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

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

MC SYLLABUS 29.1b

J.C.P. BUS (RED.)

COLLOQUIUM NUMERIEKE PROGRAMMATUUR

DEEL 1b

MATHEMATISCH CENTRUM AMSTERDAM 1976

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

ISBN 90 6196 128 9

Inhoud

Inhoud

4. SPECIALE FUNCTIES EN FOURIERTRANSFORMATIE

4.1. Speciale functies

door N.M. Temme
(Mathematisch Centrum)

4.1.1. Integralen die geschreven kunnen worden als speciale functies

Bij het berekenen van een integraal kan het verstandig zijn om na te gaan of de integraal uitgedrukt kan worden in speciale functies, waarvoor in de programmatheken algoritmen zijn opgenomen. Een rechtstreekse berekening via de integratieprocedures leidt doorgaans tot minder efficiënte methodes. In deze sectie worden enkele typen integralen behandeld met vermelding van de relatie tot de speciale functies en verwijzing naar de diverse programmatheken. Integratieprocedures zullen hier niet aan de orde komen. Het hier gegeven overzicht is niet volledig, maar vermeldt enkele in de praktijk vaak voorkomende gevallen. Voor meer informatie verwijzen we naar ABRAMOWITZ & STEGUN [1964], MAGNUS et al. [1966] en GRADSHTEYN & RYSHIK [1965]. We gebruiken voor de notatie van speciale functies die van ABRAMOWITZ & STEGUN [1964].

De subsecties 4.1.1.1, 4.1.1.2, 4.1.1.3 en 4.1.1.4 komen overeen met de hoofdstukken 5, 6, 7 en 26 in het Handbook. De functies uit deze hoofdstukken worden ook voornamelijk als integralen gedefinieerd. Andere speciale functies worden vaak via een differentiaalvergelijking gedefinieerd. Er zijn dan overigens meestal wel integraalrepresentaties, maar die worden hier verder niet besproken.

Er worden in dit hoofdstuk geen numerieke voorbeelden gegeven. Het aanroepen van de procedures en subroutines geeft namelijk vrijwel geen problemen. Wel is doorgaans kort aangegeven waar grote of kleine parameters de berekening kunnen verstoren.

4.1.1.1. Exponentiële en verwante integralen

TABEL 1

			IMSL	NAG	NUMAL
(4.1.1.1.1)	$\int_x^\infty (e^{-t}/t)dt$	$= E_1(x)$	MMDEI	S13AA A/F	EI
(4.1.1.1.2)	$\int_x^\infty e^{-t}\, t^{-n}\, dt$	$= E_n(x)$	-	-	ENX
(4.1.1.1.3)	$-\int_{-x}^\infty (e^{-t}/t)dt$	$=$	MMDEI	-	EI
	$\int_{-\infty}^{x} (e^{t}/t)dt$	$= Ei(x)$			
(4.1.1.1.4)	$\int_1^\infty t^n\, e^{-xt}\, dt$	$= \alpha_n(x)$	-	-	EI ALPHA
(4.1.1.1.5)	$\int_0^x (\sin t)/t\, dt$	$= Si(x)$	-	S13AB A/F	SINCOSINT
(4.1.1.1.6)	$\int_0^x (\cos t - 1)/t\, dt$	$= Ci(x) - \gamma - \ell n\, x$	-	S13AC A/F	SINCOSINT
(4.1.1.1.7)	$\int_0^\infty e^{-xt}(t^2+1)^{-1}\, dt$	$=$	-	-	SINCOSFG
	$\int_0^\infty (\sin t)/(x+t)dt$	$= f(x)$			SINCOSFG
(4.1.1.1.8)	$\int_0^\infty t\, e^{-xt}(t^2+1)^{-1}\, dt$	$=$	-	-	SINCOSFG
	$\int_0^\infty (\cos t)/(x+t)dt$	$= g(x)$			SINCOSFG

Opmerkingen

1. De procedure ENX voor $E_n(x)$ berekent een array e[n] = $E_n(x)$, $n = n_1,\ldots,n_2$, $0 \le n_1 \le n_2$.

2. Om overflow/underflow voor grote x te voorkomen komen in NUMAL voor $E_n(x)$ twee procedures voor: ter berekening van $E_n(x)$ en van $e^x E_n(x)$ respectievelijk. Dezelfde opmerking geldt voor MMDEI uit IMSL: naar verkiezing kan Ei(x) of e^{-x}Ei(x) worden berekend.

3. Het verband tussen Ei(x) en $E_1(x)$ is

$$E_1(x) = -Ei(x), \qquad x \in \mathbb{R}, \quad x \neq 0. \tag{4.1.1.1.9}$$

De integralen in (4.1.1.1.3) zijn (voor positieve x) Cauchy-hoofdwaarde integralen. $E_1(x)$ wordt (via (4.1.1.1.1)) meestal alleen voor $x > 0$ beschouwd, net als $E_n(x)$, $\alpha_n(x)$, Ci(x), f(x) en g(x).

4. De functies in (4.1.1.1.4) worden in een array afgeleverd. De functies $\alpha_n(x)$ zijn ook verkrijgbaar via de incomplete gammafunctie (zie 4.1.1.2). Het verband is

$$\alpha_n(x) = \Gamma(n+1,x)/x^{n+1}.$$

5. Het verband tussen f, g en Si(x), Ci(x) is als volgt:

$$\begin{aligned} f(x) &= Ci(x) \sin x - [Si(x) - \tfrac{1}{2}\pi] \cos x, \\ g(x) &= -Ci(x) \cos x - [Si(x) - \tfrac{1}{2}\pi] \sin x. \end{aligned} \tag{4.1.1.1.10}$$

Uit (4.1.1.1.7) en (4.1.1.1.8) volgt dat voor positief argument f en g monotoon dalende positieve functies zijn met $f(0) = \frac{1}{2}\pi$ en $g(x) \to +\infty$ voor $x \downarrow 0$. Inversie van (4.1.1.1.10) geeft een representatie van Ci en Si die vooral voor grote x duidelijk aangeeft hoe het gedrag van Ci en Si is:

$$\begin{aligned} Si(x) &= \tfrac{1}{2}\pi - f(x) \cos x - g(x) \sin x, \\ Ci(x) &= \qquad f(x) \sin x - g(x) \cos x. \end{aligned} \tag{4.1.1.1.11}$$

6. MMDEI uit IMSL is overgenomen uit de programmatheek FUNPACK.

7. Voor de functies in (4.1.1.1.2) en (4.1.1.1.4) kunnen op eenvoudige wijze recurrente betrekkingen worden opgesteld, waarvan de voorwaartse vorm voor α_n stabiel is. De stabiliteit van die van E_n is echter gecompliceerder.

Voorbeeld. Beschouw voor positieve waarden van x de integraal

$$(4.1.1.1.12)\quad \int_0^\infty (\sin xt)/(t^2+1)\ dt.$$

Deze integraal kan als functie van x uitgedrukt worden in de functies E_1 en Ei, met als resultaat

$$(4.1.1.1.13)\quad \tfrac{1}{2}[e^x E_1(x) + e^{-x}Ei(x)] = \tfrac{1}{2}[-e^x Ei(-x) + e^{-x}Ei(x)].$$

Hiervoor kunnen de programmatheken IMSL en NUMAL gebruikt worden.

a. IMSL. Roep twee keer MMDEI aan in de volgende vorm:

```
(4.1.1.1.14) p = - MMDEI(3,-x,n),   q = MMDEI(3,x,n).
```

Dan wordt de integraal (4.1.1.1.12) berekend door $\frac{1}{2}(p+q)$. De input parameter met waarde 3 geeft aan dat rechtstreeks het produkt van Ei en de exponentiële functie wordt berekend. (Als deze parameter de waarde 1 krijgt dan wordt Ei berekend; bij 2 is het resultaat E_1.) De output parameter n in (4.1.1.1.14) geeft een indicatie voor mogelijke storingen.

b. NUMAL. Via NUMAL wordt (4.1.1.1.12) berekend door $\frac{1}{2}(p+q)$ met

```
(4.1.1.1.15) p := - exp(x) * EI(-x),  q := exp(-x) * EI(x).
```

In dit geval wordt de functiewaarde achteraf met de exponentiële functie vermenigvuldigd. Een gevolg hiervan is dat de berekeningen stuklopen voor extreem hoge waarden van x. De grens ligt bij die x-waarde waarvoor exp(±x) overflow/underflow geeft. De IMSL-methode heeft hier geen hinder

van; in dat geval ligt de grens veel verder, namelijk waar 1/x underflow geeft.

Voor kleine waarden van x moeten de berekeningen zowel via IMSL als NUMAL gewantrouwd worden. De functies E_1 en Ei zijn voor x = 0 niet gedefinieerd, terwijl de integraal, zoals men eenvoudig verifieert voor x = 0 de waarde 0 oplevert. Door de functies E_1 en Ei wat nader te bestuderen ziet men dat ze voor $x \downarrow 0$ een logaritmisch gedrag vertonen:

$$Ei(x) = \ln x + O(1), \qquad E_1(x) = -\ln x + O(1),$$

zodat bij het samennemen van p en q uit (4.1.1.1.14) en (4.1.1.1.15) dominante singuliere termen exact tegen elkaar dienen weg te vallen. Bij de berekening zal dus een grote relatieve fout ontstaan. Indien men (4.1.1.1.13) met grote nauwkeurigheid wil evalueren voor kleine x kan men beter andere representaties gebruiken voor E_1 en Ei, bijvoorbeeld

$$E_1(x) = -\gamma - \ln x - \sum_{n=1}^{\infty} (-x)^n/(nn!),$$

$$Ei(x) = \gamma + \ln x + \sum_{n=1}^{\infty} x^n/(nn!).$$

Vermenigvuldigd met de exponentiële functies kunnen de termen nu zodanig samengenomen worden dat een goede representatie van (4.1.1.1.13) verkregen wordt, die goed konvergeert voor kleine x.

4.1.1.2. Integralen in verband met de gammafunctie

TABEL 2

		IMSL	NAG	NUMAL
(4.1.1.2.1)	$\int_0^\infty t^{x-1} e^{-t}\,dt = \Gamma(x)$	MGAMMA	S14AA A/F	GAMMA
	$\int_0^1 t^{a-1}(1-t)^{b-1}dt$	via gammafunctie		
(4.1.1.2.2)	$\int_0^\infty t^{a-1}(1+t)^{-a-b}dt = B(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$			
	$2\int_0^{\frac{1}{2}\pi} (\sin t)^{2a-1}(\cos t)^{2b-1}dt$			
(4.1.1.2.3)	$\int_0^x t^{a-1} e^{-t}\,dt = \gamma(a,x)$	MDGAM	-	INCOMGAM
(4.1.1.2.4)	$\int_x^\infty t^{a-1} e^{-t}\,dt = \Gamma(a,x)$	-	-	INCOMGAM
(4.1.1.2.5)	$\int_0^x t^{a-1}(1-t)^{b-1}dt = B_x(a,b)$	MDBETA	-	INCBETA

Opmerkingen

1. In IMSL en NUMAL komen tevens routines voor ter berekening van $\ln \Gamma(x)$, $x > 0$, respectievelijk aangeduid door MLGAMMA en LOGGAMMA.

2. Via MDGAM wordt niet de incomplete gammafunctie $\gamma(a,x)$ berekend, maar de ratio $\gamma(a,x)/\Gamma(a)$.

3. Via MDBETA en INCBETA wordt niet de incomplete betafunctie $B_x(a,b)$ berekend, maar de ratio $B_x(a,b)/B(a,b)$.

4. Voor het berekenen van een rij $B_x(a+i,b)$ of $B_x(a,b+i)$, $i = 0,1,\ldots,n$, kunnen in NUMAL de procedures IBPPLUSN, IBQPLUSN gebruikt worden.

Voorbeeld. Bereken voor positieve waarden van a, b en x de integraal

$$(4.1.1.2.6) \quad \int_0^x t^{a-1}(1+t)^{-a-b}\,dt.$$

Deze integraal kan worden uitgedrukt in de incomplete betafunctie

$$(4.1.1.2.7) \quad B_{x/(1+x)}(a,b).$$

Voor deze functie komen in IMSL en NUMAL routines voor.

a. IMSL. De berekening verloopt via MDBETA en vermenigvuldiging met B(a,b):

```
CALL MDBETA(x/(1+x),a,b,p,n),
```

waarin de output parameter p de waarde van $B_{x/(1+x)}(a,b)/B(a,b)$ krijgt en de output parameter n een indicatie geeft voor mogelijke storingen. Het resultaat voor de evaluatie van (4.1.1.2.7) wordt

$$(4.1.1.2.8) \quad p * \Gamma(a) * \Gamma(b)/\Gamma(a+b),$$

waarin de gammafuncties met MGAMMA berekend kunnen worden.

b. NUMAL. De berekening verloopt via de functieprocedure INCBETA. Het resultaat is

(4.1.1.2.9) INCBETA(x/(1+x),a,b,2 ↑ (-46)) * Γ(a) * Γ(b) / Γ(a+b),

waarin de gammafuncties via GAMMA berekend kunnen worden.

Voor grote waarden van de parameters a en b kunnen zowel in het IMSL- als NUMAL-geval moeilijkheden ontstaan, verzoorzaakt door de grote argumenten in de gammafuncties in (4.1.1.2.8) en (4.1.1.2.9). Veronderstel dat

(4.1.1.2.10) a = 200, b = ½, x = 10.

Dan veroorzaken Γ(a) en Γ(a+b) overflow, hoewel het quotiënt (waar het hier om te doen is) representeerbaar is op de machine. Een quotiënt van gammafuncties kan voor grote argumenten echter beter worden uitgerekend via

$$\Gamma(x)/\Gamma(y) = \exp(\ell n\,\Gamma(x) - \ell n\,\Gamma(y)),$$

via MLGAMMA en LOGGAMMA uit IMSL en NUMAL respectievelijk. In deze representatie kan wel wat cijferverlies ontstaan, maar overflow zal niet zo gauw voorkomen. Ook kan zo'n quotient, door de recurrente betrekking

$$\frac{\Gamma(x)}{\Gamma(y)} = \frac{(x-1)\,\Gamma(x-1)}{(y-1)\,\Gamma(y-1)}$$

te gebruiken, zonder gevaar voor overflow worden berekend.

Hiermee zijn de moeilijkheden voor (4.1.1.2.8) en (4.1.1.2.9) echter niet opgelost. Bij de uitgang van procedure INCBETA wordt namelijk door B(a,b) gedeeld (waarbij de gammafuncties rechtstreeks worden uitgerekend) om tot de ratio te komen, zodat binnen de procedure overflow zal ontstaan. Ook deze klip kan omzeild worden door de NUMAL-procedure IBPPLUSN te gebruiken, bijvoorbeeld als volgt. Neem het geval met de waarden in (4.1.1.2.10). Een aanroep

IBPPLUSN(10/11,1,·5,199,2 ↑ (-46),i),

waarin i een output array is, i[0 : 199], en de integraal in (4.1.1.2.6) wordt uiteindelijk

i[199] * Γ(½) * exp(ℓn(Γ(200)) - ℓn(Γ(200·5))).

TABEL 3

		ACCULIB	IMSL	NAG	NUMAL
(4.1.1.3.1)	$\frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,dt = \mathrm{erf}(x)$	-	MERF	S15AE A/F	ERROR-FUNCTION
(4.1.1.3.2)	$\frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2}\,dt = \mathrm{erfc}(x)$	ERFC	MERFC	S15AD A/F	ERROR-FUNCTION
(4.1.1.3.3)	$\frac{i}{\pi} \int_{-\infty}^{\infty} \frac{e^{-t^2}}{z-t}\,dt = w(z)$	W OF Z	-	-	-
(4.1.1.3.4)	$e^{-x^2} \int_0^x e^{t^2}\,dt = D(x)$ (Dawson)	-	MMDAW	-	-
(4.1.1.3.5)	$\int_0^x \cos(\frac{1}{2}\pi\, t^2)dt = C(x)$	-	-	S20AB A/F	FRESNEL
(4.1.1.3.6)	$\int_0^x \sin(\frac{1}{2}\pi\, t^2)dt = S(x)$	-	-	S20AA A/F	FRESNEL
(4.1.1.3.7)	$\int_0^\infty t^{-\frac{1}{2}}\, e^{-at}/(t^2+b^2)dt = \frac{\pi}{b}\sqrt{\frac{2}{b}}\, F\left(\sqrt{\frac{2ab}{\pi}}\right)$	via W OF Z	-	-	FG
(4.1.1.3.8)	$\int_0^\infty t^{\frac{1}{2}}\, e^{-at}/(t^2+b^2)dt = \pi\sqrt{\frac{2}{b}}\, G\left(\sqrt{\frac{2ab}{\pi}}\right)$	via W OF Z	-	-	FG

Opmerkingen

1. De functies erfc(x) en w(z) komen in ACCULIB voor in ALGOL en FORTRAN. De parameter z van w(z) is complex. In de physica wordt w(z) de plasma dispersie functie genoemd. Alle andere functies uit deze subsectie kunnen in w uitgedrukt worden. Enkele voorbeelden zijn:

$$
\begin{aligned}
&\mathrm{erfc}(z) = e^{-z^2}\, w(iz),\\
&D(z) = -\tfrac{1}{2}i\sqrt{\pi}\,(w(z) - e^{-z^2}),\\
&C(z) + iS(z) = \tfrac{1}{2}(1+i)[1-e^{\frac{1}{2}iz^2}\, w(\zeta)], \quad \zeta = \tfrac{1}{2}\sqrt{\pi}(1+i)z,\\
&G(z) + iF(z) = \tfrac{1}{2}(1+i)w(\zeta).
\end{aligned}
\tag{4.1.1.3.9}
$$

2. In NUMAL komt ook een procedure voor (NONEXPERFC) waarmee $\exp(x^2) * \mathrm{erfc}(x)$ berekend kan worden. Hiermee kan men underflow/overflow problemen voorkomen voor grote waarden van x. De combinatie van erfc en de exponentiële functie komt vaak voor, bijvoorbeeld in

$$
\int_0^\infty e^{-(ax^2+bx)}\, dx = \tfrac{1}{2}\sqrt{\frac{\pi}{a}}\; e^{b^2/a}\; \mathrm{erfc}(b/\sqrt{a}), \quad \mathrm{Re}\ a > 0.
$$

De functies F en G uit (4.1.1.3.7) en (4.1.1.3.8), die overigens geen relatie hebben met de functies f en g uit 4.1.1.1, staan als volgt in verband met C en S:

$$
\begin{aligned}
F(x) &= [\tfrac{1}{2} - S(x)]\cos(\tfrac{1}{2}\pi x^2) - [\tfrac{1}{2} - C(x)]\sin(\tfrac{1}{2}\pi x^2),\\
G(x) &= [\tfrac{1}{2} - C(x)]\cos(\tfrac{1}{2}\pi x^2) + [\tfrac{1}{2} - S(x)]\sin(\tfrac{1}{2}\pi x^2).
\end{aligned}
\tag{4.1.1.3.10}
$$

Uit (4.1.1.3.7) en (4.1.1.3.8) blijkt dat F en G voor positief argument monotoon dalende positieve functies zijn met $F(0) = G(0) = \frac{1}{2}$. Inversie van (4.1.1.3.10) geeft een representatie van C en S die vooral voor grote x duidelijk aangeeft hoe het gedrag van C en S verloopt.

$$
\begin{aligned}
C(x) &= \tfrac{1}{2} + F(x)\sin(\tfrac{1}{2}\pi x^2) - G(x)\cos(\tfrac{1}{2}\pi x^2),\\
S(x) &= \tfrac{1}{2} - F(x)\cos(\tfrac{1}{2}\pi x^2) - G(x)\sin(\tfrac{1}{2}\pi x^2).
\end{aligned}
\tag{4.1.1.3.11}
$$

Voor grote waarde van x hangt de nauwkeurigheid voornamelijk af van die van de goniometrische functies.

Voorbeeld. Bereken voor positieve x de integraal

$$(4.1.1.3.12)\quad \int_0^\infty e^{-xt}\,\sin(t^2)/t\;dt.$$

Deze integraal kan worden uitgedrukt in de C en S funkties:

$$(4.1.1.3.13)\quad \tfrac{1}{2}\pi\{[\tfrac{1}{2}-C(y)]^2+[\tfrac{1}{2}-S(y)]^2\},\qquad y=x\sqrt{2\pi}.$$

Hiervoor kunnen de routines uit NAG, NUMAL en ACCULIB gebruikt worden.

a. NAG. Maak gebruik van S20AA A/F en S20AB A/F. Op deze manier wordt het antwoord

$$\tfrac{1}{2}\pi\{[\tfrac{1}{2}-\text{S20AA A/F}(y)]^2+[\tfrac{1}{2}-\text{S20AB A/F}(y)]^2\}.$$

b. NUMAL. De aanroep

FRESNEL(y,C,S)

levert C en S als output, waarmee (4.1.1.3.13) berekend kan worden. Een andere aanpak is via de procedure FG, die na aanroep FG(y,F,G) de functies F en G levert. Onder gebruikmaking van (4.1.1.3.11) volgt na enig omwerken dat (4.1.1.3.13) gelijk is aan $\frac{1}{2}\pi[F^2(y)+G^2(y)]$. Deze berekening zal voor grote x-waarden veel nauwkeuriger zijn dan die via C en S. Immers, voor grote x is $C(x)=\frac{1}{2}+O(1/x)$, $S(x)=\frac{1}{2}+O(1/x)$, zodat in de berekening van $\frac{1}{2}-C(y)$ en $\frac{1}{2}-S(y)$ significante cijfers verloren zullen gaan. De berekening via FG is tevens veel nauwkeuriger, omdat geen goniometrische functie berekend moeten worden (zie Opmerking 3).

c. ACCULIB. De functies C en S (of F en G) komen niet als zodanig voor, maar wel via w(z), zie (4.1.1.3.9). Via F en G geeft weer de beste resultaten. Met $\zeta=\frac{1}{2}\sqrt{\pi}(1+i)y$ en $w(\zeta)=\xi+i\eta$ $(\xi,\eta\in\mathbb{R})$ volgt nu dat de integraal in (4.1.1.3.12) gelijk is aan $\frac{1}{4}\pi(\xi^2+\eta^2)$.

4.1.1.4. Functies uit de statistiek

In IMSL komen veel routines voor die corresponderen met functies die men in de statistiek aantreft, zoals binomiaal-, χ^2-, F-, Poisson-, Student's t-, hypergeometrische- en Smirnov-verdelingsfuncties.

In de NAG-programmatheek komen in dit verband in hoofdstuk S van Speciale Functies routines voor ter berekening van de errorfuncties en de normale verdelingsfuncties P en Q. Bovendien komen in hoofdstuk G routines voor ter berekening van de χ^2-, F- en Student's t-verdeling.

In NUMAL komen geen speciale functies voor onder de namen en de gedaante zoals ze vaak in de mathematische statistiek voorkomen, maar de procedures voor de errorfuncties, de incomplete betafunctie en de incomplete gammafuncties kunnen wel in dit verband gebruikt worden.

Daarentegen bevat de MC-programmatheek STATAL wel materiaal, vergelijkbaar met dat van IMSL. In onderstaande tabel zijn enkele voorbeelden genoemd. Niet alle aangegeven procedures uit STATAL zijn reeds beschikbaar. In deze programmatheek komen overigens ook procedures voor ter berekening van gammafuncties, incomplete betafuncties etc. Een vergelijking tussen IMSL en STATAL zou interessant zijn, maar wordt hier niet gegeven, aangezien dit buiten het kader van dit hoofdstuk valt.

Voor vrijwel alle verdelingsfuncties uit IMSL en STATAL wordt tevens een routine voor de inverse geleverd, terwijl in NUMAL alleen voor de errorfuncties een inverse aanwezig is.

TABEL 4

	IMSL	NAG	NUMAL	STATAL
Binomiaal	MDBIN	(via F)	(INCBETA)	BIN
χ^2	MDCH	G01BC A/F	(INCOMGAM)	CHISQ
F	MDFDRE	G01BB A/F	(INCBETA)	FISHER
hypergeometrisch	MDHYP	-	-	HYPERG
normaal	MDNOR	S15AB A/F	(ERRORFUNCTION)	NORMAL, PHI
Smirnov	MDSMR	-	-	KOLSMIRDIS
Student's t	MDTD	G01BA A/F	(INCBETA)	STUDENT
Niet-centrale t	MDTN	-	-	NCSTUDENT
Poisson	MDTPOS	(via χ^2)	(INCOMGAM)	POISSON

Voorbeelden

Binomiaal	$\sum_{j=a}^{n} \binom{n}{j} p^j q^{n-j} = I_p(a,n-a+1)$.	
χ^2	$P(\chi^2\|\nu) = \gamma(a,x)/\Gamma(a)$, $Q(\chi^2\|\nu) = \Gamma(a,x)/\Gamma(a)$.	$\nu = 2a$, $\chi^2 = 2x$.
F	$Q(F\|\nu_1,\nu_2) = I_x(\frac{1}{2}\nu_2,\frac{1}{2}\nu_1)$,	$x = \nu_2/(\nu_2+\nu_1 F)$.
normaal	$P(x) = \frac{1}{2}[1 + \mathrm{erf}(x/\sqrt{2})]$, $Q(x) = \frac{1}{2}\mathrm{erfc}(x/\sqrt{2})$.	
Student's t	$A(t\|\nu) = 1 - 2I_x(\frac{1}{2}\nu,\frac{1}{2})$,	$x = \nu/(\nu+t^2)$.
Poisson	$\sum_{j=0}^{a-1} e^{-m} \frac{m^j}{j!} = \Gamma(a,m)/\Gamma(a)$,	a geheel.

4.1.2. Voorbeelden voor gebruik van routines voor Besselfuncties

Naast de voorbeelden die in het vorige gedeelte aan de orde zijn gekomen op het gebied van functies die doorgaans als integralen voorkomen, wordt nu een aantal toepassingen besproken voor Besselfuncties. Numerieke resultaten worden in de bijlage verstrekt; ze zijn verzorgd door Koos Huibers.

De Besselfuncties kunnen in twee groepen verdeeld worden.

1. De gewone Besselfuncties.
   Aangeduid met $J_\nu(x)$ en $Y_\nu(x)$, ν de orde, x het argument. Ze zijn verwant (qua gedrag) met de sinus en de cosinus; $Y_\nu(x)$ is niet begrensd voor $x \to 0$, $J_\nu(x)$ daarentegen wel (mits $\nu \geq 0$ of $\nu \in \mathbb{Z}$).

2. De gemodificeerde Besselfuncties.
   Aangeduid met $I_\nu(x)$ en $K_\nu(x)$, ν de orde, x het argument. Ze zijn verwant (qua gedrag) met de exponentiële functies of hyperbolische functies; $K_\nu(x)$ is niet begrensd voor $x \to 0$, $I_\nu(x)$ daarentegen wel (mits $\nu \geq 0$ of $\nu \in \mathbb{Z}$). Voor $x \to \infty$ is het gedrag

(4.1.2.1) $\quad I_\nu(x) \sim e^x/\sqrt{2\pi x}$, $\quad K_\nu(x) \sim e^{-x}\sqrt{\pi/2x}$.

