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.

M C SYLLABUS

12

M C SYLLABUS

12

NUMERIEKE ALGEBRA

DOOR

T.J. DEKKER

MATHEMATISCH CENTRUM AMSTERDAM

1971

Voorwoord

Deze syllabus is ontstaan uit aantekeningen door enige studenten en mijzelf van het college Numerieke Algebra, dat ik aan de Universiteit van Amsterdam heb gegeven gedurende het eerste semester van het academische jaar 1969/1970.
Er wordt hoofdzakelijk aandacht geschonken aan numerieke oplossingsmethoden voor lineaire algebraische problemen, met name stelsels lineaire vergelijkingen en eigenwaarde-problemen.
Hierbij is een keuze gemaakt uit enige belangrijke methoden; vrijwel uitsluitend methoden geschikt voor volle (dwz. niet-schaarse) matrices komen ter sprake.
De lezer wordt verondersteld enige kennis te bezitten van de lineaire algebra, met name: vectoren, matrices, lineaire stelsels, eigenwaarden en eigenvectoren (zie bijvoorbeeld het eerste hoofdstuk uit Wilkinson (1965) of dat uit Faddeev & Faddeeva (1963)).
Voor het lezen van deze syllabus is tevens van belang dat men enige ervaring heeft in het gebruik van een rekenautomaat.
Ik betuig mijn dank aan de studenten die aan de tot stand koming van deze syllabus hebben medegewerkt, met name aan de heren W.E. Baanstra, J.C.P. Bus, W. Hoffmann, R. von Kriegenbergh en H.L. Oudshoorn, voor het opstellen en corrigeren van de tekst en aan de heer J.M. Anthonisse voor zijn waardevolle opmerkingen betreffende het hoofdstuk (6) over lineaire programmering.

Amsterdam, februari 1971.

T.J. Dekker.

Inhoud

1. Inleiding 3
2. Normen 7
3. Conditie van stelsels vergelijkingen 11
4. Getalvoorstelling en arithmetiek in een computer 15
5. Oplossen van vierkante lineaire stelsels 23
6. Lineaire programmering 45
7. Lineaire kleinste-kwadratenproblemen 57
8. Eigenwaarden en eigenvectoren 68
9. Singuliere waarden en numerieke rang van een matrix 83
10. Literatuur 88

1. Inleiding

Numerieke algebra is een onderdeel van de numerieke wiskunde waarin het onderzoek wordt gericht op het numeriek oplossen van algebraïsche problemen.
De belangrijkste problemen die aan de orde komen zijn (stelsels) algebraïsche vergelijkingen, waarin de onbekenden reële of complexe getallen of vectoren zijn. De oplossingsmethoden worden vaak ook toegepast op (stelsels) transcendente vergelijkingen, waarbij deze worden herleid tot bij-benadering-equivalente algebraïsche problemen.

Linearizering

Vaak herleidt men stelsels algebraïsche of transcendente vergelijkingen tot bij-benadering-equivalente stelsels lineaire vergelijkingen. Dit geschiedt enerzijds bij het ontwerpen van numerieke oplossingsmethoden, anderzijds bij het stellen van het probleem. Wanneer van de processen weinig bekend is (bijvoorbeeld in psychologie en economie), wordt in een eerste benadering vaak een lineair verband verondersteld; wanneer men wat meer inzicht heeft, gaat men ook niet-lineaire effecten beschouwen. Stelsels lineaire vergelijkingen vormen daarom een belangrijk bijzonder geval. Hiertoe zullen we ons in de volgende hoofdstukken beperken.

Herkomst

Stelsels vergelijkingen treden op bij:

A) discrete eindige problemen, bijvoorbeeld:

   1) verwerking van enquêtes of andere verzamelde gegevens op psychologisch, economisch of sociologisch gebied;

   2) het doorrekenen van eindige constructies, zoals electrische netwerken, bouwconstructies, enz.;

B) discretisatie van continue problemen, bijvoorbeeld differentiaal- of integraalvergelijkingen. We spreken van "discretisatie", wanneer functies gedefinieerd op een continuum worden gereduceerd tot functies gedefinieerd op een eindige deelverzameling.

