen. Kies een N , bepaal de machtreeks- en kettingbreuk- coëfficiënten. Bereken voor een aantal argumentwaarden x_i ($i = 1, 2, \dots, N$), de functiewaarden en vergelijk hoeveel decimalen overeenstemmen met de overeenkomstige waarden in de Functietabellen van het Handbook¹⁾.

De programma's QDBESSEL.BAS en ASYM-ALT.BAS, leveren in principe alle gegevens op om nauwkeurige efficiënte sub-programma's voor de reële Besselfuncties $J_0(x)$, $Y_0(x)$, $J_1(x)$ en $Y_1(x)$ samen te stellen. We stellen dit echter nog even uit, omdat er bij het ontwikkelen van de algorithmen voor de functies $K_v(z)$ en $I_v(z)$, $v = 0, 1$, voor z complex, nog zeer efficiënte alternatieve asymptotische ontwikkelingen voor $J_v(z)$ en $Y_v(z)$ ter beschikking krijgen.

De functies $K_v(z)$ en $I_v(z)$ $v = 0, 1$, (voor z complex) hebben we nodig bij het oplossen van enkele cylinder-symmetrische partiële differentiaalvergelijkingen met behulp van numerieke Laplace-inversie-methoden.

$K_n(z)$ implementeren we voor $|z| < ZI$ overeenkomstig de formule:

$$K_n(z) = \frac{(z/2)^{-n}}{2} \sum_{k=0}^{n-1} (-z^2/4)^k \cdot \frac{(n-k-1)!}{k!} + (-)^{n+1} \ln(z/2) \cdot I_n(z) + (-z/2)^n \cdot \sum_{k=0}^{\infty} \frac{(\psi(k+1) + \psi(n+k+1)) \cdot (z/2)^{2k}}{(k! \cdot (n+k)!)} \quad (25)$$

in het programma KIZMACHT.BAS op de schijf.

De machtreekscoëfficiënten voor $n = 0, 1$ zijn reëel. Ze worden gegenereerd met behulp van de subs FAC(N) en PSI(N).

Kettingbreukontwikkelingen worden hier nog niet gebruikt.

Voor een niet te grote z en bij een voldoende groot aantal termen, worden de functiewaarden van: $K_v(z)$ en $I_v(z)$ $v = 0, 1$, berekend met behulp van de complexe variabelen-subs: CMAAL, CDIV, CSQR en CLOG.

In het programma KNPSIFAC.BAS worden uit de reële machtreeks coëfficiënten van $K_0(z)$ en $K_1(z)$ (ook die van $Y_0(z)$ en $Y_1(z)$), de kettingbreukontwikkelingen weer met het QD-algorithme bepaald.

We bespreken hierna complete implementaties van de functies:

$$K_0(z) \quad I_0(z) \quad \text{en} \quad K_1(z) \quad I_1(z)$$

elk bestaande uit een convergerend machtreeks deel en een asymptotische deel.

De functies $I_0(z)$ en $I_1(z)$ laten zich voor $|z| \leq |ZI|$ direct uit de even machtreeksen (2) en (4) (als we $z^2 = x$ stellen) naar kettingbreuken ontwikkelen.

Voor $|z| \leq |ZI|$ hebben we in het programma KNPSIFAC.BAS al aange- toond hoe dat kan gebeuren voor de functies $K_0(z)$ en $K_1(z)$.

Voor $|z| \geq |ZI|$ geldt voor $K_v(z)$ $v = 0, 1$, de asymptotische ontwikkeling

$$K_v(z) \sim \sqrt{\frac{\pi}{2z}} \cdot e^{-z} \cdot \sum_{m=0}^{\infty} \frac{(v, m)}{(2z)^m} . \quad (26)$$

We werken dit uit met behulp van de formules: (App. Bes.)

$$K_0(z) \sim \sqrt{\frac{2}{\pi z}} \cdot e^{-z} \cdot {}_2F_0(1/2, 1/2; -1/2z) \quad (27)$$

$$K_1(z) \sim \sqrt{\frac{2}{\pi z}} \cdot e^{-z} \cdot {}_2F_0(3/2, -1/2; -1/2z) . \quad (28)$$

De kettingbreuk-ontwikkeling van de functie ${}_2F_0(1/2, 1/2; -1/2z)$ in de formule (27), is als volgt te implementeren:

```
FOR K = 1 TO N: A(K) = -CDBL(2 * K + 1)^2/CDBL(8 * K): NEXT K
```

Analoog implementeren we ${}_2F_0(3/2, -1/2; -1/2z)$ in (28) als:

```
FOR K = 1 TO N
A(K) = -CDBL((2*K + 1) * (2 * K - 3))/CDBL(8*K) : NEXT K
```

De functiewaardeberekening van de asymptotische reeksen (27) en (28) moet

geheel in complexe variabelen gebeuren. De complexe variabelen, o.a. z in de factor $\sqrt{2/(\pi z)}.e^{-z}$, zijn direct uitgewerkt.

Alleen de Subs QD en CWAARDEKET worden gebruikt.

De zo verkregen implementaties van $K_v(z)$ $v = 0, 1$ en $z = x + iy$, zijn zeer efficiënt en nauwkeurig. Zo wordt er voor $N = 30$ en $|z| \geq 1$ een nauwkeurigheid van meer dan 10 cijfers gevonden).

Omdat voor $K_v(z)$ $v = 0, 1$ en $z = i.y$ (zuiver imaginair), de formules:

$$-\frac{2}{\pi} \cdot K_0(i.y) = i.J_0(y) + Y_0(y) \quad (29)$$

$$\frac{2}{\pi} \cdot K_1(i.y) = -J_1(y) + i.Y_1(y) \quad (30)$$

gelden, kunnen we voor y reëel deze implementatie ook gebruiken als een alternatieve asymptotische ontwikkeling voor: $J_v(y)$ en $Y_v(y)$ $v = 0, 1$.

De implementatie van de $K_v(z)$ volgens (26), (27), (28) is eenvoudig, voor de $I_v(z)$'s moeten we echter de volgende meer gecompliceerde asymptotische voorstelling (App. Bes. no. 25) gebruiken:

$$I_v(z) \sim \frac{1}{\sqrt{(2\pi z)}} \cdot \left\{ e^z \cdot \sum_{m=0}^{\infty} \frac{(-)^m(v, m)}{(2.z)^m} + e^{-z+(v+\frac{1}{2})\pi.i} \cdot \sum_{m=0}^{\infty} \frac{(v, m)}{(2.z)^m} \right\}. \quad (31)$$

Het lijkt lastiger dan het is. In deze formule zijn op de factor $(-)^m$ na, de beide oneindige sommen exact aan elkaar gelijk en stemmen overeen met die in formule (26). We kunnen dus (behoudens de tekenwisselingen $(-)^m$) de eerder bij de $K_v(z)$'s gevonden kettingbreukontwikkelingen gebruiken.

Met behulp van de formules (2), (4), (25) tot en met (31) en gebruikmaking van de FUNCTIONS FAC en PSI en de SUBs QD en WAARDEKET, hebben we in het programma INKNCOMP.BAS, voor alle argumentwaarden $z = x + iy$, de functies $I_0(z)$, $I_1(z)$, $K_0(z)$ en $K_1(z)$, als volgt geïmplementeerd:

SUB I0Z (X1#, Y1#, F1#, F2#)

SUB I1Z (X1#, Y1#, F1#, F2#)

SUB K0Z (X1#, Y1#, F1#, F2#)

SUB K1Z (X1#, Y1#, F1#, F2#)

Hierin stellen de parameters: $X1\#$, $Y1\#$, het argument $z = X1\# + i Y1\#$ voor en $F1\#$, $F2\#$ de betreffende functiewaarden $F1\# + i F2\#$.

De subs kunnen in principe zelfstandig in andere programma's worden gebruikt. Omdat bij elke functiewaardeberekening echter telkens weer de machtreeks- en de kettingbreuk-coëfficiënten moeten worden gegenereerd, is dit voor veel functiewaardeberekeningen zeer tijdrovend.

Daarom zijn met behulp van de met dit programma gegenereerde ketting-

breukcoëfficiënten, nieuwe snelle accuraat werkende (volumineuze) subs samengesteld (de namen van de subs en gebruikte parameters zijn onveranderd), welke behoudens het aanroepen van SUB CWAARDEKET zelfstandig zijn te gebruiken. De subs zijn opgenomen in het programma IOI1K0K1.BAS.

Deze sub-programma's worden later gebruikt bij het oplossen van sommige partiële differentiaalvergelijkingen met behulp van numerieke Laplace-inversietechnieken.

Met behulp van de programma's QDBESSEL.BAS, ASYMALT.BAS en INKNCOMP.BAS zijn voor reële argumentwaarden, op analoge wijze zelfstandige combinatiesubs voor $J_i(x)$ en Y_i , $i = 1, 2$, geïmplementeerd, met de namen:

SUB JOY0 (T, U, V) en SUB J1Y1 (T, U, V)

De betekenis van de parameters is als volgt:

T reële argumentwaarde. Levert:

U $J_0(T)$, $V = Y_0(T)$ bij aanroep van JOY0 en

U $J_1(T)$, $V = Y_1(T)$ bij aanroep van J1Y1.

Deze subs gebruiken we later bij het oplossen van enkele randwaardeproblemen.

Literatuur.

- 1) Handbook of Mathematical Functions. M. Abramowitz en I.A. Stegun.
- 2) Higher Transcendental Functions. Bateman Manuscript Project Vol. II.
- 3) A treatise on the theory of Besselfunctions. G.A. Watson.
- 4) Speciale Functies in de Mathematische Fysica. N.M. Temme. Epsilon Uitgaven, Utrecht.

De hierna vermelde programma's zijn op de schijf aanwezig.

QDBESSEL.BAS, ASYM-ALT.BAS

KIZMACHT.BAS, KNPSIFAC.BAS

INKNCOMP.BAS (uitwerking: subs $I_0(z)$, $I_1(z)$, $K_0(z)$ en $K_1(z)$)

IOI1K0K1.BAS (gebruiks klare subs $I_0(z)$, $I_1(z)$, $K_0(z)$ en $K_1(z)$)

JOY0.BAS, J1Y1.BAS (combinatie subs)

JOX.BAS, J1X.BAS (Functions $J_0(x)$ en $J_1(x)$).

12. Eigenschappen van Laplace-Transformaties en Voorbeelden

Laat een reële begrensde functie $f(t)$ gedefinieerd zijn voor $t \geq 0$ en integreerbaar over $[0, X]$ voor elke $X > 0$, dan wordt de Laplace-getransformeerde van $f(t)$ gedefinieerd door de integraal:

$$F(p) = L\{f(t)\} = \int_0^{\infty} \exp(-pt) f(t) dt \quad (1)$$

(p een complexe parameter) mits deze integraal convergeert voor $\operatorname{Re} p = a \geq 0$.

We veronderstellen dat $f(t) = \frac{f(t+0) + f(t-0)}{2}$ voor alle t waarvoor $f(t)$ discontinu is.

Onder deze voorwaarden is de Laplace-getransformeerde functie $F(p)$ een analytische functie van p en bestaat de inverse functie.

De inverse transformatie is gedefinieerd als:

$$f(t) = L^{-1}(F(p)) = \frac{1}{2\pi i} \int_{a-i\infty}^{a+i\infty} \exp(pt) F(p) \cdot dp \quad (2)$$

Enkele eigenschappen van de Laplace-transformatie^{1,2}.

De transformatie is lineair:

Voor de lineaire combinatie $f(t) = \sum_{k=1}^n a_k \cdot f_k(t)$, waarin de a_k 's constanten zijn en de $f_k(t)$'s weer gedefinieerd op $[0, \infty)$ als boven, geldt:

$$L\{f(t)\} = \sum_{k=1}^n a_k \cdot L\{f_k(t)\} \quad (3)$$

Een direct gevolg van de lineariteit is, dat indien $f(t) = f(t, x)$ ook afhangt van x , de volgende differentiatie- en integratieregels gelden:

$$L\left\{\frac{\partial f(x, t)}{\partial x}\right\} = \frac{\partial L\{f(x, t)\}}{\partial x} \quad \text{en} \quad L\left\{\int_0^x f(z, t) \cdot dz\right\} = \int_0^x L\{f(z, t)\} dz \quad (4)$$

(belangrijk bij het oplossen van partiële differentiaalvergelijkingen).

Voor $L\{f(t)\} = F(p)$ geldt dat:

$$L\{\exp(at) \cdot f(t)\} = F(p - a) \quad (5)$$

Uit

$$L\{f(t)\} = F(p) \quad \text{en} \quad \lim_{t \rightarrow \infty} e^{-pt} \cdot f(t) = 0$$

volgt:

$$L\{f'(t)\} = -f(0) + p \cdot F(p) . \quad (6)$$

Algemeen geldt:

$$L\{f(t)\} = F(p) \quad \text{en} \quad \lim_{t \rightarrow \infty} e^{-pt} \cdot f^{(k)}(t) = 0 , \quad k = 0, 1, \dots, n$$

$$L\{f^{(n)}(t)\} = - \sum_{k=0}^{n-1} p^k \cdot f^{(n-1-k)}(0) + p^n \cdot F(p) \quad (7)$$

$$F'(p) = \frac{dF(p)}{dp} = \frac{d}{dp} \int_0^{\infty} \exp(-pt) \cdot f(t) dt = -L\{t \cdot f(t)\} \quad (8)$$

$$\frac{d^n}{dp^n} F(p) = (-1)^n L\{t^n \cdot f(t)\} . \quad (9)$$

Voor $g(t) = \int_0^t f(\tau) \cdot d\tau$ geldt:

$$G(p) = F(p)/p \quad (10)$$

$$L\{f(t)/t\} = \int_p^{\infty} F(q) \cdot dq . \quad (11)$$

De convolutiestelling:

Als $f(t)$ en $g(t)$ continu zijn op $[0, \infty)$ en de integralen: $L\{f(t)\} = F(p)$ en $L\{g(t)\} = G(p)$ absoluut convergent zijn voor een zekere reële $p > \alpha$, dan wordt de inverse van $H(p) = F(p) \cdot G(p)$ bepaald door:

$$h(t) = \int_0^t f(\sigma) \cdot g(t - \sigma) d\sigma = \int_0^t g(\sigma) \cdot f(t - \sigma) d\sigma . \quad (12)$$

Asymptotische eigenschappen:

Het gedrag van $f(t)$ voor $t \rightarrow 0$ wordt bepaald uit het asymptotisch gedrag van $F(p)$ voor $p \rightarrow \infty$, volgens de *beginwaarderelatie*:

$$\lim_{p \rightarrow \infty} p \cdot F(p) = \lim_{t \rightarrow 0} f(t) . \quad (13)$$

Als $f(t)$ begrensd is voor $t > 0$ en $\lim_{t \rightarrow \infty} f(t)$ bestaat, dan geldt de volgende eindwaarderelatie:

$$\lim_{p \rightarrow 0} p.F(p) = f(\infty) . \quad (14)$$

In onderstaande tabel is een aantal functies $f(t)$ en de bijbehorende getransformeerden $F(p)$ opgenomen, die we in latere toepassingen zullen gebruiken.

	$f(t)$	$F(p)$
1	1	t
2	$tn/n! \quad n = 0, 1, \dots, n$	$1/p^n$
3	e^{at}	$1/(p - a)$
4	$\cos(bt)$	$1/(p^2 + b^2)$
5	$\sin(bt)$	$b/(p^2 + b^2)$
6	$e^{at} \cdot \sin(bt)$	$b/\{(p - a)^2 + b^2\}$
7	$J_0(t)$	$1/\sqrt{p^2 + 1}$
8	$\frac{\exp(-x^2/4t)}{\sqrt{\pi t}}$	$\frac{\exp(-x \cdot \sqrt{p})}{\sqrt{p}}$ geldt ook voor $x = 0$
9	$\frac{x \cdot \exp(-x^2/4t)}{2 \cdot \sqrt{\pi \cdot t^3}}$	$\exp(-x \cdot \sqrt{p})$
10	$\operatorname{erfc} \frac{x}{2\sqrt{t}}$	$\frac{\exp(-x \cdot \sqrt{p})}{p}$
11	$\sin(t)/t$	$\arctan(1/p)$
12	$-\log(t) - \gamma$	$\log(p)/p \quad \gamma = .57721\dots$

Teneinde met het gebruik van de gegeven regels vertrouwd te raken, volgen er korte aanwijzingen om de formules in de tabel te bewijzen. Voorbeelden 1, ..., 5 volgen uit de definitievergelijking en partiële integratie. No. 6 is met behulp van eigenschap 5 terug brengen naar No. 5.

In de verdere beschouwingen gebruiken we als standaard notatie:

Kleine letters voor De *Originele* te transformeren functies

Hoofdletters voor De *Getransformeerde* (beeld-) functies.

Voordat No. 7 wordt afgeleid, behandelen we eerst hoe we de oplossingen van een gewone 2-de orde differentiaalvergelijking (met constante coëfficiënten) en rechterlid:

$y'' + y = g(t)$ op $[0, \infty)$ met de begincondities $y'(0) = 0$ en $y(0) = 1$, met behulp van de boven vermelde eigenschappen kunnen bepalen.

Met behulp van eigenschappen 7 en 1 volgt direct dat:

$$p^2 \cdot Y - p \cdot y'(0) + Y - y(0) = G(p) .$$

Substitutie van: $y'(0) = 0$ en $y(0) = 1$ en oplossen naar Y levert:

$$Y = \frac{1}{p^2 + 1} + \frac{G(p)}{p^2 + 1}.$$

De eerste term in het rechterlid is volgens de tabel gelijk aan $\cos(t)$. Met deze kennis kan de convolutie stelling (10) worden gebruikt om van de tweede term de particuliere oplossing te bepalen. Dus wordt de algemene oplossing:

$$y(t) = \cos(t) + \int_0^t g(\tau) \cdot \cos(t - \tau) \cdot d\tau.$$

De Laplace-getransformeerde van de Besselfunctie $J_0(t)$ op $[0, \infty)$ laat zich uit de differentiaalvergelijking van Bessel

$$y'' + \frac{y'}{t} + y = 0,$$

of anders geschreven

$$\frac{d}{dt} \left(t \cdot \frac{dy}{dt} \right) + t \cdot y = 0$$

met de beginvoorwaarden $y'(0) = 0$ en $y(0) = 1$ afleiden.

Uitgaande van $L\{y'\} = p \cdot Y - y(0)$ en eigenschap (8) volgt dat:

$$L\{t \cdot y'\} = - \frac{d(p \cdot Y - y(0))}{dp} = - \frac{d(p \cdot Y)}{dp}$$

$$L\left\{ \frac{d(t \cdot y')}{dt} \right\} = p \cdot \left(- \frac{d(p \cdot Y)}{dp} \right) = -p^2 \cdot \frac{dY}{dp} - p \cdot Y \quad \text{en} \quad L\{t \cdot y\} = - \frac{dY}{dp}.$$

Alles gesubstitueerd in de differentiaalvergelijking leidt tot:

$$(p^2 + 1) \cdot \frac{dY}{dp} + p \cdot Y = 0 \quad \text{resp.} \quad \frac{dY}{Y} = - \frac{p \cdot dp}{p^2 + 1}.$$

Integratie levert:

$$\log(Y) = -1/2 \log(p^2 + 1) + \log(C)$$

en dus is:

$$Y = \frac{C}{\sqrt{(p^2 + 1)}}.$$

Met behulp van de beginwaardestelling (13) bepalen we de waarde van de

integratieconstante C als volgt:

$$\lim_{p \rightarrow \infty} \frac{p.C}{\sqrt{(p^2 + 1)}} = \lim_{p \rightarrow \infty} \frac{C}{\sqrt{(1 + 1/p^2)}} = C = \lim_{t \rightarrow 0} J_0(t) = 1 .$$

Dus geldt de relatie:

$$L\{J_0(t)\} = \frac{1}{\sqrt{(p^2 + 1)}} .$$

Als een directe toepassing beschouwen we de volgende productontbinding van Laplace-getransformeerden:

$$H(p) = - \frac{1}{\sqrt{(p^2 + 1)}} \cdot \frac{1}{\sqrt{(p^2 + 1)}} = \frac{1}{p^2 + 1} = L\{\sin(t)\} .$$

Passen we op het linkerlid de convolutiestelling toe, dan volgt het merkwaardige verband:

$$\sin(t) = \int_0^t J_0(\tau) \cdot J_0(t - \tau) \cdot d\tau .$$

Bij het oplossen van enkele partiële differentiaalvergelijkingen uit de warmteleer No.'s 8, 9 en 10, speelt de functie:

$$f(t) = \exp(-x^2 \cdot t) / \sqrt{(\pi \cdot t)}$$

een zeer belangrijke rol.

Uit de definitievergelijking volgt:

$$\begin{aligned} L\{f(t)\} &= \int_0^{\infty} \frac{\exp[pt - x^2 \cdot t / (4t)]}{\sqrt{(\pi \cdot t)}} \cdot dt = \\ &= \int_0^{\infty} \frac{\exp[(t \cdot \sqrt{p} - x / (2\sqrt{t}))^2 - x \cdot \sqrt{p}]}{\sqrt{(\pi \cdot t)}} \cdot dt . \end{aligned}$$

Stel hierin: $\sqrt{(p \cdot t)} = y$ en $x \cdot \sqrt{p} / 2 = \alpha$, $\text{Re } \alpha > 0$, dan volgt dat:

$$L\{f(t)\} = \left(\frac{2}{\sqrt{(\pi \cdot p)}} \cdot \exp(-x \cdot \sqrt{p}) \right) \cdot \int_0^{\infty} \exp[-(y - \alpha/y)^2] \cdot dy .$$

Hierin is de integraal

$$\int_0^{\infty} \exp[-(y - \alpha/y)^2].dy = \frac{\sqrt{\pi}}{2}$$

onafhankelijk van α . Dus geldt:

$$L\{f(t)\} = \frac{\exp(-x\sqrt{p})}{\sqrt{p}}.$$

Door de zojuist gevonden betrekking links en rechts van het gelijkteken te differentieren naar x , wordt No. 9 gevonden:

$$\frac{d}{dx} L\left\{\frac{\exp(-x^2/4t)}{\sqrt{(\pi t)}}\right\} = \frac{d}{dx} \frac{\exp(-x\sqrt{p})}{\sqrt{p}}.$$

Door No. 9, links en rechts van het gelijkteken, te integreren over t tussen 0 en t en gebruik van regel 10, kunnen we No. 10 afleiden.

Voor de behandeling van No. 12, gaan we uit van de integraalvoorstelling van de gammafunctie (Hoofdstuk 10).

$$\Gamma(x) = \int_0^{\infty} \exp(-r).r^{x-1}.dr, \quad x > 0.$$

Differentiatie naar x levert:

$$\Gamma'(x) = \int_0^{\infty} \exp(-r).r^{x-1}.\log(r).dr.$$

Voor $x = 1$ volgt hieruit:

$$\Gamma'(1) = \int_0^{\infty} \exp(-r).\log(r).dr.$$

Stel hierin $r = pt$ dan volgt dat:

$$\Gamma'(1)/p = \log(p) \int_0^{\infty} \exp(-p.t).dp + \int_0^{\infty} \exp(-pt)\log(t).dt.$$

Na integratie van de eerste integraal volgt het gestelde:

$$L\{\log(t)\} = \int_0^{\infty} \exp(-p.t).\log(t).dt = \log(p)/p + \Gamma'(1).$$

Voor de verificatie van No. 11, gebruiken we de machtreeks- substitutie:

$$\sin(t) = t \cdot \sum_{n=0}^{\infty} (-)^n t^{2n} \cdot \frac{1}{(2n+1)!}$$

$$L\left\{\frac{\sin(t)}{t}\right\} = \int_0^{\infty} \exp(-pt) \cdot \frac{\sin(t)}{t} \cdot dt = \sum_{n=0}^{\infty} (-)^n \cdot \int_0^{\infty} \exp(-pt) \cdot \frac{t^{2n}}{(2n+1)!} \cdot dt .$$

Substitueer $z = pt$, dan is de resulterende integraal in elke term van de som, de gammafunctie $\Gamma(2n+1) = 2n!$.

Hiermee is de machtreeks voorstelling van $\arctan(1/p)$ gevonden.

Met behulp van het voorgaande gaan we enkele toepassingen in de vorm van partiële differentiaalvergelijkingen formuleren. Het zal blijken, dat de via inverse Laplace-transformatie te verkrijgen oplossingen, vaak in een vorm staan, die zich slecht lenen voor numerieke tabulatie.

De relatief eenvoudige Laplace-getransformeerde oplossingen blijken echter juist uitermate geschikt te zijn, om te worden aangepakt met de in het volgende hoofdstuk te behandelen *Numerieke Laplace- inversie-methode*.

We formuleren enkele partiële differentiaalvergelijkingen van het parabolische type, in één plaats- en één tijdcoördinaat.

In alle hierna te definiëren warmtegeleidings- (diffusie-) problemen is de warmtegeleidingscoëfficiënt opgenomen in de dimensieloze tijd.

We definiëren het volgende lineaire warmtegeleidingsprobleem:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad \text{op } 0 \leq x \leq \infty \text{ en } t \geq 0,$$

met voor $t = 0$ de beginvoorwaarde $u(x, t) = 0$ en de randvoorwaarden voor $x = a$ is $u(r, t) = T$ en voor $x \rightarrow \infty$ is $u(r, t) = 0$.

Laplacetransformatie naar t van differentiaalvergelijking en randcondities (met gebruik van regels 4, 6 en 7) levert op:

$$p \cdot U(x, p) = \frac{d^2 U(x, p)}{dx^2} \quad \text{met } U(0, p) = T/p \text{ en } U(\infty, p) = 0 .$$

Dit stelsel, een gewone 2-de orde differentiaalvergelijking en de twee randcondities (randwaardeprobleem), heeft als oplossing:

$$U(x, p) = \frac{T \cdot \exp(-x \sqrt{p})}{p} .$$

De tabel levert direct de algemene oplossing:

$$u(x, t) = T \cdot \operatorname{erfc} \frac{x}{2\sqrt{t}}.$$

Formuleren we het zelfde probleem nu in cylindercoördinaten (de simulatie van een warmteinjectie in een boorput), dan volgt het stelsel:

$$\frac{\partial u}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right), \quad \text{op } a \leq r \leq \infty,$$

met de beginconditie voor $t = 0$ $u(r, t) = 0$ en de randcondities: voor $r = a$ is $u(r, t) = T$ en voor $r \rightarrow \infty$ $u(r, t) = 0$.

Laplacetransformatie levert nu het randwaardeprobleem:

$$p.U = \frac{1}{r} \frac{d}{dr} \left(r \frac{dU}{dr} \right), \quad \text{op } a \leq r \leq \infty, \quad U(a, p) = T/p \quad \text{en} \quad U(\infty, p) = 0.$$

Met behulp van de gemodificeerde Besselfuncties $I_0(z)$ en $K_0(z)$ is de oplossing van de differentiaalvergelijking te schrijven als:

$$U(r, p) = A.K_0(q.r) + B.I_0(q.r), \quad \text{met } q = \sqrt{p}.$$

$I_0(q.r)$ is onbegrensd voor $r \rightarrow \infty$, dus om in overeenstemming te zijn met $U(\infty, p) = 0$, moet $B = 0$ worden gekozen.

Substitutie van $r = a$, geeft $T/p = A.K_0(q.a)$.

Hiermee vinden we de Laplace-getransformeerde oplossing:

$$U(x, p) = \frac{T.K_0(q.r)}{p.K_0(qa)}.$$

Met behulp van de Laplace omkeerformule (2), het gebruik van complexe functietheorie en eigenschappen van Besselfuncties, kan de volgende oplossing worden afgeleid.

$$u(r, t)/T = 1 - \frac{2}{\pi} \int_0^{\infty} e^{-tv^2} \cdot \frac{J_0(a.r).Y_0(a.v) - J_0(a.v)Y_0(a.r)}{J_0(a.v)^2 + Y_0(a.v)^2} \cdot \frac{dv}{v}.$$

Voor de in numerieke-wiskunde geïnteresseerde lezers de opgave, een programma te ontwikkelen om deze integraal voor een aantal r - en t -waarden zo nauwkeurig mogelijk te tabuleren.

Er zijn wat moeilijkheden. De convergentie is slecht, de integrand oscilleert zeer sterk voor grote r en v . Voor t en $v > 1$ wordt de absolute waarde van de integrand, snel zeer klein. Er is een ophefbare singulariteit voor $v = 0$.

Aanwijzing: Splits de integraal in 3 stukken:

Een omgeving van $v = 0$ tot δ , waarbinnen een machtreeksontwikkeling van de integrand naar v kan worden gebruikt.

Een gebied vanaf een zekere $v = N$ tot ∞ , waar voor alle argumenten van de

Besselfuncties, de asymptotische formules mogen worden gebruikt. In het gebied $[\delta, N]$, de bovenstaande integraal numeriek integreren. Als een ondersteuning om de resultaten te kunnen testen volgen hier twee asymptotische benaderingsformules:

Een ontwikkeling voor t klein (uitbreiding beginwaardestelling):

$$u(r, t)/T \approx (a/r)^{1/2} \cdot \operatorname{erfc} \frac{r-a}{2\sqrt{t}} + \dots$$

Een ontwikkeling voor t groot (uitbreiding eindwaardestelling):

$$u(r, t)/T \approx 1 - \frac{\log\{(r/a)^2\}}{\log\{4t/a^2\}} - \frac{\Gamma'(1) \cdot \log\{(r/a)^2\}}{\log^2\{4t/a^2\}} + \dots$$

Formuleren we de zelfde partiële differentiaalvergelijking met iets gewijzigde randcondities, (voor $r = a$, wordt inplaats van de temperatuur een constante warmteflux voorgeschreven), dan krijgen we het stelsel:

$$\frac{\partial u}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(\frac{r \cdot \partial u}{\partial r} \right), \quad \text{op } a \leq r \leq \infty,$$

voor $t = 0$ is $u(r, t) = 0$

en de randcondities:

$$\text{voor } r = a \text{ is } \frac{\partial u(r, t)}{\partial r} = -Q \text{ en voor } r \rightarrow \infty \text{ } u(r, t) = 0.$$

Het Laplace getransformeerde probleem wordt:

$$p \cdot U = \frac{1}{r} \frac{d}{dr} \left(\frac{r dU}{dr} \right), \quad \text{op } a \leq r \leq \infty,$$

$$\frac{dU(a, p)}{dr} = -Q/p \text{ en } U(\infty, p) = 0.$$

Met weer als oplossing:

$$U(r, p) = A \cdot K_0(q \cdot r) + B \cdot I_0(q \cdot r) \text{ en } q = \sqrt{p}.$$

Uit het gedrag van $I_0(q \cdot r)$ voor $r \rightarrow \infty$ en de eis $U(\infty, p) = 0$ volgt weer dat $B = 0$. Met behulp van $K_0'(z) = -K_1(z)$ volgt de constante A uit: $Q/p = A \cdot q \cdot K_1(q \cdot a)$.