In de tabellen vindt men een overzicht van de beschikbare Besselfuncties in NAG en NUMAL. In IMSL komen alleen routines voor ter berekening van Kelvinfuncties, waarover meer wordt gezegd in 4.1.2.3. In NUMAL komen ook nog procedures voor ter berekening van de met Besselfuncties verwante Airyfuncties en de nulpunten daarvan.

TABEL 6

(voor berekening van één of twee functies)

	NAG		NUMAL	
$J_a(x)$	$J_0(x)$	S17AA A/F	$J_0(x)$	BESS J0
	$J_1(x)$	S17AB A/F	$J_1(x)$	BESS J1
$Y_a(x)$	$Y_0(x)$	S17AC A/F	$Y_0(x)$, $Y_1(x)$	BESS Y01
	$Y_1(x)$	S17AD A/F	$Y_a(x)$, $Y_{a+1}(x)$	BESS YA01
$I_a(x)$	$I_0(x)$	S18AA A/F	$I_0(x)$	BESS I0
	$I_1(x)$	S18AB A/F	$I_1(x)$	BESS I1
			$e^{-x}I_0(x)$	NONEXP BESS I0
			$e^{-x}I_1(x)$	NONEXP BESS I1
$K_a(x)$	$K_0(x)$	S18AC A/F	$K_0(x)$, $K_1(x)$	BESS K01
	$K_1(x)$	S18AD A/F	$e^{x}K_0(x)$, $e^{x}K_1(x)$	NONEXP BESS K01
			$K_a(x)$, $K_{a+1}(x)$	BESS KA01
			$e^{x}K_a(x)$, $e^{x}K_{a+1}(x)$	NONEXP BESS KA01

TABEL 7

(hulpfuncties voor de berekening van Besselfuncties; NUMAL)

$P_0(x)$, $Q_0(x)$	BESS PQ0
$P_1(x)$, $Q_1(x)$	BESS PQ1
$P_a(x)$, $P_{a+1}(x)$, $Q_a(x)$, $Q_{a+1}(x)$	BESS PQA01

TABEL 8

(voor berekening van een rij van Besselfuncties; NUMAL)

$J_a(x)$	$J_0(x),\ldots,J_n(x)$	BESS J
	$J_{\frac{1}{2}}(x),\ldots,J_{n+\frac{1}{2}}(x)$	SPHER BESS J
	$J_a(x),\ldots,J_{n+a}(x)$	BESS JAPLUSN
$Y_a(x)$	$Y_0(x),\ldots,Y_n(x)$	BESS Y
	$Y_{\frac{1}{2}}(x),\ldots,Y_{n+\frac{1}{2}}(x)$	SPHER BESS Y
	$Y_a(x),\ldots,Y_{n+a}(x)$	BESS YAPLUSN
$I_a(x)$	$I_0(x),\ldots,I_n(x)$	BESS I
	$e^{-x}I_0(x),\ldots,e^{-x}I_n(x)$	NONEXP BESS I
	$I_{\frac{1}{2}}(x),\ldots,I_{n+\frac{1}{2}}(x)$	SPHER BESS I
	$e^{-x}I(x),\ldots,e^{-x}I_{n+\frac{1}{2}}(x)$	NONEXP SPHER BESS I
	$I_a(x),\ldots,I_{a+n}(x)$	BESS IAPLUSN
	$e^{-x}I_a(x),\ldots,e^{-x}I_{a+n}(x)$	NONEXP BESS IAPLUSN
$K_a(x)$	$K_0(x),\ldots,K_n(x)$	BESS K
	$e^{x}K_0(x),\ldots,e^{x}K_n(x)$	NONEXP BESS K
	$K_{\frac{1}{2}}(x),\ldots,K_{n+\frac{1}{2}}(x)$	SPHER BESS K
	$e^{x}K_{\frac{1}{2}}(x),\ldots,e^{x}K_{n+\frac{1}{2}}(x)$	NONEXP SPHER BESS K
	$K_a(x),\ldots,K_{a+n}(x)$	BESS KAPLUSN
	$e^{x}K_a(x),\ldots,e^{x}K_{a+n}(x)$	NONEXP BESS KAPLUSN

Opmerkingen

1. De functies uit Tabel 7 staan als volgt in verband met de gewone Besselfuncties.

$$J_a(x) = \sqrt{2/\pi x}\,[P_a(x)\cos\chi - Q_a(x)\sin\chi],$$
$$Y_a(x) = \sqrt{2/\pi x}\,[P_a(x)\sin\chi + Q_a(x)\cos\chi], \qquad (4.1.2.2)$$

met $\chi = x - (\frac{1}{2}a + \frac{1}{4})\pi$.

$P_a(x)$ en $Q_a(x)$ zijn voor $x > 0$ (en a reëel) monotoon dalende positieve functies, met $P_a(x) = O(1)$ en $Q_a(x) = O(1/x)$ voor $x \to \infty$. Uit (4.1.2.2) is het oscillerende gedrag van de functie $J_a(x)$ en $Y_a(x)$ duidelijk af te lezen.

2. Niet de in Tabel 8 aangegeven halftallige Besselfuncties worden berekend, maar de sferische Besselfuncties die ontstaan na vermenigvuldiging met $\sqrt{\pi/2x}$, dus $\sqrt{\pi/2x}\ J_{\frac{1}{2}}(x),\ldots,\sqrt{\pi/2x}\ J_{n+\frac{1}{2}}(x)$, etc.

4.1.2.1. Sommatie van een reeks met Besselfuncties

In Hoofdstuk 1 van dit colloquium, Lineaire Algebra, p.12, komt een reeks van gemodificeerde Besselfuncties voor:

$$(4.1.2.1.1)\qquad \sum_{\mu=0}^{n} (E/A)^{\mu/2}\ x_\mu I_\mu(2\sqrt{AE}),$$

waarin A en E positieve getallen zijn en $x_0,\ldots,x_n$ zijn reële coëfficiënten, waarvan de berekening in 1.1 is besproken.

In de NAG-programmatheek komen geen routines ter berekening van een rij Besselfuncties voor. De waarde van n is in 1.1 gelijk aan 15, zodat de routines voor $I_0(x)$ en $I_1(x)$ uit NAG niet toereikend zijn. Er bestaat wel een recurrente betrekking tussen Besselfuncties van opvolgende orde:

$$I_{a+1}(x) = I_{a-1}(x) - \frac{2a}{x} I_a(x),$$

maar deze is in voorwaartse richting, d.w.z. met uitgangspunt $I_0(x)$ en $I_1(x)$, niet stabiel (zie GAUTSCHI [1967]). Bij de berekening in de NUMAL-procedure wordt deze instabiliteit vermeden en de reeks in (4.1.2.1.1) kan door middel van de volgende procedure verkregen worden:

```
real procedure reeks (e,a,n,x); value e,a,n; real e,a;
                                integer n; array x;
begin real b,c,sum; integer m; array ibess[0:n]
   procedure bess i(x,n,i); code 35172;
   sum:= 0; c:= 1; b:= sqrt(e/a);
   bess i(2 * sqrt(a * e),n,ibess);
   for m:= 0 step 1 until n do
   begin sum:= sum + c * x[m] * ibess[m]; c:= c * b end;
   reeks:= sum
end reeks;
```

4.1.2.2. Berekening van nulpunten van Besselfuncties

Er komen in de programmatheken geen routines voor waarmee de nulpunten van Besselfuncties berekend kunnen worden (met uitzondering van de NUMAL-procedure voor de berekening van nulpunten van Airyfuncties). Natuur-

lijk zijn er de algemene routines voor het bepalen van oplossingen van niet-lineaire vergelijkingen, maar deze routines worden hier niet besproken. De methode die hier beschreven wordt dient ter demonstratie voor het gebruik van de routines ter berekening van Besselfuncties. De methode kan wellicht verfijnd worden, hoewel de resultaten zeer bevredigend zijn. Er wordt gebruik gemaakt van het feit, dat de Besselfuncties aan een differentiaalvergelijking voldoen. De methode is dan ook tevens van toepassing op andere functies die deze eigenschap hebben, mits een redelijke schatting van de nulpunten voorhanden is.

Voor een eerste schatting van de nulpunten van de Besselfuncties maken we gebruik van de asymptotische ontwikkeling voor de nulpunten, zie ABRAMOWITZ & STEGUN [1964, p. 371], waarvan de eerste termen zijn te vinden in de benadering

$$(4.1.2.2.1) \qquad j_{\nu,s} = \beta - \frac{\mu-1}{8\beta}\left[1 + \frac{4(7\mu-31)}{3(8\beta)^2} + \frac{32(83\mu^2 - 982\mu + 3779)}{15(8\beta)^4}\right] + O(\beta^{-7}),$$

waarin $j_{\nu,s}$ het s^{de} nulpunt van $J_\nu(x)$ voorstelt en

$$\mu = 4\nu^2, \quad \beta = (s + \tfrac{1}{2}\nu - \tfrac{1}{4})\pi.$$

(Voor $y_{\nu,s}$, het s^{de} nulpunt van $Y_\nu(x)$, kan deze formule ook gebruikt worden maar dan met $\beta = (s + \frac{1}{2}\nu - \frac{3}{4})\pi$). Deze benadering kan omgewerkt worden tot

$$(4.1.2.2.2) \qquad j_{\nu,s} = \beta - \frac{\mu-1}{8\beta}\left[\frac{1 - p/(8\beta)^2}{1 - q/(8\beta)^2}\right] + O(\beta^{-7}),$$

met

$$p = \frac{4(253\mu^2 - 3722\mu + 17869)}{15(7\mu-31)}, \quad q = \frac{8(83\mu^2 - 982\mu + 3779)}{5(7\mu-31)}.$$

Deze laatste benadering correspondeert met de (1,1) Padé-breuk van de ontwikkeling in (4.1.2.2.1). De fout in beide benaderingen is $O(\beta^{-7})$, hoewel de rationale (4.1.2.2.2) gunstiger uitvalt.

De benadering in (4.1.2.2.2) (en (4.1.2.2.1)) is zinvol (in asymptotische zin) voor $s >> \nu$, maar geeft bij kleine ν reeds aardige resultaten voor $s = 1$. Voor $s = 1$, $\nu = 0$ geeft (4.1.2.2.2)

$$j_{0,1} = 2.4052\ldots$$

met werkelijke waarde 2.404825..., dus een absolute fout van $4.3_{10}{-4}$ (terwijl (4.1.2.2.1) een absolute fout $1.6_{10}{-3}$ oplevert).

Een volgende benadering kan verkregen worden door een approximatie proces gebaseerd op de regel van Newton: als x een benadering is van een nulpunt α van f, dan is, als de benadering nauwkeurig genoeg is,

$$(4.1.2.2.3) \quad \alpha = x - (f/f') - \tfrac{1}{2}(f''/f')(f/f')^2 - \frac{1}{6}[3(f''/f')^2 - f'''/f'](f/f')^3 + \\ + O[(f/f')^4].$$

Door gebruik te maken van de differentiaalvergelijking van de Besselfuncties

$$(4.1.2.2.4) \quad y'' + \frac{1}{x}\,y' + (1 - \nu^2/x^2)y = 0,$$

met als oplossingen $J_\nu(x)$ en $Y_\nu(x)$, kunnen de afgeleiden f'', f''' verwijderd worden in (4.1.2.2.3). Tevens kan f' verwijderd worden door gebruik te maken van de relatie

$$J'_\nu(x) = \frac{\nu}{x}\,J_\nu(x) - J_{\nu+1}(x),$$

waarmee $J_{\nu+1}(x)$ in de formules terechtkomt. Over $J_{\nu+1}(x)$ wordt echter via de programmatuur eerder beschikt dan over $J'_\nu(x)$. Na wat manipulatie komen we tot de volgende rationale versie van (4.1.2.2.3) (weer een (1,1) Padé-breuk)

$$(4.1.2.2.5) \quad j_{\nu,s} = x + r\,\frac{1+pr}{1+qr},$$

met x bepaald uit (4.1.2.2.2) en

$$r = J_\nu(x)/J_{\nu+1}(x),$$

$$p = \frac{1+4x^2-4\nu^2}{6x(2\nu+1)},$$

$$q = \frac{2x^2-1-6\nu-8\nu^2}{3x(2\nu+1)}.$$

In (4.1.2.2.5) zijn termen van de orde r^4 verwaarloosd.

Na deze inleidende zaken komen we toe aan het gebruik van de programmatuur. De routines in NAG voor J_0 en J_1 zijn toereikend voor zover het nulpunten van deze functies betreft; voor andere waarden van de orde zijn we op NUMAL aangewezen.

We gebruiken daartoe de procedure BESS JAPLUSN, die een rij $J_a(x),\ldots,J_{a+n}(n)$ berekent, echter met $0 \leq a < 1$. Willen we de nulpunten van $J_a(x)$ met $a > 1$ berekenen, dan kan dat gebeuren door BESS JAPLUSN als volgt aan te roepen:

```
BESS JAPLUSN(a-entier(a),x,entier(a)+1,j),
```

waarin j een array is met grenzen 0 en entier(a)+1. Dan is $J_a(x)$ = j[entier(a)], $J_{a+1}(x)$ = j[entier(a)+1]. Een procedure NULPUNTEN BESSELJ is gegeven in de bijlage, samen met de resultaten voor a = 0,1,2. De nulpunten zijn vergeleken met de 18d resultaten van DETOURNAY & PIESSENS [1971].

Opmerkingen

1. De in de bijlage gegeven procedure NULPUNTEN BESSELJ kan gemakkelijk zo veranderd worden, dat de nulpunten van $Y_a(x)$ berekend worden. Daartoe dient b de waarde $\pi(m + \frac{1}{2}a - \frac{3}{4})$ te krijgen en in plaats van BESS JAPLUSN moet BESS YAPLUSN gebruikt worden.

2. Naast de nulpunten staan in de bijlage de functiewaarden van $J_a(x)$ afgedrukt, voor $x = j_{a,s}$. $J_a(x)$ is daarbij berekend via een nieuwe procedure JAX, welke voorkomt in de procedure TEST NULPUNTEN; JAX is gebaseerd op (4.1.2.2).

4.1.2.3. Berekening van complexe Besselfuncties

Voor de berekening van de Besselfuncties

$$J_a(z),\ I_a(z),\ Y_a(z) \text{ en } K_a(z), \qquad z \in \mathbb{C},\ a \in \mathbb{R},$$

zijn in de programmatheken IMSL, NAG en NUMAL geen routines beschikbaar. Een uitzondering vormen de subroutines voor de berekening van Kelvinfuncties uit IMSL:

MMKEL0 berekent $\mathrm{ber}_0(x)$, $\mathrm{bei}_0(x)$, $\mathrm{ker}_0(x)$, $\mathrm{kei}_0(x)$,

MMKEL1 berekent $\mathrm{ber}_1(x)$, $\mathrm{bei}_1(x)$, $\mathrm{ker}_1(x)$, $\mathrm{kei}_1(x)$,

MMKELD berekent $\mathrm{ber}_0'(x)$, $\mathrm{bei}_0'(x)$, $\mathrm{ker}_0'(x)$, $\mathrm{kei}_0'(x)$,

met $x > 0$. De relaties met de andere Besselfuncties is

$$\mathrm{ber}_\nu(x) + i\,\mathrm{bei}_\nu(x) = J_\nu(x\,e^{3\pi i/4}) = e^{\nu\pi i}\,J_\nu(x\,e^{-\pi i/4}),$$
$$\mathrm{ker}_\nu(x) + i\,\mathrm{kei}_\nu(x) = e^{-\frac{1}{2}i\pi\nu}\,K_\nu(x\,e^{\pi i/4}), \qquad (4.1.2.3.1)$$

waaruit blijkt dat de Kelvinfuncties de reële en imaginaire delen zijn van J_ν en K_ν op de bisectrices van de kwadranten in het complexe vlak. De Kelvinfuncties spelen een rol in de electriciteitsleer.

Voorbeeld. Beschouw een lange draad waardoor wisselstroom loopt met stroomdichtheid $\vec{J} = (J_x, J_y, J_z)$. Uit de electriciteitsleer volgt dat $\vec{J}$ voldoet aan de (genormeerde) parabolische differentiaalvergelijking

$$\frac{\partial^2\vec{J}}{\partial x^2} + \frac{\partial^2\vec{J}}{\partial y^2} + \frac{\partial^2\vec{J}}{\partial z^2} = \frac{\partial\vec{J}}{\partial t}\,.$$

Als de draad evenwijdig aan de z-as loopt dan is $J_x = J_y = 0$ en J_z (genoteerd door J) is onafhankelijk van z en θ, waarbij $x = r\cos\theta$, $y = r\sin\theta$. Als we de frequentie van de wisselstroom ω noemen, dan zoeken we een oplossing van de differentiaalvergelijking in de vorm $J = u(r)\sin(\omega t+\phi) = \mathrm{Im}(U(r)e^{i\omega t})$ en voor U ontstaat de vergelijking

$$\frac{d^2U}{dr^2} + \frac{1}{r}\frac{dU}{dr} - i\omega U = 0,$$

met als oplossingen (zie (4.1.2.2.4) met $\nu = 0$) $U = c_1\,J_0(ar) + c_2\,Y_0(ar)$ met $a^2 = i\omega$, of $a = e^{3i\pi/4}\sqrt{\omega}$, zodat inderdaad de Kelvinfuncties ontstaan via (4.1.2.3.1).

We zullen laten zien hoe procedures uit NUMAL gebruikt kunnen worden voor de berekening van ber_n en bei_n.

In de literatuur komen overigens in algoritmevorm wel programma's voor ter berekening van complexe Besselfuncties. Zie GAUTSCH [1964] in ALGOL voor de berekening van een rij $J_a(z),\ldots,J_{a+n}(z)$, $0 \le a < 1$, $|\arg z| < \pi$, en

SOOKNE [1973] in FORTRAN voor de berekening van de rijen $J_0(z),\ldots,J_n(z)$ en $I_0(z),\ldots,I_n(z)$.

Voor ber_n en bei_n maken we gebruik van de volgende ontwikkeling voor $n = 0,1,2,\ldots$ en $x \in \mathbb{R}$:

$$\mathrm{ber}_n(x) = \sum_{k=-\infty}^{\infty} (-1)^{n+k} J_{n+2k}(t)\, I_{2k}(t),$$

(4.1.2.3.2) $$t = x/\sqrt{2}\ ,$$

$$\mathrm{bei}_n(x) = \sum_{k=-\infty}^{\infty} (-1)^{n+k} J_{n+2k+1}(t)\, I_{2k+1}(t).$$

Voor gehele orde voldoen de Besselfuncties aan

$$J_{-n}(x) = (-1)^n J_n(x), \quad I_{-n}(x) = I_n(x),$$

zodat alle termen in de reeksen kunnen worden teruggebracht tot termen met niet-negatieve orde. Voor $n = 0$ ontstaat

$$\mathrm{ber}_0(x) = J_0(t)\, I_0(t) - 2J_2(t)\, I_2(t) + 2J_4(t)\, I_4(t) \ldots$$

$$\mathrm{bei}_0(x) = 2[J_1(t)\, I_1(t) - J_3(t)\, I_3(t) + J_5(t)\, I_5(t) \ldots],$$

en voor $n = 1$

$$\mathrm{ber}_1(x) = -[J_1(t)\{I_0(t)+I_2(t)\} - J_3(t)\{I_2(t)+I_4(t)\}\ldots],$$

$$\mathrm{bei}_1(x) = -[I_1(t)\{J_2(t)-J_0(t)\} - I_3(t)\{J_4(t)-J_2(t)\}\ldots].$$

Voor het sommeren van deze reeksen moeten twee rijen Besselfuncties $\{J_k(t)\}$ en $\{I_k(t)\}$ berekend worden. Het is daarbij noodzakelijk te weten hoe veel termen in de reeksen benodigd zijn voor een zekere nauwkeurigheid. Daarvoor moet het gedrag van de Besselfuncties voor grote orde bekend zijn. Er geldt voor $k \gg t$

$$J_k(t)\, I_k(t) \sim (\tfrac{1}{2}t)^{2k}/(k!)^2.$$

Met Stirling's benadering voor $k!$ schrijven we het rechterlid als $(te/2k)^{2k}/(2\pi k)$. Voor grote x zijn $\mathrm{ber}_n(x)$ en $\mathrm{bei}_n(x)$ van de orde $\exp(t)$ en om een relatieve nauwkeurigheid van ε te bereiken breken we de reeksen

af als k zo groot is dat

$$(te/2k)^{2k} < \varepsilon\, e^{t}.$$

(Vanwege de snelle convergentie is dan ook de rest van de reeks verwaarloosbaar klein.) De bovenstaande ongelijkheid is equivalent met

$$\frac{2k}{te} \ell n \frac{2k}{te} > -\frac{1}{e} + \frac{\ell n\ 1/\varepsilon}{te}.$$

Dit probleem correspondeert met het oplossen van de vergelijking

$$p\ \ell n\ p = y$$

met $p = 2k/te$, $y = -1/e + \ell n(1/\varepsilon)/te$ (voor $p \geq 1/e$ en $y \geq -1/e$ is er precies één oplossing in het interval $[1/e, y+1]$, welke oplossing met de NUMAL procedure ZEROIN berekend kan worden).

Een en ander is verwerkt in de ALGOL 60 procedure KELVIN, waarvoor wordt verwezen naar de bijlage, met enkele functiewaarden die vergeleken kunnen worden met resultaten uit ABRAMOWITZ & STEGUN [1964, p.430]. Vergelijking met de IMSL routines is ook mogelijk, maar de aanroep is erg elementair en wordt hier verder niet gedemonstreerd.

Andere complexe Besselfuncties kunnen middels reële functies berekend worden via analoge reeksen. Voor de ker en kei-functies wordt dit niet verder uitgewerkt.

Reeksen met Besselfuncties kunnen ook gebruikt worden ter berekening van andere speciale functies. We denken hierbij aan toepassingen op het gebied van Mathieu-functies. Voor dit soort functies is een representatie via Besselfuncties de meest aangewezen weg ter berekening.

Literatuur

ABRAMOWITZ, M. & I.A. STEGUN [1964], *Handbook of Mathematical Functions*, National Bureau of Standards Applied Mathematics Series, 55, Govern. Printing Off., Washington, D.C.

DETOURNAY, P. & R. PIESSENS [1971], *Zeros of Bessel Functions and Zeros of Cross Products of Bessel Functions*, Report TW 7, Applied Mathematics Division, Katholieke Universiteit, Leuven.

GAUTSCHI, W. [1967], *Computational Aspects of Three-term Recurrence Relations*, SIAM Rev. 9, 24-82.

GAUTSCHI, W. [1964], *Algorithm 236*, Comm. Assoc. Comp. Mach. 7, 479.

GRADSHTEYN, I.S. & I.M. RYSHIK [1965], *Table of Integrals, Series and Products*, fourth ed., Academic Press, London-New York.

MAGNUS, W., F. OBERHETTINGER & R.P. SONI [1966], *Formulas and Theorems for the Special Functions of Mathematical Physics*, Springer-Verlag, Berlin.

SOOKNE, D.J. [1973], *Bessel Functions*, J. of Res. NBS, 77B, 111-136.

