RA

stichting
mathematisch
centrum

MC

REKENAFDELING NR 29/72 SEPTEMBER

J.C.P. BUS

MINIMALISERING VAN FUNKTIES VAN MEERDERE VARIABELEN

RA

2e boerhaavestraat 49 amsterdam

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

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

Inhoud

Voorwoord

In dit rapport wordt verslag uitgebracht van enig onderzoek op het gebied van het minimaliseren van niet-lineaire funkties in meerdere variabelen.

De schrijver wil zijn dank betuigen aan Prof.Dr.T.J. Dekker voor diens stimulerende leiding, waardevolle suggesties en kritiek en aan Drs.W. Hoffmann voor diens inspirerende opmerkingen.

Hoofdstuk I. Probleemstelling en Methoden

§1. Inleiding

Zij $F : R_n \to R$ een tweemaal kontinu differentieerbare naar onder begrensde niet-lineaire funktie.
We zullen hier enige algoritmen behandelen ter bepaling van een minimum van F, aannemende dat F en zijn gradient g expliciet zijn gegeven.

Definitie 1.1
Een symmetrische $(n \times n)$-matrix A heet positief definiet als geldt:

$$y^T A y > 0 \quad , \quad \forall y \in R_n \setminus \{0\}. \tag{1.1}$$

Er geldt: de eigenwaarden van een symmetrische positief definiete matrix zijn positief.

Definitie 1.2
Een twee maal kontinu differentieerbare funktie heet konvex in een gebied, als de matrix der partiële tweede afgeleiden daar positief definiet is.

We zullen de matrix der partiële tweede afgeleiden van F steeds de Hessiaan van F noemen en aanduiden met G. De inverse van de Hessiaan G zullen we in het vervolg aanduiden met H.

Zij x een vektor, dan wordt in het vervolg met $||x||$ de Euclidische norm van x aangegeven.
Zij A een matrix, dan wordt in het vervolg met $||A||$ de spectrale norm van A aangegeven.

Voor de bepaling van een minimum van F zijn twee klassieke iteratieve methoden bekend.
Zij $x_0 = (x_0^1, x_0^2, \ldots, x_0^n)^T$ een approximatie van de plaats van een minimum van F.

1° De methode van "steepest descent" (Curry(1944)).

De k-de stap (k=0,1,2,...) van dit iteratieproces wordt gedefinieerd door:

$$x_{k+1} - x_k = -\alpha_k g(x_k) \ , \tag{1.2}$$

waarbij α_k een geschikt gekozen konstante is (k=0,1,2...) .

2° De gedempte methode van Newton voor de bepaling van een nulpunt van de gradient.

De k-de stap (k=0,1,2,...) van dit iteratieproces wordt gedefinieerd door:

$$x_{k+1} - x_k = -\alpha_k G^{-1}(x_k)\, g(x_k)\ , \tag{1.3}$$

waarbij $\alpha_k > 0$ een geschikt gekozen konstante is (k=0,1,2...).

In deze twee methoden kunnen we al direkt de volgende twee basisproblemen onderscheiden :

- de bepaling van de staprichting, d.i. de richting waarin we een nieuwe approximatie van een minimum zoeken;
- de bepaling van de staplengte, d.i. de keuze van α_k.

Hoewel de gedempte methode van Newton kwadratisch konvergeert in een gebied waarin de funktie konvex is, kan een groot bezwaar van deze methode zijn, dat in iedere iteratiestap de Hessiaan moet worden berekend en een bijbehorend lineair stelsel moet worden opgelost.
De methode van steepest descent vereist weliswaar veel minder werk per stap, maar konvergeert daarentegen soms zeer langzaam.

De hier behandelde algoritmen zijn gebaseerd op de in 1959 door Davidon geïntroduceerde Variabele-metriekmethode. In deze methode wordt een benadering van de inverse van de Hessiaan tijdens het proces opgebouwd en gekorrigeerd met behulp van de verkregen informatie over de funktie. De staprichting in de Variabele-metriekmethode wordt meestal op dezelfde wijze bepaald als in Newton's methode, waarbij dan de werkelijke inverse van de Hessiaan wordt vervangen door de verkregen benadering ervan. Om deze reden wordt de Variabele-metriekmethode ook wel quasi-Newtonmethode genoemd.

Zij x_0 een approximatie van de plaats van een minimum van F en H_0 een symmetrisch matrix. (Meestal wordt voor H_0 de eenheidsmatrix, eventueel vermenigvuldigd met een skalar, gekozen.)
Dan definiëren we de k-de stap (k=0,1,2,...) van de Variabele-metriekmethode door:

Als $g^T(x_k)\, H_k g(x_k) > 0$
dan kiezen we

$$p_k = -H_k g(x_k) \quad ; \tag{1.4}$$

anders wordt p_k op een nader te bespreken wijze gekozen (zie §2).
Vervolgens kiezen we x_{k+1} zodat:

$$x_{k+1} - x_k = \alpha_k p_k \quad , \tag{1.5}$$

waarbij $\alpha_k > 0$ een geschikt gekozen konstante is (zie §3), waarvoor we meestal o.a. eisen:

$$F(x_{k+1}) < F(x_k) \quad . \tag{1.6}$$

Tenslotte korrigeren we de approximatie van de inverse Hessiaan als volgt:

$$H_{k+1} = H_k + C_k \quad , \tag{1.7}$$

waarbij C_k een symmetrische korrektiematrix is, welke in het algemeen afhankelijk is van x_{k+1}, x_k, $g(x_{k+1})$, $g(x_k)$ en H_k. (Dikwijls worden H_0 en C_k $(k=0,1,2,\ldots)$ zodanig gekozen dat H_k positief definiet is voor $k = 0, 1, 2, \ldots$.)

We zullen de zo verkregen benadering H_k van de inverse Hessiaan in het vervolg de <u>metriek</u> noemen.

Naast de twee basisproblemen, waarvoor we ons ook bij de gedempte methode van Newton en de methode van steepest descent gesteld zagen, de bepaling van de staprichting en de bepaling van de staplengte, treedt bij de Variabele-metriekmethode als derde basisprobleem op het bijhouden van de metriek, d.i. de keuze van C_k.
In §2, 3 en 4 behandelen we deze drie problemen afzonderlijk. Daarna geven we een beschrijving van enige algoritmen en leiden we enige resultaten af betreffende de konvergentie ervan. Tenslotte zullen we in hoofdstuk IV de numerieke resultaten van de behandelde algoritmen vergelijken. Deze resultaten zullen we ook vergelijken met de resultaten

van de Variabele-metriekalgoritme van Fletcher en Powell (1963), zoals deze als ALGOL-60 procedure is gegeven door Wells (1965), en tevens met de resultaten van een algoritme gebaseerd op de gedempte methode van Newton.

§2. De bepaling van de staprichting

Als

$$g^T(x_k)\, H_k g(x_k) > 0 \quad , \tag{2.1}$$

dan wordt de staprichting p_k gedefiniëerd door

$$p_k = -H_k g(x_k) \quad . \tag{2.2}$$

Als echter niet aan (2.1) is voldaan dan geldt voor de aldus gedefinieerde p_k:

$$p_k{}^T g(x_k) \geq 0 \quad . \tag{2.3}$$

De afgeleide van F in de richting p_k, in het punt x_k, is dus positief. De funktie is daarom stijgend in de richting p_k, zodat in het algemeen geen $\alpha_k > 0$ kan worden gevonden zodanig dat aan (1.6) is voldaan.
Om dit te vermijden kiezen we, als (2.1) niet geldt, een andere staprichting.
We bespreken hiervoor drie mogelijkheden.

1° Neem voor p_k de steepest-descentrichting (zie (1.2)):

$$p_k = -g(x_k) \quad . \tag{2.4}$$

2° Methode van Greenstadt (1967).

Zijn $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ de eigenwaarden van H_k en is X de matrix van bijbehorende eigenvektoren, dan geldt:

$$H_k = X \Lambda X^T \quad ,$$

waarbij $\Lambda = \text{diag}\,(\lambda_1,\ldots,\lambda_n)$.

Stel $|\Lambda| = \text{diag}\,(|\lambda_1|,\ldots,|\lambda_n|)$, dan is $X|\Lambda|X^T = K_k$ positief definiet. We kunnen dan de staprichting definiëren door

$$p_k = -K_k g(x_k) \quad (2.5)$$

Greenstadt (1967) meldt dat Newton's methode (1.3) met deze modificatie wanneer de Hessiaan niet positief definiet is in de praktijk voor zeer slecht gekonditioneerde problemen goed voldoet.

3° Methode van Goldfeld, Quandt en Trotter (1966). (zie ook Powell (1970)).

Zij Δ_k een in de k-de iteratiestap gedefinieerde schatting van de gewenste staplengte $||x_{k+1} - x_k||$.

Laten $\eta_1 \geq \eta_2 \geq \ldots \geq \eta_n$ de eigenwaarden van $G_k = H_k^{-1}$ en $v_1, \ldots, v_n$ de bijbehorende eigenvektoren zijn.
Dan definiëren we, als $v_n^T g(x_k) \neq 0$, de staprichting p_k door:

$$p_k = -(G_k+\theta I)^{-1}\, g(x_k) \quad (2.6)$$

waarbij θ voldoet aan

$$(G_k+\theta I) \text{ is positief definiet} \quad (2.7)$$

en

$$||(G_k+\theta I)^{-1}\, g(x_k)|| = \Delta_k \quad (2.8)$$

Uit (2.6) volgt

$$p_k = -\sum_{i=1}^{n} \frac{h_i v_i}{(\theta+\eta_i)} \text{ , waarbij } h_i = v_i^T g(x_k) \quad (2.9)$$

Uit (2.9) blijkt dat niet altijd een θ kan worden gevonden die aan (2.7) en (2.8) voldoet als $h_n = v_n^T g(x_k) = 0$. Zij m de grootste index zodat $h_m \neq 0$. Dan kunnen we steeds een $\theta > -\lambda_m$ vinden zodat θ voldoet aan (2.8). Substitutie van deze waarde van θ in (2.9) geeft een staprichting p_k welke voldoet aan $p_k^T g(x_k) < 0$. De funktie is dus dalend in de richting p_k, zodat mag worden verwacht, als Δ_k goed is gekozen, dat $x_{k+1} = x_k + p_k$ voldoet aan (1.6).

We zullen dus ook als $h_n = 0$ de staprichting p_k met (2.9) berekenen, ondanks het feit dat mogelijk niet aan (2.7) is voldaan, tenzij voor zekere $r > m$ zou gelden: $\theta = -\eta_r$. In dit uitzonderlijke geval is (2.9) niet meer zinvol. We kiezen dan de staprichting met behulp van:

$$p_k = \theta' v_r - \sum_{i \neq r}^{n} \frac{h_i v_i}{(\eta_i - \eta_r)} \quad , \tag{2.10}$$

waarbij θ' zo wordt gekozen dat geldt: $||p_k|| = \Delta_k$.

Merk op dat bij deze methode de staprichting p_k afhankelijk is van de gekozen schatting van een goede staplengte Δ_k. Bij een gegeven staplengte bepalen we de beste richting in de zin van het volgende lemma.

Lemma 2.1 (Goldfeld, Quandt en Trotter (1966)).
Zij F een kwadratische funktie met niet-positief definiete Hessiaan G_k. Stel $B_k = \{x \mid ||x-x_k|| \leq \Delta_k\}$. Dan wordt de minimale waarde van F op B_k aangenomen in het punt $x_k + p_k \in B_k$, waarbij p_k wordt gegeven door (2.6).

Als p_k wordt berekend met (2.9) dan geschiedt de bepaling van θ als volgt.

Als m de grootste index is zodat $h_m \neq 0$, dan geldt voor θ

$$\theta > -\lambda_m . \tag{2.11}$$

We zullen in het vervolg gemakshalve aannemen dat $m = n$.
We kunnen voor θ ook een bovengrens aangeven.
Stel

$$f(\nu) = ||p_k||^2 - \Delta_k^2 = \sum_{i=1}^{n} \frac{h_i^2}{(\nu-\eta_i)^2} - \Delta_k^2 . \tag{2.12}$$

$f(\nu)$ is monotoon dalend voor $\nu > -\lambda_n$ en er geldt:
$f(-\lambda_n) = +\infty$ en $f(\theta) = 0$.
Kies nu

$$\nu_{max} = -\lambda_n + \sqrt{n}\,\frac{H}{\Delta_k} \quad ,$$

waarbij $H = \max_{1 \le i \le n} |h_i|$,

dan geldt: $f(\nu_{max}) \le 0$,

zodat voor het nulpunt θ van f geldt:

(2.13) $$-\lambda_n < \theta \le \nu_{max} .$$

Met een algoritme voor bepaling van een nulpunt van een reële funktie, gebaseerd op interpolatie of bisectie, kunnen we θ nu eenvoudig verkrijgen. Een geschikte procedure hiervoor is bijvoorbeeld de door Dekker en Hoffmann (1968) gegeven procedure zeroin.

Opmerkingen

1° De drie gegeven methoden geven steeds een staprichting p_k waarvoor geldt $p_k^T g(x_k) < 0$. De funktie is dus steeds dalend in de verkregen richting.

2° Voor de eerste van de drie gegeven methoden is geen extra rekenwerk nodig. Gezien echter de soms zeer trage konvergentie van de methode van steepest descent is de eerste methode slechts aan te bevelen als het zeker is dat de metriek H_k slechts af en toe niet voldoet aan (2.1).

De tweede en derde methode vergen een berekening van alle eigenwaarden en eigenvektoren van de symmetrische matrix H_k. Hiervoor is het aantal benodigde aritmetische operaties van de orde n^3.

3° Als de, met de methode van Goldfeld e.a. verkregen, stap niet voldoet aan (1.6) dan zou een kleinere stap in dezelfde richting niet noodzakelijk het optimale resultaat geven, in de zin van lemma 2.1. De funktie is echter dalend in de verkregen richting, zodat wel steeds een staplengte kan worden gevonden die aan (1.6) voldoet.

§3. De bepaling van de stapgrootte

De konstante α_k in (1.2), (1.3) of (1.5) wordt in het algemeen bepaald volgens één van de drie volgende strategieën.

1° Bepaal α_k door ééndimensionaal minimaliseren.
Zij $f(\alpha) = F(x_k+\alpha p_k)$, met p_k de staprichting (zie §2).
Kies dan α_k zodanig dat f minimaal is voor $\alpha = \alpha_k$.

Als p_k is gekozen zoals beschreven in §2, dan is steeds een dergelijke $\alpha_k > 0$ te vinden (zie §2, opmerking 1 en 3). Een voorbeeld van een Variabele-metriekalgoritme waarin deze methode wordt gebruikt is de door Fletcher en Powell (1963) gemodificeerde versie van de oorspronkelijke algoritme van Davidon (1959).
Voor de meeste Variabele-metriekalgoritmen, die gebruik maken van deze strategie voor de bepaling van de staplengte, is konvergentie te bewijzen voor een ruime klasse van niet-kwadratische funkties (zie bijvoorbeeld Broyden (1970), Powell (1971)).
Voor ééndimensionale minimalisering zijn geschikte technieken bekend (zie bijvoorbeeld: §13, Davidon (1959), Fletcher and Reeves (1964), Powell (1964) en Box, Davies and Swann (1969) p.39). Deze technieken zijn meestal gebaseerd op kwadratische of kubische interpolatie. Dit vereist echter een aanzienlijke hoeveelheid rekenwerk, wat betreft benodigde evaluaties van de funktie en haar gradient. We willen daarom proberen ééndimensionale minimalisering zoveel mogelijk te vermijden. Een mogelijkheid hiertoe is:

2° kies $\alpha_k = 1$.

Een voorbeeld van het gebruik van deze strategie is de door Powell (1970) gegeven Variabele-metriekalgoritme.
Het gevaar van deze methode is echter, dat de afstand tot het minimum op de lijn $x_k + \alpha p_k$ ($\alpha>0$) sterk wordt overschat door de keuze $\alpha = 1$, zodat mogelijk niet aan (1.6) is voldaan. Met name als voor p_k de steepest-descentrichting (2.4) wordt gekozen, dan is de keuze $\alpha_k = 1$ niet zinvol.

Om konvergentie te kunnen waarborgen is het noodzakelijk dat de funktiewaarde in iedere stap voldoende afneemt. Volgens de stelling van Taylor is het increment van de funktiewaarde in de k-de stap, ΔF_k, voor kleine δ_k ongeveer gelijk aan $\delta_k^T g(x_k)$. Hierbij is:

$$\Delta F_k = F(x_{k+1}) - F(x_k) \quad \text{en} \quad x_{k+1} = x_k + \alpha_k p_k \,. \tag{3.2}$$

Als de afstand tot het minimum echter sterk is overschat, dan zal $\Delta F_k \,/\, \delta_k^T g(x_k)$ veel kleiner zijn dan 1.
Op grond van deze overwegingen komen we tot een derde strategie voor de bepaling van α_k (Goldstein and Price (1967), Fletcher (1970)).