Indeling

Na drie inleidende hoofdstukken over normen (hoofdstuk 2), conditie van stelsels vergelijkingen (hoofdstuk 3) en getalvoorstelling en arithmetiek in een computer (hoofdstuk 4), zullen we enige belangrijke numerieke oplossingsmethoden voor verschillende problemen behandelen.
Stelsels vergelijkingen kunnen, op grond van het aantal vergelijkingen en onbekenden, als volgt worden onderverdeeld:

1) vierkante stelsels

Deze hebben evenveel vergelijkingen als onbekenden. Het aantal oplossingen is vaak eindig als het stelsel algebraïsch is en vaak één als het stelsel lineair is. Numerieke oplossing van lineaire stelsels wordt behandeld in hoofdstuk 5.

2) onderbepaalde stelsels

Deze hebben meer onbekenden dan vergelijkingen. Meestal vraagt men een oplossing die een bepaalde functie minimaliseert, waarbij naast voorwaarden in de vorm van gelijkheden ook voorwaarden in de vorm van ongelijkheden optreden. Deze problemen behoren tot het gebied van de mathematische programmering. Een belangrijk bijzonder geval vormen de lineaire-programmeringsproblemen, waarin zowel de te minimaliseren functie als de voorwaarden lineair zijn. Numerieke oplossing van deze problemen wordt behandeld in hoofdstuk 6.

3) overbepaalde stelsels

Deze hebben meer vergelijkingen dan onbekenden. Aangezien een dergelijk stelsel in het algemeen geen oplossing heeft, bepaalt men de onbekenden zodanig, dat een norm (zie hoofdstuk 2) van de residuvector minimaal is. (Dit komt neer op het invoeren van de elementen van de residuvector als extra onbekenden, waardoor het probleem tot geval 1 of 2 herleid wordt.) Een belangrijk bijzonder geval vormen de kleinste-kwadratenproblemen, waarin de Euclidische norm van de residuvector geminimaliseerd wordt. Is het stelsel vergelijkingen lineair, dan spreekt men van lineaire kleinste-kwadratenproblemen. Numerieke oplossing van deze wordt behandeld in hoofdstuk 7.

4) homogene stelsels en eigenwaarde-problemen

Een speciale klasse vormen stelsels die homogeen zijn in de onbekenden, waarbij men geïnteresseerd is in een oplossing die ongelijk is aan de nulvector. In het bijzonder geldt dat homogene vierkante lineaire stelsels slechts dan een oplossing ongelijk aan de nulvector hebben, als de determinant van de matrix van het stelsel gelijk aan nul is. De matrix van het stelsel hangt dan meestal van een scalar (eigenwaarde) af die zo bepaald moet worden, dat hieraan is voldaan en dus een bijbehorende oplossingsvector (eigenvector) gevonden kan worden. In het eenvoudigste en meest voorkomende geval hangt de matrix lineair van de eigenwaarde af. Numerieke oplossing van dergelijke problemen wordt behandeld in hoofdstuk 8. Een algemene beschouwing over het met "eigenwaarde" verwante begrip "singuliere waarde" wordt gegeven in hoofdstuk 9.

Ontwikkelingen

1) De stelsels vergelijkingen, die men wil oplossen worden steeds groter. Het oplossen hiervan is mogelijk geworden doordat computers steeds groter (in geheugencapaciteit) en sneller worden.

2) Het oplossen van stelsels vergelijkingen treedt steeds vaker op als klein deel-probleem van een grote berekening.

3) Schaarsheid (sparseness).
Naarmate de stelsels groter worden, worden ze vaak schaarser.
Een matrix A heet schaars als de meeste elementen 0 zijn.
Dit treedt bijv. op bij:

   a) Discretisatie van partiële of gewone vergelijkingen. Soms krijgt men dan bandmatrices, die gemakkelijker hanteerbaar zijn dan andere schaarse matrices.

   b) Lineaire programmering (operations research, besliskunde).

   c) Het doorrekenen van constructies, bijv. elektrische netwerken, bouwconstructies, enz.,
   Voor schaarse matrices bestaan meestal speciale numerieke oplossingsmethoden, die evenwel in dit boek niet zullen worden behandeld.