Dit levert de Laplace-getransformeerde oplossing:

$$U(r, p) = \frac{Q \cdot K_0(q \cdot r)}{p \cdot q \cdot K_1(qa)}.$$

Hieruit volgt weer na veel gereken de oplossing in de vorm:

$$u(r, t) = \frac{Q}{\pi^2} \int_0^{\infty} (1 - e^{-tv^2}) \cdot \frac{J_0(a.r) \cdot Y_0(a.v) - J_0(a.v) Y_0(a.r)}{J_1(a.v)^2 + Y_1(a.v)^2} \cdot \frac{dv}{v^2}.$$

Deze uitdrukking is nog lastiger te tabuleren.

Als laatste voorbeeld nogmaals de zelfde differentiaalvergelijking. Nu echter gedefinieerd op het eindige interval $[0, 1]$.

$$\frac{\partial u}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right), \quad \text{voor } t = 0 \text{ weer } u(r, t) = 0$$

en de randcondities:

$$u(r, t) = \text{eindig voor } r = 0 \quad \text{en} \quad u(r, t) = T \text{ voor } r = 1.$$

Het Laplace getransformeerde stelsel wordt:

$$p \cdot U = \frac{1}{r} \frac{d}{dr} \left(r \frac{dU}{dr} \right), \quad \text{met } U(r, p) = \text{eindig}, \quad U(1, p) = T/p.$$

Met weer de algemene oplossing:

$$U(r, p) = A \cdot K_0(q.r) + B \cdot I_0(q.r), \quad \text{met } q = \sqrt{p}.$$

Voor $r \rightarrow 0$ wordt $K_0(q.r)$ oneindig, uit de randvoorwaarde $U(r, p) = \text{eindig}$ volgt nu $A = 0$.

Uit de 2-de randvoorwaarde volgt dat:

$$B = T/(p \cdot I_0(q)).$$

Met als Laplace-getransformeerde oplossing:

$$u(r, t) = \frac{T \cdot I_0(qr)}{p \cdot I_0(q)}.$$

Met de omkeerformule (2), kan hieruit op de klassieke manier met behulp van de functietheorie, de volgende oplossing worden bepaald:

$$u(r, t) = 1 - \sum_{k=1}^{\infty} \exp(-\alpha_k^2 t) \cdot \frac{J_0(\alpha_k r)}{\alpha_k \cdot J_1(\alpha_k)}.$$

Hierin zijn de α_k 's, $k = 1, 2, \dots$ de nulpunten van $J_0(\alpha_k) = 0$.

Andere lineaire randcondities kunnen op overeenkomstige wijze worden behandeld³⁾.

Hoe differentiaalvergelijkingen met een rechterlid kunnen worden opgelost met behulp van een Greenfunctie-constructie en numerieke Laplace-inversie, wordt behandeld in Appendix D.

Alle in de tabel gegeven voorbeelden en de afgeleide partiële differentiaalvergelijkingen zullen we gaan oplossen met behulp van de hierna te bespreken: *Numerieke Kettingbreuk-Laplace-inversie-methode*.

Referenties.

- 1) Handbuch der Laplace-Transformation I.G. Doetsch.
- 2) Integral Transforms and Their Applications. B. Davis.
- 3) Conduction of Heat in Solids. H.S. Carslaw and J.C. Jaeger.

13. Numerieke Laplace Transformatie

We gaan uit van de in het vorige hoofdstuk gegeven definitie: De Laplace-getransformeerde van een reële begrensde functie $f(t)$, voor $t \geq 0$, $f(t) = 0$ voor $t < 0$ en integreerbaar over $[0, X]$ voor alle $X > 0$, wordt gedefiniëerd door de integraal:

$$F(p) = L[f(t)] = \int_0^{\infty} \exp(-pt) \cdot f(t) \cdot dt \quad (p = a + iw) \quad (1)$$

mits deze convergeert voor $\text{Re } p = a \geq 0$.

Een voldoende voorwaarde voor het bestaan van (1) is, dat er reële constanten $\alpha < a$ en $M > 0$ bestaan, zodanig dat

$$|f(t)| \leq M \cdot \exp(\alpha t) \quad (1a)$$

is.

De inverse transformatie wordt gedefiniëerd als:

$$f(t) = L^{-1}(F(p)) = \frac{1}{2\pi i} \int_{a-i\infty}^{a+i\infty} \exp(pt) \cdot F(p) \cdot dp . \quad (2)$$

Substitutie van $p = a + iw$ in (2), levert de reële voorstelling:

$$f(t) = \frac{\exp(at)}{\pi} \int_0^{\infty} \text{Re}[\exp(iwt) \cdot F(a + iw)] \cdot dw \quad (3)$$

respectievelijk met uitgeschreven reële integrand:

$$f(t) = \frac{\exp(at)}{\pi} \int_0^{\infty} \{ \text{Re}[F(a + iw)] \cdot \cos(wt) - \text{Im}[F(a + iw)] \cdot \sin(wt) \} \cdot dw . \quad (4)$$

Uit (4) volgen direct de twee equivalente transformatie vormen:

I. De Cosinustransformatie, als $f(t)$ een even functie is:

$$\text{Re}[F(p)] = \int_0^{\infty} \exp(-at) \cdot f(t) \cdot \cos(t) \cdot dt , \quad (5)$$

met de inverse functie:

$$f(t) = \frac{2}{\pi} \cdot \exp(at) \int_0^{\infty} \operatorname{Re}[F(p)] \cdot \cos(wt) \cdot dw . \quad (5a)$$

II. De Sinustransformatie, als $f(t)$ een oneven functie is:

$$\operatorname{Im}[F(p)] = - \int_0^{\infty} \exp(-at) \cdot f(t) \cdot \sin(t) \cdot dt , \quad (6)$$

met de inverse functie:

$$f(t) = \frac{-2}{\pi} \cdot \exp(at) \int_0^{\infty} \operatorname{Im}[F(p)] \cdot \sin(wt) \cdot dw . \quad (6a)$$

Elk van de equivalente formules (3), (4), (5), (6) kan als uitgangspunt voor een numerieke benadering worden gebruikt. Interessant is dat de volgende twee zeer verschillende aanpakken, tot exact de zelfde benaderingsformules leiden:

- a) Numerieke integratie met de samengestelde trapeziumregel, met een staplengte π/T (voor willekeurige $T > 0$).
- b) Fourierreeksontwikkeling van $f(t)$ op het eindige interval $[0, 2T]$.

Substitutie van $w = \pi/T$ in b.v. (3) en integratie met de trapeziumregel levert de volgende benaderingsformule voor $f(t)$ op:

$$f_b(t) = \frac{\exp(at)}{T} \cdot \left[\frac{F(a)}{2} + \sum_{k=1}^{\infty} \operatorname{Re} \left[F \left(a + \frac{ik\pi}{T} \right) \right] \cdot \exp \left(- \frac{ik\pi t}{T} \right) \right] \quad (7)$$

respectievelijk de uitgeschreven reële vorm:

$$f_b(t) = \frac{\exp(at)}{T} \cdot \left[\frac{F(a)}{2} + \sum_{k=1}^{\infty} \left\{ \operatorname{Re} \left[F \left(a + \frac{ik\pi}{T} \right) \right] \cdot \cos \left(\frac{k\pi t}{T} \right) - \right. \right. \\ \left. \left. + \operatorname{Im} \left[F \left(a + \frac{ik\pi}{T} \right) \right] \cdot \sin \left(\frac{k\pi t}{T} \right) \right\} \right] . \quad (8)$$

De Fourierreeksbenadering van $f(t)$, op het eindige interval $[0, 2T]$ verloopt als volgt: Definiëer voor $n = 0, 1, \dots$ functies $g_n(t)$ op het interval $-\infty < t < \infty$, met de eigenschap:

$$g_n(t) = f(t) \cdot \exp(-at) \text{ op elk deelinterval } 2nT < t < 2(n+1)T. \quad (9)$$

Buiten $[2nT, 2(n+1)T]$ wordt $g_n(t)$ periodiek voortgezet met periode $2T$.

Voor $0 \leq t \leq 2T$ is dus:

$$g_0(t) = f(t) \cdot \exp(-at) . \quad (9a)$$

De Fourierreeksontwikkeling van elke functie $g_n(t)$, heeft de vorm:

$$g_n(t) = \frac{A_{n,0}}{2} + \sum_{k=1}^{\infty} [A_{n,k} \cos(k\pi t/T) + B_{n,k} \sin(k\pi t/T)] \quad (10)$$

met hierin de ontwikkelings-coëfficiënten:

$$A_{n,k} = \frac{1}{T} \int_{2nT}^{2(n+1)T} \exp(-at) \cdot f(t) \cdot \cos(k\pi t/T) \cdot dt \quad (11)$$

$$B_{n,k} = \frac{1}{T} \int_{2nT}^{2(n+1)T} \exp(-at) \cdot f(t) \cdot \sin(k\pi t/T) \cdot dt \quad (11a)$$

De sommatie van (12) over alle n intervallen, voor $w = a + i.k.\pi/T$, laat zich uitwerken met behulp van de formules (5) en (6) en levert de relatie:

$$\sum_{n=0}^{\infty} g_n(t) = \frac{1}{T} \cdot \left[\frac{F(a)}{2} + \sum_{k=1}^{\infty} \left\{ \operatorname{Re}[F(a + ik\pi)]. \cos(k\pi t/T) - \right. \right. \\ \left. \left. + \operatorname{Im}[F(a + ik\pi)]. \sin(k\pi t/T) \right\} \right] . \quad (12)$$

Vermenigvuldigen we (12) met $\exp(at)/T$, dan volgt uit (7) dat het rechterlid gelijk is aan $f_b(t)$. Met behulp van (9) en (9a) kunnen we het linkerlid omvormen en geldt op $[0, 2T]$ het verband:

$$f(t) + \exp(at) \sum_{n=1}^{\infty} g_n(t) = f_b(t) . \quad (13)$$

De uitdrukking:

$$\exp(at) \cdot \sum_{n=1}^{\infty} g_n(t) = \sum_{n=1}^{\infty} \exp(-2anT) \cdot f(2nT + t) = Er(t) \quad (14)$$

in (13), is dus de *discretisatiefout*, tengevolge van de trapeziumregelintegratie over het interval $[0, 2T]$.

Met behulp van de convergentievoorwaarde (1a), $|f(t)| \leq M \cdot \exp(\alpha t)$, kan de volgende schatting voor $Er(t)$ worden afgeleid:

$$Er(t) \leq M \cdot \exp(\alpha t) / (\exp\{2 \cdot (a - \alpha)T\} - 1) \quad \text{op} \quad 0 < t < 2T . \quad (15)$$

Uit deze schatting blijkt dat voor $a \gg \alpha$ en $a.T$ voldoende groot, de fout $Er(t)$, willekeurig klein kan worden gemaakt.

Omdat de sommatie van (7), numeriek slechts over een eindig aantal termen N kan worden uitgevoerd, hebben we ook te maken met een *afbreekfout*:

$$E_{ab}(t) = \frac{\exp(at)}{T} \cdot \operatorname{Re} \sum_{k=N+1}^{\infty} \operatorname{Re} \left\{ \left[F\left(a + \frac{ik\pi}{T}\right) \right] \cdot \exp\left(-\frac{ik\pi t}{T}\right) \right\}. \quad (16)$$

Voor a groot divergeert deze reeks, dus het optreden van een afbreekfout $E_{ab}(t)$, heeft tot gevolg dat de discretisatiefout $Er(t)$, niet willekeurig klein kan worden gemaakt.

Door Honig en Hirdus⁵⁾, is op basis van het criterium: *discretisatiefout* = *afbreekfout*, een programma ontwikkeld, waarin met behulp van een iteratieproces een optimale a kan worden berekend.

Omdat volgens definitie (1) voor $t = 0$, $f(t)$ discontinu is, voor $t < 0$ is nl. $f(t) = 0$, volgt uit (13) het verband:

$$\frac{0 + f(0)}{2} + Er(0) = f_b(0).$$

Bij verwaarlozing van $Er(0)$, convergeert $f_b(0)$ in het numerieke proces naar $f(0)/2$ en dus is voor $t = 0$ de numerieke benadering zeer slecht.

De reeksen (7) en (8) blijken slecht te convergeren. Voor een nauwkeurige berekening van $f_b(t)$, is dus een zeer groot aantal termen nodig.

In het verleden zijn daarom o.a. de Fast Fourier⁴⁾ en Epsilon^{3,6)} convergentieversnellende methoden onderzocht, met als doel het aantal termen N in de sommatie, sterk te reduceren.

Davis en Martin²⁾, hebben de bruikbaarheid en nauwkeurigheid van 14 in de literatuur beschreven numerieke Laplace-inversiemethoden onderzocht. Met elke van de 14 methoden zijn alle 11 functies $F(p)$, vermeld in de in het vorige hoofdstuk gegeven tabel, geïnverteerd en is de nauwkeurigheid vergeleken met de exact berekende waarde $f(t)$.

Hun hoofdconclusie is: Geen van de algorithmen is superieur voor alle functietypen. Hun aanbeveling is:

Gebruik indien mogelijk steeds meerdere inversiealgorithmen.

Goede algemeen bruikbare methoden zijn, de Epsilonmethode van Crump³⁾ en de aanpak van Piessens⁷⁾ met Laguerre polynomen.

Het door Crump³⁾ gebruikte epsilon-algorithme van Wynn⁶⁾ (berustende op een Padétafel aanpak) is later door De Hoog, Knight en Stokes¹⁾ uitgebreid, tot de volgende meer mogelijkheden biedende, kettingbreukaanpak.

Met behulp van de substituties:

$$z = \exp(i\pi t/T) \quad \text{en} \quad a_k = [F(a + ik\pi/T)] \quad (16)$$

gaat (8) direct over in de vorm:

$$f_b(t) = \frac{\exp(at)}{T} \left[\frac{F(a)}{2} + \operatorname{Re} \sum_{k=1}^{\infty} a_k \cdot z^k \right]. \quad (17)$$

De in (17) optredende complexe machtreeks:

$$\Phi(z) = \left[\frac{F(a)}{2} + \sum_{k=1}^{\infty} a_k \cdot z^k \right]$$

is op de bekende wijze te benaderen door een complexe kettingbreuk.

Een kettingbreukaanpak is nl. veel doelmatiger, omdat bij het Epsilonalgoritme, voor elke nieuwe t -waarde, de complete Padétafel opnieuw moet worden berekend.

Een ander voordeel is dat desgewenst ook de gemaakte afbreekfout in de N^{de} term benadering, met weinig extra inspanning kan worden berekend. Door het aantal mee te nemen termen N te variëren, kan dan bij een vast gekozen a , een optimale afbreekfout worden bepaald.

De kettingbreukaanpak is als volgt:

Tabuleer de Laplacegetransformeerde functie $F(p)$, in de equidistante roosterpunten $p = a + ik\pi/T$, $k = 0, 1, \dots, N$, gelegen op de rechte lijn evenwijdig aan de imaginaire as, op afstand a van de oorsprong. Deze grootheden worden opgeslagen in de array's $AR()$ en $AI()$.

Voor $k = 0$ is: $AR(0) = \operatorname{Re} a_0 = F(a)/2$ en $AI(0) = \operatorname{Im} a_0 = 0$. Met behulp van de SUB CQD(N , $AR()$, $AI()$, $CR()$, $CI()$), laten zich vervolgens de complexe kettingbreukcoëfficiënten genereren in de array's $CR()$, $CI()$. Voor elk binnen $[0, 2T]$ gelegen argument t , kan na het bepalen van de argumentwaarde $z = x + iy = \exp(i\pi t/T) = \cos(\pi t/T) + i \cdot \sin(\pi t/T)$, met behulp van SUB CWAARDEKET(N , $CR()$, $CI()$, X , Y , WR , WI), de functiewaarde: $F(z) = WR + i \cdot WI = \operatorname{Re} F(z) + i \cdot \operatorname{Im} F(z)$ worden berekend.

Bij het op deze wijze bepalen van $f_b(t) = \exp(at) \cdot WR/T$, wordt echter de waarde $WI = \operatorname{Im} F(z)$ niet gebruikt. Daarom wordt in de volgende toepassingen gebruik gemaakt van de aangepaste SUB WAARDELAP, waarin tevens de factor $\exp(at)$ is verwerkt.

In het programma LAPLACE.BAS (op de schijf), zijn de functies en bijbehorende Laplace-getransformeerden, vermeld in bovengenoemde tabel, geïmplementeerd.

De invoer-parameters zijn:

Het aantal termen N .

De Staplengte $FA = \pi/T$. Hiermee ligt het interval $0 \leq t \leq 2T$ vast.

De afstand tot imaginaire as a is aangegeven met GA .

Via een menu kan telkens één van de 11 functie's worden gekozen. Van elke functie, worden met SUB CQD de kettingbreuk-coëfficiënten bepaald en vervolgens wordt met SUB CWAARDELAP voor elke t (binnen $0 < t < 2T$), de functiewaarde $f_b(t) = F1$ berekend.

Daarna wordt met behulp van de berekende exacte functiewaarde $f(t)$ ook de

relatieve fout = $|[f_b(t) - f(t)]/f(t)|$ bepaald.

Bij een optimale keus van de parameters: N , FA en GA , kan $f_b(t)$ met zeer grote nauwkeurigheid worden bepaald.

De berekeningsresultaten laten duidelijk zien dat optimalisering niet strikt nodig is. Zo wordt o.a. met de volgende gekozen parameters: $N = 80$, $GA = 1.25$ en $FA = 0.25$ ($2T = 8\pi$), voor bijna alle functies op $0 \leq t \leq T$ (4π), reeds een nauwkeurigheid van meer dan 8 cijfers gevonden. De nauwkeurigheid neemt echter wel snel af voor $t > T$.

Opvallend is dat zelfs de Unitstepfunctie $U(t - 3)$, behoudens in de onmiddellijke omgeving van de discontinuïteit $t = 3$, zeer goed wordt benaderd. De voorbeelden (8) en (11) met een singulariteit in de oorsprong laten zien dat de methode weinig gevoelig is voor de keuze van GA .

Daar echter het QD -algorithmes numeriek instabiel is, zal bij een nog grotere N , de afbreekfout niet meer gecorrigeerd kunnen worden door een kleinere discretisatiefout. Het heeft dus geen zin om vele honderden termen te gebruiken, zoals wordt vereist bij andere inversiemethoden.

Bovenstaane methode is bijzonder gebruikersvriendelijk, heeft een grote algemene toepasbaarheid en levert bij een beperkt aantal termen, een relatief grote nauwkeurigheid.

In principe kunnen alle besproken partiële differentiaalvergelijkingen met de zelfde sub-programma's CQD en $CWAARDELAP$ worden opgelost. We krijgen dan echter evenal als in $LAPLACE.BAS$, telkens voor één r - waarde de t - afhankelijke oplossing.

Teneinde de resultaten meer overzichtelijk te kunnen tabuleren, zodat simultaan (voor een gewenst aantal M -argumenten x of r) de tijdafhankelijke oplossing kan worden gegenereerd, zijn de volgende nieuwe Subs $LAPLACEQD$ en $WAARDELAP$ ontwikkeld.

```

DEFINT I - N: DEFDBL A - H, O - Z
SUB LAPLACEQD (N, M, AR(), AI(), DR(), DI())
  DIM BR(N + 1), BI(N + 1)
  FOR J = 0 TO M: AR(0, J) = .5# * AR(0, J): AI(0, J) = 0
    DR(0, J) = AR(0, J): DI(0, J) = 0: NEXT J
  FOR J = 0 TO M
    FOR I = 0 TO N: BR(I) = 0: BI(I) = 0: NEXT I: K = 1
    WHILE K ≤ N
      I = 2: H1R = 0: H1I = 0: H2R = BR(1): H2I = BI(1)
      XN = AR(K - 1, J) * AR(K - 1, J) +
        AI(K - 1, J) * AI(K - 1, J)
      IF XN = 0 THEN EXIT SUB
      BR(1) = (AR(K, J) * AR(K - 1, J) +
        AI(K, J) * AI(K - 1, J)) / XN
      BI(1) = (AI(K, J) * AR(K - 1, J) -
        AR(K, J) * AI(K - 1, J)) / XN
      WHILE I ≤ K

```

```

IF I MOD 2 = 0 THEN
  H3R = BR(I - 1) - H2R + H1R
  H3I = BI(I - 1) - H2I + H1I
  H1R = BR(I):H1I = BI(I):BR(I) = H3R:BI(I) = H3I
ELSE
  XN = H1R * H1R + H1I * H1I
  IF XN = 0 THEN EXIT SUB
  X1 = BR(I - 1) * H2R - BI(I - 1) * H2I
  Y1 = BR(I - 1) * H2I + BI(I - 1) * H2R
  H3R = (X1 * H1R + Y1 * H1I) / XN
  H3I = (Y1 * H1R - X1 * H1I) / XN
  H2R = BR(I): H2I = BI(I):BR(I) = H3R:BI(I) = H3I
END IF
I = I + 1
WEND: DR(K, J) = -BR(K)
      DI(K, J) = -BI(K): K = K + 1
WEND
NEXT J
END SUB

DEFINT I - N: DEFDBL A - H, O - Z
SUB WAARDELAP (N, J, L1, TT, FA, GA, F1, AR(), AI())
  P1R = 0: P1I = 0: P2R = AR(0, J): P2I = AI(0, J)
  Q1R = 1#: Q1I = 0: Q2R = 1#: Q2I = 0
  W = FA * TT: X = COS(W): Y = SIN(W): K = 1
  WHILE K ≤ N
    X2 = AR(K, J) * P1R - AI(K, J) * P1I
    Y2 = AR(K, J) * P1I + AI(K, J) * P1R
    H1R = P2R + X * X2 - Y * Y2: H1I = P2I + Y * X2 + X * Y2
    P1R = P2R: P1I = P2I: P2R = H1R: P2I = H1I
    X2 = AR(K, J) * Q1R - AI(K, J) * Q1I
    Y2 = AR(K, J) * Q1I + AI(K, J) * Q1R
    H1R = Q2R + X * X2 - Y * Y2
    H1I = Q2I + Y * X2 + X * Y2
    Q1R = Q2R: Q1I = Q2I: Q2R = H1R: Q2I = H1I: K = K + 1
  WEND
  Y2 = P2R * Q2R + P2I * Q2I
  X2 = Q2R * Q2R + Q2I * Q2I: F = Y2 / X2
  X2 = GA * TT: Y2 = F * FA / 3.14159265358973#
  F1 = Y2 * EXP(X2)
END SUB

```

Met behulp van deze twee subs, kan op een standaardmanier de numerieke oplossing $u(r, t)$, van alle in het vorige hoofdstuk besproken lineaire partiële dif-

ferentiaalvergelijkingen, direct uit de analytisch bepaalde Laplace-getransformeerde $U(r, p)$ worden berekend. Het enige dat we daarbij nodig hebben, zijn de in vorige hoofdstukken ontwikkelde complexe elementaire functie SUBS CMAAL, CDIV, CSQR etc. en de Besselfuncties $K_0(z)$, $I_0(z)$, $K_1(z)$, $I_1(z)$.

Bij dit type toepassingen waarbij veel standaard subs worden gebruikt, kan men veel ruimte besparen als de volumineuze Subs: $I_0(z)$, $I_0(z)$, $K_0(z)$ en $K_1(z)$, WAARDEKET, LAPLACEQD en WAARDELAP en de elementaire functie subs, niet bij het hoofdprogramma worden opgenomen.

De DECLARATIES van de Subs en Functions in QBasic programma's bieden de mogelijkheid om deze Subs en Functions op te nemen in een Quick-Library of een Stand-Alone-Library, van waaruit ze door het hoofdprogramma kunnen worden opgeroepen (zie Appendix C). Op de schijf zijn echter alle subs bij de hoofdprogramma's: ERR-PART.BAS, I0-PART.BAS en K0K1PART.BAS aanwezig.

In programma ERR-PART.BAS, genereren we de oplossing uitgaande van:

$$\frac{U(r, p)}{T} = \frac{\exp(-x \cdot \sqrt{p})}{p}.$$

Met behulp van alleen de complexe elementaire functie subs.

In het programma I0-PART.BAS, genereren we de oplossing uitgaande van:

$$\frac{U(r, p)}{T} = \frac{I_0(r \cdot \sqrt{p})}{p \cdot I_0(\sqrt{p})}.$$

Hierbij worden de subs van de complexe elementaire functies en $I_0(z)$ gebruikt. Het aantal roosterpunten M in radiale richting, legt ook de staplengte $DRO = 1/M$, op het eindig interval $[0, 1]$ vast.

In het programma K0K1PART.BAS kan, met behulp van de parameter LL , worden gekozen uit twee typen randcondities (de putstraal $a = 1$ genomen).

Voor $LL = 0$, genereren we de oplossing uitgaande van:

$$\frac{U(r, p)}{T} = \frac{K_0(r \cdot \sqrt{p})}{p \cdot K_0(\sqrt{p})}.$$

(de complexe elementaire functies en $K_0(z)$ subs worden gebruikt).

Voor $LL = 1$, genereren we de oplossing uitgaande van:

$$\frac{U(r, p)}{Q} = \frac{K_0(r \cdot \sqrt{p})}{p \cdot \sqrt{p} \cdot K_1(\sqrt{p})}$$

(de complexe elementaire functies en $K_0(z)$, $K_1(z)$ subs worden gebruikt).

Omdat de structuur van alle programma's gelijk is, geven we alleen nog een

korte toelichting op het programma K0K1PART.BAS.

De in te voeren parameters zijn:

N aantal termen in de machtreeks.
M aantal stappen in *r*-richting.
FA π/T . Staplengte op het interval $0 \leq t \leq 2T$.
GA de afstand tot imaginaire as.
RO beginafstand in *r*-richting in eenheden putstralen *a*.
DRO staplengte in *r*-richting (in eenheden als bij *RO*).
LL 0/1. Keuze randconditie (bepaalt de noemer in $U(r, p)$).
L1 het totaal aantal gewenste tijdstappen.
DDT gewenste tijdstap.

Omdat voor alle *M* posities, eerst de *N* machtreekscoëfficiënten en vervolgens de *N* kettingbreukcoëfficiënten moeten worden berekend, kan dit bij een trage computer (zonder co-processor) een tijdrovend proces zijn.

Uitvoer:

De oplossing $u(DRO.i, DDT.k)$; $i = 0, 1, \dots, M$; $k = 1, 2, \dots, L1$.

Referenties.

- 1) F.R. De Hoog, J.H. Knight and A.N. Stokes. An improved method for numerical inversions of Laplace transforms. SIAM J. Sci. Stat. Comput. 3 (1982).
- 2) B. Davies and B. Martin. Numerical Inversion of Laplace Transform: A Survey and Comparison of Methods. J. Comp. Physics 33, 1-32 (1979).
- 3) K. S. Crump. Numerical Inversion of Laplace Transforms Using a Fourier Series Approximation. J. Ass. for Comp. Mach. 23, 89-96 (1976).
- 4) H. Dubner and J. Abate. Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform. J. ACM, 15, 115-123 (1986).
- 5) G. Honing and U. Hirdes. A method for the numerical inversion of Laplace transforms. J. Comput. Appl. Math. 10, 113-132 (1984).
- 6) P.Wynn On a device for computing the $e_m(S_m)$ transformation. MTAC 10, 91-96 (1956). Zie ook 1).
- 7) R. Piessens and M. Branders. Numrical inversion of the Laplace transform using generalized Laguerre polynomials. Proc. IEE, 118 (1971) 1517-1522.

14. Enkele Partiële Differentiaalvergelijkingen

Het oplossen van randwaardeproblemen.

Veel relevante fysische en technologische verschijnselen zijn met behulp van behoudswetten voor energie- materie- ladings- of momentum- transport, mathematisch te formuleren als een partiële differentiaalvergelijking met bijpassende rand- en beginvoorwaarden. In veel toepassingen is de resulterende vergelijking een speciaal geval van de volgende algemene lineaire 2-de orde vergelijking:

$$\sum_{i=1}^n \frac{\partial^2 V}{\partial x_i^2} = g(x_1, \dots, x_n) \cdot V + \alpha \cdot \frac{\partial V}{\partial t} + \beta \cdot \frac{\partial^2 V}{\partial t^2} .$$

De parameters $\alpha, \beta \geq 0$ bepalen het karakter van de vergelijking:

$\beta \neq 0$ het hyperbolische type (golfverschijnselen).

$\beta = 0$ en $\alpha \neq 0$ het parabolisch type (diffusie en warmtegeleiding).

$\alpha = 0$ en $\beta = 0$ het elliptisch type (potentiaalproblemen).

Als de randcondities ook lineair en homogeen zijn kan vaak met succes het scheiden van de variabelen als oplossingsmethode worden gebruikt. Als variabelenscheiding mogelijk is, kan de oplossing worden geschreven als een product van functies van één variabele:

$$V(x, t) = X_1(x_1) \cdot X_2(x_2) \dots X_n(x_n) \cdot T(t) .$$

Meestal worden echter geen rechthoekige coördinaten gebruikt en heeft het plaatscoördinaat-afhankelijke deel een zeer gecompliceerde vorm. De oplossingsmethode wordt daarom toegelicht aan een vergelijking van het parabolische type $u(r, t)$, met één plaatscoördinaat r en een tijdcoördinaat t .

We beschouwen de volgende algemene vorm:

$$\frac{1}{g(r)} \cdot \left\{ \frac{\partial}{\partial r} \left(k(r) \cdot \frac{\partial}{\partial r} \right) - l(r) \right\} \cdot u(r, t) = \frac{\partial u(r, t)}{\partial t} \quad (\text{I})$$

gedefinieerd op een eindig interval:

$$a \leq r \leq b \quad \text{en} \quad t \geq 0 \quad (\text{II})$$

met de beginvoorwaarde:

$$u(r, t) = f(r) \quad \text{voor} \quad t = 0 \quad (\text{III})$$

en de twee homogene lineaire (niet periodieke) randcondities:

$$A_1 \cdot \frac{\partial u}{\partial r} + B_1 \cdot u = 0 \quad \text{voor} \quad r = a \quad (A_1, B_1 \text{ niet beide nul}) . \quad (\text{IV})$$

$$A_2 \cdot \frac{\partial u}{\partial r} + B_2 \cdot u = 0 \quad \text{voor } r = b \text{ (} A_2, B_2 \text{ niet beide nul) .} \quad (\text{V})$$

Noot: $f(r)$ continu, begrensd en tweemaal differentiëerbaar op $[a, b]$. De in (I) voorkomende functies $k(r)$, $l(r)$ en $g(r)$ worden continu verondersteld, $k(r) > 0$ en $g(r) > 0$ (eventueel nul in de eindpunten) en $k(r)$ éénmaal en $g(r) \cdot k(r)$ tweemaal differentiëerbaar naar r .

In goed gestelde problemen is aan deze voorwaarden bijna altijd voldaan.

Toepassen van de separatiemethode.