3° Zij μ een gegeven konstante zodat $0 < \mu << \frac{1}{2}$.
We zullen μ in het vervolg de dalingsparameter noemen.
Als geldt

$$\frac{F(x_k+p_k) - F(x_k)}{p_k^T g(x_k)} \geq \mu \,, \tag{3.3}$$

kies dan $\alpha_k = 1$;
bepaal anders α_k zodanig dat geldt :

$$\mu \leq \frac{\Delta F_k}{\alpha_k p_k^T g(x_k)} \leq 1 - \mu \tag{3.4}$$

In de praktijk wordt wel $\mu = 0.0001$ gekozen (Fletcher (1970)).

Een voorbeeld van een Variabele-metriekalgoritme, waarin van deze methode gebruik wordt gemaakt, is gegeven door Fletcher (1970) (zie ook §7). Een algoritme gebaseerd op de gedempte methode van Newton met een dergelijke keuze voor de staplengte is gegeven door Goldstein en Price (1967). Zij tonen tevens het volgende aan voor deze strategie (zie stell.8.2) :
stel: a de funktie F voldoet aan zekere voorwaarden (zie stell.8.2);
b de rij $\{x_k\}$ konvergeert naar een minimum van F;
c $\lim_{k\to\infty} ||G_k - G(x_k)|| = 0$,
dan bestaat een natuurlijk getal N zodat $\alpha_k = 1$ voor $k > N$.

Met deze strategie wordt dus een voldoende snel dalende rij funktiewaarden $\{F(x_k)\}$ verkregen, terwijl toch per iteratiestap meestal slechts één evaluatie van de funktie en haar gradient is vereist.

Als niet aan (3.3) is voldaan, dan kunnen we een α_k, die aan (3.4) voldoet, zoeken door bijvoorbeeld achtereenvolgens voor α_k de waarden ω, ω^2, ω^3, ... $(0<\omega<1)$ te proberen (Goldstein and Price (1967)).
We zullen er echter in de hier behandelde algoritmen de voorkeur aan geven een α_k, die aan (3.4) voldoet te bepalen met één of enkele malen kubisch interpoleren langs de lijn $x_k + \alpha p_k$ $(\alpha>0)$. Indirekt wordt dus weer een zekere vorm van ééndimensionaal minimaliseren geïntroduceerd, maar dit gebeurt uitsluitend in uitzonderlijke gevallen.
Voor de bepaling van α_k met kubische interpolatie zie §13.
In de door ons behandelde algoritmen wordt steeds gebruik gemaakt van deze laatste strategie voor de bepaling van de staplengte.

§4. Het bijhouden van de metriek

Zij

$$\delta_k = x_{k+1} - x_k \quad , \quad \gamma_k = g(x_{k+1}) - g(x_k) \; . \tag{4.1}$$

Als F een kwadratische funktie is met inverse Hessiaan H, dan geldt :

$$\delta_k = H\gamma_k \; . \tag{4.2}$$

Voor driemaal kontinu differentieerbare funkties bestaat een kwadratische approximatie welke in een omgeving van het minimum de funktie in een bepaalde precisie benadert.
Het is dan ook nuttig een benadering van de Hessiaan bepaalde eigenschappen te geven, welke de werkelijke Hessiaan zou hebben gehad als de funktie kwadratisch was. We eisen daarom dat de korrektiematrices C_k (zie (1.7)) zo worden gekozen dat geldt, voor $H_{k+1} = H_k + C_k$:

$$\delta_k = H_{k+1}\gamma_k \; . \tag{4.3}$$

Deze eigenschap wordt wel de Variabele-metriekeigenschap genoemd.

De enige korrektiematrix van rang één, in de ruimte opgespannen door δ_k en $H_k\gamma_k$, is gegeven door (Broyden (1967), Davidon (1968), Fiacco and McCormick (1968) en Murtagh and Sargent (1969)) :

$$(4.4) \qquad H_{k+1} = H_k + \frac{(\delta_k - H_k\gamma_k)\,(\delta_k - H_k\gamma_k)^T}{\gamma_k^T(\delta_k - H_k\gamma_k)} \,.$$

Voor $G_{k+1} = H_{k+1}^{-1}$ geldt dan:

$$(4.4.a) \qquad G_{k+1} = G_k + \frac{(\gamma_k - G_k\delta_k)\,(\gamma_k - G_k\delta_k)^T}{\delta_k^T(\gamma_k - G_k\delta_k)} \,.$$

Voor een Variabele-metriekalgoritme, waarin de korrektiematrix is gedefinieerd door (4.4), kunnen we een aantrekkelijke eigenschap bewijzen. Daartoe geven we eerst twee definities.

Definitie 4.1

Zij H_0 een gegeven symmetrische matrix en zij de rij matrices $\{H_k\}$ gedefinieerd door $H_{k+1} = H_k + C_k$, met C_k symmetrisch (k=0,1,...). Dan zeggen we dat C_k de erfelijkheidseigenschap heeft, als voor een kwadratische funktie F geldt:

$$(4.5) \qquad \delta_t = H_{k+1}\gamma_t \quad , \quad k \geq 0,\ t = 0, 1, \ldots, k.$$

Definitie 4.2

Een algoritme voor minimaliseren van niet-lineaire funkties in meerdere variabelen heet kwadratisch eindig, als de algoritme het minimum van een kwadratische funktie in een eindig aantal stappen bepaalt.
(In de literatuur wordt hiervoor dikwijls de verwarrende term "quadratically convergent" gebruikt.)

We bewijzen nu.

Lemma 4.1

De door (4.4) gedefinieerde rang-één korrektiematrix heeft de erfelijkheidseigenschap.

Bewijs

Zie ook Fiacco and McCormick (1968), Murtagh and Sargent (1969) en Powell (1970).

We bewijzen het lemma door volledige induktie naar k.
Voor k = 0 volgt het gestelde uit (4.3).
Stel nu : $\delta_t = H_k\gamma_t$, t = 0, 1, ..., k - 1 (induktiehypothese).
Voor een kwadratische funktie met inverse Hessiaan H geldt dan voor t < k wegens (4.2) en de induktiehypothese:

$$(\delta_k - H_k\gamma_k)^T\gamma_t = \delta_k{}^T\gamma_t - \delta_t{}^T\gamma_k = \gamma_k{}^TH\gamma_t - \gamma_t{}^TH\gamma_k = 0 .$$

Het lemma volgt dan uit (4.4), (4.3) en de induktiehypothese.

Stelling 4.1

Een Variabele-metriekalgoritme met een door (4.4) gedefinieerde rang-één korrektiematrix en één van de drie in §3 beschreven strategieën voor de bepaling van de staplengte is kwadratisch eindig als $H_k\gamma_k \neq \delta_k$ $(k \leq n)$.

Bewijs

Er geldt: $\gamma_1, \ldots, \gamma_n$ zijn lineair onafhankelijk.
Stel namelijk dat dit niet zo is, dan bestaan $\zeta_1, \ldots, \zeta_{n-1}$ zodat $\gamma_n = \zeta_1\gamma_1 + \ldots + \zeta_{n-1}\gamma_{n-1}$.
We krijgen dan met behulp van lemma 4.1 :

$$\begin{aligned} H_n\gamma_n &= \zeta_1 H_n\gamma_1 + \ldots + \zeta_{n-1}H_n\gamma_{n-1} \\ &= \zeta_1\delta_1 + \ldots + \zeta_{n-1}\delta_{n-1} = \\ &= \zeta_1 H\gamma_1 + \ldots + \zeta_{n-1}H\gamma_{n-1} = H\gamma_n = \delta_n . \end{aligned}$$

Dit is in tegenspraak met het gestelde.
Omdat $\gamma_1, \ldots, \gamma_n$ lineair onafhankelijk zijn, kunnen we op analoge manier bewijzen dat voor willekeurige $x \in R_n$ geldt:

$$H_{n+1}x = Hx ,$$

zodat $H_{n+1} = H$.
Gezien de keuze van de staplengte (volgens §3) volgt hieruit direkt stelling 4.1.

De rang-één korrektieformule (4.4) heeft twee nadelen.

1° Het is mogelijk dat H_k niet positief definiet is voor zekere k, ook al kiezen we H_0 positief definiet. Het is dus niet zeker dat steeds aan ongelijkheid (2.1) is voldaan, zodat we één van de drie in §2 gegeven strategieën voor de bepaling van een staprichting moeten gebruiken. Dit nadeel is evenwel acceptabel, omdat we de werkelijke inverse Hessiaan $H(x_k)$ zo goed mogelijk willen benaderen door H_k (k=0,1,...) en door een slechte keuze van de initiële approximatie x_0 is het mogelijk dat $H(x_m)$ niet positief definiet is voor zekere m.

2° Het is mogelijk dat H_k singulier wordt voor zekere k.
Een gevolg hiervan kan zijn dat vanaf deze k de volgende stappen nog slechts een echte deelruimte van R_n opspannen. Als het gezochte minimum niet in deze deelruimte ligt zal dus nooit konvergentie naar dit minimum kunnen optreden.
Uit (4.4) volgt voor de determinant van H_{k+1} :

$$(4.6)\quad \det(H_{k+1}) = -\frac{(\gamma_k - G_k\delta_k)^T\delta_k}{(\delta_k - H_k\gamma_k)^T\gamma_k}\ \det(H_k)\ ,$$

zodat, als $\det(H_k) \neq 0$ geldt:

$$(4.7)\quad \det(H_{k+1}) = 0 \iff (\gamma_k - G_k\delta_k)^T\delta_k = 0\ .$$

In het bijzonder, als p_k wordt gegeven door (2.2) en $f(\alpha) = F(x_k+\alpha p_k)$ is minimaal voor $\alpha = 1$, dus $p_k^T g(x_{k+1}) = 0$, dan geldt voor de ongetwijfeld goede stap $\delta_k = p_k$ dat $(\gamma_k - G_k\delta_k)^T\delta_k = 0$, dus H_{k+1} is singulier.
In de praktijk zullen we dan ook alleen de rang-één korrektieformule gebruiken als voldaan is aan

$$(4.8)\quad |(\gamma_k - G_k\delta_k)^T\delta_k| > \beta\, ||\gamma_k - G_k\delta_k||\ \ ||\delta_k||\ ,$$

voor zekere konstante β met $0 < \beta < 1$.
We zullen β in het vervolg de orthogonaliteitsparameter noemen (zie ook Powell (1970), waar $\beta = 0.2$ wordt gekozen).

De rang-één korrektiematrix (4.4) is echter de enige korrektiematrix van rang ≤ 2, in de ruimte opgespannen door δ_k en $H_k\gamma_k$, die de erfelijkheidseigenschap heeft (Fiacco and McCormick (1968), Murtagh and Sargent (1969)). Een gevolg hiervan is, dat van Variabele-metriekalgoritmen, waarin een andere korrektiematrix wordt gebruikt en waarin de staplengte wordt bepaald met methode 2 of 3 uit §3, meestal niet kan worden bewezen dat ze kwadratisch eindig zijn. Als in een dergelijke algoritme de staplengte wordt bepaald met ééndimensionaal minimaliseren (§3, methode 1), dan kan voor de meeste van dergelijke algoritmen wel worden bewezen dat ze kwadratisch eindig zijn. Een voorbeeld hiervan is de algoritme van Fletcher en Powell (1963), waarin de gebruikte korrektieformule van rang twee is.
We willen echter deze relatief dure strategie voor de bepaling van de staplengte vermijden (zie §3). Dit heeft dus tot gevolg dat, als we een andere dan de door (4.4) gegeven korrektieformule willen gebruiken, we niet langer kunnen eisen dat de zo verkregen algoritme kwadratisch eindig is.

Fletcher (1970) introduceert een klasse van rang-twee korrektieformules welke wordt gegeven door :

$$H_{k+1}^{\phi} = H_k^{\phi} + (1-\phi(k)) \left(\frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{H_k^{\phi} \gamma_k \gamma_k^T H_k^{\phi}}{\gamma_k^T H_k^{\phi} \gamma} \right)$$

$$+ \phi(k) \left[\left(I - \frac{\delta_k \gamma_k^T}{\delta_k^T \gamma_k} \right) H_k^{\phi} \left(I - \frac{\gamma_k \delta_k^T}{\delta_k^T \gamma_k} \right) + \frac{\delta_k \delta_k^T}{\delta_k \gamma_k^T} \right], \tag{4.9}$$

waarbij $0 \leq \phi(k) \leq 1$, $k = 0, 1, 2, \ldots$.

Voor $G_{k+1}^{\phi} = (H_{k+1}^{\phi})^{-1}$ krijgen we :

$$G_{k+1}^{\phi} = G_k^{\phi} + (1-\phi(k)) \left[\left(I - \frac{\gamma_k \delta_k^T}{\delta_k^T \gamma_k} \right) G_k^{\phi} \left(I - \frac{\delta_k \gamma_k^T}{\delta_k^T \gamma_k} \right) + \frac{\gamma_k \gamma_k^T}{\delta_k^T \gamma_k} \right]$$

(4.9a)

$$+ \phi(k) \left(\frac{\gamma_k \gamma_k^T}{\delta_k^T \gamma_k} - \frac{G_k^{\phi} \delta_k \delta_k^T G_k^{\phi}}{\delta_k^T G_k^{\phi} \delta_k} \right) .$$

(4.9) met $\phi(k) = 0$ geeft de in 1959 door Davidon geïntroduceerde korrektieformule. We zullen deze in het vervolg Davidon's rang-twee korrektieformule noemen.
(4.9) met $\phi(k) = 1$ zullen we in het vervolg Fletcher's rang-twee korrektieformule noemen.
Kiezen we in (4.9)

$$\phi(k) = \psi_k = \frac{\delta_k^T \gamma_k}{(\delta_k^T \gamma_k - \gamma_k^T H_k^{\phi} \gamma_k)} \quad , \tag{4.10}$$

dan verkrijgen we de rang-één korrektieformule (4.4).
Er geldt echter als H_k^{ϕ} positief definiet is en $\delta_k^T \gamma_k > 0$ (dit laatste geldt steeds als F konvex is en δ_k wordt gedefinieerd door (2.2)),dat :

$$\psi_k < 0 \quad \text{of} \quad \psi_k > 1 .$$

De rang-één formule behoort dus niet tot de door (4.9) gedefinieerde klasse.

De door (4.9) gedefinieerde klasse van korrektieformules heeft de volgende eigenschap (Fletcher (1970)).

Eigenschap 4.1

Als $H_k^{\phi} = H_k^{\psi}$ en $\phi(k) < \psi(k)$

dan geldt:

$$\det(H_{k+1}^{\phi}) \leq \det(H_{k+1}^{\psi})$$

en

$$||H_{k+1}^{\phi}|| \leq ||H_{k+1}^{\psi}|| .$$

Eigenschap 4.1 geldt voor willekeurige $\phi(k)$, dus niet alleen voor $0 \leq \phi(k) \leq 1$, waartoe de door (4.9) gedefinieerde klasse van formules is beperkt.
Voor een kwadratische funktie met positief definiete inverse Hessiaan H geldt wegens (4.2) :

$$\delta_k^T \gamma_k - \gamma_k^T H_k^{\phi} \gamma_k = \gamma_k^T (H - H_k^{\phi}) \gamma_k \ . \tag{4.11}$$

Dus als $\delta_k^T \gamma_k > \gamma_k^T H_k^{\phi} \gamma_k$ dan is H_k^{ϕ} in zekere zin kleiner dan H.
Korrektie van H_k^{ϕ} met behulp van (4.9)met $\phi(k) = 1$ lijkt dus op grond van eigenschap 4.1 een geschikte keuze.
Is daarentegen $\delta_k^T \gamma_k < \gamma_k^t H_k^{\phi} \gamma_k$, dan zou $\phi(k) = 0$ een betere keuze zijn.

We kunnen bewijzen dat een Variabele-metriekalgoritme, waarin een korrektieformule uit de door (4.9) gedefinieerde klasse wordt gebruikt en waarin de staplengte wordt berekend met methode 3 uit §3, konvergeert voor kwadratische funkties.
We geven hiertoe eerst twee lemma's.

Lemma 4.2

Zij A een symmetrische matrix met eigenwaarden $\lambda_1 \geq \ldots \geq \lambda_n$ en bijbehorende eigenvektoren $v_1, \ldots, v_n$.
Stel voor zekere vektor u:

$$A^* = A + \sigma u u^T \ , \tag{4.12}$$

dan geldt voor de eigenwaarden $\lambda_1^* \geq \ldots \geq \lambda_n^*$ van A^* :

1° (zie Fletcher (1970))
als $\sigma = 1$ dan $\lambda_1^* \geq \lambda_1 \geq \ldots \geq \lambda_n^* \geq \lambda_n$;
als $\sigma = -1$ dan $\lambda_1 \geq \lambda_1^* \geq \ldots \geq \lambda_n \geq \lambda_n^*$;

2° als $v_k^T u = 0$ voor zekere $k (1 \leq k \leq n)$, dan is λ_k een eigenwaarde van A^* met als eigenvektor v_k;

3° zij λ_k een eigenwaarde van A, dan is λ_k een eigenwaarde van A^*, dan en slechts dan als $v_k^T u = 0$ of λ_k is een meervoudige eigenwaarde van A.