4) Bovengenoemde punten 1 en 2 hebben tot gevolg dat programma's die stelsels vergelijkingen of eigenwaarde-problemen oplossen aan steeds hogere eisen moeten voldoen. Met name zal men garanties willen hebben omtrent correcte werking (vooral convergentie) en nauwkeurigheid. Aan nauwkeurigheid van de oplossing en foutenanalyse zal vooral in hoofdstuk 5 aandacht worden geschonken.

2. Normen

Voor literatuur zie o.a. Wilkinson (1963, 1965), Faddeev & Faddeeva (1963) en Householder (1964).

2.1. Vectornormen

Zij

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix}$$

een vector in R^n waarbij $x_i \in C$ (dat is de verzameling der complexe getallen), $i = 1, \ldots, n$. Dan is $x^* = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$. Op de vectorruimte R^n definiëren we een afbeelding in R (dat is de verzameling der reële getallen). Deze afbeelding geven we aan met $||x||$ (spreek uit "norm van x"), en moet aan de volgende eisen voldoen ($\underline{0}$ duidt de nulvector aan):

0) $||x|| \geq 0$
1) $||x|| = 0 \Leftrightarrow x = \underline{0}$.
2) $||\alpha x|| = |\alpha| \, ||x||$ voor alle $\alpha \in C$. (homogeniteit)
3) $||x+y|| \leq ||x|| + ||y||$. (driehoeksongelijkheid)

Aan deze drie eisen voldoet de "p-norm"

$$(2.1.1) \qquad ||x||_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}, \text{ mits } p \geq 1.$$

In de praktijk kiezen we voor p de waarden 1, 2 of ∞.
Dit levert:

a) $p = 1$, de 1-norm: $||x||_1 = \sum_{i=1}^{n} |x_i|$

b) $p = 2$, de Euclidische norm:

$$||x||_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2} = (x^* x)^{1/2}.$$

c) $p = \infty$, de <u>oneindig-norm</u> ook wel <u>maximum-norm</u> genoemd, verkregen door limietovergang:

$$||x||_\infty \underset{\text{def}}{=} \lim_{p\to\infty} ||x||_p = \max_{i=1,\ldots,n} |x_i|.$$

Immers, als $x = \underline{0}$, dan geldt blijkbaar

$$||x||_p = 0, \text{ dus } ||x||_\infty = 0 = \max_{i=1,\ldots,n} |x_i|.$$

Voor $x \neq \underline{0}$ hebben we per definitie

$$||x||_\infty = \lim_{p\to\infty} \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}. \text{ Zij } |x_k| = \max_{i=1,\ldots,n} |x_i|.$$

Dan $$||x||_\infty = \lim_{p\to\infty} \left(|x_k|^p \sum_{i=1}^{n} \frac{|x_i|^p}{|x_k|^p}\right)^{1/p} = |x_k| \lim_{p\to\infty} \left(\sum_{i=1}^{n} \frac{|x_i|^p}{|x_k|^p}\right)^{1/p}.$$

Aangezien $1 \leq \sum_{i=1}^{n} \frac{|x_i|^p}{|x_k|^p} \leq n$ volgt hieruit $||x||_\infty = \max_{i=1,\ldots,n} |x_i|$.