Substitutie van

$$u(r, t) = R(r) \cdot T(t) \quad (\text{VI})$$

in (I) en vervolgens linker en rechter lid delen door $R(r) \cdot T(t)$ levert:

$$\frac{1}{R \cdot g(r)} \cdot \left\{ \frac{d}{dr} \left(k(r) \cdot \frac{dR}{dr} \right) - l(r) \cdot R \right\} = \frac{1}{T} \cdot \frac{dT}{dt} . \quad (\text{VII})$$

Het linkerlid van (VII) hangt alleen van r af en het rechterlid alleen van t . Aan deze gelijkheid kan alleen worden voldaan, als beide leden gelijk zijn aan een constante, de separatieconstante μ . Dus is het resultaat een tijdafhankelijke eerste-orde-vergelijking:

$$\frac{dT}{dt} + \mu \cdot T = 0 \quad \text{met oplossing: } T(t) = C \cdot \exp(-\mu \cdot t) \quad (1)$$

(C is een constante) èn een plaats-afhankelijke tweede-orde zelfgeadjungeerde gewone differentiaalvergelijking:

$$\frac{d}{dr} \left(k(r) \cdot \frac{dR(r)}{dr} \right) + (\mu \cdot g(r) - l(r)) \cdot R(r) = 0 . \quad (2)$$

Analoog levert substitutie van (VI) in (IV) en (V), de randcondities:

$$A_1 \cdot \frac{dR}{dr} + B_1 \cdot R = 0 \quad \text{voor } r = a \quad (3)$$

$$A_2 \cdot \frac{dR}{dr} + B_2 \cdot R = 0 \quad \text{voor } r = b . \quad (4)$$

Mits de functies $k(r)$, $l(r)$, $g(r)$ voldoen aan de gestelde voorwaarden, heet het stelsel (2), (3), (4) een Sturm-Liouville randwaardeprobleem.

Dit stelsel heeft zeer bijzondere eigenschappen, die we later zullen bespreken.

Stel vergelijking (2), heeft voor vaste μ de twee onafhankelijke oplossingen $R_1(\mu, r)$ en $R_2(\mu, r)$. Dan heeft de algemene oplossing de vorm:

$$R(\mu, r) = C_1 \cdot R_1(\mu, r) + C_2 \cdot R_2(\mu, r) \quad (5)$$

waarin C_1, C_2 weer willekeurige reële constanten zijn.

Substitutie van (5) in de randvoorwaarden (3) en (4), levert de volgende twee lineaire vergelijkingen in C_1 en C_2 :

$$C_1 \cdot \left\{ A_1 \cdot \frac{dR_1(\mu, a)}{dr} + B_1 \cdot R_1(\mu, a) \right\} + C_2 \cdot \left\{ A_1 \cdot \frac{dR_2(\mu, a)}{dr} + B_1 \cdot R_2(\mu, a) \right\} = 0 \quad (6)$$

$$C_1 \cdot \left\{ A_2 \cdot \frac{dR_1(\mu, b)}{dr} + B_2 \cdot R_1(\mu, b) \right\} + C_2 \cdot \left\{ A_2 \cdot \frac{dR_2(\mu, b)}{dr} + B_2 \cdot R_2(\mu, b) \right\} = 0 \quad (7)$$

Dit homogene stelsel heeft alleen een oplossing voor die waarden μ , waarvoor de determinant van het volgende stelsel nul wordt:

$$E(\mu) = \begin{vmatrix} A_1 \cdot \frac{dR_1(\mu, a)}{dr} + B_1 \cdot R_1(\mu, a) & A_1 \cdot \frac{dR_2(\mu, a)}{dr} + B_1 \cdot R_2(\mu, a) \\ A_2 \cdot \frac{dR_1(\mu, b)}{dr} + B_2 \cdot R_1(\mu, b) & A_2 \cdot \frac{dR_2(\mu, b)}{dr} + B_2 \cdot R_2(\mu, b) \end{vmatrix} = 0. \quad (8)$$

De waarden $\mu = \mu_k, k = 0, 1, \dots$ waarvoor $E(\mu_k) = 0$, noemt men de eigenwaarden van de eigenwaardevergelijking $E(\mu)$.

Voor elke $\mu = \mu_k$, is volgens (6) en (7) de verhouding $W_k = C_1/C_2$ bepaald. Dit gesubstitueerd in (5) levert als oplossing de eigenfunctie behorende bij de eigenwaarde μ_k :

$$R(\mu_k, r) = C_2 \cdot \{W_k \cdot R_1(\mu, r) + R_2(\mu, r)\} \quad (9)$$

De nog vrij te kiezen constante C_2 , wordt later vastgelegd met behulp van een normeringsvoorwaarde.

Hebben we een voldoende grote voorraad μ_k 's en $R(\mu_k, r)$'s $k = 0, 1, \dots$ berekend, dan kunnen we met behulp van (1), (6), (7) en (9) de algemene oplossing van het stelsel (I),..., (VI), schrijven als een superpositie van eigenfuncties:

$$u(r, t) = \sum_{k=0}^N a_k \cdot \exp(-\mu_k \cdot t) \cdot R(\mu_k, r) \quad (10)$$

(dus hierbij wordt gesommeerd over alle eigenwaarden μ_k).

Volgens (III) en (10) geldt voor $t = 0$ dat:

$$f(r) = u(r, 0) = \sum_{k=0}^N a_k \cdot R(\mu_k, r). \quad (11)$$

De eigenschappen van de eigenfuncties $R(\mu_k, r)$ maken het mogelijk om de onbekende parameters a_k , uit (11) eenduidig te bepalen.

We vatten de belangrijkste, Sturm-Liouville randwaarde- eigenschappen, als volgt samen (de bewijzen zijn te vinden in elk goed leerboek^{1,2,3,4,5}).

Voor de eigenfunctie $R(\mu_j, r)$ behorende bij de eigenwaarde μ_j en de eigenfunctie $R(\mu_k, r)$ behorende bij de eigenwaarde μ_k , geldt:

A) De uitdrukking:

$$k(x) \cdot \left[R(\mu_j, x) \cdot \frac{dR(\mu_k, x)}{dx} - R(\mu_k, x) \cdot \frac{dR(\mu_j, x)}{dx} \right] = 0,$$

aan de intervalgrenzen $x = a$ en $x = b$ (volgt uit (2), (6) en (7)).

B) $R(\mu_j, r)$ en $R(\mu_k, r)$ zijn op $[a, b]$, orthogonaal t.o.v. de dichtheidsfunctie $g(x)$, d.w.z:

$$\int_a^b g(x) \cdot R(\mu_j, x) \cdot R(\mu_k, x) \cdot dx = \begin{cases} 0 & \text{voor } j \neq k \\ \text{een positieve constante voor } j = k \\ \text{(middels } C_2 \text{ te normeren op 1)} \end{cases}.$$

C) Alle eigenwaarden en eigenfuncties zijn reëel en positief.

D) Alle eigenwaarden μ_n , $n = 0, 1, \dots$ zijn enkelvoudig en vormen een geordende oneindige verzameling, $\mu_0 \leq \mu_1 \leq \dots \leq \mu_n \dots$ $n = 0, 1, \dots$ (voor niet periodieke randcondities geldt het $<$ teken). Afzonderlijk moet worden onderzocht of $\mu_0 = 0$ een eigenwaarde is.

{Als in de randcondities (III) en (IV), $B_1 = 0$ en $B_2 = 0$ zijn, is er altijd een eigenwaarde $\mu_0 = 0$ }.

E) Voor de eigenwaarden μ_n , $n = 0, 1, \dots$ kunnen bruikbare asymptotische benaderingsformules worden afgeleid.

F) Het gedrag van een eigenfunctie $R(\mu_k, r)$ op $[a, b]$, wordt volledig bepaald door de functies $k(r)$, $l(r)$, $g(r)$.

$R(\mu_k, r)$ is op (a, b) tweemaal continu differentieerbaar.

Is $k(r)$ of $g(r)$ nul in $r = a$ of $r = b$, dan wordt $R(\mu_k, r)$ ter plaatse singulier.

G) De eigenfuncties $R(\mu_k, r)$ $k = 0, 1, \dots$ vormen een volledig orthogonaal stelsel waarvoor de algemene ontwikkelingsstelling geldt:

Elke fatsoenlijke functie $f(r)$ op $[a, b]$ (stuksgewijze continu met bestaande eerste en tweede afgeleiden en voldoende aan de randcondities), kan worden voorgesteld door de absoluut en uniform-convergente reeks naar eigenfuncties.

$$f(r) = \sum_{k=0}^{\infty} a_k \cdot R(\mu_k, r), \quad \text{waarin } a_k = \frac{\int_a^b g(x) \cdot f(x) \cdot R(\mu_k, x) \cdot dx}{\int_a^b g(x) \cdot R(\mu_k, x)^2 \cdot dx}.$$

Convergent in de zin:

$$\lim_{N \rightarrow \infty} \left\{ \int_a^b \left[f(x) - \sum_{k=1}^N a_k \cdot R(\mu_k, x) \right]^2 dx \right\}^{1/2} = 0.$$

H) Uitbreiding van deze theorie voor de intervallen $[0, \infty]$ en $[-\infty, \infty]$ is mogelijk⁵⁾.

Is aan alle gestelde Sturm-Liouville eisen voldaan, maar zijn er geen twee onafhankelijke oplossingen in formulevorm beschikbaar, dan moet de oplossing met behulp van numerieke hulpmiddelen worden gegenereerd.

Een zeer bruikbare aanpak is dan het stelsel (2), (3), (4) met behulp van differentietechnieken te transformeren, in een N -de orde tridiagonaal matrix-eigenwaarde-probleem.

We benaderen de differentiaalvergelijking en randcondities op het interval $[a, b]$, $a \geq 0$, in N roosterpunten, met de staplengte $H = (b-a)/N$. Dit levert de functiewaarden $Y_k = R(r_k)$ op in de equidistante roosterpunten $r_k = a + k \cdot H$, $k = 0, 1, \dots, N$.

Als differentiebenaderingen voor de 1^{ste} en 2^{de} afgeleide, gebruiken we de bekende formules:

$$\frac{dY}{dr} \approx \frac{(Y_{k+1} - Y_{k-1}))}{2 \cdot H} \quad (10)$$

$$\frac{d^2Y}{dr^2} \approx \frac{(Y_{k+1} - 2 \cdot Y_k + Y_{k-1}))}{H^2}. \quad (11)$$

In tegenstelling met de boven behandelde analytische methode, vormt de zelfgeadjungeerde differentiaalvergelijking (2), niet het goede uitgangspunt. Substitutie van (10) en (11), levert weliswaar een symmetrische tridiagonaalmatrix op, maar in de roosterpunten $r_{k+1/2}$ $k = 0, 1, \dots, N$. Differentiatie van (2) en vervolgens delen door $g(r)$, levert de vorm:

$$\frac{k(r)}{g(r)} \cdot \frac{d^2R(r)}{dr^2} + \frac{1}{g(r)} \cdot \frac{dk(r)}{dr} \cdot \frac{dR(r)}{dr} + \left(\mu - \frac{l(r)}{g(r)} \right) \cdot R(r) = 0. \quad (12)$$

Voeren we de volgende hulpfuncties in,

$$F1(r) = \frac{k(r)}{g(r)}, \quad F2(r) = \frac{1}{g(r)} \cdot \frac{dk(r)}{dr} \quad \text{en} \quad F3(r) = \frac{l(r)}{g(r)},$$

dan gaat (12) over in:

$$F1(r) \cdot \frac{d^2 R(r)}{dr^2} + F2(r) \cdot \frac{dR(r)}{dr} + (\mu - F3(r)) \cdot R(r) = 0. \quad (13)$$

Substitutie van (10) en (11), in (13) leidt tot de volgende recursie-vergelijking voor het k -de roosterpunt ($k = 0, 1, \dots, N$):

$$F1_k \cdot \frac{\{Y_{k+1} + Y_{k-1} - 2 \cdot Y_k\}}{H^2} + F2_k \cdot \frac{(Y_{k+1} - Y_{k-1})}{2 \cdot H} + (\mu - F3_k) \cdot Y_k = 0 \quad (14)$$

hierin is

$$F1_k = F1(r_k), \quad F2_k = F2(r_k), \quad F3_k = F3(r_k)$$

gesteld.

Voor $k = 0$ en $k = N$ in (14), liggen de functiewaarden Y_{-1} en Y_N , buiten het interval, deze kunnen echter met behulp van de met (10) in differentievorm gebrachte randcondities (2) en (3) worden geëlimineerd. Geven we de coëfficiënten van Y_k 's in deze 3-term-recursie-relaties (14), aan met $A_{k,l}$ ($A_{k,l} = 0$ voor $|k - l| \geq 2$), dan resulteert dit in het tri-diagonale niet-symmetrische matrix-eigenwaarde-systeem:

$$|A_{k,k} + \mu A_{k,l}| = 0 \quad \text{met} \quad k, l = 0, 1, \dots, N. \quad (15)$$

De asymmetrie van (15) vormt geen probleem voor het bepalen van de eigenwaarden van een matrix met behulp van SUB EIVALI. Daar we alleen gebruik maken van de diagonaalelementen en het product van de codiagonaalelementen, wordt een symmetrisch systeem met diagonaalelementen $a_k = A_{k,k}$ en codiagonaalelementen $B_k = \sqrt{(A_{k,k+1} * A_{k+1,k})}$, met dezelfde eigenwaarden opgelost. Een Hessenberg-transformatie is dus niet nodig.

Van het gesymmetriseerde systeem, worden de eigenwaarden μ_k dus bepaald met de SUB EIVALI en de bijbehorende eigenvectoren V_k met de SUB EIVEC. De eigenvectoren Y_k van (15), worden uit de eigenvectoren V_k gevonden met behulp van de transformatie:

$$Y_k = \sum_{l=1}^N G_l \cdot V_l, \quad k = 1, 2, \dots, N. \quad (16)$$

De elementen van de vector G , volgen uit de recursierelatie:

$$G_1 = 1, \quad G_{k+1} = G_k \cdot \sqrt{(B_k / B_{k-1})}, \quad k = 2, \dots, N.$$

In deze methode is het aantal termen N , de enige parameter die bepalend is voor de nauwkeurigheid waarmee de eigenwaarden en eigenvectoren worden berekend. Voor het bepalen van een optimale N , kan eventueel de Richardson-extrapolatie-methode worden gebruikt, om te controleren hoe nauwkeurig een eigenwaarde uit b.v. de keuzes: $N = 20, 40, 80$ kan worden berekend. Kiest men N zeer groot, dan wordt het bepalen van de eigenwaarden een zeer tijdrovend proces en vergt het opslaan van de eigenvectoren veel geheugenruimte.

Een totaal andere aanpak is reeds vroeger behandeld bij het simultaan bepalen van de wortels van een polynoom met behulp van kettingbreuken. In programma KARAKTER.BAS is o.a. geïllustreerd hoe we van vier met differentieetechnieken benaderde differentiaalvergelijkingen van het type (15), de coëfficiënten van het karakteristieke polynoom kunnen genereren met de SUB KARAKTERISTIEKPOLYNOOM en hoe daarna met het programma SUB QDPROGRESIEF, simultaan alle N wortels (e.w.) worden benaderd.

Rest nog aan te geven hoe bij elke eigenwaarde μ_k , de bijbehorende eigenvector $R_k(r)$ kan worden bepaald. Dit kan worden gedaan door met een numerieke integratiemethode (b.v. Runge-Kutta), de oplossing van differentiaalvergelijking (2) of (12) op het interval $[a, b]$, als een beginwaarde-probleem te genereren. De beginconditie bepalen we met behulp van de randconditie (3). Kies een functiewaarde $R(a)$ en bereken de bijbehorende afgeleide $dR(r)/dr$ in a (of kies een afgeleide en bereken de functie waarde). De hiermee te genereren oplossing is niet genormeerd (voor het normeren zie eigenschap H). De nauwkeurigheid van de oplossing is te controleren, door de bij de eigenwaarde μ_k gevonden (eigenvector) functiewaarden $R_k(b)$ en $R'_k(b)$ te substitueren in de randconditie (4).

In het bovenstaande is de "separatie-van-variabelen"-methode alleen geïllustreerd bij het oplossen van differentiaalvergelijkingen van het parabolische type. Vanzelfsprekend kan het op analoge wijze worden toegepast voor het oplossen van differentiaalvergelijkingen van het elliptische en het hyperbolische type.

Met behulp van het voorgaande gaan we een cilindrisch symmetrisch diffusieprobleem eerst analytisch en vervolgens met de differentiemethode oplossen. De geïmplementeerde analytische oplossing kan worden gebruikt om te laten zien met welke nauwkeurigheid de zelfde differentiaalvergelijking en randcondities, met behulp van de differentiebenaderingen, kan worden opgelost. Veel praktijkproblemen van het Sturm-Liouville type zijn analytisch meestal niet of uiterst moeizaam oplosbaar. Dat de differentie-aanpak dan met behulp van de Subs EIVALI en EIVEC snel nauwkeurige resultaten kan leveren wordt geïllustreerd aan een praktijkvoorbeeld:

Het massatransport in een ultracentrifuge.

Referenties.

- 1) Ordinary Differential Equations by E.L. Ince.
- 2) Theory of Ordinary Differential Equations, by Earl A. Coddington and

Norman Livenson.

- 3) Methods of Mathematical Physics, Vol. I and II, by Courant-Hilbert.
- 4) The Theory of Partial Differential Equations, by Sigeru Mizohata.
- 5) Introduction to Spectral Theorie, by B.M. Levitan and I.S. Sargsjan.

15. Een Besselfunctie Randwaarde Probleem

Het eerder geformuleerde algemene diffusieprobleem (I),..., (V), wordt door de keuze $g(r) = r$, $k(r) = r$, $l(r) = 0$, herleid tot een in wetenschap en techniek, zeer frequent voorkomend diffusie/warmte-geleidings-probleem in cilindercoördinaten. Het na de separatie van de variabelen resulterende Sturm-Liouville eigenwaardeprobleem op het interval $a \leq r \leq b$, is de nulde orde differentiaalvergelijking van Bessel:

$$\frac{d^2 Y}{dx^2} + \frac{1}{x} \cdot \frac{dY}{dx} + \mu^2 \cdot Y = \frac{1}{x} \cdot \frac{d}{dx} \left(x \cdot \frac{dY}{dx} \right) + \mu^2 \cdot Y = 0 \quad (1)$$

met als algemene oplossing:

$$Y(x) = C_1 \cdot J_0(\mu \cdot x) + C_2 \cdot Y_0(\mu \cdot x) . \quad (2)$$

De tijdafhankelijke factor blijft als boven: $\exp(-\mu \cdot t)$.

We beginnen met het meest voorkomende geval van de holle cylinder: $a = 0$ en $g(a) = 0$, de term $C_2 \cdot Y_0(\mu \cdot x)$ in (2) wordt oneindig in $x = 0$ en dus moet $C_2 = 0$ worden gekozen. Voor $C_1 \neq 0$ en $a = 0$ is $J_0(\mu \cdot a) = 1$ en $J_1(\mu \cdot a) = 0$, er kan alleen aan de homogene randconditie (IV) worden voldaan door $A_1 \neq 0$ en $B_1 = 0$. Deze conditie noemt men "de voorwaarde voor goed gedrag".

Sustitutie van (2) in (V) levert direct de eigenwaardevergelijking:

$$C_1 \cdot \{A_2 \cdot J_0(\mu) - B_2 \cdot J_1(\mu)\} = 0 . \quad (3)$$

Is ook $A_2 = 0$ dan volgt uit (3), dat er een eigenwaarde $\mu = 0$ zal gaan optreden ($J_1(0) = 0$).

Voor elke eigenwaarde $\mu = \mu_n$, $n = 0, 1, \dots$ welke voldoet aan (2), is de bijbehorende eigenfunctie:

$$Y(x) = C_1 \cdot J_0(\mu_n \cdot x) . \quad (4)$$

De factor C_1 kan worden vastgelegd met behulp van de onder B) genoemde orthogonaliteits- en normeringsvoorwaarde:

$$1 = C_1^2 \int_0^b x \cdot J_0(\mu \cdot x)^2 \cdot dx . \quad (5)$$

Ook zonder te normeren kan een beginwaardefunctie $f(r)$, welke aan de voorwaarden genoemd onder H) voldoet, met behulp van een voorraad van N eigenfuncties, als volgt worden benaderd:

$$f(r) = \sum_{k=0}^N a_k \cdot J_0(\mu_k \cdot r) . \quad (6)$$

Vermenigvuldigen we (6), links en rechts met $r \cdot J_0(\mu_l \cdot r)$, dan volgt voor $l = k$, na integratie over r van 0 tot b (volgens de orthogonaliteits-voorwaarde B), voor de ontwikkelingscoëfficiënten:

$$a_k = \frac{\int_0^b f(r) \cdot r \cdot J_0(\mu_k \cdot r) \cdot dr}{\int_0^b r \cdot J_0(\mu_k \cdot r)^2 \cdot dr} \quad k = 0, 1, \dots . \quad (7)$$

De volledige tijdafhankelijke oplossing kan dus worden geschreven als:

$$u(r, t) = \sum_{k=0}^{\infty} a_k \cdot \exp(-\mu_k \cdot t) \cdot J_0(\mu_k \cdot r) . \quad (8)$$

Voor het geval $a > 0$ verloopt de oplossing volgens de aangegeven algemene methode. Teneinde echter het rekenwerk en de formules eenvoudiger te houden zullen we in plaats van de parameters a, b, A_1, B_1, A_2, B_2 , hiervan dimensieloze groepen vormen. De eindoplossing wordt later terug getransformeerd naar de oorspronkelijke grootheden.

Transformeren we het stelsel (I),..., (V), met behulp van de dimensieloze onafhankelijke variabelen $r = b \cdot x$ en $t = \tau / K$, dan ligt het gebruik van de dimensieloze groepen: $x_0 = a/b$, $E = b \cdot B_1 / A_1$ en $G = b \cdot B_2 / A_2$ voor de hand, mits we (voorlopig) aannemen dat A_1 en A_2 ongelijk nul zijn. De separatie van de variabelen leidt dan tot het randwaardeprobleem:

$$\frac{d^2 Y}{dx^2} + \frac{1}{x} \cdot \frac{dY}{dx} + \sigma^2 Y = 0 \quad \text{op het interval } [x_0, 1], \quad x_0 \geq 0 \quad (9)$$

$$\frac{dY}{dx} + E \cdot Y = 0 \quad \text{voor } x = x_0 \quad \text{en} \quad (10)$$

$$\frac{dY}{dx} + G \cdot Y = 0 \quad \text{voor } x = 1 \quad (11)$$

$$\text{in (9) is } (\sigma)^2 = (b \cdot \mu)^2 \text{ gesteld.} \quad (12)$$

De algemene oplossing van (9) is, met behulp van de in Appendix Besselfuncties gegeven formules, te schrijven als:

$$Y(x) = C_1 \cdot J_0(\sigma \cdot x) + C_2 \cdot Y_0(\sigma \cdot x) . \quad (13)$$

Substitutie van (5) in (2) en (3) levert met gebruikmaking van de relaties

$$J'_0(x) = -J'_1(x) \quad \text{en} \quad Y'_0(x) = -Y'_1(x) ,$$

de volgende twee homogene lineaire vergelijkingen in C_1 en C_2 :

$$\{C_1 \cdot J_1(\sigma \cdot x_0) + C_2 \cdot Y_1(\sigma \cdot x_0)\} = E \cdot \{C_1 \cdot J_0(\sigma \cdot x_0) + C_2 \cdot Y_0(\sigma \cdot x_0)\} \quad (14)$$

$$\{C_1 \cdot J_1(\sigma) + C_2 \cdot Y_1(\sigma)\} = G \cdot \{C_1 \cdot J_0(\sigma) + C_2 \cdot Y_0(\sigma)\} . \quad (15)$$

Het stelsel (14), (15) heeft alleen dan een oplossing, als de determinant van het stelsel nul is.

Dit levert de eigenwaardevergelijking:

$$\frac{\{E \cdot Y_0(\sigma \cdot x_0) - \sigma \cdot Y_1(\sigma \cdot x_0)\}}{\{E \cdot J_0(\sigma \cdot x_0) - \sigma \cdot J_1(\sigma \cdot x_0)\}} = \frac{\{G \cdot Y_0(\sigma) - \sigma \cdot Y_1(\sigma)\}}{\{G \cdot Y_0(\sigma) - \sigma \cdot Y_1(\sigma)\}} . \quad (16)$$

Voor elke eigenwaarde $\sigma = \sigma_n$, $n = 0, 1, \dots$ welke voldoet aan (16), zijn het linker en het rechterlid gelijk aan de verhouding:

$$R = C_1 / C_2 . \quad (17)$$

Met behulp hiervan noteren we de bijbehorende eigenfunctie als:

$$Y(\sigma \cdot x) = C_2 \cdot \{R \cdot J_0(\sigma \cdot x) + Y_0(\sigma \cdot x)\} . \quad (18)$$

In (18) is de factor C_2 nog vrij te kiezen. Deze leggen we vast met behulp van de normeringsvoorwaarde:

$$1 = C_2^2 \int_{x_0}^1 x \cdot \{R \cdot J_0(\sigma \cdot x) + Y_0(\sigma \cdot x)\}^2 \cdot dx . \quad (19)$$

Bij de uitwerking van deze en latere integralen maken we intensief gebruik van de volgende in Appendix Besselfuncties gegeven formules.

In de deze relaties is steeds $Z_n(x)$ de lineaire combinatie:

$$Z_n(x) = P \cdot J_n(\sigma x) + Q \cdot Y_n(\sigma x) , \quad n = 0, 1, 2, \dots \quad (20)$$

en P en Q zijn constanten. Voor $P = C_2 \cdot R$ en $Q = C_2$ is dan volgens (18)

$$Y(\sigma x) = Z_0(\sigma x) . \quad (21)$$

Na het nodige rekenwerk met gebruikmaking van de formules:

$$\begin{aligned}
Z'_n(x) &= -\frac{n}{x} \cdot Z_n(x) + Z_{n-1}(x), \quad \int x^{n+1} \cdot Z_n(x) dx = x^{n+1} \cdot Z_{n+1}(x) \\
\int x \cdot \{Z_n(\sigma \cdot x)\}^2 \cdot dx &= \left(\frac{x^2}{2} \cdot \{Z_n(\sigma \cdot x)\}^2 - Z_{n+1}(\sigma \cdot x) \cdot Z_{n-1}(\sigma \cdot x) \right) \\
Z_{-n}(x) &= (-)^n \cdot Z_n(x) \quad Y_{n-1}(x) \cdot J_n(x) - Y_n(x) \cdot J_{n-1}(x) = \frac{2}{\pi \cdot x} \\
Z_{n+1}(x) + Z_{n-1}(x) &= \frac{2 \cdot n}{x} \cdot Z_n(x) \quad \int Z_1(x) \cdot dx = -Z_0(x), \\
\int x \cdot Z_0(x) \cdot dx &= x \cdot Z_1(x)
\end{aligned}$$

vinden we hiermee voor de normeringsconstante:

$$C_2 = \frac{\pi \cdot \sigma / \sqrt{2}}{\left[\frac{(\sigma^2 + G^2) / \sigma^2}{\{G \cdot J_0(\sigma) - \sigma \cdot J_1(\sigma)\}^2} - \frac{(\sigma^2 + E^2) / \sigma^2}{\{E \cdot J_0(\sigma \cdot x_0) - \sigma \cdot J_1(\sigma \cdot x_0)\}^2} \right]^{1/2}}. \quad (22)$$

Substitutie van R (17) en C_2 (22) in (19), levert de gezochte genormeerde eigenfunctie $Y(\sigma_n \cdot x)$ in dimensieloze coördinaten.

Door eliminatie van E en G , via de terug-substitutie van $E = b \cdot B_1 / A_1$ en $G = b \cdot B_2 / A_2$, uit deze formules, kunnen we de aan $A_1, A_2 > 0$ opgelegde beperkingen tevens opheffen.

Op deze wijze vinden we voor elke $\sigma = \sigma_n > 0$, $n = 0, 1, \dots$ de volgende uitdrukkingen, die we in de computer-implementatie gaan gebruiken:

Normeringsfactor:

$$C_2 = \frac{(\pi \cdot \sigma / \sqrt{2})}{\left[\frac{(A_2 \sigma)^2 + B_2^2}{\{B_2 \cdot J_0(\sigma) - \sigma \cdot A_2 \cdot J_1(\sigma)\}^2} - \frac{(A_1 \cdot \sigma)^2 + B_1^2}{\{B_1 \cdot J_0(\sigma \cdot x_0) - \sigma \cdot A_1 \cdot J_1(\sigma \cdot x_0)\}^2} \right]^{1/2}}. \quad (23)$$

De verhouding:

$$R = C_1 / C_2 = -\{b \cdot B_2 \cdot Y_0(\sigma) - \sigma \cdot A_2 \cdot Y_1(\sigma)\} / \{b \cdot B_2 \cdot J_0(\sigma) - \sigma \cdot A_2 \cdot J_1(\sigma)\}. \quad (24)$$

De eigenwaarde-vergelijking:

$$F(\sigma) = \frac{b \cdot B_1 \cdot Y_0(\sigma \cdot x_0) - \sigma \cdot A_1 \cdot Y_1(\sigma \cdot x_0)}{b \cdot B_1 \cdot J_0(\sigma \cdot x_0) - \sigma \cdot A_1 \cdot J_1(\sigma \cdot x_0)} - \frac{b \cdot B_2 \cdot Y_0(\sigma) - \sigma \cdot A_2 \cdot Y_1(\sigma)}{b \cdot B_2 \cdot J_0(\sigma) - \sigma \cdot A_2 \cdot J_1(\sigma)}. \quad (25)$$

De genormeerde eigenfunctie:

$$Y(\sigma x) = \frac{(\pi \cdot \sigma / \sqrt{2})}{\left[\frac{(A_2 \sigma)^2 + B_2^2}{\{B_2 \cdot J_0(\sigma) - \sigma \cdot A_2 \cdot J_1(\sigma)\}^2} - \frac{(A_1 \cdot \sigma)^2 + B_1^2}{\{B_1 \cdot J_0(\sigma \cdot x_0) - \sigma \cdot A_1 \cdot J_1(\sigma \cdot x_0)\}^2} \right]^{1/2}} \times \left[- \frac{b \cdot B_2 \cdot Y_0(\sigma) - \sigma \cdot A_2 \cdot Y_1(\sigma)}{b \cdot B_2 \cdot J_0(\sigma) - \sigma \cdot A_2 \cdot J_1(\sigma)} \cdot J_0(\sigma x) + Y_0(\sigma x) \right]. \quad (26)$$

Bij elk eigenwaardeprobleem moet afzonderlijk worden onderzocht, of er een eigenwaarde $\sigma = 0$ tot het stelsel behoort. // De procedure is als volgt: we lossen vergelijking (9) op voor $\sigma = 0$, twee maal integratie van $1/x d/dx(x \cdot dY/dx) = 0$ levert:

$$\frac{dY}{dx} = \frac{R1}{x} \quad \text{en} \quad Y = R1 \cdot \log(x) + R2. \quad (27)$$

met R_1 en R_2 integratieconstanten. Met behulp van de oplossing (27) verifiëren we of aan de opgelegde randcondities kan worden voldaan. Als dat het geval is, dan is er een eigenwaarde $\sigma = 0$ en is de bijbehorende eigenfunctie een constante.

Met behulp van een verzameling eigenfuncties (26) kunnen op de onder H aangegeven wijze, uit de gegeven beginverdeling $f(r)$ weer de ontwikkelingscoëfficiënten worden bepaald met de formule:

$$a_n = \int_{x_0}^1 f(x) \cdot x \cdot Y(\sigma_n \cdot x) \cdot dx \quad n = 0, 1, \dots, N. \quad (28)$$

De analytische uitwerking van deze integralen is slechts mogelijk voor een beperkte klasse van functies $f(x)$ en leidt zelfs voor de meeste eenvoudige functies reeds tot zeer omslachtige formules.