Bewijs

ad 1°. zie Wilkinson (1965), pp.94-98.

ad 2°. Dit volgt onmiddellijk door uitschrijven.

ad 3°. Stel $A = V\Lambda V^T$,

met $\Lambda = \text{diag}\,(\lambda_1,\ldots,\lambda_n)$ en V de matrix van bijbehorende eigenvektoren van A.

Dan volgt uit (4.12) :

$$A^* = V(\Lambda+\sigma ww^T)V^T \quad , \quad \text{met} \quad w = V^T u \ .$$

De eigenwaarden van A^* zijn dus gelijk aan die van $\Lambda + \sigma ww^T$.

Het gestelde volgt nu door uitschrijven van de karakteristieke vergelijking van $\Lambda + \sigma ww^T$ en uit 2°.

Lemma 4.3 (Fletcher (1970)).

Zij F een kwadratische funktie met positief definiete Hessiaan G.

Zij H_o^{ϕ} symmetrisch en positief definiet en H_k^{ϕ} gedefinieerd door (4.9), met dus $0 \le \phi(k) \le 1$.

Stel

$$K_k^{\phi} = G^{1/2} H_k^{\phi} G^{1/2} \quad , \quad k = 0, 1, 2, \ldots \ , \tag{4.13}$$

en $\lambda_k^1 \ge \lambda_k^2 \ge \ldots \ge \lambda_k^n$ zijn de eigenwaarden van K_k^{ϕ} $(k=0,1,2,\ldots)$.

Dan geldt:

$$|\lambda_{k+1}^i - 1| \le |\lambda_k^i - 1| \ , \ i = 1, 2, \ldots, n, \ k = 0, 1, 2, \ldots \ .$$

Bewijs

Zie Fletcher (1970).

Uit de in lemma 4.3 gestelde voorwaarden hoeft nog geen konvergentie te volgen blijkens het volgende voorbeeld.

Stellen we $z_k = G^{1/2}\delta_k$, dan geldt voor een kwadratische funktie:

$\gamma_k = G^{1/2} z_k$,

zodat voor $\phi(k) = 0$ geldt:

$$(4.14) \qquad K_{k+1}^{0} = K_k^{0} - \frac{K_k^{0} z_k z_k^T K_k^{0}}{z_k^T K_k^{0} z_k} + \frac{z_k z_k^T}{z_k^T z_k} \quad .$$

Zij u een eigenvektor van K_0^{0} en kies δ_k zodanig dat $z_k = u$ voor $k = 0, 1, 2, \ldots$, dan geldt:
u is een eigenvektor van K_k^{0} $(k=0,1,2,\ldots)$
en $K_{k+1}^{0} = K_k^{0}$ voor $k \geq 1$.
De in lemma 4.3 gestelde voorwaarden zijn dus niet voldoende om konvergentie van de eigenwaarden naar 1 te bewijzen. Zoals in stelling 4.1 is ook hier een extra eis nodig, opdat de successieve δ_k de gehele ruimte opspannen.

Stelling 4.2

Zij de rij matrices $\{K_k^{\phi}\}$ gedefinieerd als in lemma 4.3.
Stel er bestaat geen natuurlijk getal N en echte deelruimte $V \subset R_n$, zodat $\forall k > N$ geldt : $\delta_k \in V$.
Dan konvergeren de eigenwaarden van K_k^{ϕ}, geordend naar grootte, monotoon naar 1 voor $k \to \infty$.

Bewijs

We geven slechts de hoofdpunten van een bewijs van de stelling voor $\phi(k) = 0$. Voor $0 < \phi(k) \leq 1$ volgt het gestelde uit de geldigheid voor $\phi(k) = 0$, op een analoge wijze, als waarop Fletcher (1970) lemma 4.3 bewijst voor $0 < \phi(k) \leq 1$.
Stel

$$M_{k+1}^{0} = K_k^{0} - \frac{K_k^{0} z_k z_k^T K_k^{0}}{z_k^T K_k^{0} z_k} \quad ,$$

dan heeft M_{k+1}^{0} een eigenwaarde 0 met bijbehorende eigenvektor z_k.
Stel de eigenwaarden van M_{k+1}^{0} zijn $\mu_1, \mu_2, \ldots, \mu_{n-1}, 0$, dan zijn de eigenwaarden van K_{k+1}^{0}, wegens lemma 4.2 (3°), $\mu_1, \mu_2, \ldots, \mu_{n-1}, 1$.
Uit de voorwaarde volgt, wegens G positief definiet en
$z_k = G^{1/2} \delta_k$:

voor elke k bestaat een $p \geq k + n - 1$ zodat $z_k, z_{k+1}, \ldots, z_p$ de gehele R_n opspannen.
Een gevolg hiervan is :
voor elke eigenvektor v van K_k^0 bestaat een l, $k \leq l \leq p$,
zodat $v^T z_l \neq 0$.

Met behulp hiervan en met lemma 4.2 en 4.3 kunnen we dan achtereenvolgens bewijzen:

a als $\lambda \neq 1$ een r-voudige eigenwaarde is van K_k^0, dan is λ ten hoogste een (r-1)-voudige eigenwaarde van K_{k+p}^0 ;

b stel dat K_k^0 geen meervoudige eigenwaarden ongelijk 1 heeft; zijn $\lambda_k^1 \geq \ldots \geq \lambda_k^n$ de eigenwaarden van K_k^0 en $\lambda_{k+p}^1 \geq \ldots \geq \lambda_{k+p}^n$ de eigenwaarden van K_{k+p}^0, dan geldt:

$$\lambda_k^i = 1 \implies \lambda_{k+p}^i = 1$$

en

$$\lambda_k^i \neq 1 \implies |\lambda_k^i - 1| > |\lambda_{k+p}^i - 1| .$$

Dit laatste volgt eenvoudig uit lemma 4.2 en 4.3 en de gestelde voorwaarde.
Stel namelijk $\lambda_k^i \neq 1$ en v_i is de bijbehorende eigenvektor. Dan bestaat een l, $k \leq l \leq p$ zodat $v_i^T z_l \neq 0$.
Wegens lemma 4.2 (3°) geldt dan $\lambda_k^i \neq \lambda_{k+p}^i$, zodat uit lemma 4.3 volgt: $|\lambda_k^i - 1| > |\lambda_{k+p}^i - 1|$.

Stelling 4.2 (voor $\phi(k)=0$) volgt nu uit a en b .

Met behulp van stelling 4.2 en een stelling van Goldstein en Price (1967) (zie ook stelling 8.2) komen we dan tot de volgende stelling betreffende de konvergentie van Variabele-metriekalgoritmen met een korrektiematrix uit de door (4.9) gedefinieerde klasse.

Stelling 4.3

Een Variabele-metriekalgoritme, met een door (4.9) ($0 \leq \phi(k) \leq 1$) gedefinieerde rang-twee korrektiematrix en de derde methode uit §3 voor de bepaling van de staplengte, konvergeert voor kwadratische funkties als voldaan is aan de in stelling 4.2 gestelde voorwaarden.

Een dergelijke algoritme is evenwel niet kwadratisch eindig.

Hoofdstuk II. De Algoritmen

§5. Inleiding

We zullen nu, met behulp van de in hoofdstuk I beschreven methoden, vier Variabele-metriekalgoritmen formuleren.
Drie ervan zijn rang-één algoritmen, d.w.z. de metriek wordt in principe gekorrigeerd met een matrix van rang één (bijvoorbeeld (4.4)). De vierde algoritme is een rang-twee algoritme, waarin dus de metriek wordt gekorrigeerd met een matrix van rang twee. Deze laatste is, op enkele details na, de door Fletcher (1970) gegeven algoritme.
De staplengte wordt in deze vier algoritmen alleen dan met ééndimensionaal kubisch interpoleren bepaald, als dit noodzakelijk is om voldoende daling van de funktiewaarde te verkrijgen (§3, methode 3).

Voor al deze algoritmen worden verondersteld te zijn gegeven:

- een algoritme voor de bepaling van de funktiewaarde en de gradient;
- een initiële approximatie x_0 van de plaats van een minimum van F;
- ε_r en ε_a, de vereiste relatieve resp. absolute nauwkeurigheid van de plaats van het berekende minimum.
  Zij u zodanig dat F(u) minimaal is en zij x^* de berekende approximatie van u, dan is vereist dat geldt:

$$||u-x^*|| \leq ||x^*|| \times \varepsilon_r + \varepsilon_a \ ;$$

- ε_g, de vereiste maximale waarde van $||g(x^*)||$;
- c, de schalingsfaktor voor de initiële approximatie $H_0 = cI$ van de inverse Hessiaan in x_0. De waarde van c dient een ruwe benadering te zijn van $||H(x_0)||$. Er moet gelden: $c > 0$;
- de dalingsparameter μ. Hiervoor moet gelden: $0 < \mu << \frac{1}{2}$ (zie §3, methode 3) ;
- F_{min}, een ondergrens voor de funktiewaarde;
- de orthogonaliteitsparameter β. Hiervoor moet gelden: $0 < \beta < 1$ (zie 4.8)). Deze parameter is alleen van belang voor de rang-één algoritmen.

§6. De rang-één algoritmen

In deze paragraaf geven we drie rang-één algoritmen, resp. algoritme A, B en C. Deze algoritmen verschillen in essentie slechts van elkaar in de keuze van de staprichting in die stappen , waarin de metriek niet aan (2.1) voldoet. In algoritme A wordt in deze stappen de richting bepaald met de methode van Greenstadt, in algoritme B met de methode van Goldfeld e.a. en in algoritme C met de steepest-descentmethode (zie §2, resp. methode 2, 3 en 1).

In deze rang-één algoritmen wordt de metriek in principe gekorrigeerd met behulp van de rang-één korrektieformule (4.4). Alleen als niet is voldaan aan (4.8), dan zullen we, om singulariteit van de metriek te vermijden, een rang-twee formule gebruiken. We kiezen dan Davidon's rang-twee formule((4.9) met $\phi(k) = 0$) of Fletcher's rang-twee formule ((4.9) met $\phi(k) = 1$). De rang-één formule (4.4) wordt gegeven door (4.9) met (vgl.(4.10)) :

$$\phi(k) = \psi_k = \frac{\gamma_k{}^T \delta_k}{\gamma_k{}^T (\delta_k - H_k \gamma_k)} \quad .$$

Als niet aan (4.8) is voldaan kiezen we als $\psi_k \geq 1$ Fletcher's rang-twee formule en als $\psi_k < 0$ Davidon's rang-twee formule, omdat we wegens eigenschap 4.1 mogen verwachten dat, als we ons beperken tot de door (4.9) gedefinieerde klasse van korrektieformules, de zo verkregen metriek de best mogelijke benadering is van die welke we met behulp van de rang-één korrektieformule zouden verkrijgen. Als $0 < \psi_k < 1$, dan kiezen we Fletcher's formule om singulariteit van de metriek te vermijden.

In de algoritmen wordt steeds H_k bijgehouden, omdat in elke iteratiestap $H_k g(x_k)$ moet worden berekend.
Om te kunnen vaststellen of de rang-één korrektieformule kan worden gebruikt is echter in iedere stap de berekening nodig van $G_k \delta_k$ (vgl.(4.8)). Als in de k-de stap als staprichting de Variabele-metriekrichting (2.2) is gekozen, dan kunnen we $G_k \delta_k$ eenvoudig berekenen omdat geldt:

$$G_k \delta_k = - \alpha_k g(x_k) \quad .$$

Als echter in de k-de stap een andere staprichting wordt gekozen, dan is het afhankelijk van de hiervoor gebruikte methode of ook in deze stap $G_k\delta_k$ eenvoudig kan worden berekend. Bij de methode van Greenstadt of van Goldfeld e.a. worden de eigenwaarden en eigenvektoren van H_k berekend (zie §2), welke we kunnen gebruiken voor de berekening van $G_k\delta_k$. Bij gebruik van de steepest-descentmethode moeten we echter om $G_k\delta_k$ te berekenen, ofwel een linear stelsel oplossen ofwel in elke stap zowel H_k als G_k bijhouden. Het aantal benodigde operaties voor het oplossen van een lineair stelsel is van de orde n^3. Houden we in iedere stap zowel H_k als G_k bij, dan is het extra aantal benodigde operaties van de orde n^2 in elke stap. In algoritme C hebben we gekozen voor het bijhouden van zowel H_k als G_k. Weliswaar is het mogelijk dat door afrondingsfouten de werkelijke inverse van H_k voor grote k aanzienlijk gaat afwijken van de bijgehouden G_k, maar in de praktijk zal dit alleen van belang zijn als de orde van H_k erg groot, of de konditie van H_k erg slecht is.

In de algoritmen wordt gebruik gemaakt van methode 3 uit §3 voor de bepaling van de staplengte. Als c, de schalingsfaktor voor de initiële metriek, veel groter is gekozen dan $||H(x_0)||$, dan zal $||H_k g(x_k)||$ voor $k = 0, 1, 2, \ldots, n-1$ veel groter zijn dan de gewenste staplengte, d.i. de afstand tot het minimum op de lijn $x_k + \alpha p_k$. Een gevolg hiervan kan zijn dat α_k voor $k = 0, 1, 2, \ldots, n-1$ steeds met kubisch interpoleren moet worden berekend. Om dit te voorkomen vermenigvuldigen we p_k in de k-de iteratiestap ($k=0,1,2,\ldots,n-1$) eerst met een faktor θ_k alvorens we nagaan of aan (3.3) is voldaan. Hierbij kiezen we θ_0 zodanig dat geldt: $\theta_0 = \min(1, \alpha_{kwadr})$, waarbij α_{kwadr} een benadering is van de plaats van het minimum van $f(\alpha) = F(x_0+\alpha p_0)$ ($\alpha>0$). We berekenen α_{kwadr} met kwadratische interpolatie met behulp van F_{min}, de gegeven ondergrens voor de funktiewaarde, $F(x_0)$ en $g(x_0)$.
Voor $k = 1, 2, \ldots, n-1$ wordt vervolgens de waarde van θ_k zo gekozen dat geldt:

$$||\theta_k p_k|| = ||\delta_{k-1}|| \quad . \tag{6.1}$$

Een ander probleem bij de gekozen strategie voor de bepaling van de staplengte doet zich voor als c veel kleiner is gekozen dan $||H(x_0)||$

of als het karakter van de funktie sterk afwijkt van een kwadratisch karakter, waardoor de met behulp van kwadratische interpolatie verkregen θ_0 mogelijk veel te klein is. De staplengte $||\alpha_k p_k||$ zal dan voor k = 0, 1, ..., n - 1, door de keuze van θ_k, veel kleiner zijn dan de afstand tot het minimum op de lijn $x_k+\alpha p_k (\alpha>0)$. Een gevolg hiervan zal zijn dat $\delta_0, \ldots, \delta_{n-1}$ niet (voldoende) onafhankelijk zijn (vgl. stelling 4.2 en ook stelling 8.1), zodat de in de n-de stap verkregen metriek, mede omdat het effekt van afrondingsfouten door de zeer kleine stappen relatief groot is, sterk zal afwijken van de werkelijke inverse Hessiaan in x_n. Deze situatie kan gemakkelijk tot instabiliteit leiden.
Om dit probleem te vermijden berekenen we α_0 met tenminste één maal kubisch interpoleren. Hiertoe is het echter noodzakelijk dat geldt:

$$0 < \alpha^* < \theta_0 \quad , \tag{6.2}$$

als $f(\alpha) = F(x_0+\alpha p_0)$ $(\alpha>0)$ minimaal is voor $\alpha = \alpha^*$.
Als niet aan (6.2) is voldaan, dan wordt θ_0 verdubbeld totdat (6.2) wel geldt. (Vergelijk ook de in §13 gegeven algoritme voor de bepaling van een minimum van $f(\alpha)$.)

Stel nu dat de in §5 opgesomde gegevens bekend zijn.
Dan formuleren we de algoritmen A, B en C als volgt.

Algoritme A

Initialiseer : $H_0 = cI$.

De k-de iteratiestap (k=0,1,2,...) van deze algoritme luidt dan:

I : Als $g^T(x_k)\, H_k g(x_k) > 0$ dan $p_k = - H_k g(x_k)$,

anders berekenen we p_k met Greenstadt's methode (§2, methode 2).

II : Als k = 0 dan

bereken $\theta_0 = \min \left(1, \dfrac{2(F_{min}-F(x_0))}{- p_0^T g(x_0)} \right)$ en

test : als $p_0^T g(x_0+\theta_0 p_0) < 0$ en

$$\frac{F(x_0-\theta_0 p_0) - F(x_0)}{\theta_0 p_0^T g(x_0)} \geq \mu \qquad \text{(vgl. (3.3))}$$

dan $\theta_0 := 2\theta_0$; ga naar test ;
anders is het minimum op de lijn $x_0 + \alpha p_0$ ($\alpha>0$)
ingesloten en we bepalen nu een α_0, $0 < \alpha_0 < \theta_0$,
met tenminste éénmaal kubisch interpoleren, of bisectie
(als $p_0^T g(x_0+\theta_0 p_0)<0$), zodanig dat aan (3.4) is voldaan.
Ga naar III.

$$\theta_k = \begin{cases} ||\delta_{k-1}|| \;/\; ||p_k|| & \text{als } k < n \\ 1 & \text{als } k \geq n \; . \end{cases}$$

Als $$\frac{F(x_k-\theta_k p_k) - F(x_k)}{\theta_k p_k^T g(x_k)} \geq \mu$$

dan kiezen we $\alpha_k = \theta_k$,
anders is de afstand tot het minimum op de lijn $x_k + \alpha p_k$ ($\alpha>0$)
door de keuze $\alpha_k = \theta_k$ overschat en we bepalen α_k, $0 < \alpha_k < \theta_k$,
met kubisch interpoleren, of bisectie, zodanig dat aan (3.4)
is voldaan.