2.2. <u>Matrixnormen</u>

Zij A een m bij n matrix, d.w.z. een matrix met m rijen en n kolommen met de elementen A_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$.
We kunnen de matrix A opvatten als vector in $R^{m\times n}$ en hierop bovengenoemde definitie van p-norm toepassen.
Duiden we de aldus verkregen norm aan met $||A||_{pp}$, dan hebben we dus

$$(2.2.1) \qquad ||A||_{pp} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |A_{ij}|^p\right)^{1/p}.$$

De norm $||A||_{22}$ heet Euclidische norm of Schur-norm en wordt gewoonlijk aangeduid met $||A||_E$.
De aldus verkregen matrixnormen voldoen blijkbaar aan bovengenoemde eisen 0 - 3.
Voorzover het product AB gedefinieerd is, zullen we aan de eisen voor matrixnormen nog toevoegen:

4) $||AB|| \leq ||A||\ ||B||$. (multiplicativiteit)

We beperken ons nu tot vierkante matrices van de orde n.
Ga na dat $||A||_{\infty\infty}$ niet aan eigenschap (4) voldoet. Wel geldt dat
$n||AB||_{\infty\infty} \leq n||A||_{\infty\infty} . n||B||_{\infty\infty}$.
Voor $||A||_{11}$ en $||A||_E$ geldt eigenschap (4) wel.Voor $||A||_E$ blijkt dit aldus:

$$||AB||_E^2 = \sum_{i,j} |\sum_k A_{ik} B_{kj}|^2 \leq \sum_{i,j} \{(\sum_k |A_{ik}|^2)^{1/2} (\sum_l |B_{lj}|^2)^{1/2}\}^2 =$$

$$= \sum_{i,j} \sum_k |A_{ik}|^2 \sum_l |B_{lj}|^2 = \sum_{i,k} |A_{ik}|^2 \sum_{lj} |B_{lj}|^2 = ||A||_E^2 \, ||B||_E^2.$$

Van de normen $||A||_{pp}$ wordt in de praktijk alleen $||A||_E$ gebruikt.
Merk op dat $||I||_E = \sqrt{n}$.

We kunnen een matrix ook opvatten als een transformatie van R^n in R^n. Dit brengt ons op de zogenaamde <u>geassocieerde normen</u>:

$$(2.2.2) \qquad ||A||_p = \sup_{x \neq \underline{0}} \frac{||Ax||_p}{||x||_p}.$$

In het engels zegt men "least upper bound", daarom wordt een geassocieerde norm in de literatuur ook wel "lub norm" genoemd.
De geassocieerde norm voldoet aan de eisen 0 - 3. Uit de definitie van de geassocieerde matrixnormen volgt direct:
$||Ax||_p \leq ||A||_p \, ||x||_p$; men zegt dat een dergelijke matrixnorm <u>consistent</u> of <u>compatibel</u> is t.o.v. de bijbehorende vectornorm.
Nu kunnen we direct eigenschap 4 verifiëren:

$$||ABx||_p \leq ||A||_p \, ||Bx||_p \leq ||A||_p \, ||B||_p \, ||x||_p,$$

verder geldt $$||AB||_p \leq \frac{||ABx||_p}{||x||_p}, \quad x \neq \underline{0}.$$

Dit levert: $$||AB||_p \leq ||A||_p \, ||B||_p.$$

Voor de waarden p = 1, 2 of ∞ komt men tot:

$$||A||_1 = \max_j \sum_{i=1}^{n} |A_{ij}|,$$

$||A||_2 = \{\lambda_{max}\ (A^*\ A)\}^{1/2}$, (d.i. de wortel uit de grootste eigenwaarde van de matrix $A^*\ A$). Hierbij is A^* de toegevoegd-complex getransponeerde van A, d.w.z. $(A^*)_{ij} = \bar{A}_{ji}$. Men gebruikt ook vaak de notatie A^H en H heet de "Hermitische" operator.

De 2-norm voor matrices noemt men ook wel de <u>spectrale norm</u>.

$$||A||_\infty = \max_i \sum_j |A_{ij}| = ||A^*||_1$$

Voor de 1-norm, ∞-norm en Euclidische norm geldt $||A|| = ||\ |A|\ ||$, waarbij de <u>absolute waarde</u> $|A|$ van A is gedefinieerd door $|A|_{ij} = |A_{ij}|$.