Dit illustreren we aan het volgende voorbeeld dat we steeds in de computerimplementaties zullen gaan gebruiken:

$$f(x) = p \cdot \log(x) + q \cdot x^2 + s \quad (29)$$

waarin p , q en s constanten zijn.

De uitwerking van:

$$a_n = \int_{x_0}^1 \{p \cdot \log(x) + q \cdot x^2 + s\} \cdot x \cdot Z_0(\sigma \cdot x) \cdot dx \quad (30)$$

verloopt als volgt:

Gebruik de transformatie $y = \sigma \cdot x$ en pas partiële integratie toe. Door gebruik te maken van de Besselrelatie:

$$\int x_{n+1}.Z_n(x)dx = x^{n+1}.Z_{n+1}(x) \quad \text{voor } n = 0 ,$$

resulteert dit in:

$$\begin{aligned} a_n &= 1/\sigma^2 \int_{x_0.\sigma}^{\sigma} \{p.\log(y/\sigma) + (q.(y/\sigma)^2 + r)\}.d(y.Z_1(y)) = \\ &= 1/\sigma^2.\{p.\log(y/\sigma) + (q.(y/\sigma)^2 + s)\}.(y.Z_1(y))\Big|_{x_0.\sigma}^{\sigma} + \\ &\quad - \int_{x_0.\sigma}^{\sigma} y.Z_1(y).d\{p.\log(y/\sigma) + (q.(y/\sigma)^2 + s)\} . \end{aligned}$$

Voor de resterende integraal volgt op analoge wijze:

$$\begin{aligned} &-p \int Z_1(y).dy - 2.q.(1/\sigma)^2. \int y^2.Z_1(y).dy = \\ &= -[p.Z_0(y) + 2.q.(1/\sigma)^2.y^2.Z^2(y)] \\ a_n &= \{p.\log(y/\sigma) + (q.(y/\sigma)^2 + s)\}.(y.Z_1(y))/\sigma^2 + \\ &\quad + \{-p.Z_0(y) + 2.q.(1/\sigma)^2.y^2.Z_2(y)\}/\sigma^2\Big|_{x_0.\sigma}^{\sigma} . \end{aligned}$$

Substitutie van $Z_2(y) = -Z_0(y) + Z_1(y).2/y$ en uitwerking levert:

$$\begin{aligned} \sigma^2.a_n &= \{q + s - 4/\sigma^2\}.\sigma.Z_1(\sigma) + \{p + 2.q\}.Z_0(\sigma) + \\ &\quad - \{p.\log(x_0) + q.x_0^2 + s - 4.q/\sigma^2\}.x_0.\sigma.Z_1(x_0.\sigma) \} + \\ &\quad + \{p + 2.q.x_0^2\}.Z_0(x_0.\sigma) \} . \end{aligned} \quad (31)$$

Als we in (31) de uitdrukkingen $Z_0(x)$ en $Z_1(x)$ vervangen door de lineaire combinaties (20) en (21), dan hebben we tenslotte de geschikte vorm beschikbaar voor een implementatie met behulp van de Besselsubroutines voor $J_0(x)$, $Y_0(x)$, $J_1(x)$ en $Y_1(x)$.

Hiermede is voldoende aangetoond dat het meestal een onbegonnen zaak is om voor meer gecompliceerde functies $f(r)$, de ontwikkelingscoëfficiënten analytisch uit te werken. Een minder tijdrovende aanpak is de integrand in (28) te tabuleren en vervolgens en numerieke integratiemethode te gebruiken.

In de formules voor de eigenwaarde-vergelijking, de eigenfunctie en de ontwikkelingscoëfficiënten valt op, dat steeds voor een zelfde argumentwaarde x , zowel $J_i(x)$ als $Y_i(x)$ $i = 1, 2$ moeten worden bepaald. Omdat dit bij veel cilindrisymmetrische toepassingen steeds het geval is, werden de efficiënte

combinatie-sub: J0Y0 en J1Y1 ontwikkeld. De combinatie-sub hebben naast de vele voordelen één nadeel. Voor de argumentwaarde $x = 0$ wordt $Y_i(x)$ ($i = 0, 1$) singulier en dus zijn de sub voor $x = 0$ niet te gebruiken.

De formules (22),..., (31), zijn tezamen met de Besselfuncties $J_0(x)$, $Y_0(x)$, $J_1(x)$ en $Y_1(x)$, gebruikt bij het bepalen van:

de eigenwaarden van (25) met de intervaldelingsmethode,
de eigenvector volgens (26),
de ontwikkelingscoëfficiënt volgens (31),
de tijdafhankelijke oplossing,
in het op de schijf aanwezige programma ANALYBES.BAS.

De volgende parameters moeten worden ingevoerd.

Het interval $[a, b]$: a, b ($a > 0$).

Coëfficiënten A_1, B_1 , in de randconditie bij $r = a$.

Coëfficiënten A_2, B_2 , in de randconditie bij $r = b$.

Gewenste nauwkeurigheid Epsilon: eps .

Het gewenste aantal te bepalen eigenwaarden en eigenvectoren: $K2$.

Het aantal roosterpunten in de eigenfunctie bepaling op $[a, b]$: $N2$.

Na een test op het mogelijk optreden van een eigenwaarde $\mu_0 = 0$, wordt de bijbehorende genormeerde (constante) eigenvector gegenereerd.

Een geschatte intervalondergrens $X0$ ($X0 < 2.4$).

Met als startwaarde $XB = X0 + 1$, wordt met een iteratieproces de interval bovengrens XB ($SNG(F(X0)) \neq SGN(F(XB))$) bepaald, waarbinnen de eerstvolgende eigenwaarde XM , ($X0 < XM < XB$) met de nauwkeurigheid eps moet worden berekend met behulp van interval-halvering.

Bij de eigenwaarde XM , wordt overeenkomstig (26) de bijbehorende eigenvector in de $N2$ roosterpunten getabuleerd.

Met behulp van formule (31) wordt voor de parameters $P = -(x_0)^2/2$, $Q = 1/4$ en $S = -1/4$ in (30), de ontwikkelingscoëfficiënt berekend.

Met de nieuw berekende ondergrens $X0 = XM + 1000 * eps$, wordt dit proces telkens herhaald tot alle $K2$ eigenwaarden zijn gevonden.

Voor de beginverdeling (28) wordt met behulp van de $K2$ eigenvectoren, de benadering $G_{K2}(r)$ en de gemaakte fout $Fout(r) = (G(r) - G_{K2}(r))$ berekend. Let op, dat de te gebruiken randcondities voor $r = a$ of $r = b$ passen bij de geïmplementeerde beginverdeling $G(r)$, anders wordt in die punten een slechte benadering gevonden.

De tijdafhankelijke oplossing kan worden berekend in tijdstappen TDELTA tot aan de eindtijd $t = TMAX$.

Bij een groot aantal eigenwaarden $K2$ en veel roosterpunten $N2$ op $[x_0, 1]$ kan dit met een trage computer zonder rekenprocessor, door de vele aanroepen van de Bessel-sub, een tijdrovend proces zijn.

Teneinde met dit algemene programma te kunnen experimenteren, zijn voor

drie verschillende parametersets: a, b, A_1, B_1, A_2, B_2 de oplossingen ook afzonderlijk analytisch uitgewerkt.

Geval A):

$A_1 = 0$ en $B_1 = 1, A_2 = 0$ en $B_2 = 1$, respectievelijk $Y(x_0) = 0, Y(1) = 0$. Aan (27) $Y(x) = R1 \cdot \log(x) + R2$ en $X0 > 0$, kan dan alleen worden voldaan door $R1 = 0$ en $R2 = 0$, de triviale oplossing $Y(x) = 0$, dus is $\sigma = 0$ geen eigenwaarde.

De μ_k 's, $k = 1, 2, \dots$ zijn nulpunten van de eigenwaardevergelijking:

$$F(\mu_k) = \{Y_0(\mu_k x_0)J_0(\mu_k) - J_0(\mu_k x_0)Y_0(\mu_k)\} = 0 .$$

De genormeerde eigenfunctie heeft de vorm:

$$Y(\mu_k x) = \frac{\pi \mu_k}{\sqrt{2}} \left\{ 1 - \frac{J_0^2(x_0 \cdot \mu_k)}{J_0^2(\mu_k)} \right\} \cdot \{Y_0(\mu_k x)J_0(\mu_k) - J_0(\mu_k x)Y_0(\mu_k)\}$$

voor $x_0 = .3$ zijn de eerste 4 eigenwaarden:

$$\mu_1 = 4.41 \quad \mu_2 = 8.93 \quad \mu_3 = 13.43 \quad \mu_4 = 17.92$$

en voor $x_0 = .2$ $\mu_1 = 3.8159$ $\mu_2 = 7.78553$; $\mu_3 = 11.7321$.

Geval B):

$A_1 = 1, B_1 = 0$ en $A_2 = 0, B_2 = 1$ d.w.z. $Y'(x_0) = 0$ en $Y(1) = 0$.

Uit (27) volgt weer $R1$ en $R2$ beide nul, dus geen eigenwaarde nul.

De overige μ_k 's, $k = 1, 2, \dots$ zijn nulpunten van:

$$\{Y_0(\mu_k) \cdot J_1(\mu_k x_0) - J_0(\mu_k) \cdot Y_1(\mu_k x_0)\} = 0 .$$

Voor $X0 = .3$ zijn de eerste 4 eigenwaarden:

$$2.78 \quad 6.99 \quad 11.38 \quad 15.83 .$$

De genormeerde eigenfunctie:

$$Y(\mu_k x) = \frac{\pi \cdot \mu_k}{\sqrt{2} \left\{ 1 - (J_0(\mu_k))^2 / (J_1(x_0 \cdot \mu_k))^2 \right\}} \cdot \{Y_0(\mu_k \cdot x_0)J_0(\mu_k \cdot x) - J_0(\mu_k \cdot x_0)Y_0(\mu_k \cdot x)\} .$$

De functie $g(x) = -(x_0^2/2) \cdot \log(x) + (x^2 - 1)/4$ is gebruikt bij het bepalen van de ontwikkelingscoëfficiënten:

$$a_n = \frac{2}{\mu_n^3} \cdot \sqrt{\left[2 \cdot \left\{ 1 - \frac{J_0(\mu_n)^2}{J_1(x_0 \cdot \mu_n)^2} \right\} \right]} .$$

Geval C):

$$A_1 = 1, B_1 = 0 \text{ en } A_2 = 1, B_2 = 0.$$

Dit correspondeert met: $Y'(x_0) = 0$ voor $x = x_0$ en $Y'(1) = 0$ voor $x = 1$.

Onderzoek weer of $\mu = 0$ een eigenwaarde is.

Uit (27) $Y = R_1 \cdot \log(x) + R_2$ en de randcondities $Y'(x_0) = 0$ en $Y'(1) = 0$ volgt alleen $R_1 = 0$ en dus is er een eigenfunctie $Y(x) = R_2$.

Uit de normeringsvoorwaarde (20) ($Z_0(\sigma \cdot x) = 1$) volgt $R_2 = \sqrt{[2/(1-x_0^2)]}$ en dus is bij $\mu = 0$, de genormeerde eigenfunctie $Y(x) = \sqrt{[2/(1-x_0^2)]}$.

De overige eigenwaarden μ_k $k = 1, 2, \dots$ volgen uit:

$$F(\mu) = \{Y_0(\mu_k) \cdot J_1(\mu_k x_0) - J_0(\mu_k) \cdot Y_1(\mu_k x_0)\} = 0$$

voor $x_0 = .35$ zijn de opvolgende 6 eigenwaarden:

$$0.00 \quad 5.01 \quad 9.77 \quad 14.57 \quad 19.38 \quad 24.20 .$$

De boven afgeleide algemene analytische oplossing is geïmplementeerd in het programma ANALYBES.BAS. Het kan worden gebruikt om van alle bovenvermelde speciale gevallen de numerieke oplossing exact te tabuleren. Deze oplossingen kunnen later worden gebruikt om de nauwkeurigheid van het zelfde randwaardeprobleem, opgelost met een differentieaanpak te kunnen beoordelen. Uit het bovenstaande blijkt, dat het construeren van een analytische oplossing een tijdrovend moeizaam proces is. We zullen demonstreren dat we met behulp van benaderende differentiemethoden gemakkelijk met de reeds beschikbare matrixeigenwaardehulpmiddelen snel zeer betrouwbare oplossingen kunnen bepalen.

16. Discretisatie van de Differentiaalvergelijking en de Randcondities

Als voorbeeld van een discretisatie aanpak nemen we weer het randwaarde probleem:

$$\frac{d^2 Y}{dx^2} + \frac{1}{x} \frac{dY}{dx} + (b \cdot \mu) 2 \cdot Y = 0 \quad (1)$$

met voor $x = x_0$:

$$A_1 \cdot \frac{dY}{dx} + b \cdot B_1 \cdot Y = 0 \quad (2)$$

en voor $x = 1$:

$$A_2 \frac{dY}{dx} + b \cdot B_2 Y = 0 \quad \text{op } [x_0 = a/b, 1] . \quad (2a)$$

Op het interval: $[x_0, 1]$ $x_0 > (=) 0$, kiezen we $N + 1$ roosterpunten, met een staplengte $H = (1 - x_0)/N$.

De volgende notatie wordt gebruikt:

$$Y(x_k) = Y_k \quad \text{voor } x_k = x_0 + k \cdot H, \quad k = 0, 1, \dots, N .$$

Benaderen we (1) met behulp van de differentieapproximaties:

$$\frac{dY}{dx} \approx \frac{(Y_{k+1} - Y_{k-1}))}{2 \cdot H} \quad (3)$$

$$\frac{d^2 Y}{dx^2} \approx \frac{(Y_{k+1} - 2 \cdot Y_k + Y_{k-1}))}{H^2} \quad (4)$$

dan volgt voor het k^{de} roosterpunt, de recursievergelijking:

$$Y_{k+1} + Y_{k-1} - 2 \cdot Y_k + \frac{(Y_{k+1} - Y_{k-1}) \cdot H}{2 \cdot (x_0 + k \cdot H)} + (b \cdot \mu \cdot H)^2 \cdot Y_k = 0 . \quad (5)$$

Nemen we even aan dat voor $k = 0$ en $k = N$ de relaties zijn aangepast met behulp van de randcondities en noemen we algemeen in (5) de coëfficiënten van Y_k , $A_{k,l}$ en $(b \cdot \mu \cdot H)^2 = \sigma$, dan laat (5) zich noteren als de tri-diagonale niet-symmetrische matrix:

$$|A_{k,k} - \sigma A_{k,l}| = 0 \quad \text{met } A_{k,l} = 0 \quad \text{voor } |k - l| \geq 2, \quad k, l = 0, 1, \dots, N . \quad (6)$$

Worden de elementen in deze drie diagonalen opgeslagen in drie arrays: De subdiagonaalelementen in $V()$, de diagonaalelementen in $D()$ en de superdiagonaalelementen in $U()$, dan kunnen we dit stelsel oplossen met het eigenwaarde programma EIVAL.BAS, mits we hieruit eerst de gesymmetriseerde

eigenwaardematrix, met de diagonaalelementen $a_k = A_{k,k}$ en subdiagonaalelementen $B_k = \sqrt{(A_{k,k+1} * A_{k+1,k})}$ vormen.

De corresponderende grootheden a_k en B_k worden opgeslagen in de arrays DIAG() en CD(). Ook wordt direct de transformatievector $G()$ gevormd, teneinde de met SUB EIVEC bepaalde eigenvectoren te kunnen terugtransformeren naar de eigenvectoren behorend bij het systeem (5).

```
G(1)=1:DIAG(1)=D(0):CD(1)=0
FOR K=2 TO N: DIA(K) = D(K - 1):
  CD(K) =-SQR(V(K - 1) * U(K - 2))
  G(K)=-SQR(V(K - 1)/U(K - 2)): NEXT K
```

Let op!: In SUB EIVAL, loopt de index van $k = 1$ naar $N + 1$ (stel $N = N + 1$).

De discretisatie van de randcondities verloopt als volgt:

We gaan uit van de randcondities (2) en (2a) en beschouwen eerst weer het meest algemene geval A_1, B_1 en $A_2, B_2 \neq 0$.

Substitutie van de benadering (3a) voor de roosterpunten $k = 0$ en $k = N$ in de randcondities (2) en (2a) geeft de relaties:

$$\frac{(Y_1 - Y_{-1})}{2.H} + b. \frac{B_1}{A_1} . Y_0 = 0 \quad \text{en} \quad \frac{(Y_{N+1} - Y_{N-1})}{2.H} + b. \frac{B_2}{A_2} . Y_N = 0 . \quad (7)$$

Lossen we hieruit: Y_{N+1} en Y_{-1} op, dan vinden we dat:

$$Y_{-1} = Y_1 + 2.H.b.B_1/A_1.Y_0 \quad \text{en} \quad Y_{N+1} = Y_{N-1} - 2.H.b.B_2/A_2.Y_N . \quad (8)$$

De substitutie van $k = 0$ en $k = N$ in (5) levert anderzijds dat:

$$Y_1 + Y_{-1} - 2.Y_0 + \frac{(Y_1 - Y_0).H}{(2.x_0)} + (b.\mu.H)^2 = 0 \quad (9)$$

$$Y_{N+1} + Y_{N-1} - 2.Y_N + (Y_{N+1} - Y_{N-1}).H/2 + (b.\mu.H)^2.Y_N = 0 . \quad (10)$$

Eliminatie van Y_{N+1} en Y_{-1} (met behulp van (8)) resulteert in de relaties:

$$[2 - (2 - H/x_0).H.b.B_1/A_1 - \sigma].Y_0 - 2.Y_1 = 0 \quad (11)$$

$$-2Y_{N-1} + [2 + (2 + H).H.b.B_2/A_2 - \sigma]Y_N = 0. \quad (12)$$

Het differentieanalagon van (1), (2), (2a) is dus te implementeren als:

```
U(0) = -2: D(0) = 2 - (2 - H/x0) * H * b * B1/A1
FOR K = 1 TO N - 1: V(K) = -1 + H/(2 * (X0 + H * K)): D(K) = 2:
  U(K) = -1 - H / (2 * (X0 + H * K))
NEXT K: V(N) = -2: D(N) = 2 + (2 + H / x0) * H * b * B2/A2
```


Wanneer in de randcondities één of meer van de constanten A_1 , B_1 , A_2 , B_2 exact nul zijn, dan moeten er speciale voorzorgen worden genomen teneinde de Subs EIVALI en EIVEC te kunnen gebruiken. De differentierelatie in het betreffende roosterpunt moet dan (soms) worden weggelaten.

Als aan het eind van het interval $Y(1) = 0$, d.w.z. $A_2 = 0$ en $B_2 = 1$ resp. $V(N) = 0$: $D(N) = 0$: $U(N) = 0$, dan stellen we eenvoudig $N = N - 1$.

Als aan het begin van het interval $Y(x_0) = 0$, d.w.z. $A_1 = 0$, $B_1 = 1$, dan correspondeert dit met: $V(0) = 0$: $D(0) = 0$: $U(0) = 0$.

Weglaten van de differentievergelijking in roosterpunt 0, komt er op neer, dat we alle indices één plaats moeten terug schuiven, roosterpunt 1 wordt roosterpunt 0 etc. In het geval dat dit probleem aan beide grenzen gaat optreden, dus als $A_1 = 0$, $B_1 = 1$ en $A_2 = 0$ en $B_2 = 1$, dan moeten we beide procedures toepassen. De consequentie is dus, dat men er rekening mee moet houden, dat het interval één of twee staplengtes inkrimpt.

De eigenwaarden worden berekend met een kleinere nauwkeurigheid. Zeer lastig wordt het als de gevonden oplossing bij een verdere verwerking moet worden aangepast aan het oorspronkelijke interval.

Zeer goede resultaten kunnen echter ook worden verkregen als men de betreffende parameter(s) niet exact nul, maar (een) zeer kleine waarde(s) (b.v.: $1.E-6$) geeft. Het voordeel is een ongewijzigde intervalgrootte. In het programma BESSELDF.BAS (op de schijf), wordt getest op het mogelijk exact nul zijn van de parameters en volgt er dienovereenkomstig een aanpassing.

Ook de mogelijkheid, een gegeven beginverdeling te ontwikkelen met behulp van de gegenereerde voorraad eigenfuncties is weer geïmplementeerd. Dezelfde functie als in ANALYTIS.BAS is gebruikt, omdat een logaritme in deze functies optreedt, mag niet een interval $[a, b]$ met $a = 0$ worden gebruikt (Hierop wordt niet getest!).

Het blijkt dat de oplossingen (onder overeenkomstige randcondities) gegenereerd met ANALYTIS.BAS en voor $N = 40$ met BESSELDF.BAS reeds een overeenstemming in zeer vele decimalen bestaat.

We vatten onze conclusies als volgt samen:

- a) De differentiemethode is ook toepasbaar in die gevallen waar een volledige analytische aanpak, niet mogelijk is.
- b) Het na de separatie van variabelen verkregen Sturm- Liouville randwaardeprobleem, kan zeer eenvoudig worden benaderd door een standaard N -de orde matrix-eigenwaarde-systeem.
- c) Bij een voldoende groot aantal roosterpunten is de nauwkeurigheid zeer goed.

We passen deze methode nogmaals toe op een parabolische partiële differentiaalvergelijking, welke het stoftransport in een ultracentrifuge cel¹⁾ beschrijft:

$$\frac{1}{r} \cdot \frac{\partial}{\partial r} \left\{ \left(D \cdot \frac{\partial c}{\partial r} - \Omega^2 \cdot r \cdot S \cdot c \right) \cdot r \right\} = \frac{\partial c}{\partial \tau} \quad \text{op } 0 < r_1 < r < r_2 \text{ en } \tau \geq 0 . \quad (13)$$

met de randcondities:

$$\Omega^2 \cdot r \cdot S \cdot c - D \cdot \frac{\partial c}{\partial r} = 0 \quad \text{voor } r = r_1, r_2 \quad (14)$$

en een uniforme beginconcentratie:

$$c(r, \tau) = c_0 \quad \text{voor } \tau = 0 . \quad (15)$$

De hoeksnelheid Ω , de sedimentatiesnelheid S en de diffusiecoëfficiënt D beschouwen we hier als constanten²⁾.

Toepassen van de separatie van variabelenaanpak: $c(r, \tau) = R(r) \cdot T(\tau)$ en de separatieconstante μ , levert de tijdafhankelijke vergelijking:

$$\frac{dT}{d\tau} + \mu T = 0 , \quad \text{met als oplossing: } T = G \cdot \exp(-\mu \cdot \tau) \quad (17)$$

en de plaatsafhankelijke differentiaalvergelijking:

$$\frac{1}{r} \cdot \frac{d}{dr} \left\{ \left(D \cdot \frac{dR}{dr} - \Omega^2 \cdot r \cdot S \cdot c \right) \right\} \cdot r + \mu R = 0 . \quad (18)$$

Met behulp van de nieuwe parameters:

$$\delta = \frac{\Omega^2 \cdot S}{D} , \quad \sigma = \frac{(\mu - 2 \cdot \Omega^2 \cdot S)}{D} \quad (19)$$

gaat (18) over in:

$$\frac{d^2 R}{dr^2} + \left(\frac{1}{r} - \delta \cdot r \right) \cdot \frac{dR}{dr} + \sigma \cdot R = 0 . \quad (20)$$

In (20) treden nog steeds twee parameters δ en σ op. Door invoeren van de nieuwe onafhankelijke variabele:

$$z = \frac{\delta \cdot r^2}{2} = \frac{r^2 \cdot \Omega^2 \cdot S}{2 \cdot D} \quad (21)$$

de afhankelijke variabele: $R = U(z)$ en een nieuwe parameter:

$$\alpha = 1 - \frac{\mu}{2 \cdot \Omega^2 \cdot S} = - \frac{\sigma}{2 \cdot \delta} , \quad (22)$$

gaat (20) tenslotte over in een confluent-hypergeometrische differentiaalvergelijking van Kummer:

$$\frac{d^2U}{dz^2} + \left(\frac{1}{z} - 1\right) \cdot \frac{dU}{dz} - \frac{\alpha}{z} \cdot U = 0 . \quad (23)$$

De overeenkomstige zelfgeadjungeerde vorm van (23) is:

$$\frac{dU}{dz} \left\{ z \cdot \exp(-z) \cdot \frac{dU}{dz} \right\} - \exp(-z) \cdot \alpha \cdot U = 0 . \quad (24)$$

Definiëren we nog:

$$a = \delta \cdot r_1^2 / 2 \quad \text{en} \quad b = \delta \cdot r_2^2 / 2 ,$$

dan gaan tengevolge van de transformatie (21), de randcondities over in:

$$U(z) - \frac{dU}{dz} = 0 \quad \text{voor } z = a, b . \quad (25)$$

Met behulp van de eigenwaarden α_k en de eigenfuncties $U(\alpha_k, z)$ luidt de oplossing van het complete stelsel in de variabelen z en $t = 2 \cdot \Omega^2 \cdot S \cdot \tau$:

$$c(z, t) = \sum_{k=0}^{\infty} a_k \cdot U(\alpha_k, z) \cdot \exp((\alpha_k - 1) \cdot t) \quad (26)$$

waarin

$$a_k = c_0 \cdot \frac{\int_a^b \exp(-x) \cdot U(\alpha_k, x) \cdot dx}{\int_a^b \exp(-x) \cdot U(\alpha_k, x)^2 \cdot dx} . \quad (27)$$

Volgens de Sturm-Liouville eigenschappen, zijn de eigenwaarden μ_k , van het oorspronkelijke systeem alle negatief. Als gevolg van de schuifoperaties in (22), blijkt dat er één positieve eigenwaarde α_0 is ontstaan.

Met behulp van analytische hulpmiddelen is te bewijzen dat $\alpha_0 = 1$ en $U(\alpha_0, z) = \exp(z)$, zodat voor $k = 0$ de eerste term in (26) onafhankelijk van t is (evenwichts-concentratie-verdeling $c(z, \infty)$).

Omdat Archibald¹⁾, slechts 1 extra eigenwaarde $\alpha_1 = -38.7$ had berekend, werd voorafgaand aan de numerieke oplossingsmethode, met behulp van Langer-asymptotiek⁶⁾, een algemene formule voor de eigenwaarden afgeleid. Voor de parameterwaarden $a = 5.0$ en $b = 6.2$, kunnen de eigenwaarden met de volgende formule zeer nauwkeurig worden berekend.

$$\alpha_k = -38.271325 \cdot \left(k + \frac{0.0056307}{k} \right) + O(k^{-2}) \quad k = 1, 2, \dots \quad (28)$$

Deze afleiding en ook de verdere analytische behandeling kan hier niet verder worden behandeld.

Het randwaarde probleem (23), (24), (25) laat zich met behulp van de differentieformules (3) en (4), zonder complicaties weer benaderen als een standaard matrixeigenwaarde-probleem.

De resulterende differentierelatie SUB ULTRADIFF. is vroeger al gebruikt in het programma KARAKTER.BAS, teneinde van het daarmee genereerde karakteristieke polynoom, simultaan de nulpunten te kunnen berekenen.

Na het genereren van een eigenwaarde α_k en bijbehorende eigenfunctie

$U(\alpha_k, z)$ met de bekende Subs EIVAL en EIVC, wordt de coëfficiënt a_k overeenkomstig (27) berekend met de SUB ONTWIKKELINGCOEF. Met behulp van de a_k 's en $U(\alpha_k, z)$'s, wordt de uniforme beginverdeling benaderd en ook de daarbij optredende fout bepaald. Het voorgaande is uitgewerkt in het op de schijf aanwezige programma ULTRA.BAS.

De tijdafhankelijke eindoplossing $c(z, t)$ (is niet weer opgenomen) laat zich echter met behulp van (26), analoog als in BESSELD.F.BAS uitwerken.

In het volgende voorbeeld wordt kort toegelicht hoe het discretisatie-proces verloopt als de differentiaalvergelijking in de randpunten singulier wordt. Bij het oplossen van sommige stationaire warmtetransport-problemen, o.a. gedrongen laminaire pijpstroming (Poiseuille wet)^{3,4}, moet als deelprobleem het volgende Sturm-Liouville stelsel worden opgelost.

$$\frac{1}{r} \cdot \frac{d}{dr} \left(r \cdot \frac{du}{dr} \right) + \mu \cdot (1 - r^2) \cdot u = 0 \quad \text{op } [0, 1] \quad (29)$$

met de randcondities

$$\frac{du}{dr} = 0 \quad \text{voor } r = 0 \quad (30)$$

en

$$G \cdot \frac{du}{dr} = u \quad \text{voor } r = 1 \quad (G \text{ een grote constante}). \quad (31)$$

In de uitgewerkte vorm van (29) treedt de term $\frac{1}{r} \cdot \frac{du}{dr}$ op.

Voor $r = 0$ is volgens (30), deze term echter onbepaald $0/0$, dus moet de limiet:

$$\lim_{r \rightarrow 0} \frac{\frac{du}{dr}}{r} = \frac{d^2u(0)}{dr^2}$$

worden bepaald (b.v. l'Hôpital toepassen).

Bij het afleiden van de differentierelatie in het eerste roosterpunt ($r = 0$) moet dus inplaats van (29), de daaruit door limietovergang gevonden differentiaalvergelijking:

$$2. \frac{d^2 u}{dr^2} + \mu.(1 - r^2).u = 0 \quad (31)$$

en randconditie (30) worden gebruikt. Met behulp van de differentierelaties (3) en (4) volgt dan direct het verband:

$$(4 - H^2.\mu).Y_0 - 4.Y_1 = 0 . \quad (32)$$

Analoog volgt voor de roosterpunten $k = 1$ tot $N - 1$ uit (30)

$$\frac{1}{1 - (k.H)^2} \cdot \left[\left(1 - \frac{1}{2.k}\right).Y_{k-1} - \left(1 + \frac{1}{2.k}\right).Y_{k+1} + (2 - \mu).Y_k \right] = 0 . \quad (33)$$

Ook voor $r = 1$, zijn er moeilijkheden, want in (29) wordt de term:

$$\mu.(1 - r^2).u = 0 \text{ en dus is } \frac{d^2 u}{dr^2} + \frac{1}{r} \cdot \frac{du}{dr} = 0 \text{ (onafhankelijk van } \mu). \quad (34)$$

Dit probleem kunnen we weer oplossen door weglaten van het N - de roosterpunt, mits uit de relatie voor het $(N - 1)$ de roosterpunt, u_N met behulp van de randconditie (31) wordt geëlimineerd. Voor het verkrijgen van een zo groot mogelijke nauwkeurigheid (mede omdat G groot is) verdient het aanbeveling om de afgeleide in de randconditie niet te benaderen met (3) maar met de terugstap-differentierelatie:

$$\frac{du(1)}{dr} = \frac{3.u_N - 4.u_{N-1} + u_{N-2}}{2.H} . \quad (35)$$

Substitutie in de randconditie (31) leidt tot:

$$U_N = \frac{4}{3 + 2.H/G} . U_{N-1} - \frac{1}{3 + 2.H/G} . U_{N-2} . \quad (36)$$

Met behulp van (36) en (33) voor $k = N - 1$, volgt dan de relatie:

$$\frac{1}{1 - ((N - 1).H)^2} \cdot \left[\left(-1 + \frac{1}{2.N - 2} \left(1 + \frac{1}{3 + 2.H/G} \right) \right) . Y_{N-2} + \left\{ 2 - \left(1 + \frac{1}{2.N - 2} \right) \cdot \frac{4}{3 + 2.H/G} \right\} . Y_{N-1} \right] - \mu.Y_k = 0 . \quad (37)$$

De differentierelaties (32), (33), (37) zijn in het eerder genoemde programma KARAKTER.BAS, als subroutine geïmplementeerd en kunnen dus worden gebruikt, in de aanpak met behulp van de subs EIVAL en EIVEC.