Bijlagen

```
"BEGIN"

  "PROCEDURE" NULPUNTENBESSELJ(A,N,J); "VALUE" A,N;
  "REAL" A; "INTEGER" N; "ARRAY" J;
  "BEGIN" "REAL" B,C,P,Q,R,X,PI,MU; "INTEGER" NA,M;
    "ARRAY" JA[0:ENTIER(A)+1];
    "PROCEDURE" BESS JAPLUSN(A,X,N,JA); "CODE" 35180;
    PI:= 4 * ARCTAN(1); MU:= 4 * A * A; NA:= ENTIER(A);
    "FOR" M:= 1 "STEP" 1 "UNTIL" N "DO"
    "BEGIN" B:= (M + A/2 - 0.25) * PI; C:= 1/B/B/64;
      X:= B - (MU-1)/B/8 *
           (1-C*4/15/(7*MU-31)*((253*MU-3722)*MU+17869))/
           (1-C*8/ 5/(7*MU-31)*(( 83*MU- 982)*MU+ 3779));
      BESS JAPLUSN(A-NA,X,NA+1,JA); R:= JA[NA]/JA[NA+1];
      P:= (1+4*X*X - MU)/X/6/(2*A+1);
      Q:= -(2*MU + 6*A + 1 - 2*X*X)/X/3/(2*A+1);
      J[M]:= X+R*(1+P*R)/(1+Q*R)
    "END"
  "END" NULPUNTENBESSELJ;

  "PROCEDURE" TEST NULPUNTEN;
  "BEGIN" "REAL" A; "INTEGER" N; "ARRAY" J[1:10];
    "REAL" "PROCEDURE" JAX(A,X); "VALUE" A,X; "REAL" A,X;
    "BEGIN" "REAL" PA,QA,PA1,QA1,CHI,PI;
      "PROCEDURE" BESS PQA01(A,X,PA,QA,PA1,QA1); "CODE" 35183;
      PI:= 4 * ARCTAN(1); CHI:= X-PI*(A/2+0.25);
      BESS PQA01(A,X,PA,QA,PA1,QA1);
      JAX:= SQRT(2/PI/X) * (PA*COS(CHI) - QA*SIN(CHI))
    "END" JAX;
    "FOR" A:= 0, 1, 2 "DO"
    "BEGIN" NULPUNTENBESSELJ(A,10,J);
      OUTPUT(61,"("3/,2ZD,2/")",A);
      "FOR" N:= 1 "STEP" 1 "UNTIL" 10 "DO"
      OUTPUT(61,"("5B,N,2B,+D.2D"+ZD,/")",
        J[N],JAX(A,J[N]))
    "END"
  "END" TEST NULPUNTEN;

  OUTPUT(61,"("2B,"("A")",11B,"("J[N]")",13B,"("JAX")",/")");
  TEST NULPUNTEN
"END"
```

A	J[N]	JAX
0		
	+2.4048255576958"+000	+0.00" +0
	+5.5200781102863"+000	+5.66"-16
	+8.6537279129110"+000	+1.19"-14
	+1.1791534439014"+001	-1.69"-15
	+1.4930917708488"+001	+1.12"-14
	+1.8071063967911"+001	+1.25"-14
	+2.1211636629879"+001	-1.39"-14
	+2.4352471530749"+001	+6.49"-15
	+2.7493479132040"+001	-1.22"-14
	+3.0634606468432"+001	-4.82"-15
1		
	+3.8317059702075"+000	-2.35"-15
	+7.0155866698156"+000	-1.00"-14
	+1.0173468135063"+001	+8.94"-15
	+1.3323691936314"+001	-5.39"-15
	+1.6470630050878"+001	+7.66"-15
	+1.9615858510468"+001	-6.04"-15
	+2.2760084380593"+001	+1.09"-14
	+2.5903672087619"+001	+1.17"-14
	+2.9046828534917"+001	-6.34"-15
	+3.2189679910975"+001	+1.34"-14
2		
	+5.1356223018407"+000	+2.00"-14
	+8.4172441403998"+000	-2.44"-15
	+1.1619841172149"+001	+5.20"-15
	+1.4795951782351"+001	-1.47"-15
	+1.7959819494988"+001	+4.85"-15
	+2.1116997053022"+001	-1.88"-14
	+2.4270112313573"+001	+7.84"-15
	+2.7420573549984"+001	-1.73"-14
	+3.0569204495516"+001	+1.37"-14
	+3.3716519509223"+001	+1.10"-14

```
"BEGIN"

  "PROCEDURE" KELVIN BE01(X,BER0,BEI0,BER1,BEI1);
  "VALUE" X; "REAL" X,BER0,BER1,BEI0,BEI1;
  "IF" X=0 "THEN"
  "BEGIN" BER0:= 1; BEI0:= BER1:= BEI1:= 0 "END" "ELSE"
  "BEGIN" "REAL" A,B,E,T,Y; "INTEGER" K,N,S;
    "PROCEDURE" BESS J(X,N,J); "CODE" 35162;
    "PROCEDURE" BESS I(X,N,I); "CODE" 35172;
    "BOOLEAN" "PROCEDURE" ZEROIN(X,Y,FX,TOLX); "CODE" 34150;
    T:= ABS(X)/SQRT(2); E:= EXP(1); Y:= (-1+35/T)/E;
    A:= "IF" Y>0 "THEN" 1 "ELSE" 1/E; B:= 1+Y;
    Y:= "IF" ZEROIN(A,B,A*LN(A)-Y,"-5) "THEN" A "ELSE" 1+Y;
    K:= 2*(1+ENTIER(T*E*Y/4));
    "BEGIN" "ARRAY" I,J[0:K];
      BESS J(T,K,J); BESS I(T,K,I);
      BER0:= BER1:= BEI0:= BEI1:= 0; S:= -1;
      "FOR" N:= 2 "STEP" 2 "UNTIL" K "DO"
      "BEGIN" S:= -S;
        BER0:= BER0 + J[N]*I[N]*S;
        BEI0:= BEI0 + J[N-1]*I[N-1]*S;
        BER1:= BER1 + J[N-1]*(I[N-2]+I[N])*S;
        BEI1:= BEI1 + I[N-1]*(J[N]-J[N-2])*S
      "END";
      BER0:= (J[0]*I[0]-2*BER0);
      BEI0:= 2*BEI0; BER1:= -BER1*SIGN(X);
      BEI1:= -BEI1*SIGN(X)
    "END"
  "END" KELVIN BE01;

  "PROCEDURE" TEST KELVIN;
  "BEGIN" "REAL" A,B,C,D; "INTEGER" K;
    "FOR" K:= 0 "STEP" 1 "UNTIL" 5 "DO"
    "BEGIN" KELVIN BE01(K,A,B,C,D);
      OUTPUT(61,"("/,ZD,2B,4(N,2B),/")",K,A,B,C,D)
    "END"
  "END" TEST KELVIN;

  OUTPUT(61,"(""(" K")",11B,"("BER0")",21B,"("BEI0")",
  21B,"("BER1")",21B,"("BEI1")",///")");
  TEST KELVIN
"END"
```

K	BER0	BEI0	BER1	BEI1
0	+1.0000000000000"+000	+0.0000000000000"+000	+0.0000000000000"+000	+0.0000000000000"+000
1	+9.8438178121309"-001	+2.4956604003666"-001	-3.9586826101971"-001	+3.0755663137554"-001
2	+7.5173418271380"-001	+9.7229162730666"-001	-9.9707765192642"-001	+2.9977543700203"-001
3	-2.2138024959870"-001	+1.9375867852661"+000	-1.7326442211285"+000	-4.8745417701608"-001
4	-2.5634165572586"+000	+2.2926903226993"+000	-1.8692484590319"+000	-2.5638216885611"+000
5	-6.2300824786664"+000	+1.1603438155015"-001	+3.5977666677672"-001	-5.7979079017926"+000

4. SPECIALE FUNCTIES EN FOURIERTRANSFORMATIE

4.2. Fouriertransformatie

door C.G. van der Laan
(Rekencentrum, RU Groningen)

4.2.0. Inleiding

In deze bijdrage aan het colloquium behandelen wij enige aspecten van de discrete Fouriertransformatie. In sectie 4.2.2. geven we een overzicht van de beschikbare programmatuur in de programmatheken voor de Control Data CYBER serie:

ACCULIB	versie 6
IMSL	editie 4
NAG	mark 4 (alleen FORTRAN-versie)
NUMAL	versie 14.

In sectie 4.2.3 definiëren we een aantal testvoorbeelden. De resultaten m.b.t. efficiëntie zijn samengevat in Tabel 3; het geheugengebruik zowel als het gemak waarmee de programmatuur te gebruiken is hebben we niet gespecificeerd. In sectie 4.2.4 gaan we wat nader in op de relatie tussen de continue en discrete Fouriertransformatie; bovendien benadrukken wij het te weinig bekende resultaat: de benadering is vrij van het aliaseffect. In de secties 4.2.5-4.2.8 hebben wij een greep gedaan uit de toepassingen.

De woorden vector en (eindige) rij worden door elkaar gebruikt; als notatie gebruiken wij een kleine letter, bijvoorbeeld: a, of een ondergeïndiceerde kleine letter gescheiden door komma's en omsloten door een haakjespaar, bijvoorbeeld: $(a_1,a_2,\ldots,a_n)$, of een verzamelingsnotatie, bijvoorbeeld: $\{a_k\}_{k=1}^{n} \in \mathbb{R}^n$. Een element van een vector geven we aan door een ondergeïndiceerde kleine letter, bijvoorbeeld a_k. Voor een functie gebruiken we ook kleine letters. Een functiewaarde geven we aan door de naam met daar achter het argument tussen haakjes, bijvoorbeeld: $a(k\,\Delta f)$. Benaderingen van vectoren en functies geven we aan met een tilde, bijvoorbeeld: $\tilde{a}$ is een benadering van a.

Een p-bovengeïndiceerde kleine letter duidt een periodieke functie of vector aan.

Voor de exponent

$$e^{i2\pi kj/N}$$

gebruiken we de notatie

$$W_N^{jk} \quad \text{of} \quad W^{jk}.$$

4.2.1. Definities

Onder een *trigonometrische reeks* verstaan we een reeks van de vorm

$$(4.2.1.1)\quad \frac{a_0}{2} + \sum_{k=1}^{\infty} \{a_k \cos k\theta + b_k \sin k\theta\} = \sum_{k=0}^{\infty}{}' (a_k \cos k\theta + b_k \sin k\theta)$$

met a_k, b_k onafhankelijk van θ.
Andere reeksen zijn bijvoorbeeld machtreeksen. De reeks

$$(4.2.1.2)\quad \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k - ib_k) z^k$$

is een machtreeks op de eenheidscirkel $\{z \mid z=e^{i\theta}, \theta \in [0,\infty) \bmod 2\pi\}$. We merken op dat (4.2.1.1) het reële gedeelte van (4.2.1.2) is en constateren een verband tussen trigonometrische reeksen en machtreeksen op de eenheidscirkel. Algemener geldt dat een trigonometrische reeks geschreven kan worden als een Laurentreeks op de eenheidscirkel. Vanwege de gelijkheid

$$(4.2.1.3)\quad \sum_{k=0}^{\infty}{}' (a_k \cos k\theta + b_k \sin k\theta) = \sum_{k=-\infty}^{\infty} c_k e^{ik\theta}$$

met

$$2c_k = a_k - ib_k,\ c_{-k} = \overline{c}_k,\ k = 0,1,\ldots, b_0 = 0$$

noemen we het rechterlid van (4.2.1.3) de *complexe representatie van een trigonometrische reeks*.

Onder een *Fourierreeks van een functie* verstaan we een trigonometrische reeks, waarvan de coëfficiënten gegeven worden door

$$(4.2.1.4)\quad a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(\theta) \cos k\theta \, d\theta$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(\theta) \sin k\theta \, d\theta.$$

De complexe representatie van de Fourrierreeks van een functie is de Laurentreeks (rechterlid (4.2.1.3)) met de coëfficiënten

$$(4.2.1.5)\quad c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta)\, e^{-ik\theta}\, d\theta.$$

De integraaluitdrukkingen (4.2.1.4) en (4.2.1.5) worden *goniometrische momenten* genoemd.

Als bijzondere gevallen van een Fourierreeks van een functie op $[-\pi,\pi]$ onderkennen we

1. f is symmetrisch ($f(-\theta) = f(\theta)$); cosinusreeks.

Zij

$$(4.2.1.6)\quad f(\theta) \sim {\sum_{k=0}^{\infty}}' a_k \cos k\theta$$

en

$$f(\theta) = f(\arccos x) = h(x)$$

dan geldt de Chebyshevontwikkeling,

$$(4.2.1.7)\quad h(x) \sim {\sum_{k=0}^{\infty}}' a_k\, T_k(x)$$

met T_k het k-de Chebyshevpolynoom van de eerste soort en

$$x = \cos\theta$$

$$(4.2.1.8)\quad a_k = \frac{2}{\pi} \int_{-1}^{1} \frac{h(x)T_k(x)}{\sqrt{1-x^2}}\, dx$$

2. f. is antisymmetrisch ($f(-\theta) = -f(\theta)$); sinusreeks.

Zij

$$(4.2.1.9)\quad f(\theta) \sim \sum_{k=1}^{\infty} b_k \sin k\theta$$

en

$$f(\theta) = f(\arccos x) = h(x)$$

dan geldt

$$(4.2.1.10)\quad h(x) \sim \sqrt{1-x^2} \sum_{k=1}^{\infty} b_k \, U_{k-1}(x)$$

met

$$x = \cos \theta$$

$$(4.2.1.11)\quad b_k = \frac{2}{\pi} \int_{-1}^{1} h(x) \, U_{k-1}(x) \, dx$$

waarin U_k het k-de Chebyshevpolynoom is van de tweede soort.

Onder de *n-de partiële som* S_n van een trigonometrische reeks verstaan we

$$(4.2.1.12)\quad S_n(\theta) = \sum_{k=0}^{n}{}' (a_k \cos k\theta + b_k \sin k\theta) = \sum_{k=-n}^{n} c_k \, e^{ik\theta}.$$

De uitdrukking

$$(4.2.1.13)\quad e^{in\theta} \, S_n(\theta)$$

is een *trigonometrisch polynoom* van de graad 2n.

Opmerkingen

1. Het ~-symbool is gebruikt om de relatie tussen een functie en zijn Fourierreeks aan te geven; we nemen aan dat de functies Riemannintegreerbaar zijn en aan voldoende gladheidsvoorwaarden voldoen (Lipschitzconditie/begrensde variatie/continu/differentiëerbaar) opdat de rij $\{S_k\}$ convergeert (puntsgewijs/uniform/in ℓ_2-norm).
2. Het $\sum''$-symbool wordt gebruikt om aan te duiden dat zowel de eerste als de laatste term voor de helft genomen moet worden; het $\sum'$-symbool wordt

gebruikt als alleen de eerste term voor de helft genomen moet worden. In beide gevallen geldt $b_0 = b_n = 0$.

3. Bij een periodieke n-vector, bijvoorbeeld: $(c_0,c_1,\ldots,c_{n-1}) \subset \{\{c_k\}_{k=0}^{\infty} \mid c_k \in \mathbb{C} \wedge c_k = c_{k \bmod n}\}$, onderscheiden we de symmetrie-eigenschappen:
Hermitese n-vector ($c_k = \overline{c}_{-k}$),
anti-Hermitese n-vector ($c_k = -\overline{c}_{-k}$),
symmetrische n-vector ($c_k = c_{-k}$),
antisymmetrische n-vector ($c_k = -c_{-k}$).

4.2.2. Overzicht van de beschikbare programmatuur

4.2.2.1. Complexe eindige trigonometrische reeks

Zij

$$(4.2.2.1.1) \quad f(\theta_j) = \sum_{k=0}^{N-1} c_k \, e^{\pm ik\theta_j}$$

te berekenen, waarin $\{c_k\}_{k=0}^{N-1}$ en de hoek θ_j gegeven zijn.
De beschikbare programmatuur kunnen we dan classificeren op grond van:
a. evaluatie voor één hoek of meerdere equidistante hoeken;
b. c_k zijn reëel of complex.

De FFT-algoritmen vallen onder a) en men kan opmerken dat alle aanwezige implementaties zijn gebaseerd op de ALGOL 60 versies van SINGLETON [1968]. Voor een overzicht van andere algoritmen wordt verwezen naar COCHRAN in RABINER e.a. [1972].

De evaluatie voor één hoek is aanwezig in NUMAL. De onderliggende algoritme is de Clenshawrecursie met de modificatie van Reinsch. Deze algoritmen kunnen worden verkregen als een gegeneraliseerde Hornerregel uit de representatie

$$(4.2.2.1.c) \quad f(\theta) = (1,e^{-i\theta}) \sum_{k=0}^{N-1} \begin{pmatrix} 2\cos\theta & 1 \\ -1 & 0 \end{pmatrix}^k \begin{pmatrix} c_k \\ 0 \end{pmatrix} \qquad \text{(Clenshawrecursie)}.$$

Door geschikte factorisatie van de matrix en transformatie van de onafhankelijke variabele verkrijgen de Reinschmodificaties:

$$(4.2.2.1.r) \quad f(\theta) = g_0(\lambda) = (1,-\lambda/2+i\sin\theta) \sum_{k=0}^{N-1} \begin{pmatrix} \lambda+1 & \lambda \\ 1 & 1 \end{pmatrix}^k \begin{pmatrix} c_k \\ 0 \end{pmatrix}, \lambda = -4\sin^2\theta/2;$$

van $\lambda \sim \pi$ is een analoge representatie te verkrijgen.

TABEL 1

Implementaties voor berekening van $\sum_{k=0}^{N-1} c_k e^{ik\theta_j}$.

Gegevens specificatie	ACCULIB	IMSL	NAG	NUMAL
$\theta_j = \theta$, evaluatie voor één hoek				
$\{c_k\}_{k=0}^{N-1} \in \mathbb{C}^N$	-	-	-	COMFOUSER/1/2
$\{c_k\}_{k=0}^{N-1} \in \mathbb{R}^n$	-	-	-	COMFOUSER/1/2
$\theta_j = j\frac{2\pi}{N}$, $j = 0,1,\ldots,N-1$ (FFT)				
$\{c_k\}_{k=0}^{N-1} \in \mathbb{C}^N$	FFTMCA/F	FFTP FFT2 FFT2RV	C06ABA/F C06ADA/F	-
$\{c_k\}_{k=0}^{N-1} \in \mathbb{R}^N$	-	FFTR	-	-

Opmerkingen (Tabel 1)

1. IMSL bevat routines voor de FFT en nog een hulproutine FFRDR2 om de permuatie van een complexe datarij, van lengte 2^m, uit te voeren (reverse binary $\leftrightarrow$ normal).
2. De routine FFT2RV, transformeert een complexe datarij, van lengte 2^m, gegeven in "reverse binary order"; het resultaat is in "normal order". FFT2 transformeert een complexe datarij, van lengte 2^m, uitgaande van "normal order"; het resultaat is in "reverse binary order". C06ABA/F, C06ADA/F, FFTMCA/F, FFTP en FFTR transformeren van "normal order" naar "normal order".
3. FFTMCA/F en C06ADA/F zijn geschikt voor "multi-variate" transformatie.
4. Het merendeel der programmatuur voor FFT vereist een datarij waarvan de lengte een macht van twee is. C06ADA/F, FFTP en FFTMCA/F laten datarijen van andere lengte toe. FFTR vereist een datarij waarvan de lengte even is.

5. Voor de inversie bevat de programmatuur C06ABA/F een extra boolean/logical parameter, invers. Bij IMSL staat in de documentatie vermeld hoe, door toepassing van conjugaties, de inversie m.b.v. de genoemde routines verkregen kan worden; bij FFTR wordt dit echter niet aangegeven.

In de Voorbeelden 0 t/m 3 geven we aan hoe de complexe Fouriertransformatie te gebruiken is als de datarij bijzondere eigenschappen bezit, bijvoorbeeld: $\in \mathbb{R}$, (anti-)Hermites, (anti-)symmetrisch. In het algemeen geldt dat een datarij van even lengte efficiënt getransformeerd kan worden.

Voorbeeld 0 (Transformatie van een reële datarij (van even lengte)).
Zij

$$(4.2.2.1.2) \qquad f(\frac{2\pi j}{N}) = \sum_{k=0}^{N-1} r_k \, W_N^{kj}, \; j = 0,1,\ldots,N-1$$

met $\{r_k\} \in \mathbb{R}^N$. De resulterende rij is Hermites.

Algoritme 0
N = even:

$$f_j = \tfrac{1}{2}\{a_j + \overline{a}_{N/2-j} - i \, W_N^j \, (a_j - \overline{a}_{N/2-j})\}, \; j = 0,1,\ldots,N/2,$$

$$f_{N-j} = \overline{f}_j, \; j = 0,1,\ldots,N/2$$

waarin

$$a_j = \sum_{k=0}^{N/2-1} (r_{2k} + i r_{2k+1}) \, W_{N/2}^{kj}, \; j = 0,1,\ldots,N/2-1,$$

$$a_{N/2} = a_0.$$

N = oneven:
Via algemene complexe transformatie (aanroepen van FFTR).

Illustratie 0
Zij

$$\{r_k\} = (1,0,3,4).$$

De resultaten worden verkregen uit

$$\{a_k\}_{k=0}^{1} = (4+4i,-2-4i)$$

als

$$\{f_k\}_{k=0}^{2} = (8,-2-4i,0).$$

Voorbeeld 1a (Transformatie van een complexe Hermitese datarij (van even lengte)).

Zij

$$(4.2.2.1.3) \qquad f(\frac{2\pi j}{N}) = \sum_{k=0}^{N-1} c_k W_N^{kj}, \; j = 0,1,\ldots,N-1$$

met de complexe Hermitese rij $\{c_k\}_{k=0}^{N-1}$. Het resultaat is reëel.

Algoritme 1a

N = even:

$$f_{2j} + if_{2j+1} = \sum_{k=0}^{N/2-1} \{c_k + c_{N/2+k} + i\, W_N^k(c_k - c_{N/2+k})\}\, W_{N/2}^{jk},$$

$$j = 0,1,\ldots,N/2-1.$$

N = oneven:

via algemene complexe transformatie (aanroepen van FFTP).

Illustratie 1a

Zij

$$\{c_k\} = (1,i,2,-i).$$

De resultaten worden verkregen uit

$$\{f_{2k} + i\, f_{2k+1}\}_{k=0}^{1} = (3-3i,\ 3+i)$$

als

$$\{f_k\}_{k=0}^{3} = (3,-3,3,1).$$

Deze illustratie is geprogrammeerd in FORTRAN en maakt gebruik van C06ABF (zie bijlage).

Voorbeeld 1b (Transformatie van een complexe anti-Hermitese datarij (van even lengte)).

Zij (4.2.2.1.3) gevraagd met een complexe anti-Hermitese datarij $\{c_k\}_{k=0}^{N-1}$. Voor het resultaat geldt Re $f_j = 0$. Het eenvoudigste is het voorbeeld te herleiden tot Voorbeeld 1a door te beschouwen $\{c_k/i\}_{k=0}^{N-1}$.

Algoritme 1b

N = even:

$$f_{2j}-i\, f_{2j+1} = \sum_{k=0}^{N/2-1} \{c_k + c_{N/2+k} - i\, W_N^k\, (c_k-c_{N/2+k})\}\, W_{N/2}^{jk},$$

$$j = 0,1,\ldots,N/2-1$$

met

$$f_{2j} = i\, \mathrm{Im}(f_{2j}-i\, f_{2j+1})$$

$$f_{2j+1} = i\, \mathrm{Re}(f_{2j}-i\, f_{2j+1}).$$

N = oneven:

via algemene complexe transformatie (aanroepen van FFTP).

Illustratie 1b

Zij

$$\{c_k\} = (3i,i,2i,i).$$

De resultaten worden verkregen uit

$$\{f_{2k} - i\, f_{2k+1}\}_{k=0}^{1} \quad (1+7i,1+3i)$$

als

$$\{f_k\}_{k=0}^{3} = (7i,i,3i,i).$$

Deze illustratie is geprogrammeerd in FORTRAN en maakt gebruik van FFT2, FFRDR2 (zie bijlage).

Voorbeeld 2a (Transformatie van een reële symmetrische datarij (van even lengte)).

Zij (4.2.2.1.2) gevraagd met een reële symmetrische datarij $\{r_k\}_{k=0}^{N-1}$. Het resultaat is reëel en symmetrisch.

Algoritme 2a

N = even:

(4.2.2.1.4) $$f_j = \tfrac{1}{2}[a_j+a_{N/2-j}+(a_j-a_{N/2-j})/(2\sin j\pi/(N/2)], \quad j = 1,2,\ldots, N/2-1,$$

$$f_{N-j} = f_j, \quad j = 1,2,\ldots,N/2-1,$$

$$f_0 = a_0 + \sum_{k=0}^{N/2-1} r_{2k+1},$$

$$f_{N/2} = a_0 - \sum_{k=0}^{N/2-1} r_{2k+1},$$

waarin

$$a_j = \sum_{k=0}^{N/2-1} (r_{2k}+i(r_{2k+1}-r_{2k-1}))\, W_{N/2}^{jk}, \quad j = 0,1,\ldots,N/2-1.$$

De berekening van de rij $\{a_k\}$ kan via Algoritme 1a.

N = oneven:

via de algemene complexe transformatie (aanroepen van FFTR).

Illustratie 2a

Zij

$$\{r_k\} = (1,2,3,2).$$

De resultaten worden verkregen uit

$$\{a_k\}_{k=0}^{1} = (4, -2)$$

als

$$\{f_k\}_{k=0}^{3} = (8,-2,0,-2).$$

Voorbeeld 2b (Transformatie van een reële antisymmetrische datarij (van even lengte)).

Zij (4.2.2.1.2) gevraagd met een reële antisymmetrische datarij $\{r_k\}_{k=0}^{N-1}$.
De resulterende rij is Hermites; bovendien geldt Re $f_j = 0$.

Algoritme 2b

N = even:

(4.2.2.1.5) $$f_j = \frac{1}{2i}\left[a_j - a_{N/2-j} - (a_j+a_{N/2-j})/(2\sin j\pi/(N/2))\right], \quad j = 1,2,\ldots,N/2-1$$

$$f_0 = 0,$$

$$f_{N-j} = -f_j,\ j = 1,2,\ldots,N/2,$$

$$a_j = \sum_{k=0}^{N/2-1} (r_{2k+1}-r_{2k-1}+i\ r_{2k})W_{N/2}^{kj},\ j = 0,1,\ldots,N/2-1.$$

De berekening van de rij $\{a_k\}$ kan via Algoritme 1a.

N = oneven:
via de algemene complex transformatie (aanroepen van FFTR).

Illustratie 2b

Zij

$$\{r_k\} = (0,2,0,-2).$$

De resultaten worden verkregen uit

$$\{a_k\}_{k=0}^{1} = (0,8)$$

als

$$\{f_k\}_{k=0}^{3} = (0,4i,0,-4i).$$

Voorbeeld 3 (Transformatie van twee reële datarijen).
Zij

$$a\left(\frac{2\pi j}{N}\right) = \sum_{k=0}^{N-1} x_k W_N^{kj},\ j = 0,1,\ldots,N-1$$

en

$$b\left(\frac{2\pi j}{N}\right) = \sum_{k=0}^{N-1} y_k W_N^{kj},\ j = 0,1,\ldots,N-1,$$

met $\{x_k\} \in \mathbb{R}^N$ en $\{y_k\} \in \mathbb{R}^N$. De resulterende rijen zijn Hermites.

Algoritme 3

$$a_j = \overline{a}_{N-j} = (c_j+\overline{c}_{N-j})/2,\ j = 0,1,\ldots,N\div 2+1$$

$$b_j = \overline{b}_{N-j} = (c_j-\overline{c}_{N-j})/2i,\ j = 0,1,\ldots,N\div 2+1$$

waarin

$$c_j = \sum_{k=0}^{N-1} (x_k+iy_k)W_N^{kj},\ j = 0,1,\ldots,N-1.$$

Illustratie 3
Zij

$$\{x_k\} = (1,2) \text{ en } \{y_k\} = (3,4).$$

De resultaten worden verkregen uit

$$\{c_k\}_{k=0}^{1} = (3+7i, -(1+i))$$

als

$$\{a_k\}_{k=0}^{1} = (3,-1) \text{ en } \{b_k\}_{k=0}^{1} = (7,-1).$$

4.2.2.2. Reële eindige trigonometrische reeks

Zij

$$(4.2.2.2.1) \qquad f(\theta_j) = \sum_{k=0}^{N/2}{}' \; (a_k \cos k\theta_j + b_k \sin k\theta_j)$$

met

$$f(\theta_j) \in \mathbb{R}, \{a_k\}_{k=0}^{N/2} \text{ en } \{b_k\}_{k=1}^{N/2-1} \text{ reëel } (b_0=b_{N/2}=0)$$

en θ_j een hoek. Als $\{a_k\}$, $\{b_k\}$ en θ_j gegeven zijn spreken we van de harmonische synthese; als $\{f_k\}$ en $\{\theta_k\}$ gegeven zijn spreken we van de harmonische analyse (inversie).
We onderscheiden naar:

a. evaluatie voor één hoek of meerdere equidistante hoeken;

b. sinusreeks(en) en/of cosinusreeks(en).

De evaluatie voor één hoek is beschikbaar in ACCULIB en NUMAL. De geïmplementeerde algoritmen berusten op de Clenshawrecursie met de modificatie van Reinsch.

In NUMAL is bovendien de recursie via een orthogonale transformatie gerealiseerd (Hornerschema). Een andere variant, de phase-shift-algoritme (NEWBERY [1973]), is nergens geïmplementeerd; de oorzaak daarvan moet gezocht worden in de inefficiëntie van de algoritme.

Voor N-1 equidistante hoeken is de harmonische analyse beschikbaar in ACCULIB, IMSL en NAG. De synthese is aanwezig in ACCULIB en NAG. Verdere details, omtrent beschikbaarheid, zijn te zien in Tabel 2.

TABEL 2

Implementaties m.b.t. $f(\theta_j) = \sum_{k=0}^{N/2} \{a_k \cos k\theta_j + b_k \sin k\theta_j\}$.

Gegevens specificatie	ACCULIB	IMSL	NAG	NUMAL
$\theta_j = \theta$, evaluatie voor één hoek $\{a_k\}_{k=0}^{N/2}$, $\{b_k\}_{k=1}^{N/2}$	FOUREV	-	-	FOUSER/1/2
$\{a_k = 0\}$, $\{b_k\}_{k=1}^{N/2}$	-	-	-	SINSER
$\{a_k\}_{k=0}^{N/2}$, $\{b_k = 0\}$	-	-	-	COSSER
$\theta_j = j\frac{2\pi}{N}$, $j = 0,1,\ldots,N-1$ $\{a_k\}_{k=0}^{N/2}$, $\{b_k\}_{k=1}^{N/2-1}$	FFTMRA/F	-	C06AAA/F	-
$\{f_j\}_{j=0}^{N-1}$	FFTMRA/F	FFCSIN	C06AAA/F	-

Opmerking (Tabel 2)

De NAG-programmatuur C06AAA/F, en FFTMRA/F bevat een parameter zodat de harmonische analyse of the harmonische synthese uitgevoerd wordt.