III : Stel $\delta_k = \alpha_k p_k$ en $x_{k+1} = x_k + \delta_k$.

IV : Bereken $G_k\delta_k$ als volgt:
als $g^T(x_k)\, H_k g(x_k) > 0$ dan $G_k\delta_k = -\alpha_k g(x_k)$,
anders met behulp van de berekende eigenwaarden en eigenvektoren van H_k.

V : Als $|(\gamma_k - G_k\delta_k)^T\delta_k| > \beta\, ||\gamma_k - G_k\delta_k||\, ||\delta_k||$,

dan berekenen we H_{k+1} met behulp van de rang-één korrektieformulie (4.4),

anders berekenen we H_{k+1} met een rang-twee korrektieformule als volgt:

als $$\frac{\gamma_k^T \delta_k}{\gamma_k^T (\delta_k - H_k \gamma_k)} > 0 \ ,$$

dan met Fletcher's rang-twee formule ((4.9) met $\phi(k)=1$) ,
anders met Davidon's rang-twee formule ((4.9) met $\phi(k)=0$).

Als voor zekere k geldt:

$$||H_{k+1} g(x_{k+1})|| \leq \varepsilon_r \, ||x_{k+1}|| + \varepsilon_a \wedge ||g(x_{k+1})|| \leq \varepsilon_g$$

$$\wedge \ g^T(x_{k+1}) \, H_{k+1} g(x_{k+1}) \geq 0 \quad \wedge \quad k \geq n \ ,$$

dan zijn we klaar.

Algoritme B

Deze algoritme verkrijgen we door in algoritme A deel I te vervangen door:

I : Als $g^T(x_k) \, H_k g(x_k) > 0$ dan $p_k = - H_k g(x_k)$,
anders:
kies Δ_k, de schatting voor een goede staplengte, als volgt:

$$\Delta_k = ||\delta_{k-1}|| \ ;$$

bereken de staprichting met behulp van de methode van Goldfeld e.a. (§2, methode 3).

Algoritme C

Deze algoritme verkrijgen we door in algoritme A de initialisatie en de delen I, IV en V te vervangen door resp.:

Initialiseer : $H_0 = cI$; $G_0 = \frac{1}{c}I$.

I : Als $g^T(x_k)\, H_k g(x_k) > 0$ dan $p_k = - H_k g(x_k)$,

anders:

$$p_k = - \frac{||\delta_{k-1}||}{||g(x_k)||}\, g(x_k) \qquad \text{(steepest-descentrichting).}$$

IV : Bereken $G_k\delta_k$ als volgt:

als $g^T(x_k)\, H_k g(x_k) > 0$ dan $G_k\delta_k = - \alpha_k g(x_k)$,
anders door vermenigvuldiging van de bijgehouden G_k met δ_k .

V : Als $|(\gamma_k - G_k\delta_k)^T\delta_k| > \beta\, ||\gamma_k - G_k\delta_k||\, ||\delta_k||$,

dan berekenen we H_{k+1} en G_{k+1} met behulp van de rang-één korrektieformule (4.4) resp. (4.4a) ,
anders berekenen we H_{k+1} en G_{k+1} met een rang-twee korrektieformule als volgt:

als $$\frac{\gamma_k{}^T\delta_k}{\gamma_k{}^T(\delta_k - H_k\gamma_k)} > 0 ,$$

dan met Fletcher's rang-twee formule ((4.9) resp. (4.9a) met $\phi(k) = 1$), anders met Davidon's rang-twee formule ((4.9) resp.(4.9a) met $\phi(k) = 0$).

§7. De rang-twee algoritme

Deze algoritme,welke we algoritme D zullen noemen, maakt voor de korrektie van de metriek gebruik van rang-twee korrektieformules uit de door (4.9) gedefinieerde klasse.
Als $\delta_k{}^T\gamma_k \geq \gamma_k{}^T H_k\gamma_k$, dan kiezen we Fletcher's rang-twee formule ((4.9) met $\phi(k)=1$), anders kiezen we Davidon's rang-twee formule ((4.9) met $\phi(k)=0$). Deze keuze wordt gemotiveerd door (4.11) en eigenschap 4.1.
Als $\delta_k{}^T\gamma_k < 0$, dan wordt de metriek niet gekorrigeerd, omdat anders niet meer kan worden gegarandeerd dat de metriek na korrektie positief

definiet is. Deze situatie kan zich alleen voordoen als er niet een gebied D bestaat waarvoor geldt:

$$x_{k-1},\ x_k \in D \quad \text{en} \quad F \text{ is konvex op } D.$$

In de praktijk kan dit bijvoorbeeld ook voorkomen door afrondingsfouten bij de evaluatie van de funktie en haar gradient, als ε_r, ε_a en ε_g (zie §5) te klein zijn gekozen.

Als staprichting nemen we in algoritme D uitsluitend de Variabele-metriekrichting (2.2), omdat we kunnen garanderen dat de metriek positief definiet is in iedere stap.
De bepaling van de staplengte geschiedt op dezelfde wijze als in algoritme A (§6).

Nemen we aan dat de in §5 opgesomde gegevens bekend zijn, behalve de orthogonaliteitsparameter β, dan formuleren we algoritme D als volgt:

Algoritme D

Initialiseer : $H_0 = cI$.

De k-de iteratiestap (k=0,1,2,...) van deze algoritme luidt:

I : $p_k = - H_k g(x_k)$.

II : Gelijk aan deel II in algoritme A (§6) .

III : Stel $\delta_k = \alpha_k p_k$ en $x_{k+1} = x_k + \delta_k$.

IV : Als $\delta_k^T \gamma_k < 0$ dan $H_{k+1} = H_k$,
anders korrigeren we de metriek als volgt:
als $\delta_k^T \gamma_k \geq \gamma_k^T H_k \gamma_k$,
dan met Fletcher's rang-twee korrektieformule ((4.9) met $\phi(k)=1$) ,
anders met Davidon's rang-twee korrektieformule ((4.9) met $\phi(k)=0$).

Als voor zekere k geldt:

$$||H_{k+1}g(x_{k+1})|| \leq \varepsilon_r ||x_{k+1}|| + \varepsilon_a \wedge ||g(x_{k+1})|| \leq \varepsilon_g \wedge k \geq n ,$$

dan zijn we klaar.

Algoritme D verschilt slechts van de door Fletcher (1970) gegeven rang-twee algoritme in deel II, met name in de bepaling van θ_k. In Fletcher's algoritme wordt θ_k als volgt berekend:

$$\theta_k = \begin{cases} \min\left(1, \dfrac{2(F_{min}-F(x_0))}{-p_0^T g(x_0)}\right) & \text{als } k = 0; \\ \min\left(1, \alpha_{k-1}, \dfrac{2(F_{min}-F(x_k))}{-p_k^T g(x_k)}\right) & \text{als } 1 \leq k < n; \\ 1 & \text{als } k \geq n . \end{cases}$$

Bij een dergelijke keuze geldt steeds $\theta_k \leq 1$.
Als c veel kleiner is gekozen dan $||H(x_0)||$ of als het karakter van de te minimaliseren funktie sterk afwijkt van een kwadratisch karakter, dan kan een strategie als in Fletcher's algoritme instabiliteit veroorzaken (vgl.§6). Dit bleek in de praktijk bij Box' funktie (§10, testfunktie 8), waarvan het karakter exponentieel is. Als we hier F_{min} gelijk kiezen aan de werkelijke minimale funktiewaarde, dan komt het soms voor, afhankelijk van de gekozen x_o, dat Fletcher's algoritme het minimum niet bereikt. Dit wordt veroorzaakt door te kleine stappen in de eerste n iteraties (vgl.§6).

Hoofdstuk III. De Konvergentie

§8. Konvergentie van de rang-één algoritmen

Een van de plezierigste eigenschappen van de rang-één korrektiematrix is de erfelijkheidseigenschap (definitie 4.1). Met behulp van deze eigenschap is het mogelijk te bewijzen dat de in §6 gegeven rang-één algoritmen onder zekere voorwaarden, kwadratisch eindig zijn (zie stelling 4.1)
Om voorwaarden voor konvergentie van de gegeven rang-één algoritmen voor niet-kwadratische funkties te kunnen geven, bewijzen we eerst een generalisatie van de erfelijkheidseigenschap voor de rang-één korrektiematrix (4.4).

Lemma 8.1
Zij $F : R_n \to R$ een funktie zodat $F \in C^3(S)$, voor een zekere gesloten konvexe verzameling $S \subset R_n$.
Zij $\{x_i\}$ een rij vektoren waarvoor geldt:

1° $\{x_i\} \in S$;

2° $\lim_{i\to\infty} ||x_{i-1} - x_i|| = 0$.

Zij verder G_0 gegeven en G_k $(k=1,2,\ldots)$ gedefinieerd door de rang-één korrektieformule (4.4a).

Zij

$$||\delta_k^T(\gamma_k - G_k\delta_k)|| > \beta\, ||\gamma_k - G_k\delta_k||\, ||\delta_k|| \tag{8.1}$$

voor $k = 0, 1, 2, \ldots$ en zekere β, $0 < \beta < 1$, (vgl.(4.8)).
Dan geldt:

$$\forall \varepsilon > 0 \quad \forall t > 0 \quad \exists N \geq 0 \text{ zodat } \forall k > N \text{ geldt:}$$

$$||G_{k+t}\delta_k - \gamma_k|| < \varepsilon ||\delta_k|| \tag{8.2}$$

Bewijs

We bewijzen het gestelde door volledige induktie naar t.
Voor t = 1 volgt het direkt uit (4.3).
Stel (8.2) geldt voor zekere t > 1, we bewijzen nu (8.2) voor t + 1.

Met behulp van de stelling van Taylor krijgen we:

$$(8.3) \qquad ||\gamma_k - G(x_k)\delta_k|| \leq \frac{1}{2} M ||\delta_k||^2 ,$$

waarbij $M = \max_{x \in S} ||F^{(3)}(x)||$.

Kies nu $\varepsilon > 0$ willekeurig.
Wegens de uniforme kontinuïteit van G op S geldt:

$$\exists\, \eta > 0 : ||x_{m_1} - x_{m_2}|| < \eta \Longrightarrow ||G(x_{m_1}) - G(x_{m_2})|| < \frac{\varepsilon\beta}{4} .$$

Kies natuurlijke getallen N_1, N_2 en N_3 zodanig dat:

$$m_1, m_2 > N_1 \Longrightarrow ||x_{m_1} - x_{m_2}|| < \eta \quad ;$$

$$k > N_2 \Longrightarrow ||\delta_k|| < \varepsilon\beta/4M \quad ;$$

$$k > N_3 \Longrightarrow ||G_{k+t}\delta_k - \gamma_k|| < \frac{\varepsilon\beta}{4} ||\delta_k|| .$$

Stel $N = \max(N_1, N_2, N_3)$.
Dan geldt voor $k > N$:

$$|(\gamma_{k+t} - G_{k+t}\delta_{k+t})^T \delta_k| = |(\gamma_{k+t} - G(x_{k+t})\delta_{k+t})^T \delta_k$$

$$- (G_{k+t}\delta_k - \gamma_k)^T \delta_{k+t} + (G(x_k)\delta_k - \gamma_k)^T \delta_{k+t} + \delta_k{}^T (G(x_{k+t}) - G(x_k))\, \delta_{k+t} |$$

$$\leq ||\gamma_{k+t} - G(x_{k+t})\, \delta_{k+t}||\, ||\delta_k|| + ||G_{k+t}\delta_k - \gamma_k||\, ||\delta_{k+t}||$$

$$+ ||G(x_k)\delta_k - \gamma_k||\, ||\delta_{k+t}|| + ||G(x_{k+t}) - G(x_k)||\, ||\delta_{k+t}||\, ||\delta_k|| .$$

Uit (8.3) en de keuze van N volgt dan:

$$(8.4)\quad |(\gamma_{k+t}-G_{k+t}\delta_{k+t})^T\delta_k| \leq$$

$$\tfrac{1}{2}M||\delta_{k+t}||^2||\delta_k||+\tfrac{\varepsilon\beta}{4}||\delta_k||\ ||\delta_{k+t}||+\tfrac{1}{2}M||\delta_k||^2||\delta_{k+t}||+\tfrac{\varepsilon\beta}{4}||\delta_{k+t}||\ ||\delta_k||$$

$$< \tfrac{3}{4}\varepsilon\beta||\delta_k||\ ||\delta_{k+t}|| .$$

Dan geldt wegens (4.4a) en (8.1):

$$||G_{k+t+1}\delta_k-\gamma_k||=||G_{k+t}\delta_k+\frac{(\gamma_{k+t}-G_{k+t}\delta_{k+t})(\gamma_{k+t}-G_{k+t}\delta_{k+t})^T\delta_k}{\delta_{k+t}^T(\gamma_{k+t}-G_{k+t}\delta_{k+t})}-\gamma_k||$$

$$\leq ||G_{k+t}\delta_k-\gamma_k||+\frac{|(\gamma_{k+t}-G_{k+t}\delta_{k+t})^T\delta_k|}{\beta||\delta_{k+t}||} .$$

Met (8.4) en de keuze van N_3 krijgen we tenslotte:

$$||G_{k+t+1}\delta_k-\gamma_k|| < \varepsilon\ ||\delta_k|| .$$

q.e.d.

Een gevolg van lemma 8.1 is

Lemma 8.2

Onder de in lemma 8.1 gestelde voorwaarden geldt:

$\forall\varepsilon > 0\ \ \forall t > 0\ \ \exists N \geq 0$, zodat $\forall k > N$ geldt:

$$||(G_{k+t}-G(x_{k+t}))\delta_k|| < \varepsilon\ ||\delta_k|| .$$

Bewijs

Neem $\varepsilon > 0$ willekeurig.
Met behulp van resp. lemma 8.1, de uniforme kontinuïteit van G en de konvergentie van de rij $\{x_k\}$, kunnen we achtereenvolgens natuurlijke getallen

N_1, N_2 en N_3 kiezen zodat:

$$k > N_1 \Longrightarrow \quad ||G_{k+t}\delta_k - \gamma_k|| < \frac{1}{3}\varepsilon\,||\delta_k|| \;;$$

$$m_1,\ m_2 > N_2 \Longrightarrow \quad ||G(x_{m_1}) - G(x_{m_2})|| < \frac{1}{3}\varepsilon \;;$$

$$k > N_3 \Longrightarrow \quad ||\delta_k|| < \frac{2\varepsilon}{3M} \;.$$

Dan vinden we voor $k > N = \max(N_1,N_2,N_3)$ met behulp van (8.3):

$$\begin{aligned} &||G_{k+t}\delta_k - G(x_{k+t})\delta_k|| \\ &\quad \le ||G_{k+t}\delta_k - \gamma_k|| + ||(G(x_k) - G(x_{k+t}))\delta_k|| + ||G(x_k)\delta_k - \gamma_k|| \\ &\quad \le ||G_{k+t}\delta_k - \gamma_k|| + ||G(x_k) - G(x_{k+t})||\ ||\delta_k|| + \tfrac{1}{2}M\,||\delta_k||^2 \\ &\quad \le \varepsilon||\delta_k|| \;. \end{aligned}$$

q.e.d.

Om tenslotte te kunnen bewijzen dat G_k konvergeert naar $G(x^*)$, met $x^* = \lim_{k\to\infty} x_k$, is het noodzakelijk een extra eis te stellen betreffende de onafhankelijkheid van de successieve δ_k voor grote k. We komen tot de volgende formulering.

Stelling 8.1

Zijn gegeven een funktie F, een rij vektoren $\{x_k\}$ en een rij matrices $\{G_k\}$. Stel dat er is voldaan aan de in lemma 8.1 gestelde voorwaarden. Stel verder dat voor de matrix $D_k = (\delta_{k-n},\ldots,\delta_{k-1})$ $(k>n)$ geldt: het konditiegetal $||D_k||\ ||D_k^{-1}||$ is begrens voor $k \to \infty$. Dan konvergeert $(G_k - G(x_k))$ voor $k \to \infty$ naar de nulmatrix.

Bewijs

Stel $E_k = G_k - G(x_k)$.
Kies $\varepsilon > 0$ willekeurig en $t = 1, \ldots, n$.