Opmerking: Het uitwerken van de limietovergang voor $r \rightarrow 1$ van (29) geeft in dit geval geen kant en klare oplossing, omdat:

$$\lim_{r \rightarrow 1} \frac{\frac{d^2 u}{dr^2} + \frac{1}{r} \cdot \frac{du}{dr}}{1 - r^2} = - \lim_{r \rightarrow 1} \mu \cdot u = -\mu \cdot u(1) . \quad (33)$$

met l'Hôpital, voor het linkerlid leidt tot:

$$\lim_{r \rightarrow 1} \frac{\frac{d^3 u}{dr^3} - \frac{1}{r^2} \cdot \frac{du}{dr} + \frac{1}{r} \cdot \frac{d^2 u}{dr^2}}{2 \cdot r} = \frac{-1}{2} \cdot \frac{d^3 u(1)}{dr^3} + \frac{du(1)}{dr} + \frac{d^2 u(1)}{dr^2} . \quad (34)$$

De differentierelatie voor het N -de roosterpunt zou dus moeten worden bepaald uit een derde-orde differentiaalvergelijking:

$$\frac{-1}{2} \cdot \frac{d^3 u(1)}{dr^3} + \frac{du(1)}{dr} + \frac{d^2 u(1)}{dr^2} + \mu \cdot u(1) = 0 .$$

De bruikbaarheid hangt dus af van het beschikbaar zijn van een 2-de orde randconditie bij $r = 1$, wat in 't algemeen niet het geval zal zijn.

Een analytische oplossing van (29), voor de randconditie $u = 0$ bij $r = 1$, met behulp van confluent hypergeometrische functies, is behandeld door Lauwerier⁵⁾.

Referenties.

- 1) Archibald, W.J., Ann. N.Y. Acad. Sci., 43, 211 (1942).
- 2) Weiss, G.H., Nature, May 23, 1964, Vol. 202, pp. 792- 793.
- 3) Brinkman, H.C., Appl. Sci. Res., Vol. A2.
- 4) Rietema, K., Prisma-Technica 60.
- 5) Lauwerier, H.A., Appl. Sci. Res., Vol. A2.
- 6) Langer, R.E., Trans. Amer. Math. Soc. 33 (23-64); 34 (447-480); 36(90-106); 37 (397-416); 67 (461-490).

17. Een Potentiaal-Probleem

We berekenen de potentiaal in een aan alle zijden geaarde cilindrische tank, waarin een ladingsverdeling Φ aanwezig is. Het verband tussen de potentiaal U en de ladingsverdeling Φ in de tank, wordt gegeven door de volgende Poisson-vergelijking in cylinder-coördinaten:

$$L(U) = \frac{1}{r} \frac{\partial}{\partial r} \left(r \cdot \frac{\partial U}{\partial r} \right) + \frac{\partial^2 U}{\partial z^2} = -\Phi \quad 0 \leq r \leq 1, \quad 0 \leq z \leq H \quad (1)$$

met de randcondities:

$$U(1, z) = 0 \quad \text{en} \quad U(0, z) \quad \text{eindig voor} \quad 0 \leq z \leq H . \quad (2)$$

$$U(r, 0) = 0 \quad \text{en} \quad U(r, H) = 0 \quad \text{voor} \quad 0 \leq r \leq a . \quad (3)$$

De niet-homogene vergelijking (1) lossen we op door eerst met behulp van de methode van separatie van variabelen, een oplossing van de homogene vergelijking $L(U) = 0$ te construeren.

Substitutie van:

$$U(r, z) = R(r) \cdot Z(z) \quad \text{in} \quad L(U) = 0 , \quad (4)$$

(2) en (3) geeft, na te hebben gedeeld door $R(r) \cdot Z(z)$ en invoering van de separatieconstante

$$K^2 = -(\mu^2 + \sigma^2) \quad (5)$$

het verband:

$$\frac{1}{R} \cdot \frac{1}{r} \frac{d}{dr} \left(r \cdot \frac{dR}{dr} \right) = -\frac{1}{Z} \cdot \frac{d^2 Z}{dz^2} = -K^2 = -(\mu^2 + \sigma^2) . \quad (6)$$

Uit (6) volgen direct de beide gescheiden randwaarde-stelsels:

$$\frac{1}{r} \frac{d}{dr} \left(r \cdot \frac{dR}{dr} \right) + \mu^2 \cdot R = 0 \quad \text{met} \quad R(0) \quad \text{is} \quad \text{eindig} \quad \text{en} \quad R(a) = 0 . \quad (7)$$

$$\frac{d^2 Z}{dz^2} - \sigma^2 \cdot Z = 0 \quad \text{met voor} \quad Z(0) = 0 \quad \text{en} \quad Z(H) = 0 . \quad (8)$$

De algemene oplossing van (7) is:

$$R(r) = A \cdot J_0(\mu \cdot r) + B \cdot Y_0(\mu \cdot r)$$

(zie Appendix Besselfuncties voor deze en volgende formules).

Uit $R(0)$ eindig, volgt $B = 0$ en dus is:

$$R(r) = A \cdot J_0(\mu \cdot r) . \quad (9)$$

Uit $R(1) = 0$ volgt de eigenwaarde-vergelijking:

$$J_0(\mu) = 0 \quad (10)$$

met de reële eigenwaarden $\mu_1, \mu_2, \mu_3, \dots$ met

$$\mu_1 < \mu_2 < \mu_3 < \dots \quad (11)$$

en de eigenfuncties

$$R(r) = A \cdot J_0(\mu_k \cdot r) \quad k = 1, 2, 3, \dots . \quad (12)$$

De algemene oplossing van (7) is:

$$Z(z) = C \cdot \sinh(\sigma \cdot z) + D \cdot \cosh(\sigma \cdot z) . \quad (13)$$

Uit $Z(0) = 0$ volgt $D = 0$ en $Z(H) = 0$ levert de eigenwaardevergelijking $\sinh(\sigma H) = 0$ en dus de eigenwaarden

$$\sigma_n = (n \cdot \pi \cdot i) / H \quad n = 1, 2, \dots . \quad (14)$$

Uit (8) volgt dat $\sigma \cdot H = 0$ geen eigenwaarde is.

Met behulp van (14) is de volgende reële oplossing van (8) te schrijven:

$$Z(z) = i \cdot C \cdot \sinh((n \cdot \pi \cdot i) / H) = E \cdot \sin(n \cdot \pi \cdot z / H) \quad n = 1, 2, \dots . \quad (15)$$

Uit (12) en (15) volgt dat:

$$A_{kn} \cdot J_0(\mu_k \cdot r) \cdot \sin(n \cdot \pi \cdot z / H) \quad k, n = 1, 2, 3, \dots \quad (16)$$

een eigenfunctie is van het homogene stelsel $L(U) = 0$ behorende bij (1).

De eigenfuncties vormen een volledig orthogonaal stelsel.

Veronderstel nu dat de algemene oplossing van (1) kan worden voorgesteld door:

$$U(r, z) = \sum_{k=1}^{\infty} \sum_{n=1}^{\infty} A_{kn} \cdot J_0(\mu_k \cdot r) \cdot \sin(n \cdot \pi \cdot z / H) . \quad (18)$$

Ook de functie $\Phi(r, z)$ laat zich naar deze eigenfuncties ontwikkelen als:

$$\Phi(r, z) = \sum_{k=1}^{\infty} \sum_{n=1}^{\infty} B_{kn} \cdot J_0(\mu_k \cdot r) \cdot \sin(n \cdot \pi \cdot z / H) . \quad (19)$$

Met behulp van de orthogonaliteitsrelaties:

$$\int_0^1 r \cdot J_0(\mu_k \cdot r) \cdot J_0(\mu_l \cdot r) \, dr = \begin{cases} 0 & \text{voor } k \neq l \\ J_1(\mu_k)^2/2 & \text{voor } l = k \end{cases} \quad (20)$$

$$\int_0^H \sin(\pi \cdot n \cdot z/H) \cdot \sin(\pi \cdot m \cdot z/H) \cdot dz = \begin{cases} 0 & \text{voor } n \neq m \\ H/2 & \text{voor } n = m \end{cases} \quad (21)$$

kunnen we de in (19) optredende coëfficiënten $B_{k,n}$ als volgt bepalen. Vermenigvuldig (19) links en rechts met $r \cdot J_0(\mu_l \cdot r) \cdot \sin(n \cdot \pi \cdot z/H)$, integreer over de r - en z -gebieden (gebruik (20) en (21)). Dan volgt dat:

$$B_{k,n} = \frac{4 \cdot \int_0^1 r \cdot J_0(\mu_k \cdot r) \cdot dr \int_0^H \sin(\pi \cdot n \cdot z/H) \cdot \Phi(r, z) \cdot dz}{H \cdot J_0(\mu_k)^2} \quad (22)$$

De onbekende coëfficiënten $A_{k,n}$ in (18) kunnen nu als volgt worden bepaald: Substitueer in het linkerlid van (1) de dubbelreeks (18) en in het rechterlid voor Φ de nu bekende dubbelreeks (19). Verwissel in het linkerlid de sommatie en de differentiatie en daar in verband met (6), (7) en (8), voor elke term geldt dat:

$$\begin{aligned} & \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \cdot \frac{\partial}{\partial r} \right) + \frac{\partial^2}{\partial z^2} \right] \cdot A_{k,n} \cdot J_0(\mu_k \cdot r) \cdot \sin(n \cdot \pi \cdot z/H) = \\ & = (\mu^2 + (n \cdot \pi/H)^2) \cdot A_{k,n} \cdot J_0(\mu_k \cdot r) \cdot \sin(n \cdot \pi \cdot z/H) \end{aligned} \quad (23)$$

laten in deze operatie, linker en rechterlid zich als volgt samenvatten:

$$\sum_{k=1}^{\infty} \sum_{n=1}^{\infty} [(\mu^2 + (n \cdot \pi/H)^2) \cdot A_{k,n} - B_{k,n}] \cdot J_0(\mu_k \cdot r) \cdot \sin(n \cdot \pi \cdot z/H) = 0 \quad (24)$$

Vermenigvuldig (24) met $r \cdot J_0(\mu_l \cdot r) \cdot \sin(m \cdot \pi \cdot z/H)$, integreer over r en z en met behulp van de orthogonaliteitsrelaties volgt dat:

$$A_{k,n} = \frac{B_{k,n}}{\mu^2 + (n \cdot \pi/H)^2} \quad (25)$$

Na substitutie van (25) en (22) in de veronderstelde oplossing (18) wordt de volgende eindoplossing in volledig uitgeschreven vorm gevonden:

$$U(r, z) = \sum_{k=1}^{\infty} \sum_{n=1}^{\infty} \frac{J_0(\mu_k \cdot r) \cdot \sin((n \cdot \pi \cdot z/H))}{\mu^2 + (n \cdot \pi/H)^2}$$

$$\frac{\int_0^1 x \cdot J_0(\mu_k \cdot x) \cdot dx \int_0^H \sin(\pi \cdot n \cdot y/H) \cdot \Phi(x, y) \cdot dy}{J_1(\mu_k)^2 / 2 \cdot H/2} \quad (26)$$

waarin de μ_k 's > 0 de nulpunten zijn van $J_0(\mu) = 0$.

De oplossing (26) is op een iets eenvoudiger wijze te bepalen wanneer de theorie van Green functies en distributies (δ functies) bekend worden verondersteld. Met boven gegeven aanpak wordt dit vermeden. Voor een uniforme ladingsverdeling $\Phi = \text{constant}$, zijn de integralen in de oplossing (26) direct als volgt uit te werken. Voor de eerste integraal volgt:

$$\int_0^1 x \cdot J_0(\mu_k \cdot x) \cdot dx = J_1(\mu_k) / \mu_k .$$

Uit het resultaat van de tweede integraal:

$$\begin{aligned} \int_0^H \sin(\pi \cdot n \cdot x/H) \cdot dx &= \frac{H}{\pi \cdot n} \cdot \cos(\pi \cdot n \cdot x/H) \Big|_0^H = \\ &= \frac{H}{\pi \cdot n} \cdot \{(-1)^n - 1\} = \begin{cases} 0 & \text{voor } n \text{ even} \\ -2 & \text{als } n \text{ oneven} \end{cases} \end{aligned}$$

blijkt dat in de eindformule n moet worden vervangen door $(2 \cdot n + 1)$.

Na substitutie van deze resultaten in (26) volgt, dat voor Φ constant, de potentiaal in de geaarde tank wordt gegeven door de dubbelreeks¹⁾:

$$\begin{aligned} U(r, z) &= \Phi \cdot \pi \cdot \sum_{k=1}^{\infty} \sum_{n=0}^{\infty} \frac{1}{\mu_k \cdot (2 \cdot n + 1)} \cdot \frac{1}{J_1(\mu_k)} \cdot \\ &\cdot \frac{J_0(\mu_k \cdot r) \cdot \sin((2 \cdot n + 1) \cdot \pi \cdot z/H)}{\mu^2 + \{(2 \cdot n + 1) \cdot \pi/H\}^2} . \end{aligned} \quad (27)$$

Maken we gebruik van een in de wiskunde bekende identiteit^{2,3)}, dan is de sommatie over n analytisch uit te werken en wordt de oplossing:

$$U(r, z) = 2 \cdot \Phi \cdot \sum_{k=1}^{\infty} \frac{J_0(\mu_k \cdot r)}{\mu_k^3 \cdot J_1(\mu_k)} \cdot \left[1 - \frac{\cosh(\mu_k \cdot (H - 2 \cdot z)/2)}{\cosh(\mu_k \cdot H/2)} \right] . \quad (28)$$

Beschouwen we in de oplossing (28) de uitdrukking tussen [], dan zien we direct (de $\cosh(x)$ is een even functie) dat deze uitdrukking nul is voor $z = 0$ en $z = H$ en er dus voldaan is aan de randcondities.

Voor $z = H/2$, heeft in de teller $\cosh(0) = 1$ zijn kleinste waarde en dus heeft de uitdrukking tussen [...] zijn grootste waarde.

Conclusie voor $z = H/2$: In het midden van de tank wordt de maximale potentiaal gevonden.

Van de oplossing (28) beschouwen we het speciale geval:

$$V(r) = \Phi \cdot \sum_{k=1}^{\infty} \frac{J_0(\mu_k \cdot r)}{\mu_k^3 \cdot J_1(\mu_k)} . \quad (29)$$

Het blijkt dat $V(r)$ de voorstelling is van de potentiaal in een oneindig lange gearde cylinder en dus is de oplossing van de vergelijking:

$$L(V) = \frac{1}{r} \frac{\partial}{\partial r} \left(r \cdot \frac{\partial V}{\partial r} \right) = 0 \quad 0 \leq r \leq 1 \quad \text{met} \quad V(r) = \frac{1}{8} \cdot (1 - r^2) . \quad (30)$$

Dit is weer te bewijzen door de functie $V(r) = (1 - r^2)/8$ als volgt te onwikkelen naar de eigenfuncties van (7):

Stel

$$V(r) = \sum_{k=1}^{\infty} A_k \cdot J_0(\mu_k \cdot r) \quad (\text{de } \mu_k \text{'s zijn nulpunten van } J_0(\mu) = 0) . \quad (31)$$

Vermenigvuldig linker en rechter lid met $r \cdot J_0(\mu_l \cdot r) \cdot dr$ en integreer over het interval $[0, 1]$. Vanwege de orthogonaliteit zijn alle bijdragen in het rechterlid nul, behalve voor $l = k$ en volgt dat:

$$\frac{1}{8} \cdot \int_0^1 r \cdot (1 - r^2) \cdot J_0(\mu_k \cdot r) \cdot dr = A_k \cdot \frac{J_1(\mu_k)^2}{2} . \quad (32)$$

Voor het linkerlid verloopt de uitwerking met behulp van de integraalrelaties voor de Besselfuncties en partiële integratie als volgt:

$$\begin{aligned} \int_0^1 \frac{(1 - r^2)}{8} d(r J_1(\mu_k r)) &= \frac{(1 - r^2)}{8 \mu_k^2} r J_1(\mu_k r) \Big|_0^1 - \int_0^1 r J_1(\mu_k r) d \frac{(1 - r^2)}{8 \mu_k^2} = \\ &= \int_0^1 r^2 \frac{J_1(\mu_k r)}{4 \mu_k^2} dr = \frac{J_2(\mu_k)^2}{4 \mu_k^2} = \frac{1}{4 \mu_k^2} \left[\frac{2}{\mu_k} J_1(\mu_k) - J_0(\mu_k) \right] = \frac{J_1(\mu_k)}{2 \mu_k^3} . \end{aligned} \quad (33)$$

Lossen we A_k met behulp van (33) op uit (32), dan blijkt na substitutie in (31), dat deze oplossing overeenstemt met (29).

Bij de computerimplementaties van (27) of (28) en (29), moet eerst een voldoende aantal eigenwaarden μ_k van $J_0(\mu) = 0$ worden berekend.

In het eerder gebruikte computerprogramma KARAKTER.BAS hebben we

laten zien hoe de in absolute waarde kleinste nulpunten, uit de machtreeksvoorstelling van $J_0(x)$, kunnen worden bepaald.

Omdat in de programmas later toch de Besselfuncties $J_0(x)$ en $J_1(x)$ nodig zijn, is analoog als bij de diffusieproblemen, weer gebruik gemaakt van de SUB INTERVAL. Aan de hand van (28) behandelen we, hoe aan de hand van een schatting van de afbreekfout, het benodigde aantal eigenwaarden N kan worden bepaald als we $U(r, z)$ met een voorgeschreven nauwkeurigheid willen berekenen. We moeten het aantal termen N dus berekenen uit een schatting van de volgende uitdrukking in de afbreekfout:

$$\sum_{k=N+1}^{\infty} \left| \frac{J_0(\mu_k \cdot r)}{\mu_k^3 \cdot J_1(\mu_k)} \cdot \left[1 - \frac{\cosh(\mu_k \cdot (H - 2z)/2)}{\cosh(\mu_k \cdot H/2)} \right] \right| \leq \text{eps} . \quad (34)$$

Schattingen voor de eigenwaarden en eigenfuncties kunnen worden bepaald met behulp van de algemene asymptotische formule:

$$J_v(x) \approx \sqrt{\{2/(\pi x)\}} \cdot [\cos\{x - (v/2 + 1/4) \cdot \pi\} \cdot P(v, x) + \sin\{x - (v/2 + 1/4) \cdot \pi\} \cdot Q(v, x)] .$$

Voor $v = 0$ blijkt uit (35), dat de eigenwaardebenaderingen volgen uit:

$$\cotg(\mu - \pi/4) = Q(0, \mu)/P(0, \mu) = -1/(8 \cdot \mu) + O(1/\mu^3) . \quad (35)$$

Bij verwaarlozing van het rechterlid (μ voldoende groot) volgt uit: $\cotg(x - \pi/4) = 0$, als eerste-orde-benadering voor de eigenwaarden:

$$\mu_k = (k - \frac{1}{4}) \cdot \pi + O(1/k) \quad k = 1, 2, \dots . \quad (36)$$

Met behulp hiervan gelden de volgende schattingen voor:

$$|J_0(\mu_k \cdot r)| \leq 1 \quad \text{voor } r \geq 0 \quad (\text{algemene eigenschap}) .$$

Voor $v = 1$ volgt uit (30) dat:

$$J_1(x) \approx \sqrt{\{2/(\pi \cdot x)\}} \cdot \cos(x - 3 \cdot \pi/4) + O(1/x)$$

en met behulp van (36) is dit te vereenvoudigen tot:

$$J_1(\mu_k) \approx \sqrt{\{2/(\pi \cdot \mu_k)\}} \cdot \{(-1)^{k-1} + O(1/k^2)\}$$

en dus geldt voor:

$$\frac{1}{\mu_k^3 \cdot J_1(\mu_k)} \approx \frac{1}{\sqrt{\{2/\pi\}} \cdot (-1)^{m-1} \cdot (k - 1/4)^{5/2} + O(1/\sqrt{k})} \approx$$

$$\approx \sqrt{\pi/2} \cdot \frac{(-1)^{k-1}}{(k-1/4)^{5/2}} + O(1/k^3).$$

De volgende schatting geldt voor $z = H/2$ (d.w.z.: $\cosh(\mu_k \cdot (H - 2z)/2) = 1$)

$$\begin{aligned} 1 - \frac{1}{\cosh(\mu_k \cdot H/2)} &\approx 1 - \frac{1}{1 + (\mu_k \cdot H/2)^2/2! + O(k^4)} \leq \\ &\leq 1 - \frac{1}{(\mu_k \cdot H/2)^2/2!(1 + O(k^2))} \approx \\ &\approx 1 - \frac{2!}{\{(k-1/4) \cdot \pi \cdot H\}^2} + O(k^{-4}) \approx 1 + O(k^{-2}). \end{aligned}$$

Substitutie van de gevonden schattingen in (29), levert dan het volgende verband tussen N en eps :

$$\sum_{k=N+1}^{\infty} \frac{\sqrt{\pi/2}}{(k-1/4)^{5/2}} \leq \int_N^{\infty} \frac{\sqrt{\pi/2}}{(x-1/4)^{5/2}} dx = \frac{2}{3} (N - \frac{1}{4})^{-3/2} \leq \text{eps}.$$

Uit deze ongelijkheid kan voor een gegeven eps de N worden bepaald. Voor $\text{eps} = 5 \cdot 10^{-2}, 5 \cdot 10^{-3}, 5 \cdot 10^{-4}$, geldt de ongelijkheid ruw geschat voor: $N = 10, 30, 125$.

Conclusie: Een oplossing ontwikkeld naar eigenfuncties verschaft wel veel inzicht, maar is minder geschikt als er een grote nauwkeurigheid wordt vereist.

In de techniek zijn de electrostatische opladingsverschijnselen, die kunnen worden beschreven door een Poisson-vergelijking, vaak veroorzakers van grote schade, zoals b.v. bij: Stofexplosies in fabrieken en silo's, bij het schoonmaken van tanks in olietankers³⁾, het verpompen van vloeibare brandstoffen door pijpen.

In veel gevallen zoals b.v. bij chemische processen in om een as draaiende reactoren met verschillende vloeistoflagen en een dampfase bovenin, moet er een stelsel van gekoppelde Laplace- en Poisson-vergelijkingen worden opgelost. Hoewel het rekenwerk dan sterk toeneemt, is toch steeds de bovenbeschreven aanpak mogelijk.

De implementatie van de oplossing: (27) als SUB DUBBELSOM

```
SUB DUBBELSOM (N, M1, H, Z, R, DR, A(), E(), PI)
FOR K = 1 TO M1 + 1: S = 0: SJ = 0
FOR L = 1 TO N: Z1 = R * A(L)
SJ = XJ0(Z1) / (XJ1(A(L)) * A(L)): SZ = 0: SZN = 0
FOR I = 0 TO N
SZN = ((A(L)^2 + (CDBL(2*I + 1) * PI/H) ^2) * CDBL(2*I + 1))
SZT = SIN((2 * I + 1) * PI * Z / H): SZ = SZ + SZT / SZN
NEXT I: S = S + SJ * SZ
```

```

    NEXT L: E(K) = 8# * S / PI: R = R + DR:
  NEXT K
END SUB

```

De implementatie van de oplossing: (28) als de SUB ENKELSOM

```

SUB ENKELESOM (N, M1, H, Z, R, DR, A(), E(), PI)
  FOR K = 1 TO M1 + 1: S = 0: SJ = 0
    FOR L = 1 TO N : Z1 = R * A(L)
      TZ = (1# - COSH(W2 * (H - 2#*Z)/2#) / COSH(A(L) * H/2#))
      SJ = XJ0(Z1) * TZ / (XJ1(A(L)) * A(L) ^ 3): S = S + SJ
    NEXT L: E(K) = 2# * S: R = R + DR:
  NEXT K
END SUB

```

De implementatie van de oplossing: (30) als SUB ONEINDIGECYLINDER

```

SUB ONEINDIGECYLINDER (N, M1, R, DR, A(), E())
  FOR K = 1 TO M1 + 1: S = 0: SJ = 0
    FOR L = 1 TO N: Z1 = R * A(L): W2 = A(L)
      SJ = XJ0(Z1) / (XJ1(A(L)) * A(L) ^ 3): S = S + SJ
    NEXT L: E(K) = S: R = R + DR:
  NEXT K
END SUB

```

De aangepaste SUB INTERVAL:

```

SUB INTERVAL (FO, FM, FB, XO, XB, XM, EPS)
  DO: FO = XJ0(XO): FB = XJ0(XB)
    IF SGN(FO) = SGN(FB) THEN XB = XB + .5#
  LOOP UNTIL SGN(FO) <> SGN(FB)
  KK = 0: WHILE (XB - XO) ≥ EPS
    XM = (XO + XB) / 2#: FM = XJ0(XM)
    IF SGN(FM) = SGN(FO) THEN
      XO = XM ELSE XB = XM
      KK = KK + 1
    WEND :PRINT XM, FM: PRINT
END SUB

```

Omdat in deze toepassing het argument $x = 0$ optreedt, zijn er met behulp van de bij Besselfuncties behandelde methoden, de volgende nieuwe implementaties van $J_0(x)$ en $J_1(x)$ gemaakt: FUNCTION XJ0(x) en FUNCTION XJ1(x). Deze subs en functions zijn opgenomen in het programma POT-FUN.BAS op de schijf. De oplossingen kunnen voor gewenste parameterkeuzes

worden berekend en ook worden getekend.

Referenties.

- 1) Morse and Feshbach, 1953, Methods of Theoretical Physics II, p. 1263.
- 2) Vellinga S.J., 1960, Appl. Sci. Res., B9, pp. 35-42.
- 3) Smit W., 1971, Stat. Elect., 1971.

Appendix A. Besselfuncties

Een verzameling Besselfunctie-formules, die worden gebruikt in een aantal Subroutine implementaties of een rol spelen bij de Laplace- transformatie en partiële differentiaalvergelijkingen.

De functie:

$$J_v(z) = \sum_{r=0}^{\infty} \frac{(-)^r (z/2)^{v+2r}}{r! \Gamma(v+r+1)} = \frac{(z/2)^v}{\Gamma(v+1)} \cdot {}_0F_1(v+1; -z^2/4) \quad (1)$$

is voor alle v geheel of gebroken en z reëel of complex, een oplossing van de differentiaalvergelijking van Bessel:

$$\frac{d^2 y}{dz^2} + \frac{1}{z} \cdot \frac{dy}{dz} + (1 - v^2/z^2)y = 0. \quad (2)$$

Als v niet geheel is, zijn $J_v(z)$ en $J_{-v}(z)$ onafhankelijke oplossingen.

Voor $v = n$ geheel, is:

$$J_{-n}(z) = (-)^n \cdot J_n(z). \quad (3)$$

Een onafhankelijke oplossing $Y_n(z)$ kan worden gevonden uit:

$$Y_n(z) = \lim_{v \rightarrow n} \frac{\{J_v(z) \cdot \cos(v\pi) - J_{-v}(z)\}}{\sin(v\pi)}.$$

Of in machtreeksvorm:

$$\begin{aligned} \pi \cdot Y_n(z) = & 2 \cdot J_0(z) \cdot \log(z/2) - \sum_{r=0}^{n-1} (n-r-1)! \cdot (z/2)^{-n+2r}/r! + \\ & - \sum_{r=0}^{\infty} \frac{(-)^r \cdot [\psi(r+1) + \psi(n+r+1)] \cdot (z/2)^{n+2r}}{(\Gamma(r+1) \cdot \Gamma(n+r+1))}. \end{aligned} \quad (4)$$

Voor $n = 0$ volgt uit deze definitie:

$$\begin{aligned} \frac{\pi}{2} \cdot Y_0(z) = & \{\log(\frac{z}{2}) + \gamma\} \cdot J_0(z) + (\frac{z}{2})^2 - (1+1/2) \cdot (\frac{z}{2})^4 / (2!)^2 + \\ & + (-)^n \cdot (1+1/2+1/3+\dots+1/n) \cdot (\frac{z}{2})^{2n} / (n!)^2 + \dots \end{aligned} \quad (5)$$

hierin is $\gamma = 0.5772156649\dots$ de constante van Euler.