In de Voorbeelden 4 t/m 6 geven we aan hoe de complexe Fouriertransformatie te gebruiken is voor de sinustransformatie, cosinustransformatie en in het algemeen de harmonische analyse en synthese.

Voorbeeld 4 (Sinustransformatie)

Zij

$$f(\frac{\pi j}{m}) = \sum_{k=1}^{m-1} b_k \sin \pi\, kj/m$$

$$= i \sum_{k=0}^{N-1} r_k W_N^{kj}, \qquad j = 1,2,\ldots,m-1 \tag{4.2.2.2.2}$$

waarin

$$r_0 = r_m = 0$$
$$r_k = -b_k/2,\ k = 1,\ldots,m-1$$
$$r_{N-k} = b_k/2,\ k = 1,\ldots,m-1$$
$$N = 2m.$$

Algoritme 4

De transformatie (4.2.2.2.2) wordt verkregen uit de transformatie (4.2.2.1.5) van Algoritme 2b.

Illustratie 4

Zij

$$\{b_k\}_{k=1}^{3} = (1,2,3).$$

De resultaten worden verkregen uit (zie Algoritme 2b)

$$\{a_k\}_{k=1}^{3} = (-2,4,-6)$$

als

$$\{f_k\}_{k=1}^{3} = (2+2\sqrt{2},-2,-2+2\sqrt{2}).$$

Voorbeeld 5 (Cosinustransformatie)

Zij

$$f(\tfrac{\pi j}{m}) = \sum_{k=0}^{m}{}'' \alpha_k \cos \pi\, kj/m$$

$$= \sum_{k=0}^{N-1} r_k W_N^{kj}, \qquad j = 0,1,\ldots,m \tag{4.2.2.2.3}$$

waarin

$$r_k = r_{N-k} = \alpha_k/2,\ k = 0,1,\ldots,m,$$
$$N = 2m.$$

Algoritme 5

De transformatie (4.2.2.2.3) wordt verkregen uit de transformatie (4.2.2.1.4) van Algoritme 2a.

Illustratie 5

Zij

$$\{\alpha_k\}_{k=0}^{4} = (2,0,0,2,4).$$

De resultaten worden verkregen uit (zie Algoritme 2a)

$$\{a_k\}_{k=0}^{3} = (3,-3,3,1)$$

als

$$\{f_k\}_{k=0}^{4} = (5, -(1+\sqrt{2}), 3, -1+\sqrt{2}, 1).$$

Voorbeeld 6a (Harmonische synthese (C06AAA/F; FFTMRA/F))

Zij

$$f(\tfrac{\pi j}{m}) = \sum_{k=0}^{m}{}'' a_k \cos \pi\, kj/m + \sum_{k=1}^{m-1} b_k \sin \pi\, kj/m$$

$$= \sum_{k=0}^{N-1} c_k W_N^{kj}, \quad j = 0,1,\ldots,N-1 \qquad (4.2.2.2.4)$$

waarin

$$c_0 = a_0/2, \quad c_m = a_m/2$$

$$c_k = \overline{c}_{N-k} = \tfrac{1}{2}(a_k - ib_k), \quad k = 1,2,\ldots,m-1$$

$$N = 2m.$$

Algoritme 6a

De transformatie (4.2.2.2.4) wordt gegeven in Algoritme 1a.

Illustratie 6a

Zij

$$\{a_k\}_{k=0}^{4} = (2,0,0,2,4) \text{ en } \{b_k\}_{k=1}^{3} = (1,2,3).$$

De resultaten worden verkregen uit (zie Algoritme 1a)

$$\{f_{2k} + i\, f_{2k+1}\}_{k=0}^{3} = (5+i(1+\sqrt{2}),1+i(-3+3\sqrt{2}),1+i(1-\sqrt{2}),$$

$$5-i(3+3\sqrt{2}))$$

als

$$\{f_k\}_{k=0}^{7} = (5,1+\sqrt{2},1,-3+3\sqrt{2},1,1-\sqrt{2},5,-(3+3\sqrt{2})).$$

Voorbeeld 6b (Harmonische analyse (FFCSIN; C06AAA/F; FFTMRA/F))

Zij

$$a_j = \frac{1}{m} \sum_{k=0}^{N-1} f(\tfrac{\pi j}{m}) \cos \pi\, kj/m,\ j = 0,1,\ldots,m$$

$$b_j = \frac{1}{m} \sum_{k=0}^{N-1} f(\tfrac{\pi j}{m}) \sin \pi\, kj/m,\ j = 1,2,\ldots,m-1$$

met

$$N = 2m.$$

Algoritme 6b

(4.2.2.2.5) $$a_j + ib_j = \frac{1}{m} \sum_{k=0}^{N-1} f(\tfrac{\pi j}{m})\, W_N^{kj},\ j = 0,1,\ldots,m.$$

Illustratie 6b

Zij

$$\{f_k\}_{k=0}^{3} = (1,0,3,4).$$

De resultaten worden verkregen uit (zie Algoritme 0)

$$\{a_k + i\, b_k\}_{k=0}^{2} = (4,-(1+2i),0)$$

als

$$\{a_k\}_{k=0}^{2} = (4,-1,0) \text{ en } b_1 = -2.$$

Nog enige opmerkingen omtrent deze sectie.

1. IMSL bevat ook programmatuur m.b.t. spectraalanalyse. De routine die gebruik maakt van de FFT heet FTFFT1. NAG bevat een routine om de circulaire convolutie uit te voeren op twee reële datarijen (C06ACA/F).
2. Het komt voor, dat de factoren

   $$W_N^k \quad , k = 0,1,\ldots,N-1$$

   $$\sin \pi k/N, \; k = 1,2,\ldots,N-1$$

   berekend moeten worden. OLIVER [1975] heeft gewezen op de algoritme van Hopgood en Litherland; een gegeneraliseerde ALGOL 68 implementatie is opgenomen in de bijlage.
3. De afleidingen van de Algoritmen zijn te vinden in COOLEY (p.280 in RABINER [1972]).

4.2.3. Testvoorbeelden

Voor het testen van de programmatuur hebben wij een drietal relaties geverifieerd.

a. vergelijken met exacte waarden;
b. verificatie van relatie van Parseval;
c. verificatie van transformatie-inversie paar.

Voor de genoemde gevallen hebben wij de "evaluatie van een trigonometrische reeks voor één punt" toegepast op de punten die ook gebruikt worden bij de FFT-implementaties, en daarmede de testvoorbeelden algemeen toepasbaar gemaakt. Bovendien is hiermede het vóór-FFT tijdperk gesimuleerd.

Voor de meting van de relatieve nauwkeurigheid gebruiken we de formule

$$\varepsilon = \| \text{ absolute fout } \|_{rms} / \| \text{exacte waarde} \|_{rms}$$

met $\|f\|_{rms} = \{ \sum_{k=1}^{N} |f_k|^2/N \}^{\frac{1}{2}}$.

In het verdere verloop zullen we de berekende grootheden met $\tilde{F}$ en de exacte grootheden met F aanduiden. De grootheid t is de tijd nodig om $\tilde{F}$ te berekenen.

De tijden en nauwkeurigheden van de programmatuur voor de drie testvoorbeelden is opgenomen in Tabel 3. De theoretische nauwkeurigheid en efficiëntie is ruwweg evenredig met $N^2 \log N$. Voor meer specifieke details zie GENTLEMAN & SANDE [1966], GENTLEMAN [1969] of RAMOS [1971]. Aan het einde van deze sectie hebben we bovendien Tabel 4 opgenomen, die de modulariteit illustreert.

4.2.3.1. Nadere specificaties van testrelaties

Probleem I $\quad f(\frac{2\pi j}{N}) = \sum_{k=0}^{N-1} c_k W_N^{kj}, \; j = 0,1,\ldots,N-1.$

Ia. Modelprobleem

Zij

$$c_k = c^k \in C,$$

dan geldt, dat de berekende waarde, $\tilde{F}$, bestaat uit de componenten

$$\tilde{F}_j = \sum_{k=0}^{N-1} c^k W_N^{kj}, \; j = 0,1,\ldots,N-1.$$

Voor de exacte waarde, F, verkrijgen wij de componenten

$$F_j = \sum_{k=0}^{N-1} c^k W_N^{kj} = \frac{(1-c^N)(1-\bar{c}\,\bar{W}_N^{kj})}{1+|c|^2-2\mathrm{Re}(cW_N^j)}, \; j = 0,1,\ldots,N-1.$$

Ib. Relatie van Parseval

De relatie van Parseval is gegeven door

$$\sum_{j=0}^{N-1} |f(\tfrac{2\pi j}{N})|^2 = N \sum_{k=0}^{N-1} |c_k|^2.$$

De berekende grootheid, $\tilde{F}$, is

$$\tilde{F} = \sum_{j=0}^{N-1} \left| f\left(\frac{2\pi j}{N}\right) \right|^2 .$$

Als exacte waarde, F, verkrijgen we, voor het speciale geval $c_k = c^k$

$$F = N\frac{(1-|c|^{2N})}{1-|c|^2} .$$

Ic. Transformatie-inversie paar

Als berekende grootheid $\tilde{F}$ nemen we de vector bestaande uit de componenten

$$\tilde{F}_j = \frac{1}{N} \sum_{k=0}^{N-1} \bar{W}_N^{jk} \sum_{\ell=0}^{N-1} W_N^{k\ell} c_\ell, \quad j = 0,1,\ldots,N-1.$$

De exacte waarde is de oorspronkelijke vector c.

Probleem II

$$f\left(\frac{\pi j}{m}\right) = \sum_{k=0}^{m}{}'' a_k \cos \pi kj/m + \sum_{k=1}^{m-1} b_k \sin \pi kj/m, \quad j = 0,1,\ldots,2m-1.$$

IIa. Modelprobleem

Zij

$$a_k = r_c^k, \; b_k = r_s^k \text{ met } |r_c| < 1, \; |r_s| < 1$$

dan wordt de berekende grootheid, $\tilde{F}$, gegeven door de vector met componenten

$$\tilde{F}_j = \sum_{k=0}^{m}{}'' r_c^k \cos \pi kj/m + \sum_{k=1}^{m-1} r_s^k \sin \pi kj/m, \quad j = 0,1,\ldots,2m-1.$$

De exacte grootheid, F, is de vector bestaande uit de componenten

$$F_j = \tfrac{1}{2}\frac{(1-r_c^2)(1-(-1)^j r_c^m)}{1+r_c^2-2r_c \cos \pi j/m} + \frac{r_s \sin \pi j/m \,(1-(-1)^j r_s^m)}{1+r_s^2-2r_s \cos \pi j/m}$$

$$j = 0,1,\ldots,N-1; \; m \geq 1.$$

IIb. Relatie van Parseval

De relatie van Parseval is gegeven door

$$\sum_{j=0}^{2m-1} f(\tfrac{\pi j}{m})^2 = m\Big(\sum_{k=0}^{m}{}'' a_k^2 + \sum_{k=1}^{m-1} b_k^2\Big).$$

Voor het speciale geval $a_k = r_c^k$, $b_k = r_s^k$, $|r_c| < 1$, $|r_s| < 1$, worden de berekende en exacte grootheden gegeven door respectievelijk

$$\tilde{F} = \sum_{j=0}^{2m-1} \Big\{ \sum_{k=0}^{m}{}'' r_c^k \cos \pi kj/m + \sum_{k=1}^{m-1} r_s^k \sin \pi kj/m \Big\}^2$$

en

$$F = m\left(\frac{1-r_c^{N+2}}{1-r_c^2} + \frac{r_s^2(1-r_s^{N-2})}{1-r_s^2} - \tfrac{1}{2}(1+r_c^N)\right), \quad N = 2m.$$

IIc. Transformatie-inversie paar

De berekende grootheid wordt gegeven, met m = N/2, door de componenten

$$\tilde{F}_k = \tilde{a}_k = \frac{1}{m}\sum_{j=0}^{N-1} \cos \pi kj/m\Big(\sum_{\ell=0}^{m}{}'' a_\ell \cos \pi \ell j/m + \sum_{\ell=1}^{m-1} b_\ell \sin \pi \ell j/m\Big),$$

$$k = 0,1,\ldots,m;$$

$$\tilde{F}_{m+k} = \tilde{b}_k = \frac{1}{m} \sum_{j=0}^{N-1} \sin\pi kj/m \Big(\sum_{\ell=0}^{m}{}'' a_\ell \cos\pi \ell j/m + \sum_{\ell=1}^{m-1} b_\ell \sin\pi \ell j/m \Big),$$

$k = 1,\ldots,m-1.$

De exacte waarden worden gegeven door de oorspronkelijke coëffiënten

$$\{a_k\}_{k=0}^{m}, \ \{b_k\}_{k=1}^{m-1}.$$

Bij een cosinustransformatie respectievelijk een sinustransformatie vinden we voor de berekende grootheden

$$\tilde{a}_k = \frac{2}{m} \sum_{j=0}^{m}{}'' \cos\pi kj/m \sum_{\ell=0}^{m}{}'' a_\ell \cos\pi \ell j/m, \qquad k = 0,1,\ldots,m;$$

$$\tilde{b}_k = \frac{2}{m} \sum_{j=1}^{m-1} \sin\pi kj/m \sum_{\ell=1}^{m-1} b_\ell \sin\pi \ell j/m, \qquad k = 1,\ldots,m-1.$$

Opmerkingen (Tabel 3)

1. De tijd is gemeten in milliseconden en is onnauwkeurig met ~ 20ms t.g.v. multi-processing van de rekenmachine; deze tijden geven slechts informatie omtrent de orde van grootte.

2. Bij FFTP zijn twee tijden vermeld. De eerste geldt voor N = 1024 en de tweede voor N = 1000. Bij N = 1000 zijn onjuiste resultaten geconstateerd (deze zijn overigens door IMSL ook gesignaleerd en in editie 5 gecorrigeerd).

3. De inversie van de getransformeerde van een reële datarij kan elegant geschieden (via Voorbeeld en Algoritme 1a en een conjugatie); bij de documentatie in de programmatheken staat dit niet vermeld en een onschuldige gebruiker moet daartoe FFT2 gevolgd door FFRDR2 aanroepen. Dit is inefficiënt.

4. De transformatie corresponderend met de inversie FFCSIN is niet aangegeven. Om toch resultaten te verkrijgen zouden we de algoritme gegeven in Voorbeeld 6a kunnen gebruiken of C06AAA/F of FFTMRA/F.

5. De testvoorbeelden zijn gedraaid met

$$r_c = r_s = c = \tfrac{3}{4}.$$

TABEL 3

Testresultaten.

Probleem/gegevens specificatie	Implementatie	Testvoorbeelden, N = 1024					
		I		II	III		
		t	ε	ε	t		ε
$f(\theta_j) = \sum_{k=0}^{N-1} c_k e^{\pm ik\theta_j}$							
$\theta_j = \theta$ (slechts één hoek)							
$\{c_k\}_{k=0}^{N-1} \in \mathbb{C}^N$	COMFOUSER 1 (NUMAL)	71000	$.6_{10}^{-13}$	$.8_{10}^{-14}$	149000	$.8_{10}^{-11}$	
	COMFOUSER 2 (NUMAL)	93000	$.6_{10}^{-13}$	$.3_{10}^{-13}$	186000	$.8_{10}^{-11}$	
$\{c_k\}_{k=0}^{N-1} \in \mathbb{R}^N$	COMFOUSER (NUMAL)	47000	$.6_{10}^{-13}$	$.3_{10}^{-13}$		-	
$\theta_j = j\frac{2\pi}{N}$, j = 0,1,...,N-1 (FFT)							
$\{c_k\}_{k=0}^{N-1} \in \mathbb{C}^N$	FFTP (IMSL)	154/272	$.4_{10}^{-13}$	$.7_{10}^{-13}$	326/552	$.7_{10}^{-13}$	
	FFT2 (IMSL)	119	$.3_{10}^{-13}$	$.5_{10}^{-13}$	239	$.5_{10}^{-13}$	
	FFT2RV (IMSL)	120	$.4_{10}^{-13}$	$.6_{10}^{-13}$	250	$.5_{10}^{-13}$	

Tabel 3 (vervolg)

Probleem/gegevens specificatie	Implementatie	Testvoorbeelden, N = 1024					
		I		II	III		
		t	ε	ε	t	ε	
	C06ABA/F (NAG)	114	$.1_{10}^{-12}$	$.3_{10}^{-13}$	229	$.2_{10}^{-13}$	
	C06ADA/F (NAG)	132	$.1_{10}^{-12}$	$.9_{10}^{-14}$	301	$.1_{10}^{-13}$	
	FFTMCA/F (ACCU)	116	$.4_{10}^{-13}$	$.8_{10}^{-13}$	229	$.6_{10}^{-13}$	
$\{c_k\}_{k=0}^{N-1} \in \mathbb{R}^N$	FFTR (IMSL)	86	$.1_{10}^{-12}$	$.5_{10}^{-13}$		-	
$f(\theta_j) = \sum_{k=0}^{N/2}{}' \{a_k \cos k\theta_j + b_k \sin k\theta_j\}$							
$\theta_i = \theta$ (slechts één hoek)							
$\{a_k\}_{k=0}^{N/2}, \{b_k\}_{k=1}^{N/2-1} \in \mathbb{R}$	FOUSER (NUMAL	23000	$.5_{10}^{-12}$	$.9_{10}^{-13}$		-	
	FOUSER 1 (NUMAL)	35000	$.5_{10}^{-12}$	$.7_{10}^{-13}$		-	
	FOUSER 2 (NUMAL)	46000	$.5_{10}^{-12}$	$.9_{10}^{-13}$		-	
	FOUREV (ACCU)	3600 (513p)	$.2_{10}^{-11}$	$.2_{10}^{-11}$		-	
$\{a_k = 0\}, \{b_k\}_{k=1}^{N/2-1} \in \mathbb{R}$	SINSER (NUMAL)	1100 (512p)	$.4_{10}^{-12}$	$.6_{10}^{-14}$	22_{10}^{3}	$.3_{10}^{-13}$	

Tabel 3 (vervolg)

Probleem/gegevens specificatie	Implementatie	Testvoorbeelden, N = 1024					
		I		II	III		
		t	ε	ε	t	ε	
$\{a_k\}_{k=0}^{N/2}$, $\{b_k = 0\} \in \mathbb{R}$	COSSER (NUMAL)	11000 (512p)	$.4_{10}^{-12}$	$.1_{10}^{-12}$	23000	$.2_{10}^{-13}$	
$f(\theta_j) = \sum_{k=0}^{N/2}{}''\{a_k \cos k\theta_j + b_k \sin k\theta_j\}$							
$\theta_j = j\frac{2\pi}{N}$, $j = 0,1,\ldots,N-1$							
$\{a_k\}_{k=0}^{N/2}$, $\{b_k\}_{k=1}^{N/2-1} \in \mathbb{R}$	FFTMRA/F(ACCU)	87	$.4_{10}^{-12}$	$.9_{10}^{-13}$	158	$.9_{10}^{-13}$	
	C06AAA/F (NAG)	64	$.4_{10}^{-12}$	$.5_{10}^{-13}$	124	$.3_{10}^{-12}$	
$\{f_j\}_{j=0}^{N-1} \in \mathbb{R}$	FFCSIN (IMSL)	63	$.4_{10}^{-12}$	$.3_{10}^{-12}$		-	
	FFTMRA/F(ACCU)	67	$.4_{10}^{-12}$	$.3_{10}^{-12}$	138	$.1_{10}^{-12}$	
	C06AAA/F NAG)	56	$.4_{10}^{-12}$	$.3_{10}^{-12}$	117	$.3_{10}^{-12}$	

TABEL 4

Routine/procedure afhankelijkheid.

Implementatie	Hulpimplementatie
C06AAA	C06AA9, C06AA8, C06AA7, C06AA6
C06AAF	C06AB8, C06AB9
C06ABA	C06AAY, C06ABZ
C06ABF	C06AA9, C06AA8, C06AA7, C06AB9
C06ACA	C06AAZ, C06AAY, C06AAX, C06ABZ
C06ACF	C06AAZ, C06AAY, C06AAX, C06AAW
C06ADA	-
C06ADF	-
COMFOUSER	-
COMFOUSER 1	-
COMFOUSER 2	COMFOUSER
COSSER	-
FFCSIN	FFRDR2, FFTR FFTP, FFT2
FFRDR2	-
FFT2	-
FFT2RV	-
FFTMCA	-
FFTMCF	-
FFTMRA	FFTMCA
FFTMRF	FFTMCF
FFTP	-
FFTR	FFRDR2, FFTP, FFT2
FOUREV	-
FOUSER	-
FOUSER 1	-
FOUSER 2	COSSER, SINSER
FTFFT1	FFRDR2, FFTP, FFTR, FFT2
SINSER	-

Opmerkingen (Tabel 4)

1. Een cross-reference tabel is alleen opgenomen in IMSL.

2. De hulpimplementaties in NAG zijn niet gedocumenteerd.

4.2.4. Verband tussen Fourierintegraal en discrete Fouriertransformatie

4.2.4.1. Niet-periodieke functies

Zij $x(t)$, $t \in (-\infty,\infty)$ en $a(f)$, $f \in (-\infty,\infty)$ een Fourierpaar d.w.z.

$$a(f) = \int_{-\infty}^{\infty} x(t)\, e^{-2\pi i f t}\, dt, \quad f \in (-\infty,\infty),$$

$$x(t) = \int_{-\infty}^{\infty} a(f)\, e^{2\pi i f t}\, df, \quad t \in (-\infty,\infty),$$

dan geldt dat

$$T\, x^P(k\,\Delta t) \text{ en } a^P(k\,\Delta f)$$

een discreet Fourierpaar vormen. In formules (DFT):

$$(4.2.4.1) \quad a^P(j\,\Delta f) = \frac{1}{N} \sum_{k=0}^{N-1} T\, x^P(k\,\Delta t)\, W_N^{kj}, \quad j = 0,1,\ldots,N-1$$

$$(4.2.4.2) \quad T\, x^P(j\,\Delta t) = \sum_{k=0}^{N-1} a^P(k\,\Delta f)\, W_N^{kj}, \quad j = 0,1,\ldots,N-1$$

met als aliasfuncties

$$x^P(t) = \sum_{k=-\infty}^{\infty} x(t + k\,T),$$

$$a^P(f) = \sum_{k=-\infty}^{\infty} a(f + k\,F),$$

$$N\,\Delta f = F, \quad N\,\Delta t = T, \quad \Delta f = 1/T.$$

(Zie COOLEY, p. 280 in RABINER [1972].)
De belangrijkste gevolgtrekking, die we hieruit kunnen maken, is dat bij discreet Fouriertransformeren we een transformatie tussen de vectoren x^P en a^P uitvoeren en niet tussen de vectoren x en a, i.e. niet tussen de discrete functiewaarden. In sectie 4.2.4.1.1. geven we een methode om van het zogenaamde *aliaseffect* af te komen. Het begrip Nyquistfrequentie is niet meer relevant

4.2.4.1.1. Berekening van a(f) in een aantal equidistante punten

Als benadering kunnen we vragen naar a in de punten $\{f_k\} = (0,\Delta f,\ldots,(N-1)\Delta f,N\Delta f,\ldots)$. Voor $a(f_k)$ kunnen we schrijven

$$(4.2.4.3)\quad a(f_k) = \frac{1}{T}\int_0^T T\, x^P(t)\, e^{-2\pi ikt/T} dt,\ k = 0,1,\ldots,N-1,N,\ldots$$

met $T = 1/\Delta f$.

Onder meer STOER [1972] en GAUTSCHI [1972] hebben laten zien dat (4.2.4.3) te schrijven is als

$$(4.2.4.4)\quad a(f_k) = \tau_k\, a^P(k\ \Delta f) + R_k^N$$

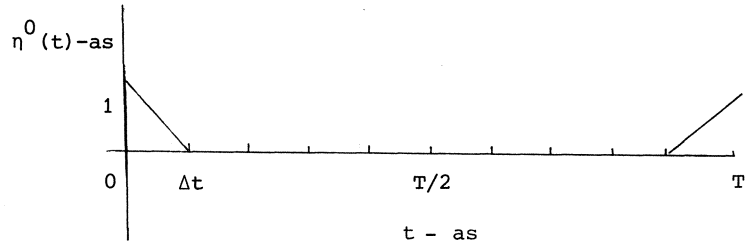
met de verzwakkingsfactoren

$$(4.2.4.5)\quad \tau_k = \frac{N}{T}\int_0^T \eta^0(t)\, e^{-2\pi ikt/T}\, dt,$$

waarin $\eta^0(t)$ een continue periodieke functie is die linear en translatie-invariant (REINSCH in STOER [1972]) de datapunten $((0,1),(\Delta t,o),\ldots,((N-1)\Delta t,0))$ approximeert, en restterm

$$R_j^N = \int_0^T \{x^p(t) - \sum_{k=0}^{N-1}(x_k^P - \eta^0(t-k\Delta t)\}\, e^{-2\pi ijt/T}\, dt.$$

Voorbeeld 7 (Berekening van $\{\tau_k\}_{k=0}^{\infty}$).
Zij het approximatieproces de lineaire interpolatie. $\eta^0(t)$ wordt dan gegeven door



Uitwerking van (4.2.4.5) geeft

$$(4.2.4.6)\quad \tau_k = \frac{N}{T}\int_{-\Delta t}^{\Delta t} (1-\frac{|t|}{\Delta t})\, e^{-2\pi ikt/T}\, dt$$

$$= \{\frac{1}{\pi k/N} \sin \pi k/N\}^2,\ k = 0,1,\ldots\ .$$

Opmerking

Voor andere τ-factoren zie GAUTSCHI [1972] of BAUER & STETTER [1959].

Voorbeeld 8 (Berekening van $a(f_j)$)
Zij

$$x(t) = e^{-t},\ t \geq 0$$
$$= 0\ ,\ t < 0$$

met Fouriergetransformeerde

$$a(f) = \frac{1}{1+2\pi if}\ .$$

De benaderingen $\tilde{a}_k$ van (4.2.4.4), met τ_k zoals gegeven door (4.2.4.6) zijn

$$(4.2.4.7)\quad \tilde{a}_k = \frac{((\sin\pi k/N)/(\pi k/N))^2}{F(1-e^{-1/F-2\pi k/N})}\ ,\ k = 0,1,\ldots\ .$$

In Tabel 5 geven we een overzicht van de resultaten.