Volgens lemma 8.2 bestaat voor elke $t = 1, \ldots, n$ een natuurlijk getal N_t zodat $\forall k > N_t$ geldt:

$$||E_{k+t}\delta_k|| < \varepsilon \, ||\delta_k|| \; .$$

Zij $N = \max_t N_t$. Dan geldt $\forall k > N$:

$$||E_{k+t}\delta_k|| < \varepsilon \, ||\delta_k|| \qquad (t=1,\ldots,n) \; .$$

Zij $M = N + n + 1$ dan volgt hieruit:

$$||E_M\delta_{M-t}|| < \varepsilon \, ||\delta_{M-t}|| \qquad (t=1,\ldots,n) \; . \tag{8.5}$$

Voor de Euclidische norm van een vektor x en de spectrale norm van een matrix A gelden de volgende afschattingen (zie Wilkinson (1965) p.58):

$$||x||_1 \geq ||x|| \geq \frac{1}{n} ||x||_1$$

$$\frac{1}{n\sqrt{n}} ||A||_1 \leq ||A|| \leq \sqrt{n} \, ||A||_1$$

Uit (8.5) volgt dan:

$$||E_M D_M||_1 = \max_t ||E_M\delta_{M-t}||_1 \leq \varepsilon n \, ||D_M||_1 \; ,$$

zodat

$$||E_M|| \leq \sqrt{n} \, ||E_M D_M||_1 \, ||D_M^{-1}||_1 \leq \varepsilon \, (n\sqrt{n})^3 ||D_M|| \, ||D_M^{-1}|| .$$

Uit de gestelde voorwaarde volgt dan:

$$||E_k|| \to 0 \qquad \text{voor } k \to \infty \; ,$$

waarmee stelling 8.1 is bewezen.

<u>Stelling 8.2</u> (Goldstein and Price (1966)).
Zij gegeven een funktie $F : R_n \to R$ en een punt $x_0 \in R_n$.
Zij $S = \{x | F(x) \leq F(x_0)\}$ en $\{G_k\}$ een gegeven rij matrices.
Stel er is voldaan aan de volgende voorwaarden:

1° er bestaat een gesloten konvexe verzameling $\hat{S} \supset S$ zodat $F \in C^3(\hat{S})$;

2° er bestaat een $\omega > 0$ zodat geldt:

$$u^T G(x) u \geq \omega \, ||u||^2 \quad , \quad \text{voor alle } u \in R_n \text{ en } x \in S \ ;$$

3° $$\lim_{k \to \infty} ||G_k - G(x_k)|| = 0 \ .$$

Zij μ een gegeven konstante, $0 < \mu << \frac{1}{2}$, en zij de rij $\{x_k\}$ gegeven door een algoritme waarvan de k-de stap wordt gedefinieerd door:

a als $k = 0$ of G_k singulier of $g^T(x_k) G_k^{-1} g(x_k) \leq 0$
dan

$$p_k = - g(x_k) \qquad \text{(steepest-descentrichting)},$$

anders

$$p_k = - G_k^{-1} g(x_k) \ ;$$

b beschouw de funktie

$$h_k(\alpha) = \frac{F(x_k + \alpha p_k) - F(x_k)}{\alpha g^T(x_k) p_k} \ ;$$

als $h_k(1) \geq \mu$ dan kiezen we $\alpha_k = 1$, anders kiezen we α_k zodanig dat geldt

(8.6) $$\mu \leq h_k(\alpha_k) \leq 1 - \mu \qquad \text{(zie ook (3.4))} \ ;$$

c stel $x_{k+1} = x_k + \alpha_k p_k$.

Dan geldt:

1° de zo gedefinieerde rij $\{x_k\}$ konvergeert naar een minimum van F;

2° er bestaat een getal N zodat $\alpha_k = 1$ voor alle $k > N$;

3 de konvergentieorde van de algoritme is tenminste superlineair en ten hoogste kwadratisch.

Zie voor het bewijs van deze stelling Goldstein and Price(1966).

De manier waarop p_k wordt gekozen als $g^T(x_k)G_k^{-1}g(x_k) \leq 0$ is niet van belang, als maar steeds kan worden voldaan aan (8.6) als $h_k(1) < \mu$. Immers, voor grote k geldt wegens voorwaarden 2 en 3 dat $g^T(x_k)G_k^{-1}g(x_k) > 0$.
Met behulp van de stellingen 8.1 en 8.2 komen we dan tot de formulering van een stelling over de konvergentie van de in §6 gegeven rang-één algoritmen.

Stelling 8.3

Zij gegeven een funktie $F : R_n \to R$ en een punt $x_0 \in R_n$.
Zij $S = \{x \mid F(x) \leq F(x_0)\}$ en stel dat F voldoet aan:

- er bestaat een gesloten konvexe verzameling $\hat{S} \supset S$ zodat $F \in C^3(\hat{S})$;
- er bestaat een $\omega > 0$ zodat

$$u^T G(x) u \geq \omega ||u||^2 , \quad \text{voor alle } u \in R_n \text{ en } x \in S.$$

Stel verder dat voor de in §6 gegeven algoritmen geldt, bij gegeven β, $0 < \beta < 1$:

- $||\delta_k^T(\gamma_k - G_k\delta_k)|| > \beta||\gamma_k - G_k\delta_k||\,||\delta_k||$, $k = 1, 2, \ldots$;
- voor de matrix $D_k = (\delta_{k-n}, \ldots, \delta_{k-1})$ $(k>n)$ geldt:

  het konditiegetal $||D_k||\,||D_k^{-1}||$ is begrensd voor $k \to \infty$.

Dan geldt:

de in §6 gegeven algoritmen konvergeren naar een minimum van F, de konvergentie is superlineair en er bestaat een getal N zodat $\alpha_k = 1$ voor alle $k > N$.

Bewijs

Dit volgt onmiddellijk uit de stellingen 8.1 en 8.2.

§9. Opmerkingen betreffende de konvergentie van de rang-twee algoritme

In §4 vermeldden we reeds dat een rang-twee algoritme, waarin de metriek wordt bijgehouden met behulp van (4.9), en waarin de staplengte wordt bepaald met methode 3 uit §3, konvergeert voor kwadratische

funkties, onder de voorwaarde dat de successieve stappen de gehele ruimte opspannen (stelling 4.3). Dit is een gevolg van stelling 4.2 en de stelling van Goldstein en Price (stelling 8.2). De konvergentie is dus superlineair.
We zullen hier geen voorwaarden afleiden waaronder de gegeven rang-twee algoritme konvergeert voor niet-kwadratische funkties.
In de praktijk blijkt wel dat de rang-twee algoritme D goed konvergeert.

Hoofdstuk IV. Numerieke Resultaten en Konklusies

§10. Numerieke resultaten

De in hoofdstuk II gegeven algoritmen zijn geschreven als ALGOL-60 procedures (zie voor twee ervan hoofdstuk V) en getest op de Elektrologica X8 computer van het Mathematisch Centrum. De machineprecisie hiervan is ongeveer 12 decimalen. In deze paragraaf zullen we enige resultaten geven van gebruik van de procedures voor het bepalen van een minimum van een aantal uit de literatuur bekende testfunkties.
We geven eerst een korte beschrijving van deze testfunkties.

1 Rosenbrock's funktie (Rosenbrock (1960)).

$$F(x) = 100(x_2-x_1^2)^2 + (1-x_1)^2 .$$

Deze funktie heeft een diep dal langs de parabool $x_2 = x_1^2$.
Als startpunt nemen we (-1.2,1).
De minimale funktiewaarde is 0 voor $x = (1,1)^T$.
De Hessiaan in het minimum is singulier.

2 Leon's funktie (Leon (1966)).

$$F(x) = 100(x_2-x_1^3)^2 + (1-x_1)^2 .$$

Deze funktie lijkt op die van Rosenbrock, maar heeft een dal langs de kromme $x_2 = x_1^3$.
Als startpunt nemen we (-1.2,-1).
De minimale funktiewaarde van deze funktie is 0 voor $x = (1,1)^T$.

3 Beale's funktie (Beale (1958)).

$$F(x) = \sum_{k=1}^{3} (c_k - x_1(1-x_2^k)) ,$$

met $c_1 = 1.5$, $c_2 = 2.25$ en $c_3 = 2.625$.
Als startpunt nemen we (0.1,0.1) .
De minimale funktiewaarde is 0 voor $x = (3,0.5)^T$.

4 Een funktie met spiraalvormig dal (Fletcher and Powell (1963)).

$$F(x) = 100((x_3-10\theta)^2 + (r-1)^2) + x_3$$

met

$$r = (x_1^2+x_2^2)^{1/2}$$

en

$$2\pi\theta = \begin{cases} \arctan(x_2/x_1) & \text{als } x_1 > 0 \\ \pi + \arctan(x_2/x_1) & \text{als } x_1 < 0 . \end{cases}$$

Als startpunt nemen we $(-1,0,0)$.
De minimale funktiewaarde is 0 voor $x = (1,0,0)^T$.

5 Wood's funktie (Colville (1968)).

$$F(x) = 100(x_2-x_1^2)^2 + (1-x_1)^2 + 90(x_4-x_3)^2 + (1-x_3)^2$$
$$+ 10.1[(x_2-1)^2 + (x_4-1)^2] + 19.8(x_2-1)(x_4-1) .$$

Deze funktie heeft niet-optimale stationnaire punten.
Als startpunt nemen we $(-3,-1,-3,-1)$.
De minimale funktiewaarde van deze funktie is 0 voor $x = (1,1,1,1)^T$.
N.B. We noemen x een niet-optimaal stationnair punt van F als geldt:
$g(x) = 0$ en $G(x)$ heeft tenminste één negatieve en tenminste één positieve eigenwaarde.

6 Powell's funktie in vier variabelen (Powell (1962)).

$$F(x) = (x_1+10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 + 10(x_1-x_4)^4.$$

Bepaling van het minimum van deze funktie is moeilijk omdat de Hessiaan in het minimum slechts van rang twee is.
Als startpunt nemen we $(3,-1,0,1)$.
De minimale funktiewaarde is 0 voor $x = (0,0,0,0)^T$.

7 Powell's funktie in drie variabelen (Powell (1964)).

$$F(x) = 3 - \frac{1}{1+(x_1-x_2)^2} - \sin\left(\frac{\pi}{2} x_2 x_3\right) - \exp\left[-\left(\frac{x_1+x_3}{x_2} - 2\right)^2\right]$$

Als startpunt nemen we (0,1,2).
De minimale funktiewaarde is 0 voor $x_1 = x_2 = x_3 = \pm\sqrt{4n+1}$, (n≥0 en geheel).

8 Box' funktie (Box (1966)).

$$F(x) = \sum_{i=1}^{10} [\exp(-ix_1/10) - \exp(-ix_2/10) - x_3(\exp(-i/10) - \exp(-i))]^2$$

De minimale funktiewaarde van deze funktie is 0 in $(1,10,1)^T$ en op de lijn $(\lambda,\lambda,0)^T$.
We hebben bij deze funktie tien verschillende startpunten gekozen:

(0,20,1), (2.5,10,10), (0,0,10), (0,10,1), (0,10,20),

(0,10,10), (0,20,0), (0,20,10), (0,20,20), (2.5,25,25).

Naast deze moeilijk te minimaliseren funkties in relatief weinig variabelen hebben we ook de procedures getest met mogelijk eenvoudiger problemen in relatief veel variabelen.

9 Trigonometrische funkties (Fletcher and Powell (1963)).

$$F(x) = \sum_{i=1}^{n} \left(E_i - \sum_{j=1}^{n} (A_{ij} \sin(x_j) + B_{ij} \cos(x_j)) \right)^2 .$$

A_{ij} en B_{ij} (i,j=1,...,n) zijn random gehele getallen tussen -100 en +100.
Stel x_j^*(j=1,...,n) zijn random getallen tussen $-\pi$ en π.
Dan berekenen we E_i (i=1,...,n) zodat $F(x^*) = 0$,
met $x^* = (x_1^*,\ldots,x_n^*)^T$.

De minimale funktiewaarde van F is dus 0 voor $x = x^*$.
Als startpunt kiezen we $x^* + 0.1\delta$, waarbij ook de komponenten van δ random getallen zijn tussen $-\pi$ en π.
Deze funktie is getest voor n = 2, 3, 4, 5, 6, 8, 10, 20, 30 (zie tabel 10.4).

Tenslotte hebben we, om het kwadratisch eindig zijn van de rang-één algoritmen te testen, met behulp van de procedures ook het minimum bepaald van enige kwadratische funkties.

10 Een kwadratische funktie in 2 variabelen.

$$F(x) = x_1^2 - 2x_1x_2 + 2x_2^2 .$$

11 Een kwadratische funktie met tridiagonale Hessiaan (Brent (1971)).

$$F(x) = x^TAx - 2x_1 ,$$

$$A = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & 0 \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & -1 \\ 0 & & & -1 & 2 \end{pmatrix} .$$

De minimale funktiewaarde is -n voor $x = (n,n-1,\ldots,1)^T$.
Deze funktie is getest voor n = 2, 3, 4, 5, 6, 8, 10, 20, 30 (zie tabel 10.6). Verschillende eigenwaarden van de inverse van deze matrix zijn zeer slecht gekonditioneerd (Gregory and Karney (1969), Wilkinson (1965) p.92).

12 Een kwadratische funktie waarvan de Hessiaan een (n×n)-segment van de Hilbertmatrix is (Brent (1971)).

$$F(x) = x^TA_nx ,$$

met $(A_n)_{ij} = 1 / (i+j-1) \quad (i,j=1,\ldots,n)$.
Het konditiegetal $\kappa(A_n) = ||A_n||\ ||A_n^{-1}||$ van deze matrices stijgt zeer snel met toenemende orde n.
Deze funktie is getest voor n = 2, 3, 4, 5, 6, 8 , 10 (zie tabel 10.6).

Voor de bijbehorende konditiegetallen geldt:

$\kappa(A_2) = 14$, $\kappa(A_3) = 3_{10}2$, $\kappa(A_4) = {}_{10}4$, $\kappa(A_5) = 2_{10}5$, $\kappa(A_6) = {}_{10}7$,

$\kappa(A_8) = {}_{10}9$, $\kappa(A_{10}) = {}_{10}14$.

De in de tabellen weergegeven resultaten van de verschillende minimaliseringsprocedures zijn verkregen met de volgende waarden voor de verschillende parameters (zie §5).

- voor ε_r en ε_a, de geëiste relatieve resp. absolute precisie waarin een minimum moet worden berekend: $\varepsilon_r = \varepsilon_a = {}_{10}-5$;
- voor ε_g, de geëiste maximale waarde van $||g(x^*)||$, waarbij x^* het berekende minimum is: $\varepsilon_g = {}_{10}-5$;
- voor c, de schalingsfaktor voor de initiële metriek: c = 1 ;
- voor F_{min}, de ondergrens voor de funktiewaarde:
  voor de testfunkties 1 t/m 7, 10, 12 : $F_{min} = \min(-1, -0.01\, F(x_0))$,
  voor de testfunkties 8 en 9, welke sommen van kwadraten zijn:
  $F_{min} = 0$ en voor testfunktie 12: $F_{min} = -2n$;
- voor β, de orthogonaliteitsparameter: $\beta = 0.01$;
  deze parameter is alleen nodig voor de rang-één algoritmen A, B en C.

In de tabellen wordt met i steeds het benodigde aantal iteraties aangeduid en met e het benodigde aantal evaluaties van de funktie en haar gradient.
Naast de resultaten van de algoritmen A, B, C en D geven we ook de resultaten van de originele Variabele-metriekalgoritme van Davidon (1959), in de gemodificeerde versie van Fletcher en Powell (1963), zoals deze als ALGOL-60 procedure is gegeven door Wells (1965).
In deze procedures zijn enige fouten verbeterd en enige details gewijzigd (zie Fletcher (1966), Hamilton and Boothroyd (1969), House (1971)).
In de tabellen 10.1, 10.2 en 10.4 worden de waarden van i en e gegeven, bij gebruik van de verschillende algoritmen voor de bepaling van het minimum van resp. de eerste zeven testfunkties, Box' funktie (testfunktie 8) uitgaande van tien verschillende startpunten en de trigonometrische funktie (testfunktie 9) voor de aangegeven orden.

tabel 10.1

testfunkties 1 t/m 7

Algoritme / funktie	A		B		C		D		Flepomin	
	i	e	i	e	i	e	i	e	i	e
Rosenbrock's funktie	42	57	42	52	42	53	40	46	21	76
Leon's funktie	51	72	48	63	56	77	47	65	39	135
Beale's funktie	12	16	12	16	12	16	12	16	8	22
funktie met spiraalvormig dal	32	39	30	41	35	46	28	32	23	56
Wood's funktie	59	85	90	130	109	148	87	99	34	101
Powell's funktie in 4 variabelen	50	56[a]	53	58[a]	53	59[a]	70	78[a]	25	74[a]
Powell's funktie in 3 variabelen	12	17	12	16	12	15	12	14	9	22

a) Volledige precisie niet bereikt (zie tekst).

Uit deze resultaten blijkt dat algoritme C twee keer het minimum niet bereikt (binnen de gestelde limiet van 200 funktie- en gradientevaluaties). Ook is het aantal benodigde funktie- en gradientevaluaties bij Wood's funktie en bij Box' funktie (startpunten (0,20,0) en (0,20,20)) aanzienlijk groter dan voor de algoritmen A of D. (In een relatief zeer groot aantal van de iteraties wordt hier de steepest-descentrichting gekozen.) Dit gedrag treedt op bij funkties die niet-optimale (bijna-) stationnaire punten hebben, zoals Wood's funktie en Box' funktie.