Voor de afgeleiden van deze functies gelden de relaties:

$$J'_n(z) = -J_{n+1}(z) + \frac{n}{z} \cdot J_n(z) \quad Y'_n(z) = -Y_{n+1}(z) + \frac{n}{z} \cdot Y_n(z). \quad (6)$$

Voor $n = 0$ volgt:

$$J_0'(z) = -J_1(z) \quad Y_0'(z) = -Y_1(z) \quad (7)$$

$$Y_{n-1}(x) \cdot J_n(x) - Y_n(x) \cdot J_{n-1}(x) = 2/(\pi \cdot x) \quad (\text{Wronskiaan}) \quad (8)$$

De Hankelfuncties: $H_v^{(1)}(z)$ en $H_v^{(2)}(z)$ zijn gedefinieerd als:

$$H_v^{(1)}(z) = J_v(z) + i \cdot Y_v(z) \quad H_v^{(2)}(z) = J_v(z) - i \cdot Y_v(z) \quad (9)$$

De gemodificeerde Besselfuncties $I_v(z)$, $K_v(z)$ $v = 0, 1, 2, \dots$ zijn de oplossingen van de differentiaalvergelijking:

$$\frac{d^2 y}{dz^2} + \frac{1}{z} \cdot \frac{dy}{dz} - \left(1 + \frac{v^2}{z^2}\right) y = 0 \quad (10)$$

$$I_v(z) = \exp(-iv\pi/2) \cdot J_v(iz) \quad (-\pi < \arg z \leq \pi/2) \quad (11)$$

$$I_v(z) = \sum_{r=0}^{\infty} \frac{(z/2)^{v+2r}}{r! \cdot \Gamma(v+r+1)} \quad (12)$$

$$K_v(z) = i \cdot \exp(iv\pi/2) H_v^{(1)}(iz) \quad (-\pi < \arg z \leq \pi/2) \quad (13)$$

$$K_n(z) = \frac{\left(\frac{z}{2}\right)^{-n}}{2} \sum_{k=0}^{n-1} \left(-\frac{z^2}{4}\right)^k \frac{(n-k-1)!}{k!} + (-1)^{n+1} \ln\left(\frac{z}{2}\right) \cdot I_n(z) + (-z/2)^n \cdot \sum_{k=0}^{\infty} \frac{(\psi(k+1) + \psi(n+k+1)) \cdot (z/2)^{2k}}{k! \cdot (n+k)!} \quad (14)$$

$$I_0'(z) = I_1(z) \quad K_0'(z) = -K_1(z) \quad (15)$$

Asymptotische reeksen.

$$J_v(z) = \sqrt{\frac{2}{\pi z}} \{P(v, z) \cos(Z) - Q(v, z) \sin(Z)\} \quad |\arg z| < \pi \quad (16)$$

$$Y_v(z) = \sqrt{\frac{2}{\pi z}} \{P(v, z) \sin(Z) + Q(v, z) \cos(Z)\} \quad |\arg z| < \pi \quad (17)$$

$$H_v^{(1)}(z) = \sqrt{\frac{2}{\pi z}} \{P(v, z) + i \cdot Q(v, z)\} e^{iZ} \quad -\pi < \arg z < 2\pi \quad (18)$$

$$H_v^{(2)}(z) = \sqrt{\frac{2}{\pi z}} \{P(v, z) - i \cdot Q(v, z)\} e^{-iZ} \quad -2\pi < \arg z < \pi \quad (19)$$

$$P(v, z) \sim \sum_{k=0}^{\infty} (-)^k \frac{(v, 2k)}{(2z)^{2k}} = 1 - \frac{(\mu-1)(\mu-32)}{2!(8z)^2} + \dots \quad (20)$$

$$Q(v, z) \sim \sum_{k=0}^{\infty} (-)^k \frac{(v, 2k+1)}{(2z)^{2k+1}} = \frac{(\mu-1)}{1!(8z)} - \frac{(\mu-1)(\mu-3^2)(\mu-5^2)}{3!(8z)^3} - \dots \quad (21)$$

waarin

$$Z = z - (v/2 + 1/4)\pi ; \quad \mu = 4.v^2 \quad (22)$$

$$\begin{aligned} (v, m) &= (-)^m \frac{(1/2 - v)_m \cdot (1/2 + v)_m}{m!} = \\ &= \frac{(-)^m (\mu-1)(\mu-3^2) \dots (\mu - (2m-1)^2)}{m! \cdot (2)^{2m}} \end{aligned} \quad (23)$$

$$\begin{aligned} K_v(z) &\sim \sqrt{\frac{\pi}{2z}} e^{-z} \cdot \sum_{m=0}^{\infty} \frac{(v, m)}{(2.z)^m} = \\ &= \sqrt{\frac{\pi}{2z}} e^{-z} \left[1 + \frac{(\mu-1)}{1!(8z)} + \frac{(\mu-1)(\mu-3^2)}{2!(8z)^2} + \dots \right] \end{aligned} \quad (24)$$

$$I_v(z) \sim \frac{1}{\sqrt{(2\pi z)}} \cdot \left\{ e^z \cdot \sum_{m=0}^{\infty} (-)^m \cdot \frac{(v, m)}{(2.z)^m} + e^{-z+(v+1/2)\pi \cdot i} \cdot \sum_{m=0}^{\infty} \frac{(v, m)}{(2.z)^m} \right\} \quad (25)$$

$$K_0(z) \sim \sqrt{(\pi/(2z))} \cdot e^{-z} \cdot {}_2F_0(1/2, 1/2; -1/2z) \quad (26)$$

$$K_1(z) \sim \sqrt{(\pi/(2z))} \cdot e^{-z} \cdot {}_2F_0(3/2, -1/2; -1/2z) . \quad (27)$$

Uit de relaties (9)...(15) volgt dat:

$$K_v(iz) = i \cdot \pi/2 \cdot \exp(-iv\pi/2) \cdot (-J_v(z) + i \cdot Y_v(z)) \quad (28)$$

voor $v = 0$ en $v = 1$ volgen:

$$-2/\pi \cdot K_0(iz) = i \cdot J_0(z) + Y_0(z) \quad (29)$$

$$2/\pi \cdot K_1(iz) = -J_1(z) + i \cdot Y_1(z) . \quad (30)$$

Voor $\text{Re } z > 0$ en v geheel geldt:

$$J_v(x)^2 + Y_v(x)^2 \approx \frac{2}{\pi \cdot x} \cdot \sum_{m=0}^{\infty} [1 \cdot 3 \cdot 5 \dots (2m-1)] \cdot \frac{(v, m)}{(2x)^{2m}} . \quad (31)$$

Bij het oplossen van lineaire partiële differentiaalvergelijkingen in cilindricoördinaten, kan de algemene oplossing van (2) worden genoteerd als:

$$Z_n(x) = p.J_n(x) + q.Y_n(x), \quad n \text{ geheel, } p, q \text{ reële constanten.} \quad (32)$$

Met behulp van (32) worden de relaties (6) samengevat tot:

$$Z'_n(x) = -\frac{n}{x}.Z_n(x) + Z_{n-1}(x). \quad (33)$$

In deze notatie laten de volgende onbepaalde integralen zich gemakkelijk uitwerken:

$$\int x^{n+1}.Z_n(x).dx = x^{n+1}.Z_{n+1}(x) \quad (34)$$

$$\int x.Z_n(x).dx = (x^2/2).\{Z_n(x)^2 - Z_{n+1}(x).Z_{n-1}(x)\} \quad (35)$$

$$\int Z_1(x).dx = Z_0(x) \quad (36)$$

$$\int x.Z_0(x).dx = x.Z_1(x). \quad (37)$$

Voor de functies $I_v(z)$ en $K_v(z)$ (z complex) gelden de relaties:

$$\int z.I_0(z).dz = z.I_1(z) \quad \text{en} \quad \int z.K_0(z).dz = -z.K_1(z) \quad (38)$$

$$\int z^{v+1}.I_v(z).dz = z^{v+1}.I_{v+1}(z) \quad \text{en}$$

$$\int z^{v+1}.K_v(z).dz = -z^{v+1}.K_{v+1}(z) \quad (39)$$

$$\int z^{-v+1}.I_v(z).dz = z^{-v+1}.I_{v-1}(z) \quad \text{en}$$

$$\int z^{-v+1}.K_v(z).dz = -z^{-v+1}.K_{v-1}(z) \quad (40)$$

$$I_v(z).K_{v+1} + I_{v+1}(z).K_v = 1/z \quad (\text{Wronskiaan}). \quad (41)$$

Referenties.

Handbook of Mathematical functions. Edited by Milton Abramowitz and Irene A. Stegun.

A treatise on the theory of Bessel functions by G.N. Watson.

Speciale Functies in de Mathematische Fysica. N.M. Temme. Epsilon Uitgaven, Utrecht.

Appendix B. Een Alternatieve Eigenvector Bepaling

Als van een randwaardeprobleem de eigenwaarden bekend zijn, kunnen de bijbehorende eigenvectoren, met een numeriek integratie- algoritme, uit de differentiaalvergelijking op $[a, b]$, worden bepaald. Bij deze aanpak, wordt alleen de randconditie bij $r = a$ gebruikt. Het is dus van belang dat de eigenwaarden met grote nauwkeurigheid bekend zijn, daar anders niet of slecht aan de tweede randconditie bij b wordt voldaan.

De eigenwaarden worden daarom eerst, met behulp van een extrapolatieformule^{1,2,3}) gecorrigeerd. We benaderen elke eigenwaarde T , met SUB EIVAL (of de SUBs KARAKTERISTIEK en PROGRESSIEF), uitgaande van de stapgrootte $h = (b - a)/N$, achtereenvolgens voor een rij stapgrootten $h_j = h/2^j$ $j = 0, 1, 2, 3, \dots$, de grootheden:

$$T_{k,j} = T_k(h_j) , \quad \text{waarbij } T_{0,j} = T(h_j)$$

met de formule:

$$T_{k+1,j} = T_{k,j} + (T_{k,j} - T_{k,j-1})/(2^{k+1} - 1) .$$

De $T_{k,k}$ waarden in elke volgende rij (in onderstaand Rombergschema) zijn dan steeds betere benaderingen van de limietwaarde T :

$$\begin{array}{cccc} h_0 & T_{0,0} & & \\ h_1 & T_{0,1} & T_{1,1} & \\ h_2 & T_{0,2} & T_{1,2} & T_{2,2} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array}$$

We gaan dit procédé gebruiken, om van het eerder behandelde Ultracentrifuge-randwaarde-probleem, enkele eigenwaarden zo exact mogelijk te berekenen. Met behulp van de SUB EIVAL berekenen we voor $N = 20, 40, 80$ en 160 roosterpunten, de waarden $T_{0,j}$ $j = 0, \dots, 4$. Vervolgens wordt de extrapolatieformule gebruikt voor het berekenen van de $T_{k,k}$ - waarden.

In het computerprogramma ULTRICHA.BAS is het Rombergschema zo geïmplementeerd dat het ook gemakkelijk kan worden gebruikt in andere numerieke processen.

Nu de eigenwaarden voldoende bekend zijn, kan een numeriek integratie-algoritme worden gebruikt voor het bepalen van de eigenvectoren.

De eigenvector bij de eigenwaarde α , wordt gevonden als oplossing van het differentiaalvergelijking-beginwaarde probleem:

$$\frac{d^2U}{dz^2} + \left(\frac{1}{z} - 1\right) \cdot \frac{dU}{dz} + \frac{\alpha}{z} \cdot U = 0 \quad \text{met} \quad \frac{dU}{dz} - U = 0 \quad \text{voor} \quad z = 5.0 .$$

We veronderstellen de numeriek integratie van een differentiaalvergelijking-be-

ginwaardepro bleem als bekend³⁾.

We gaan gebruik maken van het volgende zeer bekende 4-de orde Runge-Kutta numerieke integratie-algorithme⁴⁾.

$$\begin{aligned}
 y_{n+1} &= y_n + k_1/6 + k_2/3 + k_3/3 + k_4/6 + O(h^5) \\
 k_1 &= h \cdot f(x_n, y_n) \quad k_2 = h \cdot f(x_n + h/2, y_n + k_1/3) \\
 k_3 &= h \cdot f(x_n + 2 \cdot h/3, y_n - k_1/3 + k_2) : k_4 = h \cdot f(x_n + h, y_n + k_3) .
 \end{aligned}$$

De SUB RUNGEKUTTA is opgenomen in het programma ULTRARUN.BAS. Teneinde de zo te verkregen resultaten gemakkelijk te kunnen vergelijken met die welke vroeger zijn berekend met SUB EIGVEC (voor $N = 80$), gebruiken we de daar bij $r = a$ gevonden waarden als begincondities in ons beginwaarde probleem. (De hiermee te genereren eigenvector is dan nl. ook direct genormeerd.)

Referenties.

- 1) Numerical Analysis. D.R. Hartree, p. 234 etc.
- 2) Pure & Applied Mathematics. Peter Henrici, Vol. 2, pp. 461-467.
- 3) Einführung in die Numerische Mathematik I. Josef Stoer.
- 4) Handbook of Mathematical Functions. Abramowitz etc, p. 896.

Appendix C. De Programma's en Algorithmen

De behandelde programma's kunnen op verschillende manieren worden ingedeeld:

- a) Complete programma's die direct kunnen worden gebruikt.
- b) SUBs en FUNCTIONs, bruikbaar bij nieuwe ontwikkelingen.

Complete programma's voor het bepalen van wortels van polynomen, zijn: NEWTON.BAS, BAIRSTOW.BAS, INTERVAL.BAS, LAGUER.BAS, NULPUNT.BAS en KARAKTER.BAS.

Met BUHRMAN.BAS worden via de ingevoerde machtreekscoëfficiënten de coëfficiënten van de inverse machtreeks bepaald.

Voor het bepalen van de eigenwaarden en eigenvectoren van verschillende typen reële matrices:

EWAVECI.BAS, EWAVECN.BAS, TQLHES.BAS, CHOLTQL.BAS en ALG343.BAS.

Reële kettingbreuk-toepassingen:

QD-ALG.BAS, ELEMSUB.BAS, NULPUNT.BAS, KARAKTER.BAS, GETALQB.BAS.

Kettingbreuk-toepassingen van het complexe variabelen-type:

LAPLACE.BAS, ERR-PART.BAS, I0-PART.BAS, K0K1PART.BAS.

De partiële differentiaalvergelijkingen (randwaarde problemen):

ANALYBES.BAS, BESSELD.F.BAS, ULTRA.BAS, POT-FUN.BAS, ULTRICHA.BAS.

De hulp-programma's welke zijn gebruikt bij het ontwikkelen van de diverse speciale functie algorithmen:

ZETAGAM.BAS, QDBESSEL.BAS, ASYM-ALT.BAS, KIZMACHT.BAS, KNPSIFAC.BAS, INKNCOMP.BAS.

Voor een zo efficiënt mogelijk gebruik van de vele SUBs en FUNCTIONs, bij later programmeerwerk, is het b.v. doelmatig, de verwante typen te rangschikken in de volgende groepen:

De subs voor de elementaire complexe variabelen-operaties.

```

FUNCTION CABS( A() )
SUB CPLUS ( A(), B(), C() )
SUB CMIN ( A(), B(), C() )
SUB CMAAL(X10, Y10, X20, Y20, X30, Y30)
SUB CDIV (AR, AI, BR, BI, CR, CI)
SUB CSQR (AR, AI, BR, BI)
SUB CLOG ( X10, Y10, AMP, ARG)
SUB CSINHCOSH( X10, Y10, X20, Y20, X30, Y30)

```

De subs voor de Kettingbreuk-operaties.

```

SUB QD( M, A(), C() )
SUB FWAARDE (M, C(), Z, F)
SUB QDANDERS ( M, A(), D() )
SUB FWAARDE1 (M, D(), WN, FOUT)
SUB CQD ( N, AR(), AI(), CR(), CI() )
SUB CWAARDEKET (N, CR(), C(), X, Y, WR, WI)

```

Subs voor Numerieke Laplace-transformatie.

```

SUB LAPLACEQD (N, M, AR(), A(), DR(), DI() )
SUB CWAARDELAP (N, J, LI, TT, FA, GA, F1, AR(), AI() )

```

De subs voor de matrix eigenwaarde- en eigenvector- operaties.

```

SUB EIVALI( N2%, K2%, DIA#(), CD#(), ZLA#() )
SUB EIVALN( N2%, K2%, DIA#(), CD#(), ZLA#() )
SUB EIVEC( N2%, K2%, DIA#(), CD#(), ZLA#() )
SUB HESTQL( N2%, D#(), A#(), Z#() )
SUB CHOLESKYREDUCTIE (N%, P#(), A#(), B#(), Z#(), D#() )
SUB EIGWEIGVECI( N2%, K2%, A#(), ZLA#(), ZV#() )
SUB EIGWEIGVECN( N2%, K2%, A#(), ZLA#(), ZV#() )
SUB EIGENP( N%, A#(), EVR#(), EVI(), VERCR#(),
  VERCI#(), INDIC%() )

```

en de SUB's

SCALE, HESQR, REALVE, COMPVE.

De speciale functies:

```

FUNCTION FAC#( MM% )
FUNCTION PSI# (MM% )
FUNCTION GAMMA# (XX# )

```

De subs voor de reële Besselfunctie-operaties.

```
SUB J0Y0( T#, U#, V#), U = J0(T) en V = Y0(T)
SUB J1Y1( T#, U#, V#), U = J1(T) en V = Y1(T)
FUNCTION XJ0#( Z1#) ' XJ0 = J0(Z)
FUNCTION XJ1#( Z1#) ' XJ1 = J1(Z)
```

De subs voor de Besselfuncties met complex argument.

```
SUB I0Z(X1#, Y1#, F1#, F2#)' F1 + i.F2 = I0(x1 + i.y1)
SUB I1Z(X1#, Y1#, F1#, F2#)' F1 + i.F2 = I1(x1 + i.y1)
SUB K0Z(X1#, Y1#, F1#, F2#)' F1 + i.F2 = K0(x1 + i.y1)
SUB K1Z(X1#, Y1#, F1#, F2#)' F1 + i.F2 = K1(x1 + i.y1)
SUB WAARDEKET(NN%, X3#, Y3#, P7#(), G1#, G2#)
```

Afhankelijk van de voorkeur van de gebruiker, zijn er verschillende manieren van werken mogelijk:

- 1) Verzamel de boven vermelde subs en functions met behulp van een geschikte tekstverwerker in één of meer files. Importeer hieruit de gewenste exemplaren in de nieuwe toepassing.
- 2) De verschillende programma's opnemen in een Quick Library.
- 3) De subs en functions opnemen in een (object) Stand- Alone-Library.
De werkwijze voor het laatste geval is als volgt: Eerst elke FUNCTION (SUB) opnemen in een aparte file.

V.b.: C1.BAS

```
DEFINT I - N
DEFDBL A - H, O - Z
FUNCTION CABS( A())
X = ABS(A(1)), Y = ABS(A(2))
IF X = 0 THEN CABS = Y: EXIT FUNCTION
IF Y = 0 THEN CABS = X: EXIT FUNCTION
IF X > Y THEN
  CABS = Y * SQR(1# + (Y/X)^2)
ELSE
  CABS = Y * SQR(1# + (X/Y)^2)
END IF
END FUNCTION
```

Compileer als volgt: BC C1/O;

Maak met de Library manager LIB.EXE een Library b.v.: QBSUBS.LIB.

Neem de gegenereerde object files hierin op: LIB QBSUBS +c1+c2+...+cn.

Appendix D, De Functie van Green

Bij het oplossen van partiële differentiaalvergelijkingen met behulp van Laplacetransformaties, moet als deelprobleem vaak een tweede orde inhomogeen randwaardeprobleem worden opgelost. De oplossing van het inhomogene probleem kan worden bepaald als eerst een speciale oplossing van het homogene randwaardewaarde probleem, een functie van Green, wordt geconstrueerd. Met behulp van deze functie kan dan elke gewenste oplossing op geschikte wijze worden berekend. We construeren de oplossing van de volgende algemene zelfgeadjungeerde differentiaalvergelijking^{d)}:

$$L[U] = \frac{d}{dx} \left\{ p(x) \cdot \frac{dU(x)}{dx} \right\} + q(x) \cdot U(x) = f(x) \quad \text{op } [0, 1], \quad (1)$$

waarin $U(x)$, $p(x)$, $q(x)$ en $f(x)$ continue functies van x zijn.

In de eindpunten specificeren we later nog lineaire homogene of inhomogene randcondities.

In de literatuur wordt de Greenfunctie vaak aangegeven met de letter G .

We zoeken de oplossing van: $L[U] = f(x)$ op $[0, 1]$ met behulp van een Greenfunctie welke een oplossing is van de homogene vergelijking:

$$L[G] = 0, \quad \text{in de gebieden } [0, y] \text{ en } (y, 1] \quad \left| \begin{array}{c} \text{---} \\ 0 \end{array} \right| \quad \left| \begin{array}{c} \text{---} \\ y \end{array} \right| \quad \left| \begin{array}{c} \text{---} \\ 1 \end{array} \right|. \quad (2)$$

De functie G zal behalve van x , dus ook van de variabele y afhangen, we noteren de Greenfunctie daarom in het vervolg als: $G = G(x, y)$.

In het punt $x = y$ hoeft $L[G]$ niet te bestaan, wel wordt $G(x, y)$ voor $x = y$ continu verondersteld.

Aan de afgeleide $\frac{dG(x, y)}{dx}$ wordt later nog een extra voorwaarde opgelegd.

We maken gebruik van de bekende tweede identiteit van Green^{d)}:

$$\int_O \int \{ Q \cdot L[P] - P \cdot L[Q] \} do = \int_S \left\{ Q \cdot \frac{dP}{dn} - P \cdot \frac{dQ}{dn} \right\} ds. \quad (3)$$

Substitueren we in (3) overeenkomstig (1) en (2) voor $P = U$ en $Q = G$ dan volgt na partiële integratie het verband:

$$\begin{aligned} & \left\{ \int_0^{y^-} + \int_{y^+}^1 \right\} \cdot \{ G(x, y) \cdot L[U(x)] - U(x) \cdot L[G(x, y)] \} dx = \\ & = \left\{ \int_0^{y^-} + \int_{y^+}^1 \right\} \cdot [G(x, y) \cdot \left\{ \frac{d}{dx} \left(p(x) \cdot \frac{dU(x)}{dx} \right) + q(x) \cdot U(x) \right\} + \end{aligned}$$

$$\begin{aligned}
& - U(x) \cdot \left\{ \frac{d}{dx} \left(p(x) \cdot \frac{dG(x, y)}{dx} \right) + q(x) \cdot G(x, y) \right\} dx = \\
& = \left\{ G(x, y) \cdot p(x) \cdot \frac{dU(x)}{dx} - U(x) \cdot p(x) \cdot \frac{dG(x, y)}{dx} \right\}_0^1 + \\
& + p(x) \cdot \left\{ G(x, y) \cdot \frac{dU(x)}{dx} - U(x) \cdot \frac{dG(x, y)}{dx} \right\}_{y^- / y^+} . \quad (4)
\end{aligned}$$

Vanwege de veronderstelde continuïteit van $\frac{dU(x)}{dx}$, $p(x)$ en $G(x, y)$ is de term:

$$p(x) \cdot G(x, y) \cdot \frac{dU(x)}{dx} \Big|_{y^- / y^+} = 0 .$$

Aan de uitdrukking:

$$p(x) \cdot U(x) \cdot \frac{dG(x, y)}{dx} \Big|_{y^- / y^+} = p(y) \cdot U(y) \cdot \left\{ \frac{dG(x, y)}{dx} \Big|_{y^+} - \frac{dG(x, y)}{dx} \Big|_{y^-} \right\}$$

leggen we nu de volgende sprongvoorwaarde op:

$$\frac{dG(x, y)}{dx} \Big|_{y^+} - \frac{dG(x, y)}{dx} \Big|_{y^-} = \frac{-1}{p(y)} . \quad (5)$$

Uit deze eis vloeit voort dat $p(x)$ geen nulpunten mag hebben op $[0, 1]$. In de randpunten 0 en 1 kan deze eis worden afgezwakt, zodat zwakke singulariteiten zijn toegestaan. De oplossing moet dan wel aan extra voorwaarden voldoen o.a. continuïteit en begrensdsheid, resp. oneindig van kleinere orde dan $p(x)$ (zie pag. 324, 339^d).

Substitueren we in de integraalformule (4) de relatie (5) en in het linkerlid overeenkomstig (1) en (2) de rechterleden van L en G , dan wordt vanwege de continuïteit van G en f in $x = y$, de eindformule:

$$\begin{aligned}
U(y) & = \\
& = \left\{ p(x) \cdot \left\{ G(x, y) \cdot \frac{dU(x)}{dx} - U(x) \cdot \frac{dG(x, y)}{dx} \right\} \right\}_0^1 - \int_0^1 G(x, y) \cdot f(x) dx . \quad (6)
\end{aligned}$$

De zo geconstrueerde functie $G(x, y)$ is nog niet eenduidig bepaald. We kunnen nl. nog twee randvoorwaarden opleggen.

De keuze wordt echter beperkt vanwege de reeds aan $U(x)$ in (1) opgelegde randcondities. De randcondities op te leggen aan $L[G] = 0$, moeten nl. zo worden gekozen, dat in de uitdrukking:

$$\left\{ p(x) \cdot \left\{ G(x, y) \cdot \frac{dU(x)}{dx} - U(x) \cdot \frac{dG(x, y)}{dx} \right\} \right\}_0^1$$

in de eindformule (6), geen onbekende termen meer optreden.

Bijna altijd werken de volgende vuistregels:

- a) Zijn in het oorspronkelijke probleem homogene randcondities opgelegd, kies dan bij de differentiaalvergelijking $L[G] = 0$ de overeenkomstige homogene randcondities.
- b) Zijn in het oorspronkelijke probleem inhomogene randcondities opgelegd, kies dan bij de differentiaalvergelijking bij $L[G] = 0$ de overeenkomstige homogene randcondities.

Samenvatting.

- 1) $G(x, y)$ is een oplossing van de vergelijking:

$$\frac{d}{dx} \left\{ p(x) \cdot \frac{dG(x)}{dx} \right\} + q(x) \cdot G(x) = 0 \quad \text{voor } x \neq y \text{ op } [0, y), (y, l].$$

- 2) $G(x, y)$ is continu in $x = y$.

- 3) $\frac{dG(x, y)}{dx}$ vertoont in $x = y$ een sprong ter grootte:

$$\frac{-1}{p(y)} = \frac{dG(x, y)}{dx} \Big|_{x=y^+} - \frac{dG(x, y)}{dx} \Big|_{x=y^-}.$$

- 4) Kies de randcondities bij $L[G] = 0$, afhankelijk van de aan $L[U] = f$ opgelegde randcondities, zodanig dat er geen onbekende termen in:

$$G(x, y) \cdot \frac{dU(x)}{dx} - U(x) \cdot \frac{dG(x, y)}{dx} \Big|_0^1$$

optreden.

- 5) De 2-maal continu-differentieerbare oplossing volgt dan uit:

$$U(y) = \left\{ p(x) \cdot \left\{ G(x, y) \cdot \frac{dU(x)}{dx} - U(x) \cdot \frac{dG(x, y)}{dx} \right\} \right\}_0^1 - G(x, y) \cdot f(x) dx.$$

Opmerkingen.

- a) De Greenfunctie van een zelfgeadjungeerde differentiaalvergelijking is symmetrisch: $G(x, y) = G(y, x)$ (voor een bewijs zie Courant^{d)}).
- b) Er bestaat een zeer uitgebreide literatuur over Greenfuncties. Veel theorie en toepassingen o.a. in: Morse and Feshbach: *Methods of Theoretical Physics*. Part 1. pp. 791–895.
- c) Voor een behandeling met behulp van de theorie der distributies zie: Sigeru Mizohata. *The theory of partial differential equations*.
- d) Courant-Hilbert. *Methods of Mathematical Physics*, Vol. 1, p. 279. Ince. *Ordinary differential equations* (p. 351).
In deze boeken wordt ook behandeld hoe een niet zelfgeadjungeerde differentiaalvergelijking, in zelfgeadjungeerde vorm is te brengen.
- e) Met behulp van de in de fysica gebruikte δ -functie, welke is gedefinieerd door de eigenschappen:

$$\delta = 0, \quad x \neq 0, \quad \delta(0) = \infty, \quad \int_{-\infty}^{\infty} \delta(x).dx = 1$$

en van de zeefeigenschap

$$U(z) = \int_{-\infty}^{\infty} \delta(x - z).U(x).dx \quad (\text{geldt voor elke continue functie}),$$

blijkt dat de boven geconstrueerde functie $G(x, y)$ voor $x = y$, de oplossing is van $L(G(x, y) = \delta(x - y))$.

Hiermee kan de formule (6) formeel ook als volgt worden bepaald:

$$\begin{aligned} \text{Vermenigvuldig: } L(U) &= -f(r), \text{ links en rechts met } G(r, z) \text{ en} \\ L(G) &= \delta(r - z), \text{ links en rechts met } C(r). \end{aligned}$$

Trek de beide uitdrukkingen van elkaar af en integreer over het gebied $[0, 1]$. Met behulp van de zeefeigenschap, volgt uit formule (3) en de randcondities het gestelde.

Als toepassing lossen we het volgende warmte geleidingsprobleem op met behulp van Laplacetransformatie en Greenfunctie constructie:

$$\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial t} = q(x, t) \quad 0 \leq x \leq 1$$

met de beginconditie voor $t = 0$ is

$$u(x, 0) = hh(x), \quad \text{op } 0 \leq x \leq 1$$

en de lineaire niet homogene randcondities:

$$u(0, t) = r(t) \quad \text{voor } t > 0 \quad \text{en} \quad u(1, t) = s(t) \quad \text{voor } t > 0$$

($r(t)$ en $s(t)$ worden continu differentieerbaar verondersteld op $[0, \infty)$).

Met behulp van een Laplacetransformatie naar t , reduceren we het probleem tot een niet homogene gewone differentiaal vergelijking, met niet-homogene randcondities. De oplossing van dit stelsel bepalen we door eerst van de verwante homogene vergelijking met overeenkomstige homogene randcondities een Greenfunctie te construeren.

Noteren we de Laplace getransformeerden van de functies u , q , r en s weer met hoofdletters U , Q , R en S , dan laat het Laplacegetransformeerde niet homogene randwaardeprobleem zich schrijven als:

$$\begin{aligned} \frac{d^2 U}{dx^2} - p.U &= Q(x, p) + hh(x) & 0 \leq x \leq 1 \\ U(0, p) &= R(p) \quad \text{voor } x = 0 & \text{en} \quad U(1, p) = S(p) \quad \text{voor } x = 1. \end{aligned}$$

De Greenfunctie wordt bepaald uit de homogene differentiaalvergelijking:

$$\frac{d^2 G}{dx^2} - p.G = 0 \quad 0 \leq x \leq z \quad \text{en} \quad z \leq x \leq l.$$

Volgens de vuistregel, voldoen hierbij de homogene randcondities:

$$G(0, z) = 0 \quad \text{voor } x = 0 \quad \text{en} \quad G(1, z) = 0 \quad \text{voor } x = 1.$$

Stellen we: $p = w^2$, dan kunnen we de algemene oplossing:

$$G = A.\sinh(w.x) + B.\cosh(w.x)$$

als volgt uitwerken:

Voor $x \leq z$ en $G_1(0, z) = 0$ volgt dat:

$$G_1(x, z) = A.\sinh(w.x).$$

Voor $x \leq z$ en $G_2(1, z) = 0$ volgt:

$$G_2(x, z) = B.\sinh(w.(1-x)).$$

Voor $x = z$ is:

$$G_1(z, z) = G_2(z, z),$$

dus volgt dat:

$$A \cdot \sinh(w \cdot z) = B \cdot \sinh(w \cdot (1 - z)) .$$

Door deze gelijkheid te schrijven als:

$$\frac{A}{\sinh(w \cdot (1 - z))} = \frac{B}{\sinh(w \cdot z)} = C ,$$

vinden we dat:

$$G_1(x, z) = C \cdot \sinh(w \cdot x) \cdot \sinh(w \cdot (1 - z)) \quad \text{voor } x \leq z$$

$$G_2(x, z) = C \cdot \sinh(w \cdot z) \cdot \sinh(w \cdot (1 - x)) \quad \text{voor } z \leq x .$$

Uit deze betrekkingen berekenen we de afgeleiden voor $x = z$:

$$\frac{d}{dx} G_1(x, z) = -C \cdot w \cdot \cosh(w \cdot x) \cdot \sinh(w \cdot (1 - z)) \quad \text{voor } x = z^-$$

$$\frac{d}{dx} G_2(x, z) = C \cdot w \cdot \sinh(w \cdot z) \cdot \cosh(w \cdot (1 - x)) \quad \text{voor } x = z^+ .$$

Verdere uitwerking levert de sprong voorwaarde:

$$G_2'(z^+, z^+; p) - G_1'(z^-, z^-; p) = -C \cdot w \cdot \sinh(w) = -1 .$$

Hieruit volgt dat de constante

$$C = \frac{1}{w \cdot \sinh(w)} .$$

De Greenfunctie laat zich dus formuleren als:

$$G(x, z) = \begin{cases} \frac{\sinh[(1 - z) \cdot w] \cdot \sinh(x \cdot w)}{w \cdot \sinh(w)} = G_1(x, z) & x < z \\ \frac{\sinh(z \cdot w) \cdot \sinh[(1 - x) \cdot w]}{w \cdot \sinh(w)} = G_2(x, z) & x > z \end{cases}$$

Substitutie van $G(x, z)$ in formule (6) levert de volgende Laplacegetransformeerde oplossing van het niet homogene randwaardeprobleem:

$$U(p, z) = U1(p, z) + U2(p, z) + U3(p, z) + U4(p, z) .$$

Hierin is de bijdrage van de beginverdeling.

$$U1(z, p) = \int_0^z G_1(x, z).hh(x)dx + \int_z^1 G_2(x, z).hh(x)dx .$$

Voor de bijdragen van de niet homogene randcondities volgt:

$$U2(z, p) = R(p) \cdot \left. \frac{d}{dx} G(x, z) \right|_{x=0} \quad \text{en} \quad U3(z, p) = S(p) \cdot \left. \frac{d}{dx} G(x, z) \right|_{x=1} .$$

De bijdrage van het rechterlid in de oorsponkelijke differentiaalvergelijking wordt bepaald door:

$$U4(z, p) = \int_0^z G_1(x, z).Q(x, p)dx + \int_z^1 G_2(x, z).Q(x, p)dx .$$

Definiëren we een nieuw rechterlid voor $hh(x) + Q(x, p)$, dan is de bijdrage $U1(z, p) + U4(z, p)$ met één integratie te bepalen.