TABEL 5

Reële deel Fouriergetransformeerde van e^{-t}; $t \geq 0$.

k	Re a_k	Re a_k^P	Re $\tilde{a}_k$
0	1	$\frac{1}{F(1-e^{-1/F})}$	$\frac{1}{F(1-e^{-1/F})}$
N/2	$\frac{1}{1+\pi F^2}$	$\frac{1}{F(1+e^{-1/F})}$	$\frac{1}{F\pi^2(1+e^{-1/F})}$

4.2.4.2. Periodieke functies

Zij $f(\theta)$ periodiek met periode T. De Fouriercoëfficiënten worden gegeven door

$$(4.2.4.8) \quad c_k = \frac{1}{T}\int_0^T f(\theta)\, e^{-2\pi ik\theta/T}\, d\theta, \quad k = 0,\pm1,\pm2,\ldots .$$

De integralen (4.2.4.8) kunnen berekend worden via (4.2.4.4) met $Tx^P = f$.

Opmerking

Een andere technische uitwerking is de functie f te splitsen in een symmetrisch en antisymmetrisch gedeelte.
Van ieder gedeelte kan men de Fourierontwikkeling verkrijgen (zie sectie 4.2.5 en 4.2.6).

Voorbeeld 9 (Berekening van de Fouriercoëfficiënten van een periodieke functie)

Zij

$$f(\theta) = \sin\theta.$$

De Fouriercoëfficiënten, (4.2.4.8), worden benadered via (4.2.4.4) als

$$\tilde{c}_k = \begin{cases} \tau_1/2i & , k = 1,\ N > 2 \\ -\tau_{N-1}/2i, & k = N-1 \\ 0 & , k \neq 1,\ N-1 \end{cases}$$

In Tabel 6 geven we een overzicht van de resultaten met τ_k zoals gegeven door (4.2.4.6)

TABEL 6

Fouriercoëfficiënten van sin θ

k	c_k	c_k^P	$\tilde{c}_k$
1	$\frac{1}{2i}$	$\frac{1}{2i}$	$\frac{1}{2i}(1-(\pi/N)^2/3+\ldots)$
N-1	0	$-\frac{1}{2i}$	$-\frac{1}{2i}(\pi/N-(\pi/N)^3/3!+\ldots)^2 \times (1+\pi/N+\ldots)^2/\pi^2$
k	0	0	0, $1 < k < N-1$
$k > N$	0	periodiek mod N	$0(k^{-2})$, $k > N$

Opmerkingen

1. De berekening van integralen van het type

$$(4.2.4.9) \qquad \int_0^1 f(\theta)\, e^{ik\theta}\, d\theta$$

met f een niet-periodieke functie zijn hier niet beschouwd; de geïnteresseerde lezer wordt verwezen naar LYNESS [1974], of overweeg ontwikkeling van $g(\theta) = f(\cos\theta)$. De convergentie van de bijbehorende Fourrierreeksen is een andere zaak; literatuur hiervoor is BARY [1964] of ZYGMUND [1959].

2. Het verschijnsel van lekkage, d.w.z. de resultaten zijn verstoord omdat men de periode van een periodieke functie niet kent, kan men ontduiken door de functie als niet-periodiek te beschouwen met inachtneming van randcorrecties of door de periode te schatten.

4.2.5. Fourierontwikkeling van een symmetrische periodieke functie

Zij f symmetrisch en periodiek. De Fouriercoëfficiënten worden gegeven door

$$(4.2.5.1) \qquad c_k = \frac{1}{2\pi}\int_0^{2\pi} f(\theta)e^{-ik\theta}\, d\theta.$$

De benaderde coëfficiënten worden verkregen met behulp van (4.2.4.4) als

$$(4.2.5.2)\quad \tilde{c}_k = \frac{\tau_k}{N} \sum_{j=0}^{N-1} f(2\pi j/N)\, W_N^{kj},\ k = 0,1,\ldots .$$

De uitwerking kan efficiënt via Algoritme 2a.

Illustratie 10 (Fourierontwikkeling van $\cos^2\theta$ of Chebyshevreeks van x^2)

Zij $f(\theta) = \cos^2\theta$.
De benaderde Fouriercoëfficiënten (4.2.5.2), voor N = 4, worden gegeven door

$$\{\tilde{c}_k\}_{k=0}^{3} \quad (1/2, 0, \tau_2/2, \tau_3/2).$$

In Tabel 7 vatten we de resultaten samen met τ_k zoals gegeven in (4.2.4.6).

TABEL 7

Fouriercoëfficiënten van $\cos^2\theta$; N = 4.

k	c_k	c_k^P	$\tilde{c}_k$
0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$
1	0	0	0
2	$\frac{1}{4}$	$\frac{1}{4}$	$8/(2\pi)^2$
3	0	$\frac{1}{4}$	$4/(3\pi)^2$
4	0	0	0
5	0	$\frac{1}{4}$	$4/(5\pi)^2$

4.2.6. Fourierontwikkeling van een antisymmetrische periodieke functie

Zij f antisymmetrisch en periodiek. De Fouriercoëfficiënten worden gegeven door

$$(4.2.6.1)\quad c_k = \frac{1}{2\pi}\int_0^{2\pi} f(\theta)\, e^{-ik\theta}\, d\theta.$$

De benaderde Fouriercoëfficiënten worden verkregen met behulp van (4.2.4.4) als

$$(4.2.6.2)\quad \tilde{c}_k = \frac{\tau_k}{N}\sum_{j=0}^{N-1} f(2\pi j/N)\, \overline{W}_N^{kj},\ k = 0,1,\ldots .$$

De uitwerking kan efficiënt via Algoritme 2b.
In Voorbeeld 9 en Tabel 6 is de ontwikkeling geïllustreerd voor het geval $f(\theta) = \sin\theta$.

4.2.7. Warmtegeleiding en trillende snaar

De warmtevergelijking wordt gegeven door

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

Als randcondities kunnen we bijvoorbeeld onderscheiden

$$(4.2.7.1)\quad u = 0 \quad \text{voor } x = 0 \text{ en } x = \pi,$$
$$u = f(x) \text{ voor } t = 0.$$

$$(4.2.7.2)\quad u_x = 0 \quad \text{voor } x = 0 \text{ en } x = \pi,$$
$$u = g(x) \text{ voor } t = 0.$$

Als oplossingen verkrijgen we, LAUWERIER [1968]

$$(4.2.7.3)\quad u(x,t) = \sum_{k=1}^{\infty} a_k \sin kx\, e^{-k^2 t}$$

met

$$(4.2.7.4)\quad a_k = \frac{2}{\pi}\int_0^{\pi} f(x)\sin kx\, dx = \overline{2i\left(\frac{1}{2\pi}\int_{\pi}^{\pi} f(x)e^{-ikx}\, dx\right)},\ f(-x) = -f(x)$$

voor (4.2.7.1) en

$$(4.2.7.5)\quad u(x,t) = \sum_{k=0}^{\infty}{}' b_j \cos kx\, e^{-k^2 t}$$

met

$$(4.2.7.6)\quad b_k = \frac{2}{\pi}\int_0^{\pi} g(x)\cos kx\,dx = 2\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} g(x)e^{-ikx}\,dx\right),\ g(-x) = g(x).$$

voor (4.2.7.2).

De berekening van de integralen (4.2.7.4) en (4.2.7.6) kan geschieden via sectie 4.2.6 en 4.2.5, respectievelijk.

Illustratie 11

Zij

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t},\ 0 < x < \pi,$$

$$u_x = 0, \qquad \text{voor } x = 0 \text{ en } x = \pi,$$
$$u = \cos^2 x, \text{ voor } t = 0.$$

De oplossing wordt gegeven door

$$u(x,t) = \tfrac{1}{2}(1+\cos 2x\, e^{-4t}).$$

Een benaderde oplossing verkrijgen we via Tabel 7 als

$$\tilde{u}(x,t) = \tfrac{1}{2} + 4/\pi^2 \cos 2x\, e^{-4t} + \ldots$$

Illustratie 12

Zij

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t},\ 0 < x < \pi$$

$$u = 0 \quad , \text{ voor } x = 0 \text{ en } x = \pi$$
$$u = \sin x, \text{ voor } t = 0.$$

De oplossing wordt gegeven door

$$u(x,t) = \sin x\, e^{-t}.$$

Een benaderde oplossing verkrijgen we via Tabel 6 als

$$\tilde{u}(x,t) = (1-(\pi/N)^2/3)\sin x\, e^{-t} + \ldots$$

De beweging van een trillende snaar wordt beschreven door

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2} - F(x,t),\ 0 < x < \pi$$

$$u = 0 \qquad \text{voor } x = 0 \text{ en } x = \pi \tag{4.2.7.7}$$

$$\left.\begin{aligned} u &= f(x) \\ u_t &= g(x) \end{aligned}\right\} \text{ voor } t = 0.$$

Hier beschouwen we het speciale geval $F(x,t) = 0$; voor $F(x,t) \neq 0$ wordt verwezen naar LAUWERIER [1968]. Als oplossing verkrijgen we, LAUWERIER [1968],

$$u(x,t) = \sum_{k=1}^{\infty} (a_k \cos kt + b_k \sin kt) \sin kx$$

met

$$a_k = \frac{2}{\pi}\int_0^{\pi} f(x)\sin nx\,dx = \overline{2i\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-ikx}dx\right)},\ f(-x) = -f(x) \tag{4.2.7.8}$$

en

$$b_k = \frac{2}{\pi k}\int_0^{\pi} g(x)\sin nx\,dx = \frac{2i}{k}\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} g(x)e^{-ikx}dx\right),\ g(-x) = -g(x). \tag{4.2.7.9}$$

De berekening van de integralen (4.2.7.8) en (4.2.7.9) kan geschieden via sectie 4.2.6.

Illustratie 13

Zij

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

$u = 0$, voor $x = 0$, $x = \pi$, $t = 0$,

$u_t = \sin x$, $t = 0$.

De oplossing wordt gegeven door

$$u(x,t) = \sin t \sin x.$$

Een benaderde oplossing verkrijgen we via Tabel 6 als

$$\tilde{u}(x,t) = (1-(\pi/N)^2/3) \sin t \sin x + \ldots$$

Tenslotte zij opgemerkt dat de Fourierreeksontwikkeling ook gebruikt kan worden bij de potentiaalvergelijking.

4.2.8. Deconvolutie

Het hier gesignaleerde probleem komt neer op het oplossen van een integraalvergelijking

$$(4.2.8.1) \quad y(s) = \int_{-\infty}^{\infty} h(s-t)\, x(t)\, dt,$$

waarin $y(t)$ en $h(t)$ discreet gegeven zijn en $x(t)$ opgelost moet worden. Met behulp van de convolutiestelling kunnen we de Fouriergetransformeerde van (4.2.8.1) schrijven als

$$(4.2.8.2) \quad \int_{-\infty}^{\infty} y(t) e^{-2\pi i f t} dt = Fx \int_{-\infty}^{\infty} h(t) e^{-2\pi i f t} dt.$$

De integralen (4.2.8.2) kunnen benaderd worden door (zie sectie 4.2.4)

(4.2.8.3) $\tilde{b}_\gamma = \frac{\tau_1}{N_1} \sum_{k=0}^{N-1} T_1 y_k^p \bar{w}_{N_1}^{jk}$, $\tilde{a}_j = \frac{\tau_2}{N_2} \sum_{k=0}^{N-1} T_2 h_k^p \bar{w}_{N_2}^{jk}$

waarin

(4.2.8.4) $y_k^p = \sum_{j=-\infty}^{\infty} y(k\Delta t_1 + jT_1)$, $h_k^p = \sum_{j=-\infty}^{\infty} h(k\Delta t_2 + jT_2)$.

Het discrete analogon van (4.2.8.2), voor $N_1 = N_2$, luidt

$$\tilde{b}_j = \tilde{a}_j (Fx)_j, \quad j = 0,1,\ldots .$$

De discrete Fouriergetransformeerde van x kan men dan schrijven als

(4.2.8.5) $Fx_j = \tilde{b}_j / \tilde{a}_j$, $j = 0,1\ldots$

onder de voorwaarde $\tilde{a}_j$ *voldoende* groot.

Er dient opgemerkt te worden dat bij gebruik van gelijke τ-factoren, bij $\tilde{a}_j$ en $\tilde{b}_j$, het quotiënt een periodieke functie is.
De berekening van een benadering van x(t) kan verlopen als volgt.
Zij

$$r(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i f t} dt$$

dan geldt voor de inverse

(4.2.8.6) $x(t) = \int_{-\infty}^{\infty} r(f) e^{2\pi i f t} df.$

Deze integraal kan benaderd worden met

(4.2.8.7) $\tilde{x}_j = \frac{\tau_j}{N} \sum_{k=0}^{N-1} Fr_k^p \, w_N^{jk}$

waarin

$$r_k^p = \sum_{j=-\infty}^{\infty} r(k\Delta f + jF)$$

respectievelijk

$$Fr_k^p = r(k\Delta f)$$

voor een niet-periodieke, respectievelijk periodieke r(f).

Opmerkingen

1. De geschetste methode is, in Groningen, in gebruik bij het elimineren van de onscherptes die veroorzaakt worden door de meetinstrumenten. Zoals in (4.2.8.5) al naar voren kwam is er nog onderzoek nodig voor een criterium wanneer $\tilde{a}_j$ te klein is. MILLER (ongepubliceerd) stelt een "Roll-off" procedure voor.
2. Een andere manier van benaderen nl.

$$\int_{-\infty}^{\infty} h(s-t)x(t)dt = \int_{-N}^{N} h(s-t)x(t)dt + R_N$$

en dan de integraal discretiseren geeft aanleiding tot het stelsel

$$\sum_{j=-M}^{M} h(s_i-t_j)x(t_j) = y(s_i),\ i = -M,\ldots,M.$$

Een voor de hand liggende manier is dit stelsel op te lossen via directe methodes. Daarbij kampt men al gauw met de grootte van de matrix. Dit stelsel kan ook via een discrete convolutiestelling en beperkingen, opgelegd aan de h, opgelost worden m.b.v. Fouriertransformaties. Wij hebben (4.2.8.5) wat anders afgeleid i.v.m. de relatie tussen het continue en discrete probleem. In dit verband moet men de opmerkingen plaatsen omtrent matrix-inversie via de FFT.

Slotwoord

De leden van de werkgroep approximatie van functies wil ik bedanken voor hun kritische opmerkingen; Adri den Arend wil ik bedanken voor het invullen van Tabel 3 en enige suggesties tot verbeteringen.

Literatuur

BARY, N.K. [1964], *A treatise on trigonometric series*, Pergamon Press.

BAUER, F.L. & H.J. STETTER [1959], *Zur numerischen Fourier-Transformation*, Numer. Math. 1, 208-220.

BRIGHAM, E.O. [1974], *The fast Fourier transform*, Prentice Hall.

GAUTSCHI, W. [1972], *Attenuation factors in practical Fourier analysis*, Numer. Math. 18, 373-400.

GENTLEMAN, W.M. [1969], *An error analysis of Goertzel's (Watt's) method for computing Fourier coefficients*, Comput. J. 12, 160-165.

GENTLEMAN, W.M. & G. SANDE [1966], *Fast Fourier transforms - for run and profit*, Proc. AFIPS 29, 563-578.

HELMS, H.D. & L.R. RABINER (eds.) [1973], *Literature in digital signal processing*, IEEE, Wiley.

LANCZOS, C. [1966], *Discourse on Fourier series*, Oliver & Boyd.

LAUWERIER, H. [1968], *Randwaarde problemen*, deel 2, MC Syllabus 3.2.

LYNESS, J.N. [1974], *Computational techniques based on the Lanczos representation*, Math. Comp. 28, 81-123.

MILLER, G.F. (persoonlijke mededeling), *"Roll-of" procedure for integral equations of the convolution type.*

NEWBERY, A.C.R. [1973], *Error analysis for Fourier series evaluation*, Math. Comp. 27, 123, 639-644.

OLIVEIRA-PINTO, F. [1971], *Trigonometric curve fitting to equally or unequally spaced data, Algorithm 69* (ALGOL 60), Comp. J. 14, 213-214.

OLIVER, J. [1975], *Stable methods for evaluating the points cos* $i\pi/N$, J. Inst. Maths. Applics. 16, 247-257.

OUDSHOORN, H.L. [1976], *Fast Fourier Transform*, ACCU-reeks 15.

RABINER, L.R. & C.M. RADER (eds.) [1972], *Digital signal processing*, IEEE, Wiley.

RAMOS, G.U. [1971], *Roundoff error analysis of the FFT*. Math. Comp. 25, 757-768.

SINGLETON, R.C. [1968], *Algorithms 338,339,345*, Comm. ACM.

STETTER, H. [1966], *Numerical approximation of Fourier-transforms*, Numer. Math. 8, 235-249.

STOER, J. [1972], *Einführung in die Numerische Mathematik I*, Heidelberger Taschenbücher 105.

ZYGMUND, A. [1959], *Trigonometric series*, Cambridge University Press.

Index

Inhoud

```
      PROGRAM ILS1A (OUTPUT,TAPE6=OUTPUT)
C ILLUSTRATION 1A BY USE OF NAG ROUTINE C06ABF
      REAL B(2),A(2),R(2),SQR2
      COMPLEX C(4),E(2),COM
      N=4
      M=2
      M1=M
      SQR2=SQRT(2.0)
      C(1)=(1.,0.)*SQR2
      C(2)=(0.,1.)*SQR2
      C(3)=(2.,0.)*SQR2
      C(4)=(0.,-1.)*SQR2
C EXPONENTIAL TWIDDLE FACTORS
      E(1)=(1.,0.)
      E(2)=(0.,1.)
      DO 10 K=1,M,1
      COM=C(K)+C(M+K)+(0.,1.)*(C(K)-C(M+K))*E(K)
      A(K)=REAL(COM)
      B(K)=AIMAG(COM)
   10 CONTINUE
      CALL C06ABF(A,B,M,.FALSE.,M1,R)
      WRITE (6,100) A(1),B(1),A(2),B(2)
  100 FORMAT(2H (,4(E8.2,1X),1H))
      END
```

```
( .30E+01 -.30E+01  .30E+01  .10E+01 )
```

```
      PROGRAM ILS1B (OUTPUT,TAPE6=OUTPUT)
C ILLUSTRATION BY IMSL ROUTINE FFT2 (FFRDR2)
      INTEGER M,N
      INTEGER IWK(2)
      COMPLEX C(4),E(2)
      N=4
      M=2
      C(1)=(0.,3.)
      C(2)=(0.,1.)
      C(3)=(0.,2.)
      C(4)=(0.,1.)
C EXPONENTIAL TWIDDLE FACTORS
      E(1)=(1.,0.)
      E(2)=(0.,1.)
      DO 10 K=1,M,1
      C(K)=C(K)+C(M+K)-(0.,1.)*(C(K)-C(M+K))*E(K)
   10 CONTINUE
      CALL FFT2(C,1,IWK)
      CALL FFRDR2(C,1,IWK)
      WRITE (6,100) AIMAG(C(1)),REAL(C(1))
     *             ,AIMAG(C(2)),REAL(C(2))
  100 FORMAT(2H (,4(E8.2,2H*I,1X),1H))
      END
```

```
( .70E+01*I  .10E+01*I  .30E+01*I  .10E+01*I )
```

Purpose : calculation of the twiddle factors:
$\{e^{i\theta k}\}_{k=0}^{N}$

Data : integer $N \geq 0$;
real theta $[0,2\pi)$

Results : vector z of length N + 1, such that z is a numerical approximation of the values $\{e^{i\theta k}\}_{k=0}^{N}$

Algorithm : set up part

by means of standard functions:
$z[k] = e^{i(kt) \pmod{2\pi}}$, k = 1, 2, 4, 8, ... , $m \leq N < 2m$;

fill up part

k = m/4, m/8,...,4, 2, 1

j = 3k, 5k,...,m-3k, m-k

$$z[j] = \begin{cases} 2\mathrm{Re}(z[k]) * z[j-k] - z[j-2k], & |2\mathrm{Re}[k])| \leq 1 \\ (z[j-k]+z[j+k])/2\mathrm{Re}(z[k]), & |2\mathrm{Re}(z[k])| > 1 \end{cases}$$

Remaining part

$z[k] = z[m] * z[k-m]$, k = m+1, m+2, ... , N.

Authors: Hollenberg, J.P. and C.G. van der Laan.

```
'BEGIN''INT'N;

    'PROC'EXPTWI=('REAL'T,'INT'N)'REF'[]'COMPL';
    'IF'N'LT'0'OR'N'GT'MAXINT
    'THEN'STOP
    'ELSE''REAL'TPI=2*PI,
        'INT'K:=1,'REAL'KTH:=T,'HEAP'[0:N]'COMPL'Z;
        'CO' SET UP PART 'CO'
        Z[0]:=1,0'I'0,0;
        'WHILE'K'LE'N
        'DO'Z[K]:=COS(KTH)'I'SIN(KTH);
            K*:=2;
            'IF''REAL'H=(KTH*:=2)-TPI;H'GT'0'THEN'KTH:=H'FI'
        'OD';
        'INT'M=K'OVER'2;
        'CO' FILL UP PART 'CO'
        K:=K'OVER'4;
        'WHILE'K:=K'OVER'2;K'GE'1
        'DO''IF''REAL'TCT=2*RE'OF'Z[K];'ABS'TCT'LE'1
            'THEN''FOR'J'FROM'3*K'BY'2*K'TO'M
                'DO'Z[J]:=TCT*Z[J-K]-Z[J-2*K]'OD'
            'ELSE''FOR'J'FROM'3*K'BY'2*K'TO'M
                'DO'Z[J]:=(Z[J-K]+Z[J+K])/TCT'OD'
            'FI'
        'OD';
        'CO' REMAINING PART 'CO'
        'COMPL'ZM=Z[M];
        'FOR'K'FROM'M+1'TO'N
        'DO'Z[K]:=ZM*Z[K-M]'OD';
        Z
    'FI';


    'FOR'N'FROM'0'TO'10
    'DO'PRINT(("NEWLINE,"N=",N,NEWLINE,
        EXPTWI(PI/4,N),NEWLINE))'OD'
'END'
```

5. NIET-LINEAIRE VERGELIJKINGEN EN OPTIMALISERING

5.1. Stelsels niet-lineaire vergelijkingen

door J.C.P. Bus
(Mathematisch Centrum)

5.1.1. Inleiding

Vele problemen in de toepassingsgebieden van de wiskunde kunnen zo worden geformuleerd dat oplossing van een stelsel niet-lineaire vergelijkingen nodig is om de oplossing van het probleem te kunnen verkrijgen. Een voorbeeld hiervan is het oplossen van tweepunts randwaardeproblemen (zie hoofdstuk 3.1).

$$u'' = f(t,u) \qquad 0 \leq t \leq 1$$

$$u(0) = \alpha, \quad u(1) = \beta$$

met een eindige-differentiemethode of eindige-elementenmethode als f(t,u) niet-lineair is in u.

We zullen ons in dit hoofdstuk beperken tot stelsels van n niet-lineaire vergelijkingen in n onbekenden. Dus voor een gegeven

(5.1.1.1) $F: \mathbb{R}^n \to \mathbb{R}^n$

moet een $z \in \mathbb{R}^n$ worden berekend, zodat $F(z) = 0$.

We kunnen een onderscheid maken tussen twee klassen van algoritmen voor de oplossing van dit probleem:

1. *Lokale algoritmen.* Voor deze algoritmen moeten één of meer redelijke schattingen voor de oplossing worden gegeven. Indien deze schattingen niet voldoende dicht bij een oplossing liggen, dan is de kans dat geen oplossing wordt gevonden groot.
2. *Globale algoritmen.* Voor deze algoritmen zijn geen schattingen van de oplossing nodig of slechts ruwe schattingen.

Bij het gebruik van numerieke programmatuur zijn twee aspekten voor de gebruiker van belang:

1. De *robuustheid* van de programmatuur.
   In hoeverre kunnen met de betreffende programmatuur (nauwkeurige) oplossingen worden berekend van moeilijke problemen.
2. De *efficiëntie* van de programmatuur.
   We zeggen dat een programma een zeker probleem efficiënter oplost dan een ander programma, als het werk dat moet worden gedaan door het eerste programma minder is dan dat met het tweede programma.

Het blijkt in de praktijk dat robuustheid veelal ten koste van de effi-

ciëntie gaat. Het is daarom van belang dat de gebruiker zich enig idee vormt over de moeilijkheidsgraad van het op te lossen probleem. Het zal immers zo zijn dat voor relatief eenvoudige problemen niet erg robuuste programmatuur nodig is, zodat op de efficiëntie kan worden gewonnen.

Tenslotte moet worden opgemerkt dat de efficiëntie van programmatuur voor het oplossen van niet-lineaire stelsels van nog andere eigenschappen van het probleem afhangt, zoals:

1. de orde van het stelsel,
2. het werk dat nodig is om de funktie te evalueren,
3. de beschikbaarheid van analytische afgeleiden en het werk dat nodig is om deze te evalueren,
4. de specifieke struktuur van het probleem.

5.1.2. Probleem en methoden

Zij F gegeven door (5.1.1.1). Dan kunnen we de lokale methoden voor het oplossen van het stelsel niet-lineaire vergelijkingen $F(x) = 0$ ruwweg verdelen in de volgende vier klassen.

1. *Newton-achtige methoden.* Bij deze methoden wordt de matrix van partiële eerste afgeleiden, de jacobiaan, gevraagd. In elke iteratiestap moet voor een gegeven argumentvektor deze matrix worden berekend (eventueel numeriek) en wordt een lineair stelsel met deze matrix opgelost. Voor grote orde n is het benodigde werk per stap, afgezien van de evaluatie van de funktie en de jacobiaan, van de orde n^3.

2. *Sekantenmethode.* Bij deze methoden wordt de jacobiaan niet gevraagd maar tijdens het proces benaderd. Veelal worden bij deze methode, in tegenstelling tot de meeste anderen, $n + 1$ startpunten gevraagd. Het werk per iteratiestap is, evenals bij Newton-achtige methoden, van de orde n^3 voor grote n, omdat een lineair stelsel moet worden opgelost.

3. *Quasi-Newton methode.* Hier wordt de inverse van de jacobiaan bijgehouden. Het werk per iteratiestap is daarom voor grote n veel minder dan voor methoden uit de eerste twee klassen (orde n^2). Meestal wordt wel een benadering van de jacobiaan berekend en geïnverteerd aan het begin van het proces.

4. *Methoden van komponentsgewijze approximatie.* Deze methoden zijn gebaseerd op suksessieve approximatie van de funktiekomponenten als funktie van één der variabelen. Het hangt af van de wijze waarop dit wordt uitgewerkt

hoeveel werk per iteratiestap nodig is. Voor één van de beschikbare programma's welke hierop zijn gebaseerd is dit van de orde n^4 voor grote n.

De globale methoden kunnen we als volgt indelen.

5. *Kontinuatiemethoden.* Het probleem wordt getransformeerd tot een serie van niet-lineaire stelsels, welke worden opgelost met één van de lokale methoden.

6. *Direkte-zoekmethoden.* Door in suksessieve koördinaatrichtingen of random-richtingen te zoeken wordt geprobeerd de oplossing te vinden.