Als x_k, voor zekere k, in een omgeving ligt van een niet-optimaal (bijna-) stationnair punt u, waarbij $F(x_k) > F(u)$, en er is niet voldaan aan $g^T(x_k)\, H_k g(x_k) > 0$ (2.1), dan wordt in algoritme C de steepest-descentrichting (2.4) gekozen in de k-de stap. Omdat de afgeleide in die richting (bijna) gelijk 0 is, wordt de staplengte zeer klein. De situatie veranderd dan zo weinig dat ook in de volgende stap niet aan (2.1) is voldaan. Dit blijft vele stappen achtereen het geval. Als x_k in een omgeving van u ligt is het waarschijnlijk dat voor zekere k niet aan (2.1) is voldaan, omdat H(u) niet positief definiet is.

Het blijkt dus dat de algoritme C naar de steepest-descentmethode neigt, juist in die gevallen waar deze methode uiterst langzaam konvergeert (vgl. §1 en §2 opmerking 2).

In §6 werd als mogelijk bezwaar van algoritme C genoemd, dat de bijgehouden approximatie van de inverse metriek mogelijk op den duur te veel gaat afwijken van de werkelijke inverse van de metriek. In de praktijk werd dit echter niet als een bezwaar ondervonden.

Algoritme B vertoont een analoog gedrag als algoritme C bij de bepaling van het minimum van Wood's funktie en Box' funktie. Dus ook de methode van Goldfeld e.a. (§2, methode 3) blijkt, voor de bepaling van de staprichting als niet aan (2.1) is voldaan, in de praktijk niet altijd goed te voldoen.

Alleen algoritme A vertoonde in geen enkele van de tests een gedrag zoals beschreven voor algoritme C.

De methode van Greenstadt (zie §2) lijkt dan ook de beste van de drie in §2 gegeven methoden voor de bepaling van de staprichting als niet aan (2.1) is voldaan.

tabel 10.2

Box' funktie

startpunt \ Algoritme	A		B		C		D		Flepomin	
	i	e	i	e	i	e	i	e	i	e
0, 20, 1	17	25	-	-	17	26	45	50	17	63
2.5, 10, 10	11	17	11	17	11	17	17	23	20	61
0, 0, 10	13	16	13	16	13	16	13	16	33	126
0, 10, 1	13	19	12	17	13	19	20	24	6	15
0, 10, 20	23	29	23	29	23	29	31	36	18	70
0, 10, 10	20	24	22	26	21	25	27	30	14	67
0, 20, 0	19	32	-	-	47	59	23	35	-	-
0, 20, 10	20	28	48	64	-	-	47	52	24	88
0, 20, 20	24	32	27	38	73	81	41	49	19	92
2.5, 25, 25	26	37	30	46	25	40	46	63	35	124

De rang-één algoritmen bereiken bij de bepaling van het minimum van Powell's funktie in 4 variabelen een precisie van $3_{10}-5$, terwijl $_{10}-5$ werd geëist. De rang-twee algoritme D bereikt een precisie van $5_{10}-5$ en Flepomin bereikt slechts $2_{10}-4$. De oorzaak hiervan is niet gelegen in de manier waarop de stap wordt berekend, maar in het feit dat een zo eenvoudig stopkriterium is gekozen. Powell's funktie in 4 variabelen is een zeer slecht gekonditioneerd probleem. De Hessiaan van deze funktie is in het minimum slechts van rang twee, zodat in twee richtingen de funktiewaarde op een afstand, groter dan de tolerantie, van het minimum, mede door afrondingsfouten,niet of nauwelijks afwijkt van de minimale funktiewaarde. Om in dergelijke extreme situaties te kunnen garanderen dat de gevraagde precisie wordt bereikt, is een zó veilig stopkriterium nodig, dat in de gevallen waarin geen problemen optreden veel onnodige evaluaties van de funktie en haar gradient zouden worden gedaan (zie Powell (1964), Brent (1971)). We hebben daarom voor een eenvoudig stopkriterium gekozen (zie §6 en 7), temeer omdat slecht gekonditioneerde problemen kunnen worden herkend aan de uiteindelijk verkregen metriek, welke in norm relatief groot zal zijn. In tabel 10.3 geven we voor de algoritmen A en D, bij twee verschillende waarden voor de vereiste absolute en relatieve precisie (ε_a resp. ε_r, zie §5), de bereikte precisie, het benodigde aantal evaluaties van de funktie en haar gradient (e) en de orde van grootte van $||H^*||$, als H^* de uiteindelijk verkregen metriek is. Deze getallen zijn verkregen bij de bepaling van het minimum van Powell's funktie in 4 variabelen.

tabel 10.3

Powell's funktie in 4 variabelen

Algoritme	$\varepsilon_r = \varepsilon_a$	bereikte precisie	e	orde($\|\|H^*\|\|$)
A	10^{-5}	$3_{10}-5$	56	10^{8}
	10^{-7}	$3_{10}-6$	67	10^{10}
D	10^{-5}	$5_{10}-5$	78	10^{7}
	10^{-7}	$3_{10}-7$	107	10^{12}

tabel 10.4

Trigonometrische funkties

Algoritme	A		B		C		D		Flepomin	
orde	i	e	i	e	i	e	i	e	i	e
2	6	10	6	10	6	10	7	11	6	26
3	9	12	9	12	9	12	9	12	11	36[b]
4	12	16	12	16	12	16	13	17	9	38
5	13	18	13	18	13	18	14	18	9	30
6	18	23	17	21	19	26	19	23	48	123
8	20	26	20	26	20	26	21	25	16	46
10	26	35	29	35	29	35	51	60[c]	19	54
20	42	48	42	51	44	52	43	48	32	87
30	85	103[b]	74	91	--	--	127	140	43	109

b) De algoritme vindt een ander minimum dan het bedoelde.

c) Gedurende 12 opeenvolgende stappen wordt de metriek niet gekorrigeerd omdat $\delta_k^T \gamma_k < 0$ (zie §7) .

Als we de algoritmen A en D, welke de staplengte bepalen met methode 3 uit §3, vergelijken met Flepomin, waarin de staplengte met ééndimensionaal minimaliseren (§3, methode 1) wordt bepaald, dan blijkt dat de eerste meestal aanzienlijk efficiënter zijn, wat betreft het benodigde aantal evaluaties van de funktie en haar gradient.
Bij het testen bleek dat de gekozen strategie voor de bepaling van de staplengten in de eerste n iteratiestappen (vgl. §6 en 7) goed voldoet. Niet alleen werd nu door de algoritmen A en D in alle gevallen een minimum bereikt, wat bij de door Fletcher (1970) gekozen strategie voor met name Box' funktie, als $F_{min} = 0$ wordt gekozen, niet geldt, maar ook waren de resultaten in het algemeen iets beter met de door ons gekozen strategie.

Tenslotte moeten we, met betrekking tot de in de tabellen 10.1, 10.2 en 10.4 gegeven resultaten, enige opmerkingen maken wat betreft de waarde van de orthogonaliteitsparameter $\beta(0<\beta<1)$. Als β groot wordt gekozen, dan zal het gedrag van de rang-één algoritmen sterk overeenkomen met dat van algoritme D, d.w.z. in relatief veel iteratiestappen wordt een rang-twee korrektieformule gebruikt en in relatief zeer weinig iteratiestappen zal een andere staprichting dan de Variabele-metriekrichting (2.2) worden gekozen. Kiezen we daarentegen β klein, dan zal in weinig stappen een rang-twee korrektieformule worden gebruikt, met het gevaar dat de metriek singulier wordt als β zeer klein wordt gekozen, en in wat meer iteratiestappen zal een andere staprichting dan de Variabele-metriekrichting worden gekozen.
In tabel 10.5 geven we voor $\beta = 0.1$ en $\beta = 0.01$ het aantal iteratiestappen waarin een rang-twee korrektieformule wordt gebruikt en het aantal iteratiestappen waarin niet de Variabele-metriekrichting wordt genomen. Deze aantallen zijn uitgedrukt in percentages van het totaal aantal iteratiestappen .
Deze percentages zijn het gemiddelde van de percentages voor alle in de tabellen 10.1, 10.2 en 10.4 gegeven testproblemen.

Uit deze tabel blijkt dat in de praktijk de keuze $\beta = 0.01$ geschikt is. Bij deze waarde is het gedrag van algoritme A duidelijk dat van een rang-één algoritme, terwijl toch relatief weinig een andere dan de Variabele-metriekrichting wordt gekozen.

tabel 10.5.

Algemeen gedrag van de rang-één algoritme A, bij verschillende keuze van β.

β	rang-twee korrektie (%)	niet-Vm-richting (%)
0.1	46	4.3
0.01	5	9.2

Tabel 10.6 geeft de resultaten van de verschillende algoritmen voor de bepaling van het minimum van kwadratische funkties (testfunkties 10, 11 en 12). Zij n de orde van het probleem. Uit de resultaten voor testfunktie 11, de kwadratische funktie met tridragonale Hessiaan, blijkt dat Flepomin en de rang-één algoritmen, als hierbij in tenminste n stappen de rang-één korrektieformule wordt gebruikt, kwadratisch eindig zijn (definitie 4.2), in tegenstelling tot de rang-twee algoritme D. De rang-één algoritmen bereiken in de praktijk steeds het minimum in de eerstvolgende stap, nadat de metriek n maal is gekorrigeerd met behulp van de rang-één korrektieformule.
Geen van de beschouwde algoritmen bereikt de vereiste precisie bij de kwadratische funkties met een segment van de Hilbertmatrix als Hessiaan voor $n \geq 6$. Dit is te wijten aan de keuze van het stopkriterium en de zeer slechte konditie van deze matrices (vgl. verklaring bij Powell's funktie in 4 variabelen).

Om een indruk te geven van het verschil tussen Variabele-metriekalgoritmen enerzijds en een gedempte-Newton-algoritme (§1) anderzijds, geven we in tabel 10.7 de resultaten van een gedempte-Newton-algoritme voor enkele testfunkties. In deze algoritme wordt de staplengte op dezelfde wijze bepaald als in de behandelde Variabele-metriekalgoritmen (§3, methode 3).

tabel 10.6

Kwadratische funkties

Algoritme	A		B		C		D		Flepomin	
orde	i	e	i	e	i	e	i	e	i	e
testfunktie 10										
2	3	5	3	5	3	5	3	5	3	6
kwadratische funkties met tridiagonale Hessiaan										
2	3	5	3	5	3	5	3	5	3	5
3	4	7	4	6	4	6	6	8	4	8
4	5	8	5	7	5	8	10	12	5	10
5	6	9	6	8	6	9	12	14	6	14
6	7	10	7	9	7	10	14	16	7	16
8	10	13	10	13	10	13	19	21	9	20
10	12	15	12	14	12	15	23	25	11	26
20	26	29	26	29	24	27	42	44	21	46
30	43	46	46	51	40	45	66	68	31	65
kwadratische funkties met een segment van de Hilbertmatrix als Hessiaan										
2	3	6	3	6	3	6	3	6	3	9
3	4	8	4	8	4	8	4	8	4	13
4	5	10	5	10	5	10	5	10	5	17
5	7	13	7	13	7	13	8	14	5	17[a]
6	7	13[a]	7	13[a]	7	13[a]	7	13[a]	7	23[a]
8	12	21[a]	11	20[a]	11	20[a]	9	17[a]	9	29[a]
10	11	18[a]	11	18[a]	11	18[a]	11	18[a]	12	36[a]

a) Volledige precisie niet bereikt. Vergelijk verklaring bij Powell's funktie in 4 variabelen (tabel 10.1 en 10.3).

tabel 10.7

gedempte-Newton

funktie	i	e
kwadratische funktie (testfunktie 10)	2	3
Rosenbrock's funktie	20	26
Powell's funktie in 4 variabelen	29	30[a]
Leon's funktie	70	121
Wood's funktie	61	65

a) Volledige precisie niet bereikt (vgl.tabel 10.1).

Stellen we dat voor de berekening van de Hessiaan $\frac{1}{2}n(n-1)$ funktie-evaluaties nodig zijn en voor de berekening van de gradient n.(Dit zijn vrij algemeen aanvaarde normen.) Dan moeten we, om de resultaten uit tabel 10.7 met die uit de tabellen 10.1 en 10.6 te kunnen vergelijken, het aantal evaluaties in tabel 10.7 met $\frac{1}{2}(n-1)$ vermenigvuldigen. Er blijkt dan duidelijk dat de Variabele-metriekmethode voor de bepaling van een minimum van een niet-lineaire funktie in meerdere variabelen veelal veruit te verkiezen zal zijn boven een gedempte-Newton-methode.

§11. Konklusies

In §10 zagen we reeds dat, van de in hoofdstuk II beschreven rang-één algoritmen, algoritme A het best voldoet.
We geven daarom naast de rang-twee algoritme D, alleen de rang-één algoritme A als ALGOL-60 procedure (hoofdstuk V). Uit de in §10 gegeven numerieke resultaten kunnen we konkluderen dat de algoritmen A en D voor niet-kwadratische funkties ongeveer even efficiënt kunnen worden

genoemd, wat betreft het benodigde aantal evaluaties van de funktie en haar gradient.
In tegenstelling tot wat door Fletcher (1970) wordt gesuggereerd blijkt dus dat ook met de rang-één korrektieformule (4.4) een efficiënte Variabele-metriekalgoritme kan worden gegeven.
We moeten hierbij echter wel beseffen, dat bij gebruik van algoritme A mogelijk in enige stappen de berekening van de eigenwaarden en -vektoren van de metriek is vereist. Het aantal aritmetische operaties dat hiervoor nodig is, is van de orde n^3. Afgezien hiervan is het aantal operaties dat per iteratiestap is vereist slechts van de orde n^2. In de praktijk blijkt echter dat het aantal stappen waarin de eigenwaarden en -vektoren van de metriek moeten worden berekend, d.i. het aantal keren dat niet de Variabele-metriekrichting als staprichting wordt gekozen, steeds relatief klein is t.o.v. het aantal iteratiestappen. Voor niet te moeilijk te minimaliseren funkties, zoals bijvoorbeeld de trigonometrische funkties (testfunktie 9), is dit aantal zelfs meestal 0. Gemiddeld over alle niet-kwadratische testproblemen is het aantal stappen waarin de eigenwaarden en -vektoren van de metriek moeten worden berekend slechts 9.2% van het totaal aantal iteratiestappen (vgl. tabel 10.5).
Bovendien is het pas werkelijk bezwaarlijk, dat in enige iteratiestappen een berekening van orde n^3 operaties is vereist, als het aantal operaties, dat nodig is voor een evaluatie van de funktie en haar gradient, van lagere orde is dan n^3.

Samenvattend kunnen we tenslotte het volgende stellen.
Voor minimalisering van niet-lineaire tweemaal kontinu differentieerbare funkties in meerdere variabelen, waarvan de gradient expliciet is gegeven, worden hier twee Variabele-metriekalgoritmen gegeven. Een hiervan is een rang-één algoritme (§6, algoritme A, voor ALGOL-60 procedure zie §15), de andere is een rang-twee algoritme (§7, algoritme D, voor ALGOL-60 procedure zie §16).
Deze beide algoritmen zijn meestal aanzienlijk efficiënter dan de door Fletcher en Powell (1963) gegeven algoritme.
Voor problemen in een groot aantal variabelen, waar een evaluatie van

de funktie en haar gradient niet duur is (aantal vereiste operaties maximaal van orde n^2) moeten we het gebruik van algoritme D aanbevelen. Voor problemen in relatief weinig variabelen, of waar een evaluatie van de funktie en haar gradient duur is, is in het algemeen geen duidelijke voorkeur aan te geven voor algoritme A of D.
Algoritme A is onder zekere voorwaarden kwadratisch eindig (stelling 4.1), terwijl tevens konvergentie werd bewezen van deze algoritme voor een klasse van niet-kwadratische funkties (stelling 8.3).
Van algoritme D is slechts konvergentie bewezen voor kwadratische funkties (stelling 4.3).