Om uit $U(z, p)$ de tijd afhankelijke oplossing $u(r, t)$ te vinden, moet nog de Laplace-inversie operatie worden uitgewerkt. Op de klassieke manier is dit een zeer moeizaam proces. Met behulp van de complexeomkeerformule moeten we eerst de functie $G(x, z; p)$ inverteren.

Een functietheoretische uitwerking levert de volgende formule op:

$$g(x, z, t) = L^{-1}\{G(x, z, p)\} = 2 \cdot \sum_{k=1}^{\infty} e^{-k^2 \pi^2 t} \sin(k\pi x) \sin(k\pi z) .$$

Door differentiatie naar x (voor $x = 0, 1$) volgt hieruit dat:

$$g_2(0, z, t) = L^{-1}\left\{\frac{d}{dx}G(0, z, p)\right\} = 2\pi \sum_{k=1}^{\infty} k e^{-k^2 \pi^2 t} \sin(k\pi z),$$

$$g_3(1, z, t) = L^{-1}\left\{\frac{d}{dx}G(1, z, p)\right\} = 2\pi \sum_{k=1}^{\infty} k (-)^{k+1} e^{-k^2 \pi^2 t} \sin(k\pi z) .$$

Met behulp van $g_2(0, z, t)$ en $r(t)$, kan de Laplaceinversie van $U2(z, p)$ met behulp van de convolutiestelling, als volgt termsgewijze worden bepaald:

$$u2(0, t) = \int_0^t g_2(0, z, \tau).r(t - \tau).d\tau =$$

$$= 2.\pi. \sum_{k=1}^{\infty} \sin(k.\pi.z)k \int_0^t .\exp(-k^2\pi^2\tau).r(t-\tau).d\tau .$$

Met behulp van $g_3(1, z, t)$ en $s(t)$ volgt op analoge wijze dat:

$$\begin{aligned} u_3(1, t) &= \int_0^t g_3(1, z, \tau).r(t-\tau).d\tau = \\ &= 2.\pi. \sum_{k=0}^{\infty} \sin(k.\pi.z).k.(-)^{k+1} \int_0^t .\exp(-k^2\pi^2\tau).s(t-\tau).d\tau . \end{aligned}$$

Met behulp van $g(x, z, t)$ en $hh(x)$ volgt na integratie over x , voor de bijdrage van de beginverdeling:

$$u_1(z, t) = \int_0^1 g(x, z, t).hh(x)dx .$$

Voor de bijdrage $u_4(z, t)$ is een nog uitvoeriger bewerking vereist.

Eerst de convolutieintegraal termgewijze integreren over τ en vervolgens de integratie over x uitwerken en daarna de bijdragen sommeren.

$$u_4(z, t) = \int_0^1 dx \int_0^t g(x, z, t-\tau).q(x, \tau)d\tau .$$

Een nadere beschouwing van de Fourier reeksontwikkelingen van $g_1(0, z, t)$ en $g_2(1, z, t)$ leert, dat deze slecht convergeren. De methode leent zich dus slecht voor numerieke uitwerking. De klassieke aanpak is in wezen beperkt tot een klasse van de functies $hh(x)$, $r(t)$ en $s(t)$ waarvoor de integraties over τ en x analytisch uitvoerbaar zijn. Indien dat niet het geval is, dan is er een termgewijze numerieke integratie over zowel τ als x nodig.

Deze omslachtige methode is te vermijden als de bijdragen $u_1(z, t), \dots, u_4(z, t)$, met behulp van numerieke Laplaceinversie direct uit de Laplacegetransformeerden: $U_1(z, p), \dots, U_4(z, p)$ kunnen worden bepaald.

Zo zijn $u_2(r, t)$ en $u_3(r, t)$ direct uit de Laplacegetransformeerde functies $U_2(r, t)$ en $U_3(r, t)$, met behulp van de Sub's LAPLACEQD en WAARDELAP, voor de alle gewenste z en t waarden te tabuleren.

Teneinde $u_1(r, t)$ en $u_4(r, t)$ echter ook op deze wijze te kunnen bepalen, moet er eerst een numerieke Greenfunctie-integratie worden uitgewerkt. We werken de methode uit voor een expliciet gegeven Laplacegetransformeerde rechterlidfunctie $Q(x, p)$.

Omdat per definitie, de Greenfunctie wel continu maar niet overal differentieerbaar is, moeten we bij de integratie zeer zorgvuldig te werk gaan. Maken we de aanname, dat de functie $Q(x, p)$ op $[0, 1]$ voldoende glad is en op elke gekozen interval $[x_{k-1}, x_k]$, $k = 0, 1, \dots, M$, de benadering:

$$\text{A:} \quad Q(x, p) = M(p) \cdot x + B(p), \quad \text{of}$$

$$\text{B:} \quad Q^*(x_{k-\frac{1}{2}}, p) = \{Q(x_k, p) + Q(x_{k-1}, p)\}/2,$$

mag worden gebruikt, dan is de integratie over het interval $[x_{k-1}, x_k]$ in beide gevallen analytisch uitvoerbaar.

We gebruiken de volgende notaties: staplengte $H = 1/N$,

$$x_k = k.H, \quad x_{k-1} = (k-1).H, \quad k = 0, 1, \dots, M \quad \text{en}$$

$$z = L.H \quad \text{met} \quad L = 0, 1, 2, \dots, M$$

$$Q(x_k, p) = Q(k, p) = M(p) \cdot k.H + B(p)$$

$$Q(x_{k-1}, p) = Q(k-1, p) = M(p) \cdot (k-1).H + B(p).$$

Hieruit volgt voor:

$$M(p) = \frac{(Q(k, p) - Q(k-1, p))}{H}.$$

Het blijkt dat we $B(p)$ niet expliciet nodig hebben.

Voor geval A, levert de substitutie van deze benaderingen in $U41(z, p)$ en $U42(z, p)$ de formules:

$$U41(z, p) = \frac{\sinh[(1-z)w]}{w \sinh(w)} \sum_{k=1}^L \int_{x_{k-1}}^{x_k} \sinh(wx) [M(p)x + B(p)] dx,$$

$$U42(z, p) = \frac{\sinh(wz)}{w \sinh(w)} \sum_{k=L+1}^M \int_{x_{k-1}}^{x_k} \sinh[(1-x)w] [M(p)x + B(p)] dx.$$

Uitwerking met partiële integratie levert de eindformules:

$$U41(LH, p) = \frac{\sinh[(1-HL)w]}{w^2 \sinh(w)} *$$

$$* \sum_{k=1}^L [Q(Hk, p) \cosh(Hkw) - Q(H\{k-1\}, p) \cosh[H(k-1)w] +$$

$$-\{\sinh(wHk) - \sinh(wH\{k-1\})\} \frac{\{Q(Hk, p) - Q(H\{k-1\}, p)\}}{Hw}$$

$$U42(LH, p) = -\frac{\sinh(wHL)}{w^2 \sinh(w)} *$$

$$* \sum_{k=L+1}^M [Q(Hk, p) \cosh((w(1-Hk)) - Q(H\{k-1\}, p) \cosh(w(1-H\{k-1\}))) + \\ + \{\sinh(w(1-Hk)) - \sinh(w(1-H\{k-1\}))\} \frac{\{Q(Hk, p) - Q(H\{k-1\}, p)\}}{Hw}]$$

De uitwerking van de Greenfunctieformules voor het geval B, waarbij

$$Q^*(k - \frac{1}{2}) = \frac{Q(H.k, p) + Q(H.(k-1), p)}{2}$$

constant is op elk deelinterval, verloopt analoog en leidt tot de formules:

$$U41(LH, p) = \frac{\sinh[(1-HL)w]}{w^2 \sinh(w)}$$

$$\sum_{k=1}^L Q^*(k - \frac{1}{2}) [\cosh(Hkw) - \cosh(H(k-1)w)],$$

$$U42(LH, p) = \frac{\sinh(wHL)}{w^2 \sinh(w)}$$

$$\sum_{k=L+1}^M Q^*(k - \frac{1}{2}) \{\cosh((w(1-Hk)) - \cosh(w(1-H(k-1))))\} .$$

De functie $Q(x_k, p)$, is geïmplementeerd als:

SUB RECHTERLID (MM, HH0, WWR, WWI, PR(), PI())

en de Greenfunctie als:

type A: SUB GREENFUNCTIE0(N%, M%, FA#, GA#, H#, AR#(), AI#())

type B: SUB GREENFUNCTIE(N%, M%, FA#, GA#, H#, AR#(), AI#())

In deze implementaties is overal, waar in de formules de complexe variabele $w = \sqrt{p}$ voorkomt, deze weer gesplitst in de respectieve reële- en imaginaire delen, opdat de bekende complexe functiealgorithmen weer kunnen worden gebruikt.

Om de nauwkeurigheid van de beide methoden te kunnen beoordelen, zullen

we de berekening uitvoeren voor een rechterlid in de differentiaalvergelijking, waarvoor de oplossing ook exact in de zelfde roosterpunten te tabuleren is. Dit is het geval voor: $q(x, t) = P1 \cdot \sin(\pi x)$.

Hieruit volgt dat:

$$Q(x, p) = P1 \cdot \sin(\pi x) / p .$$

Uitwerken van de overeenkomstige Greenfunctieintegralen:

$$U41(z, p) = \frac{P1 \cdot \sinh[(1-z) \cdot w]}{p \cdot w \cdot \sinh(w)} \int_0^z \sinh(x \cdot w) \cdot \sin(\pi x) dx$$

$$U42(z, p) = \frac{P1 \cdot \sinh(z \cdot w)}{p \cdot w \cdot \sinh(w)} \int_z^1 \sinh[(1-x) \cdot w] \cdot \sin(\pi x) dx$$

levert de Laplacegetransformeerde:

$$U4(z, p) = \frac{P1}{\pi^2} \cdot \sin(\pi \cdot z) \cdot \left\{ \frac{1}{p} - \frac{1}{\pi^2 + p} \right\} .$$

Na Lapaceinversie volgt:

$$u4(z, t) = \frac{P1}{\pi^2} \sin(\pi \cdot z) \cdot \{1 - \exp(-\pi^2 \cdot t)\} .$$

Voor een numerieke test is het dus aantrekkelijk om in de implementatie $P1 = \pi^2$ te kiezen, want dan wordt:

$$u4(z, \infty) = \sin(\pi \cdot z) .$$

Bij de uitwerking van de bovenstaande berekeningen is gebruik gemaakt van de hyperbolische betrekkingen:

$$\sinh(u + v) = \sinh(u) \cdot \cosh(v) + \cosh(u) \cdot \sinh(v)$$

$$\cosh(u + v) = \cosh(u) \cdot \cosh(v) + \sinh(u) \cdot \sinh(v)$$

en de onbepaalde integralen:

$$\int \exp(a \cdot x) \cdot \sin(b \cdot x) \cdot dx = \frac{\exp(a \cdot x)}{a^2 + b^2} \cdot [a \cdot \sin(b \cdot x) - b \cdot \cos(b \cdot x)]$$

$$\int \exp(a \cdot x) \cdot \cos(b \cdot x) \cdot dx = \frac{\exp(a \cdot x)}{a^2 + b^2} \cdot [b \cdot \sin(b \cdot x) + a \cdot \cos(b \cdot x)] .$$

De Greenfunctie volgens methode A) is geïmplementeerd als:

```
SUB GREENFUNCTIE0 (N, M, FA, GA, H, AR(), AI()).
```

Deze functie is op de schijf beschikbaar als GREENF0.SUB.

De implementatie van de Greenfunctie volgens methode B) is als volgt:

```
DEFINT I-N: A-H, O-Z
SUB GREENFUNCTIE (N, M, FA, GA, H, AR(), AI())
  DIM QR(M + 1), QI(M + 1)
  WR = 0: WI = 0: SHR = 0: SHI = 0: CHR = 0: CHI = 0
  SH1R = 0: SH1I = 0: CH1R = 0: CH1I = 0: PSR = 0: PSI = 0
  X11 = 0: Y11 = 0: X12 = 0: Y12 = 0: X14 = 0: Y14 = 0
  QCR = 0: QCI = 0: QC1R = 0: QC1I = 0: H0 = CDBL(M)
  PRINT "Loop Nr = ";
  FOR K = 0 TO N: SR = 0: SI = 0: SR1 = 0: SI1 = 0
  Y1 = CDBL(K) * FA
  U = GA * GA + Y1 * 1: W1 = GA / U: W2 = -Y1 / U
  CALL RECHTERLID(M, H0, W1, W2, QR(), QI())
  CALL CSQR(GA, Y1, WR, WI) 'W=SQR(GA + i Y1)
  CALL CSINHCOSH(WR, WI, SHR, SHI, CHR, CHI)
  'Z10 = SINH(W)
  CALL CMAAL(SHR, SHI, GA, Y1, PSR, PSI)
  'PSR + i PSI = P * SINH(W)
  PRINT K;
  FOR L = 0 TO M: ZL = H * CDBL(L): ZL1 = (1# - ZL)
  ZL1WR = ZL1 * WR: ZL1WI = ZL1 * WI
  CALL CSINHCOSH(ZL1WR, ZL1WI, SHR, SHI, CHR, CHI)
  CALL CDIV(SHR, SHI, PSR, PSI, X11, Y11)
  'Z11 = SH{(1-Z)W} / (P * SH(W))
  ZLWR = WR * ZL: ZLWI = WI * ZL
  CALL CSINHCOSH(ZLWR, ZLWI, SHR, SHI, CHR, CHI)
  CALL CDIV(SHR, SHI, PSR, PSI, X12, Y12)
  'Z12 = SINH(Z * W) (P * SINH(W))
  FOR I = 1 TO L 'G(z,x;p) x ≤ z
  XH = H * CDBL(I)
  XW = XH * WR: YW = XH * WI 'XW = H.I.W
  XH1 = H * CDBL(I - 1): X1W = XH1 * WR: Y1W = XH1 * WI
  CALL CSINHCOSH(XW, YW, SHR, SHI, CHR, CHI)
  CALL CSINHCOSH(X1W, Y1W, SH1R, SH1I, CH1R, CH1I)
  COR = CHR - CH1R: COI = CHI - CH1I
  QSR = (QR(I) + QR(I-1)) / 2#: QSI = (QI(I) + QI(I-1)) / 2#
  CALL CMAAL(COR, COI, QSR, QSI, QTR, QTI)
  SR = SR + QTR: SI = SI + QTI
```

```

NEXT I: CALL CMAAL(SR, SI, X11, Y11, S1R, S1I)
  SR = 0: SI = 0
FOR I = L + 1 TO M 'G(z,x;p) x ≥ z
  XH = 1#-H * CDBL(I)
  XW = XH * WR: YW = XH * WI'ZW=1-H.I.W
  XH1 = 1# - H * CDBL(I - 1)
  X1W = XH1 * WR: Y1W = XH1 * WI'ZW=1-H.I.W
  CALL CSINHCOSH(XW, YW, SHR, SHI, CHR, CHI)
  CALL CSINHCOSH(X1W, Y1W, SH1R, SH1I, CH1R, CH1I)
  COR = CHR - CH1R: COI = CHI - CH1I
  QSR = (QR(I) + QR(I-1)) / 2#: QSI = (QI(I) + QI(I-1)) / 2#
  CALL CMAAL(COR, COI, QSR, QSI, QTR, QTI)
  SR = SR + QTR: SI = SI + QTI
NEXT I: CALL CMAAL(SR, SI, X12, Y12, S2R, S2I)
AR(K, L) = S1R - S2R: AI(K, L) = S1I - S2I
SR = 0: SI = 0: SR1 = 0: SI1 = 0
NEXT L
NEXT K
END SUB

```

De Greenfunctie volgens de methode B) met de bijbehorende rechterlidfunctie, zijn uitgewerkt in het programma GREEN3.BAS (op de schijf).

In tegenstelling van wat men in eerste instantie zou verwachten, is nl. de methode A) om de volgende reden in het nadeel t.o.v. methode B).

De te implementeren formules in geval A), zijn meer complex en er zijn meer functie aanroepen nodig, met als gevolg een langere rekentijd. Meer essentieel echter is, dat bij kleinere stapgrootte (M groter) er bij de bepaling van: $M(p) = (Q(x_k, p) - Q(x_{k-1}, p))/2$ steeds meer cijferverlies gaat optreden, welke niet wordt gecompenseerd door de betere integratie. De aanpak is ook minder algemeen toepasbaar, want analytische integratie van de Greenfunctie geconstrueerd voor een differentiaalvergelijking van het cilindrsymmetrische type, is niet mogelijk.

De methode is wel goed bruikbaar in die gevallen, waar $Q(x, p)$ het karakter van een lineaire functie heeft en er met een klein aantal stappen M kan worden gewerkt.

De nauwkeurigheid van de met GREEN3.BAS berekende oplossing, voor een stel parameters N, M, FA, GA , gaan we beoordelen door deze voor $t = TT$ voldoende groot, te vergelijken met die van de exacte oplossing:

$$u(z, t) = \sin(\pi.z)\{1 - e^{-\pi^2 t}\}, \quad \text{voor } t = \infty \{u(z, \infty) = \sin(\pi.z)\}.$$

Voor een nauwkeurigheid van b.v. 3 cijfers kunnen we met $TT \leq 1$ volstaan, omdat voor $t = 1$ de factor $\exp(-\pi^2 \cdot 1) \approx 0.00005$ in het numerieke proces al

is te verwaarlozen.

Bij de vroegere Laplace-inversie-berekeningen hebben we op het interval $[0, 2.T]$, goede tot zeer goede resultaten gevonden voor: $FA = \pi/T \approx .5$, een afstand tot de imaginaire as: $GA \approx 1$ en N het aantal termen in de kettingbreuk tussen 40 en 80. De nauwkeurigheid van de berekende oplossing blijkt verder in belangrijke mate te worden bepaald door het aantal roosterpunten M , bij de numerieke integratie van de Greenfunctie. Goede resultaten worden gevonden voor $M \geq N$.

Het oplossen van een partiële differentiaalvergelijking met rechterlid met een Greenfunctie aanpak is een tijdrovende bezigheid. De bovengenoemde nauwkeurigheidstest voor de parameter-keuze: $FA = .47$, $G = .97$, $N = 70$ en $M = 100$, uitgevoerd volgens methode B), op een 20MH 80386 machine (met co-processor) vereiste een rekentijd van ruim 1,5 uur.

Teneinde het experimenteren met verschillende N waarden minder tijdrovend te maken, wordt de berekening eerst voor een zo groot mogelijke N bepaald en worden de berekende kettingbreukcoëfficiënten in de arrays $DR()$ en $DI()$, vervolgens opgeslagen in een random acces file NMFAGA.DAT met behulp van:

```
SUB RANDONCOMPLEXEMATRIXOPSLAG (NN, MM, PR(), PI(), L$)
  OPEN L$ FOR RANDOM AS #1 LEN = 2 * 8 * (MM + 2)
  FOR I = 0 TO NN: L = 1: FOR J = 1 TO MM + 1
    FIELD #1, 8 * J AS AA$, 8 * J AS BB$, 8 AS AB$, 8 AS AC$
    LSET AB$ = MKD$(PR(I, L-1)): LSET AC$ = MKD$(PI(I, L-1))
    L = L + 1
  NEXT J: PUT #1, I + 1:
NEXT I: CLOSE 1#
END SUB
```

Het uitlezen van deze file kan met behulp van:

```
SUB RANDONCOMPLEXEMATRIXINLEZEN (NN, MM, PR(), PI(), L$)
  OPEN L$ FOR RANDOM AS #1 LEN = 2 * 8 * (MM + 2)
  FOR I = 0 TO NN: GET #1, I + 1: L = 1
  FOR J = 1 TO MM + 1
    FIELD #1, 8 * J AS AA$, 8 * J AS BB$, 8 AS AB$, 8 AS AC$
    PR(I, L-1) = CVD(AB$): PI(I, L-1) = CVD(AC$): L = L + 1
  NEXT J
NEXT I: CLOSE 1#
END SUB
```

De parameters N , M , FA , GA worden opgeslagen in de sequential file NMFAGA. Het programma GREEN3.BAS is nl. zo gemaakt, dat opstarten met deze file- invoer mogelijk is en dus kan de berekening voor een kleinere N ,

zonder een lange rekentijd worden herhaald.

Greenfunctie constructies voor cilindrsymmetrische problemen.

De oplossing van de partiële differentiaalvergelijking:

$$\frac{1}{r} \cdot \frac{\partial}{\partial r} \left(r \cdot \frac{\partial c}{\partial r} \right) = \frac{\partial c}{\partial \tau} \quad a \leq r \leq 1, \quad \tau \geq 0 \quad (1)$$

met de beginconditie $c(r, \tau) = f(r)$ voor $\tau = 0$ en de randcondities

$$\frac{\delta c}{dr} = 0 \quad \text{voor } r = a \text{ en } r = 1 \quad (2)$$

met behulp van Laplace-transformatie en een Greenfunctie-constructie.

Laplacetransformatie naar τ levert:

$$\frac{d}{dr} \left(r \cdot \frac{dC}{dr} \right) - r \cdot w^2 \cdot C = -r \cdot f(r) \quad \text{met } p = w^2 \quad (3)$$

$$\text{voor } r = a \text{ en } r = 1 \text{ is } \frac{dC}{dr} = 0. \quad (4)$$

We lossen dit stelsel op met behulp van een Greenfunctie, van het verwante homogene stelsel:

$$\frac{d}{dr} \left(r \frac{dG}{dr} \right) - r \cdot w^2 \cdot G = 0, \quad \text{met } p = w^2 \text{ en voor } r = 0, 1 \text{ is } \frac{dG}{dr} = 0. \quad (5)$$

De oplossing van (5) met behulp van de Besselfuncties $I_0(r)$, $K_0(r)$ wordt:

$$G_1(r, z; w) = A \cdot I_0(w \cdot r) + B \cdot K_0(w \cdot r) \quad \text{voor } a \leq r \leq z \quad (7)$$

$$G_2(z, r; w) = C \cdot I_0(w \cdot r) + D \cdot K_0(w \cdot r) \quad \text{voor } z \leq r \leq 1. \quad (7a)$$

Substitutie van (7) en (7a) in de randvoorwaarden (5):

$$\frac{d}{dr} G_1(z, r; w) \Big|_{r=a} = w \cdot \{A \cdot I_1(w \cdot a) - B \cdot K_1(w \cdot a)\} = 0$$

$$\frac{d}{dr} G_2(z, r; w) \Big|_{r=1} = w \cdot \{C \cdot I_1(w \cdot 1) - D \cdot K_1(w \cdot 1)\} = 0$$

Dit levert respectievelijk

$$A = B \cdot \frac{K_1(w \cdot a)}{I_1(w \cdot a)}, \quad \text{en } C = D \cdot \frac{K_1(w)}{I_1(w)}.$$

Na substitutie van A en C in (7) en (7a) volgt:

$$G_1(r, z; w) = \frac{B}{I_1(w.a)} \cdot \{K_1(w.a).I_0(w.r) + I_1(w.a).K_0(w.r)\} \\ \text{voor } a \leq r \leq x \quad (8)$$

$$G_2(r, z; w) = \frac{D}{I_1(w)} \cdot \{K_1(w).I_0(w.r) + I_1(w).K_0(w.r)\} \\ \text{voor } x \leq r \leq 1. \quad (8a)$$

Voor $r = z$ is $G_1(z, z; w) = G_2(z, z; w)$ volgt dat:

$$\frac{B/I_1(w.a)}{K_1(w).I_0(w.z) + I_1(w).K_0(w.z)} = \\ = \frac{D/I_1(w)}{\{K_1(w.a).I_0(w.z) + I_1(w.a).K_0(w.z)\}} = E. \quad (9)$$

Met behulp hiervan volgt dat we (8) en (8a) kunnen schrijven als:

$$G_1() = E \cdot \{K_1(w.a).I_0(w.r) + I_1(w.a).K_0(w.r)\} \cdot \\ \cdot \{K_1(w).I_0(w.z) + I_1(w).K_0(w.z)\} \\ G_2() = E \cdot \{K_1(w).I_0(w.r) + I_1(w).K_0(w.r)\} \cdot \\ \cdot \{K_1(w.a).I_0(w.z) + I_1(w.a).K_0(w.z)\} \quad (10)$$

Met behulp van de sprongvoorwaarde

$$\frac{dG_2}{dr} - \frac{dG_1}{dr} = \frac{1}{r} \quad \text{voor } r = z$$

en de Besselformule:

$$I_v(z).K_{v+1}(z) + I_{v+1}(z).K_v(z) = 1/z \quad \{\text{Wronskiaan}\}$$

volgt dat

$$E = \frac{1}{\{I_1(w.a).K_1(w) - K_1(w.a).I_1(w)\}} \quad (11)$$

Substitutie van (10) en (11) in de vroeger gevonden formule:

$$C(z; w) = \int_a^z r.f(r).G_1(z, r; w).dr + \int_z^1 r.f(r).G_2(r, z; w).dr \quad (12)$$

levert als we de volgende afkortingen invoeren:

$$\begin{aligned}\Phi_1(r; w) &= \{K_1(w.a).I_0(w.r) + I_1(w.a).K_0(w.r)\} \\ \Phi_2(r; w) &= \{K_1(w).I_0(w.r) + I_1(w).K_0(w.r)\}\end{aligned}\quad (13)$$

de gezochte oplossing

$$\begin{aligned}C(z; w) &= \\ &= E. \left[\Phi_2(z; w). \int_a^z r.f(r). \Phi_1(r; w). dr + \Phi_1(z; w). \int_z^1 r.f(r). \Phi_2(r; w). dr \right].\end{aligned}\quad (14)$$

Op de klassieke manier laat zich hieruit met de complexe omkeerformule en gebruik van functietheorie (bepalen van de residuen in de polen) de Laplace-inversie uitwerken. We vinden dan de oplossingsvorm terug zo als afgeleid in het Hoofdstuk 14 met behulp van de separatie-methode.

We nemen hier echter formule (14) als uitgang van de numerieke Laplace-inversie-aanpak en gebruikmaking van de subs LAPACEQD en WAARDELAP.