Het is afhankelijk van de gebruikte methode of een probleem wel of niet moeilijk oplosbaar is. Voor lokale methoden en kontinuatiemethoden geven echter de grootheden

(5.1.2.1) $\|F''(x)\|\|[F'(x)]^{-1}\|$

en

(5.1.2.2) $\|F'(x)\|\|[F'(x)]^{-1}\|$,

voor x in een gebied dat het startpunt en de oplossing bevat, een goede aanduiding voor de moeilijkheidsgraad. Indien deze grootheden groot zijn t.o.v. 1, kan worden verwacht dat het probleem moeilijk oplosbaar is voor deze methoden. Zeker als deze grootheden in de oplossing van de grootte-orde $1/\varepsilon$ zijn, met ε de rekenprecisie, moet ernstig rekening worden gehouden met het feit dat methoden uit de klassen 1 t/m 5 geen oplossing kunnen berekenen.

5.1.3. Beschikbare programmatuur

In de programmatheken ACCULIB, IMSL, NAG en NUMAL is betrekkelijk weinig programmatuur beschikbaar voor het oplossen van stelsels niet-lineaire vergelijkingen. De beschikbare programma's zijn alle gebaseerd op lokale methoden. Voor niet-lineaire stelsels worden gegeven:

ACCULIB:	KNEWTON (ALGOL en FORTRAN) (sekantenmethode)
IMSL :	ZSYSTM (FORTRAN) (komponentsgewijze approximatie)

NAG : C05NAA/F
(een variant op de quasi-Newton methode)

C05PAA/F
(een variant op Newton's methode, modificatie volgens Marquardt)

NUMAL : quanewbnd, quanewbnd1
(quasi-Newton methode voor de speciale situatie dat de jacobiaan een bandmatrix is)

Alleen C05PAA/F vereist het programmeren van de jacobiaan, J(x), voor gegeven vektor x. Deze routine berekent echter in feite een minimum van de som van kwadraten van de funktiewaarden. De gebruiker wordt gevraagd om $J^T(x)J(x)$ en $J^T(x)F(x)$ voor een gegeven vektor x te programmeren. Afgezien van het extra werk dat van de gebruiker wordt gevraagd, kan de gebruikte methode voor het oplossen van het lineaire stelsel via de normaalvergelijking en Cholesky-ontbinding gemakkelijker tot instabiel gedrag leiden dan wanneer dit zou zijn gedaan met behulp van Householder-orthogonalisatie.

In de NUMAL zijn vooralsnog slechts procedures beschikbaar voor het oplossen van speciale niet-lineaire stelsels, namelijk stelsels waarvan de jacobiaan een bandmatrix is.

De routine ZSYSTM uit de IMSL-programmatheek vraagt steeds bij een zekere argument-vektor slechts één funktiekomponent. Dit kan inefficiënt zijn, indien één evaluatie van de gehele funktievektor minder werk vergt dan n evaluaties van komponenten. Tenslotte moet van deze routine worden opgemerkt dat het werk per iteratiestap voor grote n, van de orde n^4 is, zodat de routine zeer inefficiënt is voor funkties in veel variabelen, die met betrekkelijk weinig werk te berekenen zijn.

Natuurlijk kan een niet-lineair stelsel worden opgevat als niet-lineair kleinste-kwadratenprobleem, zodat van de hiervoor beschikbare programmatuur ook gebruik kan worden gemaakt. Hiervoor zij verwezen naar hoofdstuk 5.2.

5.1.4. Klassifikatie van problemen en keuze van de programmatuur

We onderscheiden problemen naar:

1. het aantal variabelen;
2. het werk dat moet worden gedaan om een funktievektor te evalueren;
3. de beschikbaarheid van analytische expressies voor de jacobiaan en het werk dat moet worden gedaan voor evaluatie hiervan;

4. de moeilijkheidsgraad;
5. speciale struktuur van de jacobiaan.

ad 1. We noemen een probleem klein als het aantal variabelen kleiner is dan 15, anders noemen we het groot.

ad 2. Voor kleine problemen is het onderscheid tussen een duur en een goedkoop probleem moeilijk aan te geven. De gebruiker zal zich hiervoor een zeker gevoel moeten ontwikkelen. Voor grote problemen maken we onderscheid op grond van het aantal elementaire aritmetische operaties als funktie van het aantal variabelen n.

a. zeer goedkoop: van orde n;
b. goedkoop : van de orde n^2;
c. duur : van de orde n^3;
d. zeer duur : van de orde n^4 of meer.

ad 3. Het werk dat moet worden gedaan om de jacobiaan te evalueren kan worden gegeven in verhouding tot het werk dat nodig is voor een funktieevaluatie.

ad 4. Als schattingen voor de uitdrukkingen (5.1.2.1) en (5.1.2.2) bekend zijn dan kan dit de keuze van het goede programma voordelig beïnvloeden.

ad 5. Heeft de jacobiaan een bandstruktuur? Is de jacobiaan ijl en is het mogelijk door verwisselingen de struktuur te vereenvoudigen?, enz.

Alvorens een programma te kiezen voor het oplossen van een speficiek probleem, dient de gebruiker het probleem op de boven beschreven wijze te klassificeren. Indien niets omtrent de moeilijkheidsgraad is te zeggen, is het veelal het beste om eerst de meest efficiënte routine te kiezen voor het specifieke probleem, om vervolgens, als deze faalt, een robuuster routine te kiezen.

Uit BUS [1975] blijkt, dat het wenselijk is te kunnen beschikken over betrouwbare implementaties van Newton's methode, quasi-Newton methode, sekantenmethode en Brown's methode van komponentsgewijze approximatie. Ook zijn methoden nodig die gebruik maken van de specifieke struktuur van problemen. Geen enkele van de vier besproken programmatheken bieden echter deze keuzemogelijkheid.

We kunnen, gebruikmakend van de resultaten uit BUS [1975] de volgende regels geven:

1. Als analytische uitdrukkingen voor de jacobiaan beschikbaar zijn en de

evaluatie hiervan is ongeveer even duur als een of enkele funktie-evaluaties, dan is de meest efficiënte methode die van Newton. Hiervoor zijn beschikbaar: C06PAA/F (NAG) en gssnewton (NUMAL) (vgl. hoofdstuk 5.2).

2. Als het aantal variabelen zeer groot is en de jacobiaan ijl, dan zijn speciale hierop gerichte methoden aan te bevelen. Als de jacobiaan een bandstruktuur heeft dan is quanewbnd1 (NUMAL) beschikbaar.
3. Voor kleine problemen waarvan de jacobiaan niet beschikbaar is, is Brown's methode van komponentsgewijze approximatie de aangewezen methode. Deze is als routine ZSYSTM in IMSL (FORTRAN) beschikbaar. In ALGOL 60 is geen implementatie in één van de programmatheken beschikbaar. Een tweede keus voor deze klasse van problemen is C05NAA/F (NAG).
4. Voor grote goedkope en zeer goedkope problemen is Brown's methode (ZSYSTM uit IMSL) zeer inefficiënt. Eén van de andere methoden verdient hier de voorkeur.
5. Voor grote dure en zeer dure problemen, waarvan de jacobiaan niet beschikbaar of goedkoop is, zijn quasi-Newton methoden (C05NAA/F uit NAG) aan te bevelen. Hiervoor zijn ook de sekantenmethode of Brown's methode als tweede en derde keus geschikt.
6. Als een probleem niet wordt opgelost met een zekere routine, dan is een tweede poging met een andere methode slechts zinvol, indien redelijke zekerheid bestaat dat de laatste robuuster is dan de eerste. Met name de routines ZSYSTM en C05NAA/F kunnen zeer robuust genoemd worden. Hoewel voor KNEWTON en C05PAA/F geen testresultaten zijn verkregen is de verwachting dat deze routines aanzienlijk minder robuust zijn.
7. Indien bij routines een waarde moet worden meegegeven voor de stapgrootte bij de benadering van de jacobiaan met voorwaartse difrentieformules, dan is het aanbevelingswaardig deze vooral niet te klein te kiezen. Bij de gegeven machineprecisie van 10^{-14} is 10^{-5} veelal een redelijke keuze; dit hangt echter sterk af van de niet-lineariteit van de funktie.

5.1.5. Een voorbeeld

Als voorbeeld beschouwen we een probleem van diffusie in verschillende gebieden, gescheiden door een bewegende grens (zie AMES [1973]). Stel dat diffusie plaatsvindt in een half-oneindig medium en dat het oppervlak in $x = 0$ een konstante koncentratie C_1 heeft. Neem aan dat de koncentratie op grote afstand gelijk C_2 is. De diffusiekoëfficiënt is:

$$D = D_1, \quad C_x < c < C_1$$
$$D = D_3, \quad C_y < c < C_x$$
$$D = D_2, \quad C_2 < c < C_y.$$

De diffusiekoëfficiënt is diskontinu in de koncentraties C_x en C_y, welke optreden in $x = X(t)$ en $x = Y(t)$. Deze funkties van t willen we bepalen. We krijgen voor de verschillende gebieden:

$$\frac{\partial c_1}{\partial t} = D_1 \frac{\partial^2 c_1}{\partial x^2}, \qquad 0 < x < X, \qquad c_1 = C_1 \text{ voor } x = 0, \tag{5.1.5.1}$$

$$\left.\begin{aligned} c_1 &= c_3 = C_x \qquad (5.1.5.2)\\ D_1 \frac{\partial c_1}{\partial x} &= D_3 \frac{\partial c_3}{\partial x} \qquad (5.1.5.3)\end{aligned}\right\} \quad x = X(t),$$

$$\frac{\partial c_3}{\partial t} = D_3 \frac{\partial^2 c_3}{\partial x^2}, \qquad X < x < Y, \tag{5.1.5.4}$$

$$\left.\begin{aligned} c_3 &= c_2 = C_y \qquad (5.1.5.5)\\ D_3 \frac{\partial c_3}{\partial x} &= D_2 \frac{\partial c_2}{\partial x} \qquad (5.1.5.6)\end{aligned}\right\} \quad x = Y(t),$$

$$\frac{\partial c_2}{\partial t} = D_2 \frac{\partial^2 c_2}{\partial x^2}, \qquad Y < x < \infty, \qquad c_2 = C_2 \text{ voor } x \to \infty, \tag{5.1.5.7}$$

waarbij c_1, c_2 en c_3 de koncentraties zijn in de gebieden met diffusiekoëfficiënt D_1, D_2 en D_3, respektievelijk.

Met behulp van een partikuliere oplossing van deze differentiaalvergelijkingen en randvoorwaarden krijgen we voor konstanten A, B en E:

$$c_1 = C_1 + A \operatorname{erf}[x/2(D_1 t)^{\frac{1}{2}}], \qquad 0 < x < X,$$
$$c_2 = C_2 + B \operatorname{erf} c[x/2(D_2 t)^{\frac{1}{2}}], \qquad Y < x < \infty,$$
$$c_3 = C_x + E\{\operatorname{erf}[x/2(D_3 t)^{\frac{1}{2}}] - \operatorname{erf}[X(t)/2(D_3 t)^{\frac{1}{2}}]\}, \qquad X < x < Y.$$

Substitutie van (5.1.5.2) en (5.1.5.5) geeft uitdrukkingen voor A, B en E in X(t) en Y(t). Een gevolg hiervan is dat we kunnen zeggen dat X en Y evenredig zijn met $t^{\frac{1}{2}}$, dus

$$X = k_1 t^{\frac{1}{2}}, \qquad Y = k_2 t^{\frac{1}{2}},$$

zodat we uiteindelijk de volgende twee niet-lineaire vergelijkingen in de variabelen k_1 en k_2 krijgen:

$$\frac{D_1^{\frac{1}{2}}(C_x - C_1)e^{-k_1^2/4D_1}}{\operatorname{erf}(k_1/2D_1^{\frac{1}{2}})} - \frac{D_3^{\frac{1}{2}}(C_y - C_x)e^{-k_1^2/4D_3}}{\operatorname{erf}(k_2/2D_3^{\frac{1}{2}}) - \operatorname{erf}(k_1/2D_3^{\frac{1}{2}})} = 0$$

$$\frac{D_2^{\frac{1}{2}}(C_y - C_2)e^{-k_2^2/4D_2}}{\operatorname{erfc}(k_2/2D_2^{\frac{1}{2}})} + \frac{D_3^{\frac{1}{2}}(C_y - C_x)e^{-k_2^2/4D_3}}{\operatorname{erf}(k_2/2D_3^{\frac{1}{2}}) - \operatorname{erf}(k_1/2D_3^{\frac{1}{2}})} = 0.$$

Voor oplossing van dit probleem is gebruik van de routine ZSYSTM het meest geschikt, zoals blijkt uit de aanwijzingen in sektie 5.1.4.

We kozen als data:

$$C_2 = 0, \quad C_1 = 1, \quad C_y = 0.5 \text{ en } C_x = 0.75$$

$$D_1 = 1.4, \; D_2 = 0.9, \; D_3 = 4.5,$$

en als startpunt $(5,1)^T$.

Programma en uitvoer zijn gegeven in de bijlage.

Literatuur

AMES, W.F. [1973], *Nonlinear algebraic equations in continuum mechanics*, in: Byrne, G.D. & Hall, C.A., Numerical Solution of Systems of Nonlinear Algebraic equations, Acad. Press.

BUS, J.C.P. [1975], *A comparative study of programs for solving nonlinear equations*, Mathematisch Centrum, NW 25/75.

```
      PROGRAM EXAMP(INPUT,OUTPUT)
      DIMENSION X(2),PAR(6),IFORM(6),WA(8),T(3),C1K(10),C2K(10),
     +          C3K(10)
      EQUIVALENCE (PAR(1),D1),(PAR(2),D2),(PAR(3),D3),
     +            (PAR(4),C21),(PAR(5),CY1),(PAR(6),CX1)
      EXTERNAL F
      DATA EPS,NSIG,ITMAX /1E-5, 4, 100/
      DATA T /.01, 100., 1E+4/
      DATA IFORM /6L D1 ? ,6L D2 ? ,6L D3 ? ,
     +            10H C2 / C1 ?,10H CY / C1 ? ,10H CX / C1 ? /
      DO 10 I=1,6
      PRINT 100,IFORM(I)
      READ 110,PAR(I)
10    PRINT 105,IFORM(I),PAR(I)
100   FORMAT(A10)
105   FORMAT(A10,E14.7)
110   FORMAT(F7.3)
      PRINT 120
120   FORMAT(" K1 ? ")
121   FORMAT(" K1 ? ",E14.7)
      READ 110,X(1) $ PRINT 121,X(1)
      PRINT 125
125   FORMAT(" K2 ? ")
126   FORMAT(" K2 ? ",E14.7)
      READ 110,X(2) $ PRINT 126,X(2)

      CALL ZSYSTM ( F, EPS, NSIG, 2, X, ITMAX, WA, PAR, IER )

      PRINT 130,IER
130   FORMAT("0IER= ",I8)
      PRINT 135,X(1),X(2)
135   FORMAT("1K1= ",E14.7,/," K2= ",E14.7,/)
      C1=1
      C2= C21 * C1
      CY= CY1 * C1
      CX= CX1 * C1
      A= (CX - C1) / ERF ( X(1) / (2 * SQRT( D1 )))
      B= (CY - C2) / ERFC( X(2) / (2 * SQRT( D2 )))
      E= (CY - CX) / (ERF( X(2) / (2 * SQRT( D3 ))) -
     +                ERF( X(1) / (2 * SQRT( D3 ))))
      PRINT 140,A,B,E
140   FORMAT("1A= ",E14.7,/," B= ",E14.7,/," E= ",E14.7)
      IT=1
      XG= X(1) * SQRT( T(IT) )
      YG= X(2) * SQRT( T(IT) )
      DO 20 I=1,10
      XK= .1 * I * XG
20    C1K(I)= C1 + A * ERF( XK / (2 * SQRT( D1 * T(IT) )))
      DO 30 I=1,10
      XK= .1 * I * (YG - XG) + XG
30    C3K(I)= CX + E * (ERF( XK / (2 * SQRT( D3 * T(IT) ))) -
     +                  ERF( XG / (2 * SQRT( D3 * T(IT) ))) )
      DO 40 I=1,10
      XK= .1 * I * YG + YG
40    C2K(I)= C2 + B * ERFC( XK / (2 * SQRT( D2 * T(IT) )))
      PRINT 150,T(IT),XG,YG
150   FORMAT("0T= ",E14.7,/," X= ",E14.7,/," Y= ",E14.7)
      PRINT 160
160   FORMAT("0        C1K            C2K            C3K")
      DO 60 I=1,10
60    PRINT 170,C1K(I),C2K(I),C3K(I)
170   FORMAT(3E14.7)
      END
```

```
      FUNCTION ERF(Y)
      CALL MERF(Y,XERF)
      ERF= XERF
      RETURN
      END

      FUNCTION ERFC(Y)
      CALL MERFC(Y,XERFC)
      ERFC= XERFC
      RETURN
      END

      FUNCTION F(X,K,PAR)
      DIMENSION X(2),PAR(6)
      DATA TOL /1.0E-7/
      IF( K.EQ.1) GOTO 10
      IF( K.EQ.2) GOTO 50
      CALL DISPLA("K=",K)
      STOP "EXECUTION ERROR"
10    A1= ERF( ERFPAR( X(1),PAR(1)))
      B1= SQRT(PAR(1)) * (PAR(6) - 1) * EXP( EXPPAR(X(1),PAR(1)))
       IF (ABS(B1 * TOL).GE.ABS(A1)) CALL ERROR(A1,B1)
20    A2= ERF( ERFPAR(X(2),PAR(3))) - ERF( ERFPAR(X(1),PAR(3)))
      B2= SQRT(PAR(3)) * (PAR(5) - PAR(6)) * EXP( EXPPAR(X(1),PAR(3)))
       IF (ABS(B2 * TOL).GE.ABS(A2)) CALL ERROR(A2,B2)
30    F= B1 / A1 - B2 / A2
      RETURN
50    A1= ERFC( ERFPAR(X(2),PAR(2)))
      B1= SQRT(PAR(2)) * (PAR(5) - PAR(4)) * EXP( EXPPAR(X(1),PAR(1)))
       IF(ABS(B1 * TOL).GE.ABS(A1)) CALL ERROR(A1,B1)
60    A2= ERF( ERFPAR(X(2),PAR(3))) - ERF( ERFPAR(X(1),PAR(3)))
      B2= SQRT(PAR(3)) * (PAR(5) - PAR(6)) * EXP( EXPPAR(X(2),PAR(3)))
       IF(ABS(B2 * TOL).GE.ABS(A2)) CALL ERROR(A2,B2)
70    F= B1 / A1 + B2 / A2
      RETURN
      END

      FUNCTION ERFPAR(X,Y)
      ERFPAR = X / ( 2 * SQRT(Y))
      RETURN
      END

      FUNCTION EXPPAR(X,Y)
      EXPPAR = - X ** 2 / ( 4 * Y )
      RETURN
      END
```

```
      SUBROUTINE ERROR(A,B)
      ENTRY ERROR1
      PRINT 100,B,A
      CALL REMARK("ERROR1 SEE OUTPUT")
      STOP "EXECUTION ERROR"
      ENTRY ERROR2
      PRINT 110,B,A
      CALL REMARK("ERROR2 SEE OUTPUT")
      STOP "EXECUTION ERROR"
      ENTRY ERROR3
      PRINT 120,B,A
      CALL REMARK("ERROR3 SEE OUTPUT")
      STOP "EXECUTION ERROR"
      ENTRY ERROR4
      PRINT 130,B,A
      CALL REMARK("ERROR4 SEE OUTPUT")
      STOP "EXECUTION ERROR"
  100 FORMAT("0<< ERROR1 >>",/," TELLER: ",E20.13,/," NOEMER: ",E20.13)
  110 FORMAT("0<< ERROR2 >>",/," TELLER: ",E20.13,/," NOEMER: ",E20.13)
  120 FORMAT("0<< ERROR3 >>",/," TELLER: ",E20.13,/," NOEMER: ",E20.13)
  130 FORMAT("0<< ERROR4 >>",/," TELLER: ",E20.13,/," NOEMER: ",E20.13)
      END
```

```
D1 ?         .1400000E+01
D2 ?         .9000000E+00
D3 ?         .4500000E+01
C2 / C1 ? 0.
CY / C1 ?   .5000000E+00
CX / C1 ?   .7500000E+00
K1 ?    .1000000E+01
K2 ?    .5000000E+01



IER=         0

K1=    .3295865E+00
K2=    .1450215E+01

A=   -.1601070E+01
B=    .1787438E+01
E=   -.8811802E+00

T=    .1000000E-01
X=    .3295865E-01
Y=    .1450215E+00

        C1K             C2K             C3K
 .9748399E+00    .4190332E+00    .7239545E+00
 .9496895E+00    .3478227E+00    .6980518E+00
 .9245587E+00    .2859197E+00    .6723271E+00
 .8994570E+00    .2327322E+00    .6468148E+00
 .8743943E+00    .1875634E+00    .6215483E+00
 .8493802E+00    .1496495E+00    .5965602E+00
 .8244242E+00    .1181946E+00    .5718818E+00
 .7995359E+00    .9240115E-01    .5475434E+00
 .7747247E+00    .7149565E-01    .5235737E+00
 .7500000E+00    .5474844E-01    .5000000E+00
```

5. NIET-LINEAIRE VERGELIJKINGEN EN OPTIMALISERING

5.2. Minimaliseren zonder nevenvoorwaarden

door J.C.P. Bus
(Mathematisch Centrum)

5.2.1. Inleiding

In dit hoofdstuk zullen we programmatuur beschouwen voor het minimaliseren van een gegeven funktie

(5.2.1.1) $F: \mathbb{R}^n \to \mathbb{R}$,

waarbij geen beperkingen zijn opgelegd aan de variabelen.

Wij zullen hierbij een aparte beschouwing wijden aan *niet-lineaire kleinste-kwadratenproblemen*, waarbij

$$(5.2.1.2) \quad F(x) = \sum_{i=1}^{m} (f_i(x))^2,$$

met $(f_1(x),\ldots,f_m(x))^T$ bijvoorbeeld de residuvektor bij curve-fitting problemen.

Gewone minimaliseringsproblemen komen in de praktijk weinig voor zonder nevenvoorwaarden. Het is echter soms mogelijk door transformaties het probleem te formuleren als een vrij minimaliseringsprobleem. Ook maken bepaalde technieken voor het minimaliseren van funkties onder nevenvoorwaarden (boete-funktie methoden) gebruik van vrije minimaliseringstechnieken.

Ook bij deze problemen kunnen we, zoals in hoofdstuk 5.1.1, onderscheid maken tussen lokale en globale algoritmen. Een extra probleem hierbij is dat lokale algoritmen een lokaal minimum vinden, dat mogelijk niet het absolute minimum is van de betreffende funktie. Dit geldt echter ook voor direkte zoekmethoden, welke volgens de in hoofdstuk 5.1.1 gegeven omschrijving globale algoritmen zijn. Het probleem een absoluut minimum te vinden van een willekeurige naar onder begrensde funktie is echter nog maar weinig behandeld en er bestaat nog nauwelijks programmatuur voor. Wij zullen ons daarom beperken tot algoritmen welke lokale minima bepalen en waarvoor een redelijke beginschatting is vereist.

5.2.2. Probleem en methoden

Zij F gegeven door (5.2.1.1). Wij onderscheiden voor de bepaling van een minimum van deze funktie drie klassen van methoden.

1. Direkte zoekmethoden; deze methoden vereisen slechts de funktie.
2. Gradiëntmethoden ; hiervoor is zowel de funktie als de gradiënt vereist.
3. Tweede-orde methoden; deze vereisen funktie, gradiënt en hessiaan (tweede afgeleide).

5.2.2.1. Direkte zoekmethoden

De belangrijkste representanten van deze klasse zijn:

1. *Sequentiële methoden.* Deze methoden zijn gebaseerd op de evaluatie van de funktie in een aantal symmetrisch liggende punten op een n-dimensionale geometrische figuur. Een voorbeeld is de *simplex-methode.* De konvergentie van deze methoden kan traag zijn, vooral bij scherpe lange dalen.
2. *Lineaire methoden.* Uitgaande van een verzameling van n onafhankelijke richtingen, wordt achtereenvolgens het lijnminimum bepaald in deze richtingen. Indien deze verzameling vast wordt gekozen kan dit zeer inefficiënt zijn als geen van deze richtingen lijkt op de lokale principale as van de funktie.
   Voorbeelden van methoden in deze klasse zijn de methode van *Hooke en Jeeves* en die van *Rosenbrock.* Bij beide methoden wordt tijdens het proces de verzameling van onafhankelijke richtingen aangepast.
3. *Methode van gekonjugeerde richtingen.* Deze methoden zijn in feite speciale lineaire methoden. De onafhankelijke richtingen worden gekonjugeerd gekozen. Voor kwadratische funkties termineert deze methode in n iteratiestappen. Een goed voorbeeld van een dergelijke methode is de algoritme, gegeven door POWELL [1964] en een modificatie hiervan, gegeven door BRENT [1973].

In het algemeen is de konvergentie van direkte zoekmethoden traag. De meest betrouwbare en dikwijls ook efficiënte methode is meestal die van Powell, zoals gemodificeerd door Brent.

De methode van Hooke & Jeeves kan voordelen hebben als de funktie een som van termen is, die elk van slechts enkele variabelen afhangt, zodat bij verandering van richting slechts enkele termen hoeven worden herberekend.

De Simplex-methode kan bij curve-fitting voordelen bieden, omdat de covariantiematrix kan worden verkregen uit de uiteindelijke simplex-configuratie.

5.2.2.2. Gradiëntmethoden

We maken hierbij onderscheid tussen kleinste-kwadratenproblemen en andere minimaliseringsproblemen.

Kleinste-kwadratenproblemen:

Zij

$$F(x) = \sum_{i=1}^{m} [f_i(x)]^2,$$

dan wordt aangenomen dat de matrix der partiële eerste afgeleiden, $J(x)$, gedefinieerd door

$$(J(x))_{ij} = \frac{\partial f_i}{\partial x_j}$$

is gegeven. De bekendste methode voor het bepalen van een minimum van F is de *methode van Gauss-Newton*. Deze is gebaseerd op het bepalen van een nulpunt van de gradiënt van F, met behulp van de methode van Newton, waarbij de tweede afgeleide van F wordt benaderd door

$$2J^T(x)J(x).$$

Om problemen van divergentie van deze methode te voorkomen, welke optreden als de matrix $J(x)$ (bijna) singulier is, wordt deze methode veelal gemodificeerd volgens ideeën van Levenberg en Marquardt. De essentie van deze modificatie is dat de tweede afgeleide van F nu wordt benaderd door

$$2J^T(x)J(x) + \lambda I,$$

waarbij I de eenheids-matrix is en λ een reële parameter. De zo verkregen methode wordt veelal *Marquardt's methode* genoemd.

Andere minimaliseringsproblemen:

We nemen hierbij aan dat de gradiënt $\nabla F(x)$ is gegeven. De eenvoudigste gradiëntmethode is de *methode van steilste daling* ("steepest descent"). Bij deze methode wordt steeds in de richting van de negatieve gradiënt een betere approximatie van het minimum gezocht. Deze methode kan, vooral bij funkties met lange scherpe dalen, zeer traag konvergeren.