Elke matrixnorm $||A||_p$, waarvoor $||Ax||_p \leq ||A||_p\ ||x||_p$, is een bovengrens voor de <u>spectrale radius</u> (d.w.z. maximale modulus der eigenwaarden) van A. De 2-norm is in de theorie der lineaire algebra het belangrijkst en levert ook de scherpste bovengrens voor de eigenwaarden. Zij is echter het lastigste te berekenen, om welke reden men in de praktijk meestal een van de andere genoemde normen gebruikt. Dit maakt geen groot verschil, zoals blijkt uit de volgende relaties:

$$||A||_2 \leq ||A||_E \leq \sqrt{n}\ ||A||_2$$

$$||\ |A|\ ||_2 \leq ||\ |A|\ ||_E = ||A||_E \leq \sqrt{n}\ ||A||_2$$

$$||A||_2^2 \leq ||A^*\ A||_1 \leq ||A^*||_1\ ||A||_1 = ||A||_\infty\ ||A||_1$$

$$||A||_\infty \leq n \max_{i,j} |A_{ij}| \leq n\ ||A||_1.$$

3. Conditie van stelsels vergelijkingen

Voor literatuur zie o.a. Wilkinson (1963, 1965) en Faddeev & Faddeeva (1963).

3.1. Algemene stelsels

Algemeen kunnen we een stelsel schrijven als $f_i(x_1, \ldots, x_n) = 0$, $i = 1, 2, \ldots, m$, waarbij n het aantal onbekenden en m het aantal vergelijkingen is. In vector-notatie wordt dit geschreven als $f(x) = \underline{0}$ (dat is de nulvector). Vaak hangen $f_1, \ldots, f_m$ van enige parameters $a_1, \ldots, a_p$ af: $f_i(a_1, \ldots, a_p, x_1, \ldots, x_n) = 0$, $i = 1, 2, \ldots, m$.
In vector-notatie is dit: $f(a,x) = \underline{0}$.
Bij elk stelsel kunnen we ons afvragen hoe de conditie van dat stelsel is.

Hierbij verstaan we onder de conditie van een stelsel een maat voor de gevoeligheid van een oplossing van dat stelsel voor variaties in de parameters. Indien een oplossing hiervoor zeer gevoelig is, spreken we van een slecht-geconditioneerd stelsel.
De conditie van een stelsel kan op verschillende manieren worden uitgedrukt in een of meer getallen.

We voeren de getallen K_{rs} in, die aangeven in welke mate de variabele x_r gevoelig is voor storingen in de parameter a_s. Deze getallen definieren we als volgt: $K_{rs} = \frac{\partial x_r}{\partial a_s}$, $r = 1, \ldots, n$, $s = 1, \ldots, p$.

Willen we één conditiegetal hebben voor het stelsel $f(a,x) = \underline{0}$, dan zal dit een of andere functie van de getallen K_{rs} moeten zijn, bijvoorbeeld een functie van de norm van de matrix (K_{rs}).

3.2. Vierkante lineaire stelsels

Een belangrijk bijzonder geval van een stelsel vergelijkingen is dat waarin het stelsel vierkant en lineair is. We schrijven zo'n stelsel in de vorm

$$Ax = b \,, \tag{3.2.1}$$

waarbij A een n bij n matrix en x en b n-vectoren zijn.
Als A niet-singulier is, is de oplossingsvector gelijk aan $x = A^{-1}b$, maw.

$$x_r = \sum_{i=1}^{n} (A^{-1})_{ri} b_i .$$

Veronderstellen we dat de elementen b_i, $i = 1, \ldots, n$, de parameters van dit stelsel zijn, dan geldt

$$K_{rs} = \frac{\partial x_r}{\partial b_s} = (A^{-1})_{rs} .$$

Maw. de matrix K blijkt dan gelijk aan A^{-1} te zijn.

3.3. Conditie-getal

Een vraag die we ons kunnen stellen is:
Hoe groot is de fout δx in de oplossing x indien we de vector b vervangen door de vector $b + \delta b$?
Zij x de vector waarvoor geldt $Ax = b$.
Nu geldt $A(x + \delta x) = b + \delta b$,

$Ax + A\delta x = b + \delta b$, dus $A\delta x = \delta b$, maw.

$\delta x = A^{-1}\delta b$.