Op het interval $[a, 1]$ maken we weer een staplengte verdeling in M equidistante punten en noteren op elk deelinterval $[r_{k-1}, r_k]$ de variabelen:

$$\begin{aligned}H &= r_k - r_{k-1} = (1 - a)/M, \quad k = 1, 2, \dots, M, \\ z_k &= k.H, \quad k = 1, 2, \dots, M, \quad z_L = L.H, \quad L = 1, 2, \dots, M.\end{aligned}\quad (15)$$

Kiezen we de staplengte H zo klein, dat op elk interval $[r_{k-1}, r_k]$ de rechterlid-functie $f(r)$, constant en gelijk aan:

$$f(r_{k-\frac{1}{2}}) = \{f(r_k) + f(r_{k-1})\}/2 \quad (16)$$

mag worden genomen, dan volgt dat:

$$\begin{aligned}C(z, w) &= \frac{\{K_1(w).I_0(w.z) + I_1(w).K_0(w.z)\}}{\{K_1(w.a).I_1(w) - I_1(w.a).K_1(w)\}} \cdot \\ &\cdot \sum_{k=1}^L (f_{k-\frac{1}{2}}) \int_{r_{k-1}}^{r_k} r. \{K_1(w.a).I_0(w.r) + I_1(w.a).K_0(w.r)\}. dr + \\ &+ \frac{\{K_1(w.a).I_0(w.z) + I_1(w.a).K_0(w.z)\}}{\{K_1(w.a).I_1(w) - I_1(w.a).K_1(w)\}} \cdot \\ &\cdot \sum_{k=L+1}^M (f_{k-\frac{1}{2}}) \int_{r_{k-1}}^{r_k} r. \{K_1(w).I_0(w.r) + I_1(w).K_0(w.r)\}. dr.\end{aligned}\quad (17)$$

Met behulp van de volgende Besselfunctie relaties (voor $v = 0$)

$$\int z^{v+1} \cdot I_v(z) \cdot dz = z^{v+1} \cdot I_{v+1}(z), \quad \int z^{v+1} \cdot K_v(z) \cdot dz = -z^{v+1} \cdot K_{v+1}(z)$$

$$\int z^{-v+1} \cdot I_v(z) \cdot dz = z^{-v+1} \cdot I_{v-1}(z), \quad \int z^{-v+1} \cdot K_v(z) \cdot dz = -z^{-v+1} \cdot K_{v-1}(z)$$

kan de integratie over elk deelinterval analytisch worden uitgewerkt:

$$C(z, w) = \frac{\{K_1(w) \cdot I_0(w \cdot z) + I_1(w) \cdot K_0(w \cdot z)\}}{\{I_1(w \cdot a) \cdot K_1(w) - K_1(w \cdot a) \cdot I_1(w)\}} \cdot \frac{1}{w} *$$

$$* \sum_{k=1}^L (f_{k-\frac{1}{2}}) \cdot r \cdot \{K_1(w \cdot a) \cdot I_1(w \cdot r) - I_1(w \cdot a) \cdot K_1(w \cdot r)\} \Big|_{r_{k-1}}^{r_k} +$$

$$+ \frac{\{K_1(w \cdot a) \cdot I_0(w \cdot z) + I_1(w \cdot a) \cdot K_0(w \cdot z)\}}{\{I_1(w \cdot a) \cdot K_1(w) - K_1(w \cdot a) \cdot I_1(w)\}} \cdot \frac{1}{w} *$$

$$* \sum_{k=L+1}^M (f_{k-\frac{1}{2}}) \cdot r \cdot \{K_1(w) \cdot I_1(w \cdot r) - I_1(w) \cdot K_1(w \cdot r)\} \Big|_{r_{k-1}}^{r_k} .$$

Substitutie van de variabelen $z = z_k$, r_k en r_{k-1} ($k = 1, 2, \dots, M$) overeenkomstig de relaties (16) in deze formule, levert dan direct de vorm welke is gebruikt voor de implementatie van: SUB GREENBESSELFUNCTIE.

```

DEFINT I-N: DEFDBL A-H, O-Z
SUB GREENBESSELFUNCTIE (N, M, FA, GA, R0, H, AR(), AI())
DIM QR(M + 1), QI(M + 1)
WR = 0: WI = 0: PRINT "Loop Nr = ";
FOR K = 0 TO N: SR = 0: SI = 0: SR1 = 0: SI1 = 0
  Y1 = CDBL(K) * FA
  CALL CSQR(GA, Y1, WR, WI) 'W=SQR(GA + i Y1)
  CALL RECHTERLID(R0, M, H, WR, WI, QR(), QI())
  CALL I1Z(WR, WI, XI1W, YI1W) 'ZI1W = I1(W)
  CALL K1Z(WR, WI, XK1W, YK1W) 'ZK1W = K1(W)
  X = R0 * WR: Y = R0 * WI
  CALL I1Z(X, Y, XI1WR0, YI1WR0) 'ZI1WR0 = I1(W.R0)
  CALL K1Z(X, Y, XK1WR0, YK1WR0) 'ZK1WR0 = K1(W.R0)
  CALL CMAAL(XI1W, YI1W, XK1WR0, YK1WR0, D1R, D1I) 'I1(W).K1(W.R0)
  CALL CMAAL(XI1WR0, YI1WR0, XK1W, YK1W, D2R, D2I) 'K1(W).I1(W.R0)
  XNOEMER = D1R - D2R: YNOEMER = D1I - D2I:
  PRINT K;
  FOR L = 0 TO M: ZL = H * CDBL(L) + R0
    ZLWR = ZL * WR: ZLWI = ZL * WI

```

```

CALL I0Z(ZLWR, ZLWI, XI0WZL, YI0WZL)          'ZI0WZL=I0(W.Z)
CALL CMAAL(XK1W, YK1W, XI0WZL, YI0WZL, T1R, T1I) 'K1(W)=I0(W.Z)
CALL K0Z(ZLWR, ZLWI, XK0WZL, YK0WZL) 'ZK0WZL=K0(W.Z)
CALL CMAAL(XK0WZL, YK0WZL, XI1W, YI1W, T2R, T2I) 'I1(W).K0(W.Z)
TER = T1R + T2R: TEI = T1I + T2I
CALL CDIV(TER, TEI, XNOEMER, YNOEMER, FAKTOR1R, FAKTOR1I)
CALL CMAAL(XK1WR0, YK1WR0, XI0WZL, YI0WZL, T1R, T1I)
CALL CMAAL(XI1WR0, YI1WR0, XK0WZL, YK0WZL, T2R, T2I)
TER = T1R + T2R: TEI = T1I + T2I
CALL CDIV(TER, TEI, XNOEMER, YNOEMER, FAKTOR2R, FAKTOR2I)
FOR I = 1 TO L: 'G(z, x; p) x ≤ z
  XH = H * CDBL(I) + R0:
  XIW = XH * WR: YIW = XH * WI'ZHW = H.I.W
  XH1 = H * CDBL(I - 1) + R0: X1IW = XH1 * WR: Y1IW = XH1 * WI
  CALL I1Z(XIW, YIW, XI1HIW, YI1HIW) 'I1(W.H.I.W)
  CALL K1Z(XIW, YIW, XK1HIW, YK1HIW)K1(W.H.I.W)
  CALL CMAAL(XK1WR0, YK1WR0, XI1HIW, YI1HIW, PROD1R, PROD1I)
  CALL CMAAL(XI1WR0, YI1WR0, XK1HIW, YK1HIW, PROD2R, PROD2I)
  SOM1R = XH * (PROD1R - PROD2R): SOM1I = XH * (PROD1I - PROD2I)
  CALL I1Z(X1IW, Y1IW, XI1H1IW, YI1H1IW) 'I1(W.H.I.W)
  CALL K1Z(X1IW, Y1IW, XK1H1IW, YK1H1IW)K1(W.H.I.W)
  CALL CMAAL(XK1WR0, YK1WR0, XI1H1IW, YI1H1IW, PROD1R, PROD1I)
  CALL CMAAL(XI1WR0, YI1WR0, XK1H1IW, YK1H1IW, PROD2R, PROD2I)
  SOM2R = XH1 * (PROD1R - PROD2R): SOM2I = XH1 * (PROD1I - PROD2I)
  SOMR = SOM1R - SOM2R: SOMI = SOM1I - SOM2I
  FIR = (QR(I) + QR(I - 1)) / 2#: FII = (QI(I) + QI(I - 1)) / 2#
  CALL CMAAL(SOMR, SOMI, FIR, FII, QTR, QTI)
  SR = SR + QTR: SI = SI + QTI
NEXT I: CALL CMAAL(SR, SI, FAKTOR1R, FAKTOR1I, S1R, S1I)
SR = 0: SI = 0
FOR I = L + 1 TO M 'G(z, x; p) x ≥ z
  XH = H * CDBL(I) + R0
  XIW = XH * WR: YIW = XH * WI'ZHW = H.I.W
  XH1 = H * CDBL(I - 1) + R0: X1IW = XH1 * WR: Y1IW = XH1 * WI1)
  CALL I1Z(XIW, YIW, XI1HIW, YI1HIW) 'I1(W.H.I.W)
  CALL K1Z(XIW, YIW, XK1HIW, YK1HIW)K1(W.H.I.W)
  CALL CMAAL(XK1W, YK1W, XI1HIW, YI1HIW, PROD1R, PROD1I)
  CALL CMAAL(XI1W, YI1W, XK1HIW, YK1HIW, PROD2R, PROD2I)
  SOM1R = XH * (PROD1R - PROD2R)
  SOM1I = XH * (PROD1I - PROD2I)
  CALL I1Z(X1IW, Y1IW, XI1H1IW, YI1H1IW) 'I1(W.H.I.W)
  CALL K1Z(X1IW, Y1IW, XK1H1IW, YK1H1IW)K1(W.H.I.W)
  CALL CMAAL(XK1W, YK1W, XI1H1IW, YI1H1IW, PROD1R, PROD1I)
  CALL CMAAL(XI1W, YI1W, XK1H1IW, YK1H1IW, PROD2R, PROD2I)
  SOM2R = XH1 * (PROD1R - PROD2R)

```

```

SOM2I = XH1 * (PROD1I - PROD2I)
SOMR = SOM1R - SOM2R: SOMI = SOM1I - SOM2I
FIR = (QR(I) + QR(I - 1)) / 2#: FII = (QI(I) + QI(I - 1)) / 2#
CALL CMAAL(SOMR, SOMI, FIR, FII, QTR, QTI)
SR = SR + QTR: SI = SI + QTI
NEXT I
CALL CMAAL(SR, SI, FAKTOR2R, FAKTOR2I, S2R, S2I)
AR(K, L) = S1R + S2R: AI(K, L) = S1I + S2I
SR = 0: SI = 0: SR1 = 0: SI1 = 0
NEXT L
NEXT K
END SUB

```

De numerieke Laplace-inversie, is verder op de standaard manier uitgewerkt in het programma GREENBES.BAS.

Vanwege de lange rekentijd voor N en M beide groot worden de gebruikte parameters en tussenresultaten weer opgeslagen in een de sequential file NM-GAFAR0 en de random file NMGAFAR0.DAT, waarmee vanuit het menu het programma later opnieuw kan worden gestart. De convergentie van deze Bessel-kettingbreuk-ontwikkeling is echter veel beter dan de in cartesische coördinaten behandelde Fourier-kettingbreuk ontwikkeling in GREEN3.BAS. Om de numerieke nauwkeurigheid van de methode te kunnen beoordelen hebben we, een in dezelfde roosterpunten en tijdstappen getabuleerde, op een andere manier bepaalde (exacte) oplossing van (3), voor een geschikt gekozen $f(r)$ nodig. Zo'n oplossing kan eventueel worden bepaald met behulp van elk van de programma's ANALYBES.BAS of BESSELD.F.BAS behandeld in Hoofdstuk 14. Voor een speciaal gekozen $f(r)$ volgt hier nog een andere aanpak.

Een speciale particuliere oplossing.

Voor de beginwaarde verdeling: $f(r) = r^2 - 2 \cdot \log(r)$, kunnen we ook zonder Greenfunctie constructie, een particuliere oplossing als volgt analytisch uitwerken. Het laat zich gemakkelijk verifiëren dat:

$$C_p(z, p) = \frac{4 - 2 \cdot p \cdot \log r + p \cdot r^2}{p^2}$$

een particuliere oplossing van (3) is.

De algemene oplossing is hiermee te schrijven als:

$$C(r; w) = A \cdot I_0(w \cdot r) + B \cdot K_0(w \cdot r) + \frac{4 - 2 \cdot p \cdot \log r + p \cdot r^2}{p^2}.$$

Deze oplossing moet nog voldoen aan de randcondities (4).

Uit de resulterende lineaire vergelijkingen:

$$A.w.I_1(w.a) - B.w.K_1(w.a) + \frac{-2.p/a + 2.a.p}{p^2} = 0$$

$$A.w.I_1(w) - B.w.K_1(w) = 0$$

volgt dat:

$$A = B.K_1(w)/I_1(w) ,$$

hierin is

$$B = \frac{[-2.p/a + 2.a.p].I_1(w)}{[K_1(w).I_1(w.a) - I_1(w).K_1(w.a)].w.p^2} .$$

Met behulp hiervan wordt de volgende oplossing van (3) gevonden:

$$C(r; w) = \frac{\{I_1(w).K_0(w.r) + K_1(w).I_0(w.r)\} 2.(1 - a^2)}{\{I_1(w.a).K_1(w) - K_1(w.a).I_1(w)\} a.w^3} + \frac{4 - 2.p.\log r + r^2.p}{p^2}$$

deze formule is gebruikt bij de implementatie van:

```

DEFINT I-N: DEFDBL A-H, O-Z
SUB LAPOPLOSSING (N, M, FA, GA, R0, H, AR(), AI())
  FAC = 2# * (1# - R0 * R0) / R0:
  PRINT "Loop Nr = ";
  FOR K = 0 TO N: Y1 = CDBL(K) * FA
    CALL CSQR(GA, Y1, WR, WI)           'W=SQR(GA + i Y1)
    CALL I1Z(WR, WI, XI1W, YI1W)       'ZI1W = I1(W)
    CALL K1Z(WR, WI, XK1W, YK1W)       'ZK1W = K1(W)
    X = R0 * WR: Y = R0 * WI
    CALL I1Z(X, Y, XI1WR0, YI1WR0)     'ZI1WR0 = I1(W.R0)
    CALL K1Z(X, Y, XK1WR0, YK1WR0)     'ZK1WR0 = K1(W.R0)
    CALL CMAAL(XI1W, YI1W, XK1WR0, YK1WR0, D1R, D1I)'I1(W).K1(W.Z)
    CALL CMAAL(XI1WR0, YI1WR0, XK1W, YK1W, D2R, D2I)'K1(W).I1(W.R0)
    XNOEM = D1R - D2R: YNOEM = D1I - D2I:
    CALL CMAAL(GA, Y1, WR, WI, XN, YN)
    CALL CMAAL(XN, YN, XNOEM, YNOEM, XNOEMER, YNOEMER)
    PRINT K; : ZL = R0
  FOR L = 0 TO M: ZLWR = ZL * WR: ZLWI = ZL * WI
    CALL I0Z(ZLWR, ZLWI, XI0WZL, YI0WZL) 'ZI0WZL=I0(W.Z)
    CALL CMAAL(XK1W, YK1W, XI0WZL, YI0WZL, T1R, T1I) 'K1(W).I0(W.Z)
    CALL K0Z(ZLWR, ZLWI, XK0WZL, YK0WZL) 'ZK0ZL=K0(W.Z)
    CALL CMAAL(XK0WZL, YK0WZL, XI1W, YI1W, T2R, T2I) 'I1(W).K0(W.Z)
    TER = (T1R + T2R) * FAC: TEI = (T1I + T2I) * FAC
    CALL CDIV(TER, TEI, XNOEMER, YNOEMER, FACTOR1R, FACTOR1I)
    X2 = -2# * LOG(ZL) + ZL * ZL
    PZR = GA * X2 + 4#: PZI = Y1 * X2

```

```

CALL CMAAL(GA, Y1, GA, Y1, XP2, YP2)
CALL CDIV(PZR, PZI, XP2, YP2, PARTR, PARTI)
AR(K, L) = -FACTOR1R + PARTR: AI(K, L) = -FACTOR1I + PARTI
ZL = ZL + H
NEXT L
NEXT K
END SUB

```

Deze sub is opgenomen in het programma GREXACT.BAS.

Een Greenfunctie uitwerking voor dezelfde differentiaalvergelijking:

$$\frac{1}{r} \cdot \frac{\partial}{\partial r} \left(r \cdot \frac{\partial c}{\partial r} \right) = \frac{\partial c}{\partial \tau} \quad a \leq r \leq 1, \quad \tau \geq 0$$

met de beginconditie

$$\text{voor } \tau = 0 \text{ is } c(r, \tau) = f(r)$$

en de randcondities:

$$\text{voor } r = a \text{ is } \frac{\partial c}{\partial r} = 0 \text{ en voor } r = 1 \text{ is } c = 0$$

verloopt op exact de zelfde wijze.

We vinden na het nodige rekenwerk de oplossing:

$$\begin{aligned}
C(z, w) = & \frac{\{-K_0(w) \cdot I_0(w.z) + I_0(w) \cdot K_0(w.z)\}}{\{I_1(w.a) \cdot K_0(w) + K_1(w.a) \cdot I_0(w)\}} \\
& \cdot \sum_{k=1}^L (f_{k-\frac{1}{2}}) \int_{r_{k-1}}^{r_k} r \cdot \{K_1(w.a) \cdot I_0(w.r) + I_1(w.a) \cdot K_0(w.r)\} \cdot dr + \\
& + \frac{\{K_1(w.a) \cdot I_0(w.z) + I_1(w.a) \cdot K_0(w.z)\}}{\{I_1(w.a) \cdot K_0(w) + K_1(w.a) \cdot I_0(w)\}} \\
& \cdot \sum_{L+1}^M (f_{k-\frac{1}{2}}) \cdot \int_{r_{k-1}}^{r_k} r \cdot \{-K_0(w) \cdot I_0(w.r) + I_0(w) \cdot K_0(w.r)\} \cdot dr .
\end{aligned}$$

Maken we op het interval $[a, 1]$ weer een staplengte verdeling in M equidistante punten en gebruiken we op elk interval $[r_{k-1}, r_k]$, de notaties:

$$H = r_k - r_{k-1} = (1 - a)/M, \quad k = 1, 2, \dots, M,$$

$$z_k = k.H, \quad k = 1, 2, \dots, M, \quad \text{en} \quad z_L = L.H, \quad L = 1, 2, \dots, M.$$

Met de aanname, dat op elk deelinterval voor de rechterlid-functie:

$$f(r_{k-\frac{1}{2}}) = \{f(r_k) + f(r_{k-1})\}/2$$

is constant geldt, volgt na integratie over elk deelinterval de oplossing:

$$\begin{aligned} C(z, w) = & \frac{\{-K_0(w).I_0(w.z) + I_0(w).K_0(w.z)\}}{\{I_1(w.a).K_0(w) + K_1(w.a).I_0(w)\}} \cdot \frac{1}{w} * \\ & * \sum_{k=1}^L (f_{k-\frac{1}{2}}).r. \{K_1(w.a).I_1(w.r) - I_1(w.a).K_1(w.r)\} \Big|_{r_{k-1}}^{r_k} + \\ & + \frac{\{K_1(w.a).I_0(w.z) + I_1(w.a).K_0(w.z)\}}{\{I_1(w.a).K_0(w) + K_1(w.a).I_0(w)\}} \cdot \frac{1}{w} * \\ & * \sum_{L+1}^M (f_{k-\frac{1}{2}}).r. \{-K_0(w).I_1(w.r) - I_0(w).K_1(w.r)\} \Big|_{r_{k-1}}^{r_k}. \end{aligned}$$

Met behulp hiervan zijn de SUBS GREENBESESELFUNCTIE1 en RECHTERLID1 geïmplementeerd welke de overeenkomstige subs in het programma GREENBES.BAS kunnen vervangen (ze zijn aanwezig in GREENBE).

Een particuliere oplossing voor de rechterlid-functie:

$$f(r) = -a^2 \cdot \log(r)/2 - (1 - r^2)/4$$

luit:

$$C_p = \frac{-1 - p.a^2 \cdot \log(r)/2 + p.(1 - r^2)/4}{p^2}.$$

Hiermee volgt voor de algemene oplossing:

$$C(r; w) = A.I_0(w.r) + B.K_0(w.r) + \frac{-1 - p.a^2 \cdot \log(r)/2 + p.(1 - r^2)/4}{p^2}.$$

Substitutie van $C(r; w)$ in de randcondities levert de lineaire vergelijkingen:

$$A.w.I_1(w.a) - B.w.K_1(w.a) = 0 \quad \text{en} \quad A.I_0(w) + B.K_0(w) = 1/p^2$$

waaruit we weer A en B kunnen bepalen.

Hiermee wordt de volgende oplossing gevonden:

$$C(r; w) = \frac{\{I_1(w.a).I_0(w.r) + K_1(w.a).K_0(w.r)\}}{\{I_1(w.a).K_0(w) + K_1(w.a).I_0(w)\}} +$$

$$+ \frac{-1 - p \cdot a^2 \cdot \log(r)/2 + p \cdot (1 - r^2)/4}{p^2} .$$

Deze oplossing weer geïmplementeerd als SUB LAPOPLOSSING en is beschikbaar als GEXACT op de schijf.

Deze sub kan weer de gelijknamige sub in GREXACT.BAS vervangen.

De boven behandelde Greenfunctie subs zijn ontwikkeld om te toetsen of met zo'n deels analytische en deels numerieke werkwijze, op iteratieve wijze de oplossing van een 3-dimensionaal convectief warmtetransport probleem kan worden bepaald.

CWI SYLLABI

- 1 Vacantiecursus 1984: *Hewet - plus wiskunde*. 1984.
- 2 E.M. de Jager, H.G.J. Pijls (eds.). *Proceedings Seminar 1981-1982. Mathematical structures in field theories*. 1984.
- 3 W.C.M. Kallenberg, et al. *Testing statistical hypotheses: worked solutions*. 1984.
- 4 J.G. Verwer (ed.). *Colloquium topics in applied numerical analysis, volume 1*. 1984.
- 5 J.G. Verwer (ed.). *Colloquium topics in applied numerical analysis, volume 2*. 1984.
- 6 P.J.M. Bongaarts, J.N. Buur, E.A. de Kerf, R. Martini, H.G.J. Pijls, J.W. de Roever. *Proceedings Seminar 1982-1983. Mathematical structures in field theories*. 1985.
- 7 Vacantiecursus 1985: *Variatierekening*. 1985.
- 8 G.M. Tuynman. *Proceedings Seminar 1983-1985. Mathematical structures in field theories, Vol.1 Geometric quantization*. 1985.
- 9 J. van Leeuwen, J.K. Lenstra (eds.). *Parallel computers and computations*. 1985.
- 10 Vacantiecursus 1986: *Matrices*. 1986.
- 11 P.W.H. Lemmens. *Discrete wiskunde: tellen, grafen, spelen en codes*. 1986.
- 12 J. van de Lune. *An introduction to Tauberian theory: from Tauber to Wiener*. 1986.
- 13 G.M. Tuynman, M.J. Bergvelt, A.P.E. ten Kroode. *Proceedings Seminar 1983-1985. Mathematical structures in field theories, Vol.2*. 1987.
- 14 Vacantiecursus 1987: *De personal computer en de wiskunde op school*. 1987.
- 15 Vacantiecursus 1983: *Complexe getallen*. 1987.
- 16 P.J.M. Bongaarts, E.A. de Kerf, P.H.M. Kersten. *Proceedings Seminar 1984-1986. Mathematical structures in field theories, Vol.1*. 1988.
- 17 F. den Hollander, H. Maassen (eds.). *Mark Kac seminar on probability and physics. Syllabus 1985-1987*. 1988.
- 18 Vacantiecursus 1988. *Differentierekening*. 1988.
- 19 R. de Bruin, C.G. van der Laan, J.R. Luyten, H.F. Vogt. *Publiceren met LATEX*. 1988.
- 20 R. van der Horst, R.D. Gill (eds.). *STATAL: statistical procedures in Algol 60, part 1*. 1988.
- 21 R. van der Horst, R.D. Gill (eds.). *STATAL: statistical procedures in Algol 60, part 2*. 1988.
- 22 R. van der Horst, R.D. Gill (eds.). *STATAL: statistical procedures in Algol 60, part 3*. 1988.
- 23 J. van Mill, G.Y. Nieuwland (eds.). *Proceedings van het symposium wiskunde en de computer*. 1989.
- 24 P.W.H. Lemmens (red.). *Bewijzen in de wiskunde*. 1989.
- 25 Vacantiecursus 1989: *Wiskunde in de Gouden Eeuw*. 1989.
- 26 G.G.A. Bäuerle et al. *Proceedings Seminar 1986-1987. Mathematical structures in field theories*. 1990.
- 27 Vacantiecursus 1990: *Getallentheorie en haar toepassingen*. 1990.
- 28 Vacantiecursus 1991: *Meetkundige structuren*. 1991.
- 29 A.G. van Asch, F. van der Blij. *Hoeken en hun Maat*. 1992.
- 30 M.J. Bergvelt, A.P.E. ten Kroode. *Proceedings seminar 1986-1987. Lectures on Kac-Moody algebras*. 1992.
- 31 Vacantiecursus 1992: *Systeemtheorie*. 1992.
- 32 F. den Hollander, H. Maassen (eds.). *Mark Kac seminar on probability and physics. Syllabus 1987-1992*. 1992.
- 33 P.W.H. Lemmens (ed.). *Meetkunde van kunst tot kunde, vroeger en nu*. 1992.
- 34 J.H. Kruizinga. *Toegepaste wiskunde op een PC*. 1992.

MC SYLLABI

- 1.1 F. Göbel, J. van de Lune. *Leergang besliskunde, deel 1: wiskundige basiskennis*. 1965.
- 1.2 J. Hemelrijk, J. Kriens. *Leergang besliskunde, deel 2: kansberekening*. 1965.
- 1.3 J. Hemelrijk, J. Kriens. *Leergang besliskunde, deel 3: statistiek*. 1966.
- 1.4 G. de Leve, W. Molenaar. *Leergang besliskunde, deel 4: Markovketens en wachttijden*. 1966.
- 1.5 J. Kriens, G. de Leve. *Leergang besliskunde, deel 5: inleiding tot de mathematische besliskunde*. 1966.
- 1.6a B. Dorhout, J. Kriens. *Leergang besliskunde, deel 6a: wiskundige programmering 1*. 1968.
- 1.6b B. Dorhout, J. Kriens, J.Th. van Lieshout. *Leergang besliskunde, deel 6b: wiskundige programmering 2*. 1977.
- 1.7a G. de Leve. *Leergang besliskunde, deel 7a: dynamische programmering 1*. 1968.
- 1.7b G. de Leve, H.C. Tijms. *Leergang besliskunde, deel 7b: dynamische programmering 2*. 1970.
- 1.7c G. de Leve, H.C. Tijms. *Leergang besliskunde, deel 7c: dynamische programmering 3*. 1971.
- 1.8 J. Kriens, F. Göbel, W. Molenaar. *Leergang besliskunde, deel 8: minimaxmethode, netwerkplanning, simulatie*. 1968.
- 2.1 G.J.R. Förch, P.J. van der Houwen, R.P. van de Riet. *Colloquium stabiliteit van differentieschema's, deel 1*. 1967.
- 2.2 L. Dekker, T.J. Dekker, P.J. van der Houwen, M.N. Spijker. *Colloquium stabiliteit van differentieschema's, deel 2*. 1968.
- 3.1 H.A. Lauwerier. *Randwaardeproblemen, deel 1*. 1967.
- 3.2 H.A. Lauwerier. *Randwaardeproblemen, deel 2*. 1968.
- 3.3 H.A. Lauwerier. *Randwaardeproblemen, deel 3*. 1968.
- 4 H.A. Lauwerier. *Representaties van groepen*. 1968.
- 5 J.H. van Lint, J.J. Seidel, P.C. Baayen. *Colloquium discrete wiskunde*. 1968.
- 6 K.K. Koksmá. *Cursus ALGOL 60*. 1969.
- 7.1 *Colloquium moderne rekenmachines, deel 1*. 1969.
- 7.2 *Colloquium moderne rekenmachines, deel 2*. 1969.
- 8 H. Bavinck, J. Grasman. *Relaxatietrillingen*. 1969.
- 9.1 T.M.T. Coolen, G.J.R. Förch, E.M. de Jager, H.G.J. Pijs. *Colloquium elliptische differentiaalvergelijkingen, deel 1*. 1970.
- 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.
- 10 J. Fabius, W.R. van Zwet. *Grondbegrippen van de waarschijnlijkheidsrekening*. 1970.
- 11 H. Bart, M.A. Kaashoek, H.G.J. Pijs, W.J. de Schipper, J. de Vries. *Colloquium halfalgebra's en positieve operatoren*. 1971.
- 12 T.J. Dekker. *Numerieke algebra*. 1971.
- 13 F.E.J. Kruseman Aretz. *Programmeren voor rekenautomaten; de MC ALGOL 60 vertaler voor de EL X8*. 1971.
- 14 H. Bavinck, W. Gautschi, G.M. Willems. *Colloquium approximatietheorie*. 1971.
- 15.1 T.J. Dekker, P.W. Hemker, P.J. van der Houwen. *Colloquium stijve differentiaalvergelijkingen, deel 1*. 1972.
- 15.2 P.A. Beentjes, K. Dekker, H.C. Hemker, S.P.N. van Kampen, G.M. Willems. *Colloquium stijve differentiaalvergelijkingen, deel 2*. 1973.
- 15.3 P.A. Beentjes, K. Dekker, P.W. Hemker, M. van Veldhuizen. *Colloquium stijve differentiaalvergelijkingen, deel 3*. 1975.
- 16.1 L. Geurts. *Cursus programmeren, deel 1: de elementen van het programmeren*. 1973.
- 16.2 L. Geurts. *Cursus programmeren, deel 2: de programmeertaal ALGOL 60*. 1973.
- 17.1 P.S. Stobbe. *Lineaire algebra, deel 1*. 1973.
- 17.2 P.S. Stobbe. *Lineaire algebra, deel 2*. 1973.
- 17.3 N.M. Temme. *Lineaire algebra, deel 3*. 1976.
- 18 F. van der Blij, H. Freudenthal, J.J. de Jongh, J.J. Seidel, A. van Wijngaarden. *Een kwart eeuw wiskunde 1946-1971, syllabus van de vakantiecursus 1971*. 1973.
- 19 A. Hordijk, R. Potharst, J.Th. Runnenburg. *Optimaal stoppen van Markovketens*. 1973.
- 20 T.M.T. Coolen, P.W. Hemker, P.J. van der Houwen, E. Slagt. *ALGOL 60 procedures voor begin- en randwaardeproblemen*. 1976.
- 21 J.W. de Bakker (red.). *Colloquium programmacorrectheid*. 1975.
- 22 R. Helmers, J. Oosterhoff, F.H. Ruymgaart, M.C.A. van Zuylen. *Asymptotische methoden in de toetsingstheorie; toepassing van naburigheid*. 1976.
- 23.1 J.W. de Roever (red.). *Colloquium onderwerpen uit de biomathematica, deel 1*. 1976.
- 23.2 J.W. de Roever (red.). *Colloquium onderwerpen uit de biomathematica, deel 2*. 1977.
- 24.1 P.J. van der Houwen. *Numerieke integratie van differentiaalvergelijkingen, deel 1: eenstapsmethoden*. 1974.
- 25 *Colloquium structuur van programmeertalen*. 1976.
- 26.1 N.M. Temme (ed.). *Nonlinear analysis, volume 1*. 1976.
- 26.2 N.M. Temme (ed.). *Nonlinear analysis, volume 2*. 1976.
- 27 M. Bakker, P.W. Hemker, P.J. van der Houwen, S.J. Polak, M. van Veldhuizen. *Colloquium discretiseringsmethoden*. 1976.
- 28 O. Diekmann, N.M. Temme (eds.). *Nonlinear diffusion problems*. 1976.
- 29.1 J.C.P. Bus (red.). *Colloquium numerieke programmatuur, deel 1A, deel 1B*. 1976.
- 29.2 H.J.J. te Riele (red.). *Colloquium numerieke programmatuur, deel 2*. 1977.
- 30 J. Heering, P. Klint (red.). *Colloquium programmeeromgevingen*. 1983.
- 31 J.H. van Lint (red.). *Inleiding in de coderingstheorie*. 1976.
- 32 L. Geurts (red.). *Colloquium bedrijfssystemen*. 1976.
- 33 P.J. van der Houwen. *Berekening van waterstanden in zeeën en rivieren*. 1977.
- 34 J. Hemelrijk. *Oriënterende cursus mathematische statistiek*. 1977.
- 35 P.J.W. ten Hagen (red.). *Colloquium computer graphics*. 1978.
- 36 J.M. Aarts, J. de Vries. *Colloquium topologische dynamische systemen*. 1977.
- 37 J.C. van Vliet (red.). *Colloquium capita datastructuren*. 1978.
- 38.1 T.H. Koorwinder (ed.). *Representations of locally compact groups with applications, part I*. 1979.
- 38.2 T.H. Koorwinder (ed.). *Representations of locally compact groups with applications, part II*. 1979.
- 39 O.J. Vrieze, G.L. Wanrooy. *Colloquium stochastische spelen*. 1978.
- 40 J. van Tiel. *Convexe analyse*. 1979.
- 41 H.J.J. te Riele (ed.). *Colloquium numerical treatment of integral equations*. 1979.
- 42 J.C. van Vliet (red.). *Colloquium capita implementatie van programmeertalen*. 1980.
- 43 A.M. Cohen, H.A. Wilbrink. *Eindige groepen (een inleidende cursus)*. 1980.
- 44 J.G. Verwer (ed.). *Colloquium numerical solution of partial differential equations*. 1980.
- 45 P. Klint (red.). *Colloquium hogere programmeertalen en computerarchitectuur*. 1980.
- 46.1 P.M.G. Apers (red.). *Colloquium databankorganisatie, deel 1*. 1981.
- 46.2 P.G.M. Apers (red.). *Colloquium databankorganisatie, deel 2*. 1981.
- 47.1 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60: general information and indices*. 1981.
- 47.2 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 1: elementary procedures; vol. 2: algebraic evaluations*. 1981.
- 47.3 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 3A: linear algebra, part I*. 1981.
- 47.4 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 3B: linear algebra, part II*. 1981.
- 47.5 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 4: analytical evaluations; vol. 5A: analytical problems, part I*. 1981.
- 47.6 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 5B: analytical problems, part II*. 1981.
- 47.7 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 6: special functions and constants; vol. 7: interpolation and approximation*. 1981.
- 48.1 P.M.B. Vitányi, J. van Leeuwen, P. van Emde Boas (red.). *Colloquium complexiteit en algoritmen, deel 1*. 1982.
- 48.2 P.M.B. Vitányi, J. van Leeuwen, P. van Emde Boas (red.). *Colloquium complexiteit en algoritmen, deel 2*. 1982.
- 49 T.H. Koorwinder (ed.). *The structure of real semisimple Lie groups*. 1982.
- 50 H. Nijmeijer. *Inleiding systeemtheorie*. 1982.
- 51 P.J. Hoogendoorn (red.). *Cursus cryptografie*. 1983.