Een belangrijke klasse van gradientmethoden zijn de *quasi-Newton* of *variabele-metriek* methoden (DAVIDON [1959], FLETCHER & POWELL [1963]). Deze methoden zijn gebaseerd op Newton's algoritme voor de bepaling van een nulpunt van de gradiënt, waarbij de inverse hessiaan wordt benaderd en bijgehouden met behulp van korrektieformules, die geen extra funktie- of gradiëntevaluaties vragen. Er zijn een groot aantal quasi-Newton algoritmen

voorgesteld in de literatuur, welke voornamelijk verschillen in de korrektieformule die wordt gebruikt, en de wijze waarop bij gegeven richting een approximatie van het minimum van de funktie in die richting wordt verkregen. Het hangt ook sterk af van deze keuzen of de verkregen algoritme efficiënt en betrouwbaar is.

Tenslotte moet de methode van *gekonjugeerde gradiënten* worden genoemd (FLETCHER & REEVES [1964]), waarbij de zoekrichtingen, afhankelijk van de gradiënt zo worden gekozen, dat opvolgende richtingen gekonjugeerd zijn.

De gradiëntmethoden zijn behalve de methode van steilste daling, meestal efficiënter dan direkte zoekmethoden als de afgeleide van de funktie gegeven is. Het konvergentiegedrag is in de praktijk veelal superlineair.

5.2.2.3. Tweede-orde methoden

Voor deze methoden is de gradiënt en de hessiaan van de funktie vereist. We maken geen speciaal onderscheid tussen kleinste-kwadratenproblemen en andere minimaliseringsproblemen, omdat meestal geen tweede afgeleide van een som van kwadraten beschikbaar is. Voor dergelijke problemen is meestal de residuvektor $(f_1(x),\ldots,f_m(x))^T$ (vgl. 5.2.1.2)) gegeven en de tweede afgeleide hiervan is al een tensor.

Voor algemene problemen noemen we als tweede-orde methode de methode van Newton voor het bepalen van een nulpunt van de gradiënt, waarbij moet worden opgemerkt dat de hessiaan symmetrisch is, zodat het oplossen van het lineaire stelsel met een speciaal daarvan gebruik makende algoritme kan worden gedaan.

5.2.3. Beschikbare programmatuur

We zullen nu een overzicht geven van de programmatuur welke beschikbaar is in de programmatheken ACCULIB, IMSL, NAG en NUMAL.

5.2.3.1. Programmatuur voor niet-lineaire kleinste-kwadratenproblemen

A. Geen afgeleide vereist

ZXSSQ (IMSL) : Methode van Marquardt waarbij de jacobiaan wordt berekend met voorwaartse-differentieformules.

E04FAA/F (NAG) : Gauss-Newton methode waarbij de jacobiaan op speciale wijze wordt benaderd (PECKHAM [1970]).

E04FBA/F (NAG) : Gauss-Newton methode, waarbij de jacobiaan wordt benaderd zonder dat dit extra funktie-evaluaties vergt (POWELL [1964]).

B. Afgeleide vereist

E04GAA/F (NAG) : Marquardt's methode.

marquardt (NUMAL) : Marquardt's methode.

gssnewton (NUMAL) : Gauss-Newton methode.

5.2.3.2. Programmatuur voor algemene minimaliseringsproblemen zonder nevenvoorwaarden.

A. Geen afgeleide vereist

OPTDZM (ACCULIB) : Lineaire methode van Hooke & Jeeves.

ZXMIN (IMSL), E04CDA/F (NAG) : Quasi-Newton methode waarbij de gradiënt wordt benaderd met voorwaartse-differentieformules.

E04CAA/F (NAG) : Methode van gekonjugeerde richtingen (POWELL [1964]).

E04CCA/F (NAG) : Simplex-methode van Nelder & Mead.

praxis (NUMAL) : Methode van gekonjugeerde richtingen met modificaties volgens BRENT [1973].

B. Eerste afgeleide vereist

FMFP (ACCULIB) : Quasi-Newton methode (FLETCHER & POWELL [1963]).

E04DCA/F (NAG) : Quasi-Newton methode (POWELL [1970]).

E04DDA/F (NAG) : Quasi-Newton methode (GILL & MURRAY [1972]).

E04DBA/F (NAG) : Methode van gekonjugeerde gradiënten (FLETCHER & REEVES [1964]).

rnk1min (NUMAL), flemin (NUMAL) : Quasi-Newton methoden (BUS [1972]).

C. Eerste en tweede afgeleiden nodig

E04EAA/F (NAG) : Gemodificeerde Newton methode.

Opmerkingen

1. De routines in ACCULIB zijn zowel in ALGOL 60 als in FORTRAN beschikbaar.
2. In NAG zijn nog twee routines beschikbaar welke in mark 5 zullen worden weggehaald. Deze zijn hier niet genoemd.
3. Zowel in IMSL als in NAG zijn nog routines beschikbaar, speciaal voor "curve-fitting" problemen. Deze vallen als zodanig buiten de scope van dit hoofdstuk en zijn daarom niet genoemd.
4. Het is in principe mogelijk om de procedures in NUMAL, die de afgeleiden vereisen, te gebruiken met een numerieke benadering van deze afgeleiden.
5. De routines in NAG hebben de mogelijkheid om eenvoudig tussenuitvoer te verkrijgen, wat voor het verkrijgen van een goed inzicht in het probleem soms van belang kan zijn.
6. Als een routine zowel de funktie als de gradiënt vereist, dan is het mogelijk om deze als twee gescheiden routines mee te geven in de parameterlijst, maar ook om de gebruiker een routine te laten geven die bij gegeven argumentvektor zowel de funktie als de gradiënt berekent. Het laatste is vooral voordelig als een groot deel van het werk voor de berekening van de funktie en de gradiënt hetzelfde is. Is dit echter niet het geval, dan kan vooral de lijnminimalisatie in de routine veelal efficiënter worden gedaan als de funktie apart kan worden berekend. in NAG zijn beide mogelijkheden aanwezig, in NUMAL slechts de tweede.
7. Bij gebruik van de routines in NAG is het soms mogelijk door slim programmeren efficiënter te rekenen. Dit gaat echter wel ten koste van de overzichtelijkheid van de parameterlijst.

5.2.4. De keuze van de programmatuur

We zullen geen uitputtende evaluatie geven van de beschikbare programmatuur. Daarvoor zijn niet voldoende testresultaten voorhanden.

We zullen ons beperken tot een aantal opmerkingen, die de gebruiker de mogelijkheid geven om zijn keuze in ieder geval te beperken tot slechts enkele routines.

1. Indien de funktie een som van kwadraten is, dan is het veelal efficiënter om een routine te gebruiken die speciaal hiervoor is geschreven.
2. Indien de eerste en eventueel zelfs de tweede afgeleide kan worden gegeven, dan is het zeer aan te bevelen een routine te kiezen die hiervan ook gebruik maakt. Dit is meestal veel efficiënter.

3. Routines die afgeleiden benaderen met differentieformules zijn dikwijls efficiënter dan direkte zoekmethoden, maar zij kunnen gemakkelijk afbreken als de funktie zich er niet toe leent. In de volgende twee gevallen is gebruik van dergelijke routines niet aan te bevelen:
   a. de te benaderen afgeleide varieert zeer sterk; de tweede afgeleide is groot in norm;
   b. de fout in de funktie veroorzaakt door meten en/of afronden is vrij groot t.o.v. de machineprecisie.

   Het is niet aan te bevelen tweede-orde methoden te gebruiken als zowel de gradiënt als de hessiaan met differentieformules worden benaderd. Gebruik van tweede-orde methoden, waarbij de gradiënt is gegeven en de hessiaan met differentieformules wordt berekend, hoeft niet efficiënter te zijn dan gebruik van een gradiëntmethode. In de praktijk blijkt voor grote n dat bijvoorbeeld quasi-Newton methoden meestal minder gradiëntevaluaties nodig hebben om een oplossing te berekenen dan een op deze wijze gebruikte tweede-orde methode. In een dergelijke situatie heeft een quasi-Newton methode dus de voorkeur als evaluatie van de gradiënt duur is (als het aantal aritmetische operaties bijv. van de orde n^3 is, met n het aantal variabelen).
4. De routine E04DBA/F gebruikt zeer weinig geheugenruimte. Dit is slechts ongeveer 3n plaatsen, terwijl de overige methoden veelal tenminste $\frac{1}{2}n^2$ geheugenplaatsen nodig hebben.
5. De procedure marquardt levert direkt de covariantiematrix $[J^T(x)J(x)]^{-1}$ in de berekende oplossing af, welke kan worden gebruikt voor schattingen van de nauwkeurigheid van de oplossing. De routine E04GAA/F levert een ontbonden vorm van $J^T(x)J(x)$ waarmee de covariantiematrix betrekkelijk goedkoop kan worden berekend. De routine ZXSSQ levert slechts een benadering van J(x), zodat de gebruiker zelf nog moet inverteren.

5.2.5. Een curve-fitting probleem

VAN DOMSELAAR [1974] beschrijft een curve-fitting probleem dat wordt verkregen bij de analyse van het hartinfarct. Het model, dat het uitstortingsproces van een enzym, afkomstig van afgestorven hartweefsel na een hartinfarct beschrijft, heeft de volgende vorm:

$$v_1 \frac{dx_1}{dt} = -kv_1x_1 + (x_2-x_1)p + R(t)$$

(5.2.5.1)

$$v_2 \frac{dx_2}{dt} = (x_1-x_2)p,$$

waarbij v_1 en v_2 de volumina binnen, resp. buiten de bloedvaten zijn, k de afbraakkonstante, R(t) de enzymaktiviteit beschrijft en p de permeabiliteitskonstante is. Met behulp van de analytische oplossing van deze differentiaalvergelijking komen we tot de volgende te fitten modelfunktie in de parameters p_1, p_2, p_3 en p_4:

$$g(t;p_1,p_2,p_3,p_4) =$$

$$c \frac{p_4(p_1-a)}{a(p_4-p_1)} \frac{d_1(t)}{d_{lim}} \exp(-p_4t) + c \frac{d_2(t)}{d_{lim}} \exp(-at) - g_{lim},$$

met g_{lim} de normale uitstorting,

$$d_1(t) = \int_0^t \frac{1}{\tau} \exp\left[-0.5\left(\frac{\ln(\tau)-p_3}{p_2}\right)^2 - p_4\tau\right]d\tau,$$

$$d_2(t) = \int_0^t \frac{1}{\tau} \exp\left[-0.5\left(\frac{\ln(\tau)-p_3}{p_2}\right)^2 + a\tau\right]d\tau,$$

$$d_{lim} = \lim_{t\to\infty} d_2(t).$$

Met de data, gegeven door VAN DOMSELAAR [1971] en het startpunt $(0.14, 0.2, 2.40, 0.28)^T$, is geprobeerd dit probleem op te lossen met de procedures gssnewton, marquardt, flemin en rnk1min uit NUMAL, waarbij de jacobiaan werd benaderd met voorwaartse differenties. De resultaten zijn gegeven in tabel 1.

TABEL 1

residunorm	aantal benodigde funktieevaluaties			
	gssnewton	marquardt	flemin	rnk1min
59.652	-	2	57	15
52.968	-	7	59	16
49.314	-	19	62	41
49.233	-	23	101	-

Het bleek dat gssnewton al spoedig afbrak wegens een singuliere jacobiaan. Evenzo blijken de algemene minimaliseringsroutines flemin en rnk1min aanzienlijk inefficiënter wat betreft het aantal funktieevaluaties, dan marquardt die speciaal voor kleinste-kwadratenproblemen is geschreven.

Literatuur

BOX, M.J., D. DAVIES & W.H. SWANN [1969], *Non-linear optimization techniques*, ICI-monograph no. 5, Oliver & Boyd.

BRENT, R.P. [1973], *Algorithms for minimization without derivatives*, Prentice-Hall.

BUS, J.C.P. [1972], *Minimalisering van funkties van meerdere variabelen*, Mathematisch Centrum, NR 29/72.

BUS, J.C.P., B. VAN DOMSELAAR & J. KOK [1975], *Nonlinear least squares estimation*, Mathematisch Centrum, NW 17/75.

DAVIDON, W.C. [1959], *Variable metric method for minimization*, Argonne Nat. Lab. Rep. ANL-5990.

DIXON, L.C.W. [1974], *Nonlinear optimization: A survey of the state of the art*, in: Evans, D.J. (ed.), Software for numerical mathematics, Academic Press.

VAN DOMSELAAR, B. [1974], *Een mathematische analyse van het hartinfarct*, Mathematisch Centrum, NN 4/74.

FLETCHER, R. & M.J.D. POWELL [1963], *A rapidly convergent descent method for minimization*, Comp. J. 6, 162-168.

FLETCHER, R. & C.M. REEVES [1964], *Function minimization by conjugate gradients*, Comp. J. 7, 149-154.

GILL, P.E. & W. MURRAY [1972], *Quasi-Newton methods for unconstrained optimization*, JIMA 9, 91-108.

MARQUARDT, D.W. [1963], *An algorithm for least squares estimation of nonlinear parameters*, SIAM, J. 11, 431-441.

PECKHAM, G. [1970], *A new method for minimizing a sum of squares without calculating gradients*, Comp. J. 13, 418-420.

POWELL, M.J.D. [1964], *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, Comp. J. 7, 155-162.

POWELL, M.J.D. [1970], *A FORTRAN subroutine for unconstrained minimization requiring first derivatives of the objective function*, UKAEA Res. Gp. Rep. AERE R6469.

5. NIET-LINEAIRE VERGELIJKINGEN EN OPTIMALISERING

5.3. Constrained minimization via unconstrained minimization

door F.A. Lootsma
(Technische Hogeschool, Delft.)

5.3.1. Introduction

This section is concerned with the problem of minimizing a function of n variables over an area defined by a finite set of inequalities and equalities. In its general form the problem can be written as:

$$\begin{array}{lll} & \text{minimize} & f(x_1,\ldots,x_n) \\ (5.3.1) & \text{subject to} & g_i(x_1,\ldots,x_n) \geq 0;\ i = 1,\ldots,m; \\ & & h_j(x_1,\ldots,x_n) = 0;\ j = 1,\ldots,p; \end{array}$$

where f denotes the objective function, and $g_1,\ldots,g_m$, $h_1,\ldots,h_p$ the constraint functions. A particular problem to be found in this area is the classical equality-constrained Lagrange problem:

$$\begin{array}{lll} & \text{minimize} & f(x_1,\ldots,x_n) \\ (5.3.2) & \text{subject to} & h_j(x_1,\ldots,x_n) = 0;\ j = 1,\ldots,p. \end{array}$$

In the last few decades, considerable attention has also been given to the inequality-constrained problem:

$$\begin{array}{lll} & \text{minimize} & f(x_1,\ldots,x_n) \\ (5.3.3) & \text{subject to} & g_i(x_1,\ldots,x_n) \geq 0;\ i = 1,\ldots,m, \end{array}$$

which could be investigated with tools from the theory of linear inequalities. The idea of using unconstrained-minimization methods to solve constrained-minimization problems is not very new. The first suggestion is due to COURANT [1943]. He proposed to reduce the computational process for solving a constrained problem to sequential unconstrained minimization of a penalty function combining in a particular way the objective function, the constraint functions and a so-called controlling parameter. In the nineteen fifties several methods of this nature have been suggested. However, the theoretical basis and the extensive numerical experience, both necessary to convert the idea into a workable tool for constrained minimization, were obtained in the nineteen sixties. Particularly, FIACCO and McCORMICK [1968] contributed to the development in that decade. In this section, we shall briefly sketch some of the ideas underlying the methods in question. For more details, the reader is also referred to the author's

monograph (LOOTSMA [1970]) which contains a classification of these methods according to the manner in which they approach the boundary of the contraint set (in practical situations, minimum solutions will almost never be found in the interior), as well as an ALGOL 60 procedure for constrained minimization via a (mixed) penalty function. An improved version (the ALGOL 60 procedure minifun published by LOOTSMA [1972b]) is also available in the NAG library (the ALGOL 60 procedure E04HAA and the FORTRAN subroutine E04HAF). It is a matter of course that several new concepts have to be introduced as soon as we turn to constrained-minimization problems. Let us consider the original problem (5.3.1). Any point $x \in E_n$ satisfying the constraints is termed a *feasible solution* of (5.3.1), and the set F of all feasible solutions is generally referred to as the *constraint set* of (5.3.1). A feasible solution $\bar{x}$ is a *local minimum solution* of (5.3.1) if there is an ε-neighbourhood $N(\bar{x},\varepsilon)$ of $\bar{x}$ such that $f(\bar{x}) \leq f(x)$ for all $x \in F \cap N(\bar{x},\varepsilon)$. A feasible solution $\bar{x}$ is a *global minimum solution* of (5.3.1) if $f(\bar{x}) \leq f(x)$ for all $x \in F$. The reader will be able to formulate definitions of weak and strong minimum solutions by analogy with the definitions of unconstrained minimum solutions.

5.3.2. Necessary conditions for constrained minima

There is an extensive theory on necessary and sufficient conditions for constrained minima, usually referred to as *Lagrange theory* if we are concerned with problem (5.3.2), and as *Kuhn-Tucker theory* if the problem under consideration is given by (5.3.3). We start off by considering the Lagrange problem (5.3.2), and we suppose that a local minimum solution $\bar{x}$ to (5.3.2) exists. It is well known that, if the functions $f, h_1,\ldots,h_p$ admit of continuous first derivatives in E_n and if the gradients of the functions $h_1,\ldots,h_p$ at the point $\bar{x}$ are linearly independent, then a vector $\bar{w} \in E_p$ can be found such that $(\bar{x},\bar{w})$ is a stationary point of the *Lagrangian function*

$$(5.3.4) \qquad f(x) - \sum_{j=1}^{p} w_j h_j(x).$$

The stationary points of (5.3.4) are characterized by the (n+p) non-linear equations

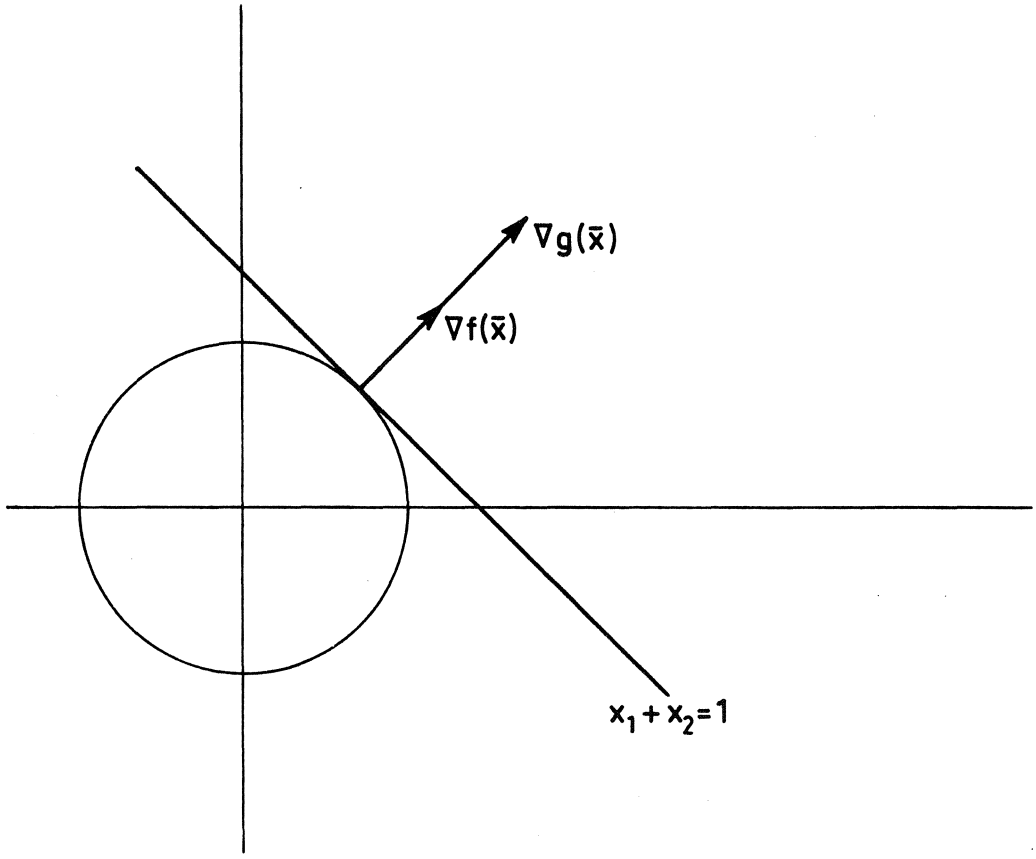


Fig. 1.

At a local minimum solution the gradient of the objective function is a linear combination of the constraint gradients.

$$\nabla f(x) - \sum_{j=1}^{p} w_j \nabla h_j(x) = 0$$

(5.3.5)

$$h_j(x) = 0;\ j = 1,\ldots,p,$$

with (n+p) variables $x_1,\ldots,x_n$, $w_1,\ldots,w_p$. Thus, the gradient $\nabla f(\bar{x})$ is a *linear combination* of the gradients $\nabla h_j(\bar{x})$, $j = 1,\ldots,p$. The reader may verify (see fig. 1) that the problem

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}(x_1^2+x_2^2) \\ &\text{subject to} && x_1 + x_2 = 1 \end{aligned}$$

has the minimum solution $(\frac{1}{2},\frac{1}{2})$, and that the gradient of the objective function is a multiple of the constraint gradient at $(\frac{1}{2},\frac{1}{2})$.

The technique of solving the system (5.3.5) by unconstrained minimization of the Lagrangian function (5.3.4) in order to obtain a solution of the Lagrange problem (5.3.2) is called the *Lagrangian-multiplier technique*. The components $\bar{w}_j$, $j = 1,\ldots,p$, are known as the Lagrangian multipliers.

The above ideas can also be applied to the inequality-constrained problem (5.3.3). First, we convert (5.3.3) into an equality-constrained problem by the introduction of *quadratic slacks*, i.e. we write

(5.3.6)
$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && g_i(x) - s_i^2 = 0;\ i = 1,\ldots,m. \end{aligned}$$

The slack variables s_i, $i = 1,\ldots,m$, are unrestricted. Now, the Lagrangian function for solving problem (5.3.6) can be written as

$$f(x) - \sum_{i=1}^{m} u_i[g_i(x) - s_i^2], \tag{5.3.7}$$

and any stationary point $(\bar{x},\bar{s},\bar{u})$ of (5.3.7) satisfies the equations

$$\nabla f(\bar{x}) - \sum_{i=1}^{m} \bar{u}_i \nabla g_i(\bar{x}) = 0, \tag{5.3.8}$$

(5.3.9) $g_i(\bar{x}) - \bar{s}_i^2 = 0;\ i = 1,\ldots,m,$

(5.3.10) $2\,\bar{s}_i\bar{u}_i = 0;\ i = 1,\ldots,m.$

The slacks $\bar{s}_i$ can be eliminated. We rewrite (5.3.9) as

(5.3.11) $g_i(\bar{x}) \geq 0;\ i = 1,\ldots,m.$

Moreover, if $\bar{u}_i \neq 0$, it must be true that $\bar{s}_i = 0$ so that $g_i(\bar{x}) = 0$. Thus, formula (5.3.10) can be replaced by

(5.3.12) $\bar{u}_i g_i(\bar{x}) = 0,\ i = 1,\ldots,m.$

Finally, we note that the Lagrangian function (5.3.7) cannot have a local minimum at $(\bar{x},\bar{s},\bar{u})$ if $\bar{u}_i < 0$ for some i. In this *heuristic manner*, we have obtained the following results. If $\bar{x}$ is a local minimum solution, then a vector $u \in E_m$ can be found such that

(5.3.13) $\nabla f(\bar{x}) - \sum_{i=1}^{m} \bar{u}_i \nabla g_i(\bar{x}) = 0,$

(5.3.14) $\bar{u}_i g_i(\bar{x}) = 0;\ i = 1,\ldots,m.$

(5.3.15) $\bar{u}_i \geq 0;\ i = 1,\ldots,m.$

These are the well-known Kuhn-Tucker relations for inequality-constrained problems, and they can be interpreted as follows. If $g_i(\bar{x}) > 0$, then $\bar{u}_i = 0$, and the gradient of the i-th constraint (which is *inactive* at $\bar{x}$) does not yield any contribution in (5.3.13). In fact, (5.3.13) and (5.3.15) state that the gradient of the objective function is a *non-negative* linear combination of the gradients corresponding to the constraints which are *active* at $\bar{x}$. This result is also illustrated in fig. 2. A rigorous establishment of the Kuhn-Tucker relations may be found in LOOTSMA [1970].

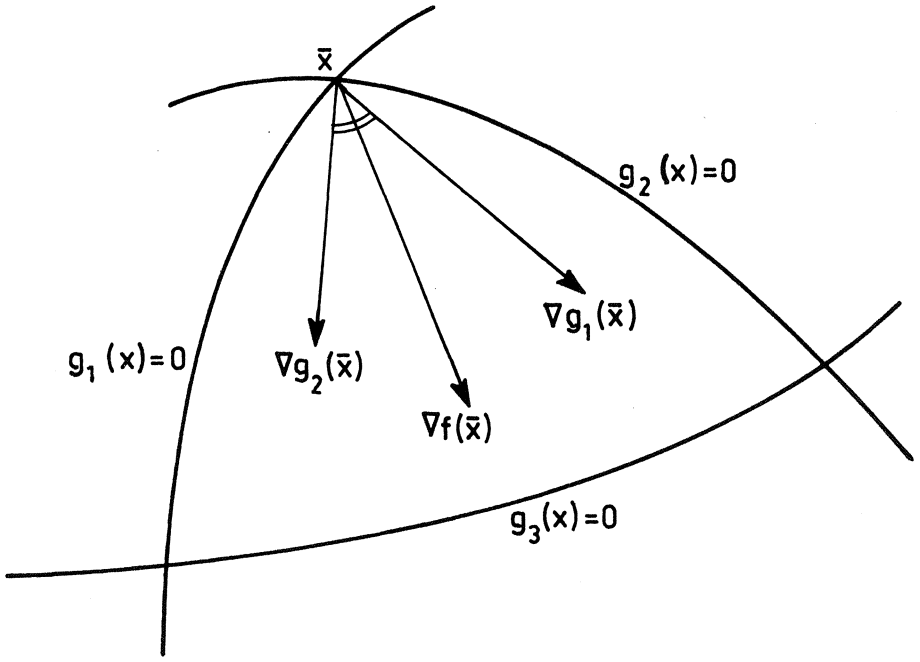


Fig. 2.

At a minimum solution $\bar{x}$ of an inequality-constrained problem, the gradient of the objective function is a non-negative linear combination of the gradients corresponding to the constraints which are active at $\bar{x}$ (Kuhn-Tucker relations).