Hoofdstuk V. De Procedures

§12. Twee hulpprocedures

symmatvec

Zij A de symmetrische matrix van zekere orde n ≥ max(i,u), waarvan de bovendriehoek is gegeven in array a [1 : n × (n+1) : 2], zodanig dat a[(j-1) × j : 2 + i] het (i,j)-de element is van A (1≤i≤j≤n).
Zij b een vektor gegeven in array b[1:u], met l ≥ 1.
Dan wordt symmatvec voor i ≥ 1 gedefinieerd door:

$$\text{symmatvec} := \sum_{j=1}^{u} A_{ij} b_j \ .$$

symmatvec maakt gebruik van de procedures vecvec en seqvec (Dekker (1968)).

inimat

Na een aanroep van inimat geldt voor de symmetrische matrix H, van orde n, waarvan de bovendriehoek wordt gegeven door array h[1 : n × (n+1) : 2], zodanig dat h[(j-1) × j : 2 + i] het (i,j)-de element is van H(1≤i≤j≤n), dat:

$$H_{ij} = 0 \quad \text{als } i \neq j$$

en (1≤i,j≤n)

$$H_{ij} = d \quad \text{als } i = j \ .$$

```
real procedure symmatvec(l, u, i, a, b); value l, u, i; integer l, u, i;
array a, b;
begin integer k, m;
    m:= if l > i then l else i; k:= m × (m - 1) : 2;
    symmatvec:= vecvec(l, if i ≤ u then i - 1 else u, k, b, a)
    + seqvec(m, u, k + i, 0, a, b)
end symmatvec;


procedure inimat(h, n, d); value d; integer n; real d; array h;
begin integer i, j, k;
    k:= 0;
    for j:= 1, j + k while k < n do
    begin for i:= 0 step 1 until k do h[i + j]:= if i = k then d else 0;
        k:= k + 1
    end
end inimat;
```

§13. De procedure linemin

Zij $F : R_n \rightarrow R$ een niet-lineaire kontinu differentieerbare naar onder begrensde funktie met gradient g.
Stel

$$f(\alpha) = F(x_0+\alpha p) \quad , \alpha > 0 \quad , \tag{13.1}$$

waarbij x_0 een gegeven punt in R_n is en p een gegeven richting.
Stel er is gegeven een $\alpha_0 > 0$ en

$$f(0) = F(x_0) \quad , \quad f(\alpha_0) = F(x_0+\alpha_0 p) \quad ,$$
$$f'(0) = p^T g(x_0) \quad \text{en} \quad f'(\alpha_0) = p^T g(x_0+\alpha_0 p),$$

zodat geldt:

$$f'(0) < 0 \quad .$$

Zij $f(\alpha)$ minimaal voor $\alpha = \alpha_{min}$ $(\alpha_{min}>0)$.
Dan berekent linemin een zekere benadering van α_{min} .
De hiervoor gebruikte iteratieve methode wordt als volgt gedefinieerd.

Zij $y_0 = u_0 = 0$, $v_0 = \alpha_0$.

Dan wordt de k-de (k=0,1,2,...) stap gegeven door:

als $f'(v_k) \geq 0$, dan bepalen we een volgende approximatie met kubisch interpoleren, waarbij we de volgende door Davidon (1959) geïntroduceerde formule voor de bepaling van het minimum van een derdegraads vergelijking gebruiken:

$$z = 3 \times (f(u_k)-f(v_k))/(v_k-u_k) + f'(u_k) + f'(v_k) \quad ;$$

$$w = (z^2-f'(u_k) \times f'(v_k))^{1/2} \quad ;$$

$$y = v_k - (v_k-u_k) \times \{(f'(v_k)+w-z) \; / \; (f'(v_k)-f'(u_k)+2w)\} \quad ;$$

zij $\varepsilon_k = ||x_0 + y_k p||\ \varepsilon_r + \varepsilon_a$;

als $(y-u_k) < \varepsilon_k$ dan $y_{k+1} = u_k + \varepsilon_k$ anders, als $(v_k-y) < \varepsilon_k$ dan $y_{k+1} = v_k - \varepsilon_k$, anders $y_{k+1} = y$;

als $f'(y_{k+1}) \geq 0$ dan $v_{k+1} = y_{k+1}$; $u_{k+1} = u_k$;
anders $v_{k+1} = v_k$; $u_{k+1} = y_{k+1}$;

als $f'(v_k) < 0$ dan:

als $$\frac{f(v_k) - f(0)}{v_k f'(0)} > \mu$$

dan geldt $0 < v_k < \alpha_{min}$ en we stellen :

$$u_{k+1} = v_k \quad \text{en} \quad y_{k+1} = v_{k+1} = 2v_k ,$$

anders geldt $u_k < \alpha_{min} < v_k$ en we stellen :

$$y_{k+1} = \frac{1}{2}(u_k+v_k) \qquad \text{en}$$

$$\text{als } f'(y_{k+1}) \geq 0 \quad \text{of} \quad \frac{f(y_{k+1}) - f(0)}{y_{k+1} f'(0)} < \mu$$

dan $v_{k+1} = y_{k+1}$; $u_{k+1} = u_k$;

anders $v_{k+1} = v_k$; $u_{k+1} = y_{k+1}$.

In linemin zijn drie stopkriteria van belang:

1° $\frac{1}{2}|v_k - u_k| < ||x_0 + y_{k+1} p||\ \varepsilon_r + \varepsilon_a$, voor zekere gegeven ε_r, $\varepsilon_a > 0$;

2° $$\mu \leq \frac{f(y_k) - f(0)}{y_k f'(0)} \leq 1 - \mu \qquad \text{(vgl. (3.4))}$$

en $0 < \alpha_{min} < v_k$;

3° het aantal benodigde funktieëvaluaties overschrijdt een zekere bovengrens.

De gebruiker van de procedure moet kiezen of deze drie stopkriteria allen worden gebruikt, of dat alleen wordt gestopt als voldaan is aan 1° of 3° .
Als het proces na de k-de iteratiestap stopt op grond van één van de drie stopkriteria, dan wordt als resultaat $\alpha_{ber} = y_{k+1}$ afgeleverd.

We bespreken nu de betekenis van de formele parameters die in de procedure voorkomen:

n : Een variabele van type integer. De waarde van n moet gelijk zijn aan de orde van het probleem, d.i. het aantal variabelen van de funktie F.

x : Een array van dimensie x[1 : n], waarin x_0 (zie 13.1)) moet worden meegegeven en waarin $x_0 + \alpha_{ber}p$ wordt afgeleverd.

d : Een array van dimensie d[1 : n], waarin de richting p moet worden meegegeven.

nd : Een variabele van type real, waarin ||p|| moet worden meegegeven.

alfa : Een variabele van type real, waarin α_0 moet worden meegegeven en waarin α_{ber} wordt afgeleverd.

g : Een array van dimensie g[1 : n], waarin de gradient $g(x_0+\alpha_{ber}p)$ wordt afgeleverd.

funct: Een procedure van type real, waarvan de heading moet luiden:

real procedure funct(n,x,g) ; integer n ; array x, g ;
De dimensies van de arrays zijn: x, g[1 : n] .
Een aanroep van funct met in x de vektor X, moet als effekt hebben:

1° funct := F(X) ;

2° in g wordt de gradient g(X) afgeleverd.

f0 : Een variabele van type real,waarin f(0) moet worden meegegeven.

f1 : Een variabele van type real, waarin $f(\alpha_0)$ moet worden meegegeven en waarin $f(\alpha_{ber})$ wordt afgeleverd.

df0 : Een variabele van type real, waarin $f'(0)$ moet worden meegegeven.

df1 : Een variabele van type real, waarin $f'(\alpha_0)$ moet worden meegegeven en waarin $f'(\alpha_{ber})$ wordt afgeleverd.

evlmax: Een variabele van type integer, waarin het maximaal toegestane aantal funktieëvaluaties, d.i. aanroepen van funkt, moet worden meegegeven. In evlmax wordt het benodigde aantal funktie-evaluaties afgeleverd.

strongsearch: Een variabele van type boolean met de volgende betekenis:

if strongsearch then
het proces stopt alleen als aan stopkriterium 1 of 3 is voldaan
else het proces stopt als aan een van de drie gegeven stopkriteria is voldaan.

input : Een array van dimensie input [1 : 3], waarin moeten zijn gegeven:

input [1] : de relatieve tolerantie ε_r (zie stopkriterium 1°) ;

input [2] : de absolute tolerantie ε_a (zie stopkriterium 1°) ;

input [3] : de parameter μ. Hiervoor moet gelden :
$0 < \mu << \frac{1}{2}$ (zie stopkriterium 2° en (3.4)).
In de praktijk is $\mu = 0.0001$ meestal een geschikte keuze.

linemin maakt gebruik van de procedure vecvec (Dekker (1968)).

```
procedure linemin(n, x, d, nd, alfa, g, funct, f0, f1, df0, df1,
evlmax, strongsearch, input); value n, nd, f0, df0, strongsearch;
integer n, evlmax; boolean strongsearch; real nd, alfa, f0, f1, df0, df1;
array x, d, g, input;
real procedure funct;
begin integer i, evl;
    boolean notinint;
    real f, oldf, df, olddf, mu, alfa0, q, w, y, z, reltol, abstol,
    eps, aid;
    array x0[1:n];
    reltol:= input[1]; abstol:= input[2]; mu:= input[3]; evl:= 0;
    alfa0:= 0; oldf:= f0; olddf:= df0; y:= alfa; notinint:= true;
    for i:= 1 step 1 until n do x0[i]:= x[i];
    eps:= (sqrt(vecvec(1, n, 0, x, x)) × reltol + abstol) / nd;
    q:= (f1 - f0) / (alfa × df0);
int: if notinint then notinint:= df1 < 0 ∧ q > mu; aid:= alfa;
    if df1 ≥ 0 then
    begin z:= 3 × (oldf - f1) / alfa + olddf + df1;
        w:= sqrt(z ↑ 2 - olddf × df1);
        alfa:= alfa × (1 - (df1 + w - z) / (df1 - olddf + w × 2));
        if alfa < eps then alfa:= eps else
        if aid - alfa < eps then alfa:= aid - eps
    end cubic interpolation
    else if notinint then
    begin alfa0:= alfa:= y; olddf:= df1; oldf:= f1 end
    else alfa:= 0.5 × alfa; y:= alfa + alfa0;
    for i:= 1 step 1 until n do x[i]:= x0[i] + d[i] × y;
    eps:= (sqrt(vecvec(1, n, 0, x, x)) × reltol + abstol) / nd;
    f:= funct(n, x, g); evl:= evl + 1; df:= vecvec(1, n, 0, d, g);
    q:= (f - f0) / (y × df0);
    if (if notinint ∨ strongsearch then true else q < mu ∨ q > 1 - mu)
    ∧ evl < evlmax then
    begin if notinint ∨ df > 0 ∨ q < mu then
        begin df1:= df; f1:= f end
        else
        begin alfa0:= y; alfa:= aid - alfa; olddf:= df; oldf:= f end;
        if alfa > eps × 2 then goto int
    end;
    alfa:= y; evlmax:= evl; df1:= df; f1:= f
end linemin;
```

§14. Enige procedures voor de korrektie van de metriek

rnkoneupd

Zij H de symmetrische matrix, van orde n, waarvan de bovendriehoek is gegeven in array h[1 : n × (n+1) : 2], zodanig dat h[(j-1) × j : 2 + i] het (i,j)-de element van H is (1≤i≤j≤n) .
Zij v de vektor die is gegeven in array v[1 : n] .
Dan berekent rnkoneupd de symmetrische matrix H^* waarvoor geldt:

$$H^* = H + c \times vv^T .$$

De bovendriehoek van H^* wordt afgeleverd in h.

davupd

Zij H de symmetrische matrix, van orde n, waarvan de bovendriehoek is gegeven in array h[1 : n × (n+1) : 2], zodanig dat h[(j-1) × j : 2 + i] het(i,j)-de element van H is (1≤i≤j≤n).
Zij v de vektor die is gegeven in array v[1 : n].
Zij w de vektor die is gegeven in array w[1 : n].
Dan berekent davupd de symmetrische matrix H^* waarvoor geldt:

$$H^* = H + c1 \times vv^T + c2 \times ww^T$$

De bovendriehoek van H^* wordt afgeleverd in h.

fleupd

Zij H de symmetrische matrix, van orde n, waarvan de bovendriehoek is gegeven in array h[1 : n × (n+1) : 2] , zodanig dat h[(j-1) × j : 2 + i] het (i,j)-de element van H is (1≤i≤j≤n) .
Zij v de vektor die is gegeven in array v[1 : n].
Zij w de vektor die is gegeven in array w[1 : n].

Dan berekent fleupd de symmetrische matrix H^* waarvoor geldt:

$$H^* = H - c1 \times vw^T - c1 \times wv^T + (1+c1/c2) \times c1 \times vv^T.$$

De bovendriehoek van H^* wordt afgeleverd in h.

```
procedure rnkoneupd(h, n, v, c); value c; integer n; real c; array h, v;
begin integer i, j, k;
    real vk;
    k:= 0;
    for j:= 1, j + k while k < n do
    begin k:= k + 1; vk:= v[k] × c;
        for i:= 0 step 1 until k - 1 do h[i + j]:= h[i + j] + v[i + 1]
        × vk
    end
end rnkoneupd;


procedure davupd(h, n, v, w, c1, c2); value c1, c2; integer n;
real c1, c2; array h, v, w;
begin integer i, j, k;
    real vk, wk;
    k:= 0;
    for j:= 1, j + k while k < n do
    begin k:= k + 1; vk:= v[k] × c1; wk:= w[k] × c2;
        for i:= 0 step 1 until k - 1 do h[i + j]:= h[i + j] + v[i + 1]
        × vk - w[i + 1] × wk
    end
end davupd;


procedure fleupd(h, n, v, w, c1, c2); value c1, c2; integer n;
real c1, c2; array h, v, w;
begin integer i, j, k;
    real vk, wk, aid;
    aid:= (1 + c1 / c2) × c1; k:= 0;
    for j:= 1, j + k while k < n do
    begin k:= k + 1; vk:= - w[k] × c1 + v[k] × aid; wk:= v[k] × c1;
        for i:= 0 step 1 until k - 1 do h[i + j]:= h[i + j] + v[i + 1]
        × vk - w[i + 1] × wk
    end
end fleupd;
```

§15. De procedure rnkonemin

De procedure rnkonemin is een ALGOL-60 versie van de in §6 beschreven rang-één algoritme A, voor de bepaling van een minimum van funkties van meerdere variabelen.
Zij u de plaats van een minimum van de gegeven funktie F en x^* de berekende benadering daarvan.
We bespreken nu de betekenis van de formule parameters welke in de procedure voorkomen.

n : Een variabele van type integer, waarvan de waarde gelijk moet zijn aan het aantal variabelen van F.

x : Een array van dimensie x[1 : n].
In x moet een initiële approximatie van de plaats van een minimum worden meegegeven.
De berekende approximatie x^* wordt in x afgeleverd.

g : Een array van dimensie g[1 : n], waarin de gradient $g(x^*)$ wordt afgeleverd.

h : Een array van dimensie h[1 : n × (n+1) : 2] .
In h wordt de bovendriehoek van de uiteindelijk verkregen metriek H^* afgeleverd, zodanig dat h[(j-1) × j : 2 + i] het (i,j)-de element van H^* is $(1 \leq i \leq j \leq n)$.

funct : Een procedure van type real, waarvan de heading moet luiden:

```
real procedure funct(n,x,g); integer n; array x, g ;
```

De dimensies van de arrays zijn: x, g[1 : n].
Een aanroep van funct met in x de vektor X, moet als effekt hebben:

1° funct := F(X) ;

2° in g wordt de gradient g(X) afgeleverd.

input : Een array van dimensie input [0 : 8], waarin enige parameters voor de besturing van het proces moeten worden meegegeven (vgl.§5).

input [0]: De machineprecisie.
Voor de X8 is een geschikte waarde: input [0]= 10^{-12}.

input [1]: ε_r, de vereiste relatieve precisie van x^*.
Er moet gelden : input [1] > input [0].

input [2]: ε_a, de vereiste absolute precisie van x^*.

Opmerking ε_r en ε_a mogen niet al te klein worden gekozen, maar moeten overeenstemmen met de nauwkeurigheid waarin de berekende funktie en gradient de plaats van het minimum bepalen.
Voor x^* zal meestal gelden:

$$||x^* - u|| < ||x^*|| \varepsilon_r + \varepsilon_a .$$

Hiervoor kan echter geen garantie worden gegeven (vgl. verklaring bij Powell's funktie in 4 variabelen, tabel 10.1 of 10.3).

input [3]: De dalingsparameter μ (vgl. (3.3)).
Er moet gelden: $0 < \text{input [3]} < \frac{1}{2}$.
In de praktijk is gewoonlijk een geschikte waarde:
input [3] = 0.0001.

input [4] ε_g, de vereiste maximale waarde van $||g(x^*)||$.
Zij $g_b(x)$ de berekende gradient in x, dan moet gelden:

$$\text{input [4]} > ||g(x) - g_b(x)|| ,$$

voor alle x in een zekere omgeving van x^*.

input [5]: F_{min}, een ondergrens voor de funktiewaarde.

input [6]: c, de schalingsfaktor voor de initiële metriek.
De waarde van c dient een ruwe schatting te zijn van $||G^{-1}(x_0)||$, met G de Hessiaan van F.
Er moet gelden: input [6] > 0 .
De keuze input [6] = 1 zal in de meeste gevallen ook een goed resultaat geven.

input [7]: Het maximaal toegestane aantal evaluaties van de funktie en haar gradient.
Indien voor de berekening van het minimum in de vereiste precisie meer funktie- en gradientevaluaties nodig zijn, dan de in input [7] gegeven waarde, dan wordt het proces afgebroken.

input [8]: β, de orthogonaliteitsparameter (vgl.(4.8)) .
Er moet gelden:

$$\frac{1}{n}\left(\frac{\text{input [0]}}{\text{input [1]}}\right)^{1/2} \leq \text{input [8]} < 1 \ .$$

In de praktijk is dikwijls een geschikte waarde : input [8] = 0.01.

output: Een array van dimensie output [0 : 4], waarin enige bijprodukten worden afgeleverd.

output [0]: $||H^*g(x^*)||$
Het is niet noodzakelijk dat geldt:

$$||x^* - u|| < \text{output [0]} \ ,$$

maar in de meeste gevallen zal dit wel gelden (vgl. ook de verklaring bij tabel 10.3).

output [1]: $||g(x^*)||$.

output [2]: Het benodigde aantal evaluaties van de funktie en haar gradient.

output [3]: Het aantal keren dat de procedure linemin is aangeroepen.

output [4] Het aantal keren dat de staprichting is berekend met Greenstadt's methode, m.a.w. het aantal stappen waarin de eigenwaarden en -vektoren van de metriek zijn berekend.

rnkonemin is een procedure van type <u>real</u>. Na een aanroep van rnkonemin geldt:

rnkonemin = $F(x^*)$.

rnkonemin maakt gebruik van de procedures:
symmatvec, inimat (§12), linemin (§13), rnkoneupd, davupd, fleupd (§14), vecvec, matvec, tamvec (Dekker (1968)) en eigsym (Dekker and Hoffmann (1968)).

```
real procedure rnkonemin(n, x, g, h, funct, input, output); value n;
integer n; array x, g, h, input, output;
real procedure funct;
begin integer i, it, n1, n2, cntl, cnte, evl, evlmax;
    boolean ok;
    real f, f0, fmin, mu, dg, dg0, ghg, gs, nrmdelta, alfa, macheps,
    reltol, abstol, eps, tolg, orth, aid;
    array v, delta, gamma, s, p[1:n];
    macheps:= input[0]; reltol:= input[1]; abstol:= input[2];
    mu:= input[3]; tolg:= input[4]; fmin:= input[5];
    alfa:= input[6]; evlmax:= input[7]; orth:= input[8]; n1:= n + 1;
    n2:= n × (n + 1) : 2; cntl:= cnte:= 0; inimat(h, n, alfa);
    f:= funct(n, x, g); evl:= 1; ok:= true;
    dg0:= vecvec(1, n, 0, g, g); nrmdelta:= sqrt(dg0) × alfa;
    dg0:= - alfa × dg0;
    for i:= 1 step 1 until n do delta[i]:= - g[i] × alfa;
    for it:= 1, it + 1 while (nrmdelta > eps ∨ dg > tolg ∨ it ≤ n1 ∨
    ¬ok) ∧ evl < evlmax do
    begin if ¬ok then
        begin array vec[1:n,1:n], th[1:n2], em[0:9];
            em[0]:= macheps; em[2]:= aid:= sqrt(macheps × reltol);
            em[4]:= orth; em[6]:= aid × n; em[8]:= 5; cnte:= cnte + 1;
            for i:= 1 step 1 until n2 do th[i]:= h[i];
            eigsym1(th, n, n, v, vec, em);
            for i:= 1 step 1 until n do
            begin aid:= - tamvec(1, n, i, vec, g);
                s[i]:= aid × abs(v[i]); v[i]:= aid × sign(v[i])
            end;
            for i:= 1 step 1 until n do
            begin delta[i]:= matvec(1, n, i, vec, s);
                p[i]:= matvec(1, n, i, vec, v)
            end;
            dg0:= vecvec(1, n, 0, delta, g);
            nrmdelta:= vecvec(1, n, 0, delta, delta)
        end calculating greenstadts direction;
        for i:= 1 step 1 until n do
        begin s[i]:= x[i]; v[i]:= g[i] end;
        if it > n then alfa:= 1 else
        begin if it ≠ 1 then alfa:= alfa / nrmdelta else
            begin alfa:= 2 × (fmin - f) / dg0;
                if alfa > 1 then alfa:= 1
            end
        end;
        for i:= 1 step 1 until n do x[i]:= x[i] + delta[i] × alfa;
        f0:= f; f:= funct(n, x, g); evl:= evl + 1;
        dg:= vecvec(1, n, 0, delta, g);
        if it = 1 ∨ f0 - f < -mu × dg0 × alfa then
        begin i:= evlmax - evl; cntl:= cntl + 1;
            linemin(n, s, delta, nrmdelta, alfa, g, funct, f0, f,
            dg0, dg, i, false, input); evl:= evl + i;
            for i:= 1 step 1 until n do x[i]:= s[i]
        end lineminimization;
```

```
        for i:= 1 step 1 until n do
        begin gamma[i]:= g[i] - v[i]; if ¬ok then v[i]:= - p[i] end;
        dg:= dg - dg0; if alfa ≠ 1 then
        begin for i:= 1 step 1 until n do
            begin delta[i]:= delta[i] × alfa; v[i]:= v[i] × alfa end;
            nrmdelta:= nrmdelta × alfa; dg:= dg × alfa
        end;
        for i:= 1 step 1 until n do
        begin p[i]:= gamma[i] + v[i];
            v[i]:= symmatvec(1, n, i, h, gamma);
            s[i]:= delta[i] - v[i]
        end;
        gs:= vecvec(1, n, 0, gamma, s); ghg:= vecvec(1, n, 0, v, gamma);
        aid:= dg / gs;
        if vecvec(1, n, 0, delta, p) ↑ 2 > vecvec(1, n, 0, p, p) ×
        (orth × nrmdelta) ↑ 2 then rnkoneupd(h, n, s, 1 / gs) else if
        aid ≥ 0 then fleupd(h, n, delta, v, 1 / dg, 1 / ghg) else
        davupd(h, n, delta, v, 1 / dg, 1 / ghg);
        for i:= 1 step 1 until n do delta[i]:= - symmatvec(1, n, i, h,
        g); alfa:= nrmdelta;
        nrmdelta:= sqrt(vecvec(1, n, 0, delta, delta));
        eps:= sqrt(vecvec(1, n, 0, x, x)) × reltol + abstol;
        dg:= sqrt(vecvec(1, n, 0, g, g));
        dg0:= vecvec(1, n, 0, delta, g); ok:= dg0 ≤ 0
    end iteration;
    output[0]:= nrmdelta; output[1]:= dg; output[2]:= evl;
    output[3]:= cntl; output[4]:= cnte; rnkonemin:= f
end rnkonemin;
```

§16. De procedure flemin

De procedure flemin is een ALGOL-60 versie van de in §7 beschreven rang-twee algoritme D, voor de bepaling van een minimum van een funktie in meerdere variabelen.
De betekenis van de formele parameters welke in de procedure voorkomen is de volgende (vgl.§15).

n : Een variabele van type integer, waarin de orde van het probleem moet worden meegegeven.

x : Een array van dimensie x[1 : n], waarin de initiële approximatie van de plaats van een minimum moet worden meegegeven en waarin de berekende approximatie x^* wordt afgeleverd.

g : Een array van dimensie g[1 : n], waarin de gradient $g(x^*)$ wordt afgeleverd.

h : Een array van dimensie h[1 : n × (n+1) : 2], waarin de bovendriehoek van de uiteindelijk verkregen metriek H^* wordt afgeleverd, zodanig dat h[(j-1) × j : 2 + i] het (i,j)-de element van H^* is ($1 \leq i \leq j \leq n$).

funct : Een procedure van type real, waarvan de heading moet luiden:

real procedure funct (n,x,g); integer n; array x, g;

De dimensies van de arrays zijn x, g[1 : n].
Een aanroep van funct, met in x de vektor X, moet als effekt hebben:
1° funct := F(X);
2° in g wordt de gradient g(X) afgeleverd.

input : Een array van dimensie input [1 : 7], waarin enige parameters voor de besturing van het proces moeten worden meegegeven (vgl.§15).
input [1] : ε_r, de vereiste relatieve precisie.
input [2] : ε_a, de vereiste absolute precisie.
input [3] : De dalingsparameter μ.

input [4] : ε_g, de vereiste maximale waarde van $||g(x^*)||$.
input [5] : F_{min}, een ondergrens voor de funktiewaarde .
input [6] : c, de schalingsfaktor voor de initiële metriek .
input [7] : Het maximaal toegestane aantal evaluaties van de funktie en haar gradient.
Voor de in input [1 : 7] mee te geven waarden gelden dezelfde eisen als in §15.

output : Een array van dimensie output [0 : 4], waarin enige bijprodukten worden afgeleverd (vgl.§15).
output [0] : $||H^*g(x^*)||$.
output [1] : $||g(x^*)||$.
output [2] : Het benodigde aantal evaluaties van de funktie en haar gradient.
output [3] : Het aantal keren dat linemin is aangeroepen.
output [4] : Als voor alle $k \leq N$, waarbij N het benodigde aantal iteratiestappen is, geldt:

$$g^T(x_k)H_k g(x_k) > 0 ,$$

dan output [4] := 0 .
Als voor zekere k geldt:

$$g^T(x_k)H_k g(x_k) \leq 0 ,$$

dan wordt het proces afgebroken met
output [4] = -1 .
Deze laatste situatie kan zich voordoen door afrondfouten als ε_r of ε_a te klein zijn gekozen (zie §15, opmerking bij input [2]), of bijvoorbeeld door inkorrekt programmeren van de funktie of haar gradient.

flemin is een procedure van type real.
Na een aanroep van flemin geldt:

$$\text{flemin} = F(x^*) .$$

flemin maakt gebruik van de procedures:
symmatvec, inimat (§12), linemin (§13), davupd, fleupd (§14) en vecvec (Dekker (1968)).

```
real procedure flemin(n, x, g, h, funct, input, output); value n;
integer n; array x, g, h, input, output;
real procedure funct;
begin integer i, it, n1, cntl, evl, evlmax;
    real f, f0, fmin, mu, dg, dg0, nrmdelta, alfa, reltol, abstol, eps,
    tolg, aid;
    array v, delta, s[1:n];
    reltol:= input[1]; abstol:= input[2]; mu:= input[3];
    tolg:= input[4]; fmin:= input[5]; alfa:= input[6];
    evlmax:= input[7]; output[4]:= 0; inimat(h, n, alfa); n1:= n + 1;
    f:= funct(n, x, g); evl:= 1; cntl:= 0; dg0:= vecvec(1, n, 0, g, g);
    nrmdelta:= sqrt(dg0) × alfa; dg0:= - alfa × dg0;
    for i:= 1 step 1 until n do delta[i]:= - g[i] × alfa;
    for it:= 1, it + 1 while (nrmdelta > eps ∨ dg > tolg ∨ it ≤ n1) ∧
    evl < evlmax do
    begin for i:= 1 step 1 until n do
        begin s[i]:= x[i]; v[i]:= g[i] end;
        if it > n1 then alfa:= 1 else
        begin if it ≠ 1 then alfa:= alfa / nrmdelta else
            begin alfa:= 2 × (fmin - f) / dg0;
                if alfa > 1 then alfa:= 1
            end
        end;
        for i:= 1 step 1 until n do x[i]:= x[i] + delta[i] × alfa;
        f0:= f; f:= funct(n, x, g); evl:= evl + 1;
        dg:= vecvec(1, n, 0, delta, g);
        if it = 1 ∨ f0 - f < - mu × dg0 × alfa then
        begin i:= evlmax - evl; cntl:= cntl + 1;
            linemin(n, s, delta, nrmdelta, alfa, g, funct, f0, f,
            dg0, dg, i, false, input); evl:= evl + i;
            for i:= 1 step 1 until n do x[i]:= s[i]
        end lineminimization;
        if alfa ≠ 1 then
        for i:= 1 step 1 until n do delta[i]:= delta[i] × alfa;
        for i:= 1 step 1 until n do v[i]:= g[i] - v[i];
        for i:= 1 step 1 until n do s[i]:= symmatvec(1, n, i, h, v);
        aid:= vecvec(1, n, 0, v, s); dg:= (dg - dg0) × alfa;
        if dg ≤ 0 then n1:= n1 + 1 else if dg > aid then fleupd(h, n,
        delta, s, 1 / dg, 1 / aid) else davupd(h, n, delta, s, 1 /
        dg, 1 / aid);
        for i:= 1 step 1 until n do delta[i]:= - symmatvec(1, n, i, h,
        g); alfa:= nrmdelta × alfa;
        nrmdelta:= sqrt(vecvec(1, n, 0, delta, delta));
        eps:= sqrt(vecvec(1, n, 0, x, x)) × reltol + abstol;
        dg:= sqrt(vecvec(1, n, 0, g, g));
        dg0:= vecvec(1, n, 0, delta, g); if dg0 > 0 then
        begin output[4]:= - 1; goto exit end
    end iteration;
exit: output[0]:= nrmdelta; output[1]:= dg; output[2]:= evl;
    output[3]:= cntl; flemin:= f
end flemin;
```

Literatuur

Abadie J. (ed.)
Integer and nonlinear programming.
North-Holland (1970).

Beale E.M.L.
On an iterative method for finding a local minimum of a function of more then one variable.
Techn. Report No.25, Statistical Techniques Research Group Princeton Univ. (1958).

Box M.J.
A comparison of several current optimization methods, and the use of transformations in constrained problems.
Comp. J. 9 (1966) p.67-77.

Box M.J. , Davies D. and Swann W.H.
Non-linear optimization techniques.
ICI Monograph No.5,Oliver and Boyd (1969).

Brent P.
Algorithms for finding zero's and extrema of functions with calculated derivatives. Section 7.
Stanford Univ. Stan-CS-71-198 (1971).

Broyden C.G.
Quasi-Newton methods and their application to function minimization.
Math. Comp. 21 (1967) p.368-381.

Broyden C.G.
The convergence of a class of double-rank minimization algorithms. Parts I and II.
J. Inst. Maths. Apps. 6 (1970 a) p.76-90 en 222-231.

Broyden C.G.

The convergence of single-rank quasi-Newton methods.
Math. Comp.24 (1970 b) p.365-382.

Colville A.R.

A comparative study of nonlinear programming codes.
IBM New York Scientific Center Tech. Report 320-2949. (1968).

Curry H.

The method of steepest descent for nonlinear minimization problems.
Quart. Appl. Math. 2 (1944) p.258-261.

Davidon W.C.

Variable metric method for minimization.
Argonne Nat.Lab. Report ANL-5990. (1959).

Davidon W.C.

Variance algorithm for minimization.
Comp. J. 10 (1968) p.406-410.

Davidon W.C.

Variance algorithm for minimization.
In Fletcher (1969).

Dekker T.J.

ALGOL 60 procedures in numerical algebra. Part 1.
Mathematical Centre Tracts 22 (1968).

Dekker T.J. and Hoffmann W.

ALGOL 60 procedures in numerical algebra. Part 2.
Mathematical Centre Tracts 23 (1968).

Fiacco A.V. and McCormick G.P.

Nonlinear programming: sequential unconstrained minimization techniques.
Wiley (1968).

Fletcher R.
Certification of Algorithm 251.
Comm. ACM 9 (1966) p.686.

Fletcher R. (ed.)
Optimization.
Academic Press (1969).

Fletcher R.
A new approach to variable metric algorithms.
Comp. J. 13 (1970) p.317-322.

Fletcher R. and Powell M.J.D.
A rapidly convergent descent method for minimization.
Comp. J. 6 (1963) p.163-168.

Fletcher R. and Reeves C.M.
Function minimization by conjugate gradients.
Comp. J. 7 (1964) p.149-154.

Goldfeld S.M., Quandt R.E. and Trotter H.F.
Maximization by improved quadratic hill-climbing and other methods.
Econometrica 34 (1966) p. 541-551.

Goldstein A.A.
On steepest descent.
SIAM J. on Control Ser. A 3 (1965) p.541-551.

Goldstein A.A. and Price J.F.
An effective algorithm for minimization.
Numer. Math. 10 (1967) p.184-189.

Greenstad J.L.
On the relative efficiencies of gradient methods.
Math. Comp. 21 (1967) p.360-367.

Gregory R.T. and Karney D.L.
A collection of matrices for testing computational algorithms.
Interscience (1969).

Hamilton P.A. and Boothroyd J.
Remark on algorithm 251.
Comm. ACM 12 (1969) p.512-513.

House F.R.
Remark on algorithm 251.
Comm. ACM 14 (1971) p.358.

Leon A.
A comparison of eight known optimizing procedures.
In Lavi A. and Vogl T.P. (eds.)
Recent advances in optimization techniques.
Wiley (1966).

Murtagh B.A. and Sargent R.W.H.
A constrained minimization method with quadratic convergence.
In Fletcher (1969).

Powell M.J.D.
An iterative method for finding stationary values of a function of several variables.
Comp. J. 5 (1962) p.147-151.

Powell M.J.D.
An efficient method for finding the minimum of a function of several variables without calculating derivatives.
Comp. J. 7 (1964) p.155-162.

Powell M.J.D.
Rank one methods for unconstrained optimization.
In Abadie (1970).

Powell M.J.D.

On the convergence of the variable metric algorithm .
J. Inst. Maths. Applics 7 (1971) p.21-36.

Rabinowitz P. (ed.)

Numerical methods for nonlinear algebraic equations.
Gordon and Breach (1970).

Rosenbrock H.H.

An automatic method for finding the greatest of least value of a function.
Comp. J. 3 (1960) p.175-184.

Wells M.

Algorithm 251 : Function minimization.
Comm. ACM. 8 (1965) p.169-170.

Wilkinson J.H.

The algebraic eigenvalue problem.
Clarendon Press (1965).