5.3.3. Simple transformations

The preceeding paragraph contains the suggestion to solve a constrained-minimization problem via unconstrained minimization of the associated Lagrangian function. By such a method the dimensionality of the problem is considerably increased. The original problems (5.3.2) and (5.3.3) are problems in the n-dimensional vector space E_n, but the Lagrangian functions (5.3.4) and (5.3.7) are functions of (n+p) and (n+2m) variables respectively. However, if there are simple constraints in the problem under consideration, conversion into unconstrained problems *of the same dimension* is also relatively simple:

a) If the problem under consideration is presented as one of minimization subject to non-negativity constraints imposed on the variables x_j, then the variables can be replaced by squares of new variables y_j. Thus, the constrained problem

$$\begin{aligned} &\text{minimize} && f(x_1,\ldots,x_n) \\ &\text{subject to} && x_j \geq 0;\ j = 1,\ldots,n, \end{aligned}$$

can be written as the unconstrained problem

$$\text{minimize} \quad f(y_1^2,\ldots,y_n^2).$$

Alternatively, we can replace x_j by $\exp(y_j)$ in order to maintain non-negativity.

b. An obvious extension of the above ideas is to replace the constrained problem

$$\begin{aligned} &\text{minimize} && f(x_1,x_2,x_3) \\ &\text{subject to} && x_3 \geq x_2 \geq x_1 \geq 0 \end{aligned}$$

by the unconstrained problem

$$\text{minimize } f(y_1^2, y_1^2+y_2^2, y_1^2+y_2^2+y_3^2).$$

c. Simple constraints of the form

$$0 \leq x_j \leq 1$$

can easily be taken into account by the substitution of

$$x_j = \sin^2 y_j .$$

d. General bounds on the variables presented in the form

$$a_j \leq x_j \leq b_j$$

can be handled by the substitution of

$$x_j = a_j + (b_j - a_j)\sin^2 y_j ,$$

or

$$x_j = \tfrac{1}{2}(b_j + a_j) + \tfrac{1}{2}(b_j - a_j)\sin y_j .$$

For some more details of these simple transformations (substitutions into the objective function) the reader is referred to BOX [1966].

5.3.4. Exterior methods

We shall now turn to a *general* approach whereby a constrained problem is converted into a *sequence* of unconstrained problems *of the same dimension*. The original suggestion of COURANT [1943] for solving the equality-constrained Lagrange problem (5.3.2) is to form the *exterior penalty function* (loss function)

$$(5.3.16) \qquad L_r(x) = f(x) + r^{-1} \sum_{j=1}^{p} h_j^2(x)$$

and to minimize this function *successively* for a sequence of positive, decreasing values $r_0, r_1, r_2, \ldots$ of the *controlling parameter* r. Let x(r) denote a point which minimizes the function L_r over E_n for a fixed value $r > 0$. Then the proposal is based on the idea that any limit point of the sequence $x(r_0)$, $x(r_1)$, $x(r_2), \ldots$ is a minimum solution of the constrained

problem (5.3.2) if the sequence $\{r_k\}$ converges to 0. We may illustrate the idea with the example

$$\text{(5.3.17)} \qquad \begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 \\ \text{subject to} & x_1 + x_2 = 1. \end{array}$$

The exterior penalty function for solving this problem is given by

$$L_r(x_1,x_2) = x_1^2 + x_2^2 + r^{-1}(x_1+x_2-1)^2.$$

The unconstrained minimum solution of L_r for $r > 0$ is obtained by putting the first-order derivatives of L_r equal to 0 and by solving the minimum solution $(x_1(r),x_2(r))$ from the resulting equations. Thus,

$$x_1(r) + r^{-1}\{x_1(r)+x_2(r)-1\} = 0,$$

$$x_2(r) + r^{-1}\{x_1(r)+x_2(r)-1\} = 0.$$

This will immediately lead to

$$x_1(r) = x_2(r),$$

$$x_1(r) + r^{-1}\{2\, x_1(r)-1\} = 0,$$

whence

$$\text{(5.3.18)} \qquad x_1(r) = x_2(r) = \frac{1}{r+2}.$$

The reader may verify that

$$\lim_{r\downarrow 0} x_1(r) = \lim_{r\downarrow 0} x_2(r) = \tfrac{1}{2},$$

and that $(\frac{1}{2},\frac{1}{2})$ is precisely the minimum solution of the original problem (5.3.17). So, if we employ the sequence $r = 1,\frac{1}{2},\frac{1}{4},\ldots$ then we approximate the constrained minimum solution with the sequence $(\frac{1}{3},\frac{1}{3})$, $(\frac{2}{5},\frac{2}{5})$, $(\frac{4}{9},\frac{4}{9}),\ldots$. We may note that a sequence with faster convergence can be obtained by

extrapolation techniques (with the same sequence $r = 1, \frac{1}{2}, \frac{1}{4}, \ldots$). These techniques are based on the observation that the trajectory $(x_1(r), x_2(r))$ can be expanded in a Taylor series about $r = 0$. We have indeed

$$x_1(r) = \frac{1}{r+2} = \frac{1}{2} - \frac{1}{4}r + \frac{1}{8}r^2 \ldots \tag{5.3.19}$$

and it is well known in numerical mathematics how one might employ this result to accelerate the computations. Linear extrapolation, for instance, with the points

$$x(r_0) = \begin{pmatrix} 0.3333 \\ 0.3333 \end{pmatrix}$$

and

$$x(r_1) = \begin{pmatrix} 0.4000 \\ 0.4000 \end{pmatrix}$$

as grid points, yields the point

$$\begin{pmatrix} 0.4667 \\ 0.4667 \end{pmatrix}$$

as the first-order approximation to the minimum solution

$$\begin{pmatrix} 0.5000 \\ 0.5000 \end{pmatrix}.$$

Quadratic extrapolation based on the grid points $x(r_0)$, $x(r_1)$, and

$$x(r_2) = \begin{pmatrix} 0.4444 \\ 0.4444 \end{pmatrix}$$

yields the second-order approximation

$$\begin{pmatrix} 0.4962 \\ 0.4962 \end{pmatrix}$$

to the minimum solution of the original problem (5.3.17), This example clearly demonstrates the acceleration due to polynomial extrapolation towards r = 0.

The above ideas have been used to design a method for solving inequality constraints. A thorough study was carried out by ZANGWILL [1967] and FIACCO and McCORMICK [1968]. The problem under consideration is problem (5.3.3) and an *exterior penalty function* (loss function) for solving it is given by

$$(5.3.20) \qquad L_r(x) = f(x) + r^{-1} \sum_{i=1}^{m} \min^2[0, g_i(x)].$$

The right-hand term (the loss term) contains the squared constraint violations which are penalized by the weight r^{-1}. The computational method using (5.3.20) to solve problem (5.3.3) is similar to the above sketched method for solving (5.3.2). It is also based on the idea that any limit point of the sequence $\{x(r_k)\}$ of penalty-function minima is a solution of the original problem (5.3.3) if $\{r_k\}$ is a decreasing null sequence.

5.3.5. Interior methods

It can be demonstrated that the unconstrained minima of the functions (5.3.16) and (5.3.20) do not belong to the constraint set of the problems (5.3.2) or (5.3.3) respectively. This may also explain why these functions are referred to as *exterior* functions. For the inequality-constrained problem (5.3.3), however, there is also an *interior approach* which does not violate the boundaries of the constraint set. The *interior penalty function* (barrier function) originally proposed by FRISCH [1955] to solve (5.3.3) is given by

$$(5.3.21) \qquad B_r(x) = f(x) - r \sum_{i=1}^{m} \ln g_i(x).$$

This function is defined in the set

$$R^0 = \{x \mid g_i(x) > 0;\ i = 1,\ldots,m\},$$

but it has a *positive singularity* at the boundary of the constraint set R of problem (5.3.3). Moreover, this function is undefined outside R. Under mild conditions the function B_r has an unconstrained minimum solution x(r) over R^0 for any $r > 0$. The interesting property established some years after the appearance of FRISCH' memorandum is that any limit point of the sequence $\{x(r_k)\}$ of barrier-function minima is a minimum solution of problem (5.3.3) provided that $\{r_k\}$ is a decreasing null sequence.
To illustrate matters we consider the problem

$$(5.3.22) \qquad \begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 \\ \text{subject to} & x_1 + x_2 \geq 1. \end{array}$$

The *logarithmic* barrier function (5.3.21) for solving this problem reduces to

$$(5.3.23) \qquad B_r(x_1,x_2) = x_1^2 + x_2^2 - r \ln(x_1+x_2-1).$$

Differentiating this function with respect to x_1 and x_2, and putting the partial derivatives equal to 0 we obtain that

$$2\, x_1(r) - \frac{1}{x_1(r) + x_2(r) - 1} = 0,$$

$$2\, x_2(r) - \frac{r}{x_1(r) + x_2(r) - 1} = 0,$$

whence

$$x_1(r) = x_2(r)$$

and

$$x_1(r) = \frac{2 \pm \sqrt{1 + 4r}}{8} \quad .$$

The reader may verify that only the positive sign in the above formula can be used (since interior penalty-function minima must strictly satisfy

the constraints). Thus,

$$x_1(r) = x_2(r) = \tfrac{1}{4} + \tfrac{1}{4}\sqrt{1 + 4r}.$$

It follows easily that

$$\lim_{r \downarrow 0} x_1(r) = \lim_{r \downarrow 0} x_2(r) = \tfrac{1}{2}.$$

Using the sequence $r = 1, \frac{1}{2}, \frac{1}{4}, \ldots$ we approximate the minimum solution $(\frac{1}{2}, \frac{1}{2})$ with the sequence $(\frac{1}{4}(1+\sqrt{5}), \frac{1}{4}(1+\sqrt{5}))$, $(\frac{1}{4}(1+\sqrt{3}), \frac{1}{4}(1+\sqrt{3}))$, $(\frac{1}{4}(1+\sqrt{2}), \frac{1}{4}(1+\sqrt{2})), \ldots$. Moreover, the trajectory $(x_1(r), x_2(r))$ can be expanded in a Taylor series about $r = 0$, so that extrapolation techniques may be used to accelerate the computations. It is easy to verify that

$$x_1(r) = x_2(r) = \tfrac{1}{4}[2 + 2r - \frac{r^2}{2} + \ldots] ,$$

thus providing a basis for extrapolation (a series expansion in terms of r). Some differences and similarities between the exterior and interior methods will now be clear. Both types of methods use a controlling parameter r in the penalty function to balance the contribution of the objective function and the constraints. Exterior methods, however, penalize constraint violations with an increasing weight, whereas interior methods penalize the approach of the boundary with a decreasing weight. The singularity of B_r at the boundary of R acts as a barrier in order to prevent the unconstrained-minimization methods for finding x(r) from obtaining a point outside the constraint set R.

Another well-known barrier function was originally proposed by CARROLL [1959] and subsequently studied by FIACCO and McCORMICK [1963,1968]. This so-called inverse barrier function is given by

$$f(x) + r \sum_{i=1}^{m} \frac{1}{g_i(x)} ,$$

and it has similar properties as the logarithmic barrier function. At present, however, it is less frequently used, in view of certain advantages of the logarithmic barrier function.

An important property of the method based on the logarithmic barrier function (5.3.21) can easily be derived from the observation that the gradient of B_r has to vanish at a minimizing point x(r). Thus, the relation $\nabla B_r(x(r)) = 0$ leads to

$$(5.3.24) \qquad \nabla f(x(r)) - r \sum_{i=1}^{m} \frac{\nabla g_i(x(r))}{g_i(x(r))} = 0.$$

Introducing the vector $u(r) \in E_m$ with components

$$u_i(r) = \frac{r}{g_i(x(r))} \; ; \; i = 1,\ldots,m,$$

we can convert (5.3.24) into the system

$$(5.3.25) \qquad \nabla f(x(r)) - \sum_{i=1}^{m} u_i(r) \nabla g_i(x(r)) = 0$$

$$(5.3.26) \qquad u_i(r)\, g_i(x(r)) = r; \; i = 1,\ldots, m$$

$$(5.3.27) \qquad u_i(r) > 0; \; i = 1,\ldots, m.$$

Obviously, there is a striking similarity with the Kuhn-Tucker relations (5.3.13) - (5.3.15). In fact, it can be shown that the pair $(x(r), u(r))$ constitutes an approximation to the pair $(\bar{x},\bar{u})$ which satisfies the Kuhn-Tucker relations. This phenomenon has appeared to be an extremely useful starting point for the analysis of penalty-function methods.

5.3.6. Mixed penalty functions

Although the computational processes of exterior and interior methods are basically the same, they present particular advantages and disadvantages. It is not our purpose to list all the different features (some advantages of the interior methods would require a thorough analysis). In this report we shall restrict ourselves to the most striking properties.

Interior methods are safer than exterior methods since the computations are entirely concerned with feasible solutions within the constraint

set (in practice, the user does not always properly define his objective function outside the constraint set). Interior methods have the following drawbacks, however:

(a) They cannot handle equality constraints.

(b) They can only start from strictly feasible solutions.

(c) The linear search of the unconstrained-minimization techniques to find barrier-function minima is complicated by the requirement that unfeasible solutions must be avoided.

Exterior methods have obvious advantages (they can handle equality constraints, they can start from any point, and the linear search is not complicated by the presence of constraints), but they will invariably leave the constraint set (unless the objective function has an unconstrained minimum in the constraint set), so that one may easily run into troubles if the objective function is not properly defined there.

A natural remedy seems to be a penalty function which is a combination of a loss function and a barrier function. Thus, we are led to a *mixed penalty function*, which incorporates some of the inequalities into a barrier term, and the remaining inequalities as well as the equalities into a loss term. So, a mixed penalty function for solving the general problem (5.3.1) is given by

$$M_r(x) = f(x) + r\, b(x) + r^{-1} [\ell(x) + e(x)] \tag{5.3.28}$$

combining the logarithmic barrier term

$$b(x) = - \sum_{i \in I_1} \ln g_i(x),$$

and the loss terms

$$\ell(x) = - \sum_{i \in I_2} \min^2 [0, g_i(x)],$$

$$e(x) = \sum_{j=1}^{p} h_j^2(x).$$

The index sets I_1 and I_2 are defined by

$$I_1 = \{i | g_i(x^0) > 0;\ 1 \leq i \leq m\},$$

$$I_2 = \{i | g_i(x^0) \leq 0;\ 1 \leq i \leq m\},$$

where x^0 denotes the starting point of the computations for solving (5.3.1). The choice of I_1 and I_2 implies that constraints which are strictly satisfied at the start will remain satisfied throughout the computations. Thus, the typical safety of barrier function methods is preserved as much as possible. On the other hand, the easy starting facilities of the loss-function methods are also available, since the computations may start from any point, feasible or not.

A useful refinement of the employment of a mixed penalty function seems to be the following one. First, the infeasibilities are *reduced* by the unconstrained minimization of

$$(5.3.29) \qquad \sum_{i=1}^{m} \min^2(0, g_i(x)) + \sum_{j=1}^{p} h_j^2(x),$$

and the minimum of this process is used as the starting point to solve the problem (5.3.1) via the mixed penalty function (5.3.28). The initial minimization of (5.3.29) will frequently yield a point which satisfies the inequalities $g_i(x) \geq 0;\ 1,\ldots,m$, and it will certainly lead to a mixed penalty function which is better behaved.

The author (LOOTSMA [1972a]) has published his computational results with a variety of test problems to show the relative efficiency of several unconstrained-minimization methods when they are used to solve constrained problems. Recently, STAHA [1973] gave a detailed account of the relative performance of penalty-function techniques with respect to various other well-known methods for constrained minimization. As a result, penalty-function methods seem to be competitive, and they have the considerable advantages of conceptual simplicity and generality.

Literatuur

BOX, M.J. [1966], *A Comparison of Several Current Optimization Methods, and the Use of Transformations in Constrained Problems*. The Comp. J. 8, 67-77.

CARROLL, C.W. [1961], *The Created Response Surface Technique for Optimizing Non-linear Restrained Systems*, Opns. Res. 9, 169-184.

COURANT, R. [1943], *Variational Methods for the Solution of Problems of Equilibrium and Vibrations*, Bull. Am. Math. Soc. 49, 1-23.

FIACCO, A.V. & G.P. McCORMICK [1963], *Programming under Non-linear Constraints by Unconstrained Minimization: a Primal-Dual Method.* Research Analysis Corporation, McLean, Va., USA, RAC-TP-96.

FIACCO, A.V. & G.P. McCORMICK [1968], *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, New York.

FRISCH, R. [1955], *The Logarithmic Potential Method for Solving Linear Programming Problems*. Memorandum of the University Institute of Economics, Oslo, Norway.

LOOTSMA, F.A. [1970], *Boundary Properties of Penalty Functions for Constrained Minimization*, Philips Res. Repts. Suppl. No. 3.

LOOTSMA, F.A. [1972a], *Penalty Function Performance of Several Unconstrained Minimization Techniques*, Philips Res. Repts. 27, 358-385.

LOOTSMA, F.A. [1972b], *The ALGOL 60 Procedure minifun for Solving Nonlinear Optimization Problems*, Philips Research Laboratories, Eindhoven, Netherlands. Report 4761.

STAHA, R.L. [1973], *Constrained Optimization via Moving Exterior Truncations*. Thesis. University of Texas, Austin, Texas, U.S.A.

ZANGWILL, W.I. [1967], *Nonlinear Programming via Penalty Functions.* Management Science 13. 344-358.

UITGAVEN IN DE SERIE MC SYLLABUS

Onderstaande uitgaven zijn verkrijgbaar bij het Mathematisch Centrum, 2e Boerhaavestraat 49 te Amsterdam-1005, tel. 020-947272.

MCS 1.1 F. GÖBEL & J. VAN DE LUNE, *Leergang Besliskunde, deel 1: Wiskundige basiskennis*, 1965. ISBN 90 6196 014 2.

MCS 1.2 J. HEMELRIJK & J. KRIENS, *Leergang Besliskunde, deel 2: Kansberekening*, 1965. ISBN 90 6196 015 0.

MCS 1.3 J. HEMELRIJK & J. KRIENS, *Leergang Besliskunde, deel 3: Statistiek*, 1966. ISBN 90 6196 016 9.

MCS 1.4 G. DE LEVE & W. MOLENAAR, *Leergang Besliskunde, deel 4: Markovketens, en wachttijden*, 1966. ISBN 90 6196 017 7.

MCS 1.5 J. KRIENS & G. DE LEVE, *Leergang Besliskunde, deel 5: Inleiding tot de mathematische besliskunde*, 1966. ISBN 90 6196 018 5.

MCS 1.6a B. DORHOUT & J. KRIENS, *Leergang Besliskunde, deel 6a: Wiskundige programmering 1*, 1968. ISBN 90 6196 032 0.

MCS 1.7a G. DE LEVE, *Leergang Besliskunde, deel 7a: Dynamische programmering 1*, 1968. ISBN 90 6196 033 9.

MCS 1.7b G. DE LEVE & H.C. TIJMS, *Leergang Besliskunde, deel 7b: Dynamische programmering 2*, 1970. ISBN 90 6196 055 X.

MCS 1.7c G. DE LEVE & H.C. TIJMS, *Leergang Besliskunde, deel 7c: Dynamische programmering 3*, 1971. ISBN 90 6196 066 5.

MCS 1.8 J. KRIENS, F. GÖBEL & W. MOLENAAR, *Leergang Besliskunde, deel 8: Minimaxmethode, netwerkplanning, simulatie*, 1968. ISBN 90 6196 034 7.

MCS 2.1 G.J.R. FÖRCH, P.J. VAN DER HOUWEN & R.P. VAN DE RIET, *Colloquium stabiliteit van differentieschema's, deel 1*, 1967. ISBN 90 6196 023 1.

MCS 2.2 L. DEKKER, T.J. DEKKER, P.J. VAN DER HOUWEN & M.N. SPIJKER, *Colloquium stabiliteit van differentieschema's, deel 2*, 1968. ISBN 90 6196 035 5.

MCS 3.1 H.A. LAUWERIER, *Randwaardeproblemen, deel 1*, 1967. ISBN 90 6196 024 X.

MCS 3.2 H.A. LAUWERIER, *Randwaardeproblemen, deel 2*, 1968. ISBN 90 6196 036 3.

MCS 3.3 H.A. LAUWERIER, *Randwaardeproblemen, deel 3*, 1968. ISBN 90 6196 043 6.

MCS 4 H.A. LAUWERIER, *Representaties van groepen*, 1968. ISBN 90 6196 037 1.

MCS 5 J.H. VAN LINT, J.J. SEIDEL & P.C. BAAYEN, *Colloquium discrete wiskunde*, 1968. ISBN 90 6196 044 4.

MCS 6 K.K. KOKSMA, *Cursus ALGOL 60*, 1969. ISBN 90 6196 045 2.

MCS 7.1 *Colloquium Moderne rekenmachines, deel 1*, 1969. ISBN 90 6196 046 0.

MCS 7.2 *Colloquium Moderne rekenmachines, deel 2*, 1969. ISBN 90 6196 047 9.

MCS 8 H. BAVINCK & J. GRASMAN, *Relaxatietrillingen*, 1969. ISBN 90 6196 056 8.

MCS 9.1 T.M.T. COOLEN, G.J.R. FÖRCH, E.M. DE JAGER & H.G.J. PIJLS, *Elliptische differentiaalvergelijkingen, deel 1*, 1970. ISBN 90 6196 048 7.

MCS 9.2 W.P. VAN DEN BRINK, T.M.T. COOLEN, B. DIJKHUIS, P.P.N. DE GROEN, P.J. VAN DER HOUWEN, E.M. DE JAGER, N.M. TEMME & R.J. DE VOGELAERE, *Colloquium Elliptische differentiaalvergelijkingen, deel 2*, 1970. ISBN 90 6196 049 5.

MCS 10 J. FABIUS & W.R. VAN ZWET, *Grondbegrippen van de waarschijnlijkheidsrekening*, 1970. ISBN 90 6196 057 6.

MCS 11 H. BART, M.A. KAASHOEK, H.G.J. PIJLS, W.J. DE SCHIPPER & J. DE VRIES, *Colloquium Halfalgebra's en positieve operatoren*, 1971. ISBN 90 6196 067 3.

MCS 12 T.J. DEKKER, *Numerieke algebra*, 1971. ISBN 90 6196 068 1.

MCS 13 F.E.J. KRUSEMAN ARETZ, *Programmeren voor rekenautomaten; De MC ALGOL 60 vertaler voor de EL X8*, 1971. ISBN 90 6196 069 X.

MCS 14 H. BAVINCK, W. GAUTSCHI & G.M. WILLEMS, *Colloquium Approximatietheorie*, 1971. ISBN 90 6196 070 3.

MCS 15.1 T.J. DEKKER, P.W. HEMKER & P.J. VAN DER HOUWEN, *Colloquium Stijve differentiaalvergelijkingen, deel 1*, 1972. ISBN 90 6196 078 9.

MCS 15.2 P.A. BEENTJES, K. DEKKER, H.C. HEMKER, S.P.N. VAN KAMPEN & G.M. WILLEMS, *Colloquium Stijve differentiaalvergelijkingen, deel 2*, 1973. ISBN 90 6196 079 7.

MCS 15.3 P.A. BEENTJES, K. DEKKER, P.W. HEMKER & M. VAN VELDHUIZEN, *Colloquium Stijve differentiaalvergelijkingen, deel 3*, 1975. ISBN 90 6196 118 1.

MCS 16.1 L. GEURTS, *Cursus Programmeren, deel 1: De elementen van het programmeren*, 1973. ISBN 90 6196 080 0.

MCS 16.2 L. GEURTS, *Cursus Programmeren, deel 2: De programmeertaal ALGOL 60*, 1973. ISBN 90 6196 087 8.

MCS 17.1 P.S. STOBBE, *Lineaire algebra, deel 1*, 1974. ISBN 90 6196 090 8.

MCS 17.2 P.S. STOBBE, *Lineaire algebra, deel 2*, 1974. ISBN 90 6196 091 6.

MCS 17.3 N.M. TEMME, *Lineaire algebra, deel 3*, 1976. ISBN 90 6196 123 8.

MCS 18 F. VAN DER BLIJ, H. FREUDENTHAL, J.J. DE IONGH, J.J. SEIDEL & A. VAN WIJNGAARDEN, *Een kwart eeuw wiskunde 1946-1971, Syllabus van de Vakantiecursus 1971*, 1974. ISBN 90 6196 092 4.

MCS 19 A. HORDIJK, R. POTHARST & J.TH. RUNNENBURG, *Optimaal stoppen van Markovketens*, 1974. ISBN 90 6196 093 2.

MCS 20 T.M.T. COOLEN, P.W. HEMKER, P.J. VAN DER HOUWEN & E. SLAGT, *ALGOL 60 procedures voor begin- en randwaardeproblemen*, 1976. ISBN 90 6196 094 0.

MCS 21 J.W. DE BAKKER (red.), *Colloquium Programmacorrectheid*, 1975. ISBN 90 6196 103 3.

* MCS 22 R. HELMERS, F.H. RUYMGAART, M.C.A. VAN ZUYLEN & J. OOSTERHOOF, *Asymptotische methoden in de toetsingstheorie toepassingen van naburigheid*, 1976. ISBN 90 6196 104 1.

MCS 23.1 J.W. DE ROEVER (red.), *Colloquium Onderwerpen uit de biomathematica, deel 1*, 1976. ISBN 90 6196 105 X.

* MCS 23.2 J.W. DE ROEVER (red.), *Colloquium Onderwerpen uit de biomathematica, deel 2*, 1976. ISBN 90 6196 115 7.

* MCS 24.1 P.J. VAN DER HOUWEN, *Numerieke integratie van differentiaalvergelijkingen, deel 1: Eenstapsmethoden* , 1974. ISBN 90 6196 106 8.

MCS 25 *Colloquium Structuur van Programmeertalen*, 1976. ISBN 90 6196 116 5.

MCS 26.1 N.M. TEMME (red.), *Nonlinear Analysis, volume 1*, 1976. ISBN 90 6196 117 3.

MCS 26.2 N.M. TEMME (red.), *Nonlinear Analysis, volume 2*, 1976. ISBN 90 6196 121 1.

MCS 27 M. BAKKER, P.W. HEMKER, P.J. VAN DER HOUWEN, S.J. POLAK & M. VAN VELDHUIZEN, *Colloquium Discreteringmethoden*, 1976. ISBN 90 6196 124 6.

* MCS 28 O. DIEKMAN, N.M. TEMME (EDS.), *Nonlinear Diffusion Problems*, 1976. ISBN 90 6196 126 2.

MCS 29.1 J.C.P. BUS (red.), *Numerieke Programmatuur, deel 1A, deel 1B*, 1976. ISBN 90 6196 128 9.

* MCS 29.2 J.C.P. Bus (red.), *Numerieke Programmatuur, deel 2*, ISBN 90 6196 144 0.

MCS 31 J.H. VAN LINT (red.), *Inleiding in de Coderingstheorie*, 1976. ISBN 90 6196 136 X.

MCS 32 L. GEURTS (red.), *Colloquium Bedrijfssystemen*, 1976. ISBN 90 6196 137 8.

* MCS 33 P.J. VAN DER HOUWEN, *Differentieschema's voor de berekening van waterstanden in zeeën en rivieren*, ISBN 90 6196 138 6.

* MCS 34 J. HEMELRIJK, *Orienterende cursus mathematische statistiek*, IBSN 90 6196 139 4.

De met een * gemerkte uitgaven moeten nog verschijnen.