Dus geldt voor een vectornorm en een daarmee consistente matrixnorm:

$$||\delta x|| \leq ||A^{-1}||\ ||\delta b|| = ||A^{-1}||\ ||b||\ ||\delta b|| \ /\ ||b|| .$$

Uit $Ax = b$ volgt $||b|| \leq ||A||\ ||x||$.

Dit levert $||\delta x|| \leq ||A^{-1}||\ ||A||\ ||x||\ ||\delta b|| \ /\ ||b||$.

3.3.1. Definitie

De constante $||A^{-1}||_p ||A||_p$ noemen we het conditie getal van de matrix A t.o.v. het "lineaire-stelsel probleem" horende bij de p-norm.

Notatie: $\text{cond}_p(A)$ of $\text{cond}(A)$.
Hierbij is p in de praktijk gelijk aan 1 of 2 ("spectrale conditiegetal") of ∞.

Voor de relatieve fout die we in x maken geldt nu:

$$(3.3.2) \qquad ||\delta x|| \;/\; ||x|| \leq \mathrm{cond}(A)\; ||\delta b|| \;/\; ||b||.$$

3.4. Een bovengrens voor de fout in de oplossing

Naast de relatieve fout in x uitgedrukt in de relatieve fout in b zijn we ook geïnteresseerd in de relatieve fout in x uitgedrukt in de relatieve fout van de matrix A.
M.a.w. hoe groot is de fout δx in de oplossing x indien we de matrix A vervangen door de matrix $A + \delta A$?
Laat weer gelden $Ax = b$.
Dan is $(A + \delta A)(x + \delta x) = b$,

$$\delta x = - A^{-1}(\delta A)(x + \delta x) \text{ waaruit volgt}$$

$$||\delta x|| \leq ||A^{-1}||\; ||\delta A||\; ||x+\delta x||,$$

$$||\delta x|| \leq (||A||\; ||A^{-1}||\; ||\delta A|| \;/\; ||A||)\; ||x+\delta x||.$$

Stellen we nu

$$(3.4.1) \qquad \gamma = \mathrm{cond}(A)\; ||\delta A|| \;/\; ||A||,$$

dan hebben we dus:

$$||\delta x|| \leq \gamma\; ||x+\delta x|| \leq \gamma\; (||x|| + ||\delta x||).$$

Nemen we aan, dat de storing δA zo klein is, dat $\gamma < 1$, dan volgt hieruit

$$||\delta x|| \;/\; ||x|| \leq \gamma \;/\; (1-\gamma).$$

Resumerend hebben we dus:

$$(3.4.2) \qquad \frac{||\delta x||}{||x||} \leq \frac{\mathrm{cond}(A)\; ||\delta A|| \;/\; ||A||}{1 - \mathrm{cond}(A)\; ||\delta A|| \;/\; ||A||},$$

als $\mathrm{cond}(A)\; ||\delta A|| \;/\; ||A|| < 1$.

In de praktijk bepalen we van A^{-1} en dus ook van cond(A) slechts een numerieke benadering. Stel dat een matrix B gevonden is als benadering voor de inverse van A, en dat AB = I + R. We noemen R de residu-matrix. Indien $||R|| < 1$, vinden we de volgende begrenzing voor cond(A);

(3.4.3) $$\text{cond}(A) = ||A||\ ||A^{-1}|| \leq ||A||\ ||B||\ /\ (1 - ||R||).$$

Bewijs

Uit $B = A^{-1} + A^{-1}R$ volgt

$$||B|| \geq ||A^{-1}|| - ||A^{-1}R||.$$

Wegens $||A^{-1}R|| \leq ||A^{-1}||\ ||R||$ volgt hieruit

$||B|| \geq ||A^{-1}||\ (1-||R||)$, waaruit (3.4.3) onmiddellijk volgt.

Substitutie van (3.4.3) in (3.4.2) levert een (praktisch berekenbare) bovengrens voor de fout.

4. Getalvoorstelling en arithmetiek in een computer

Voor literatuur zie o.a. Wilkinson (1963).

Aangezien het geheugen van een computer eindig is, is het noodzakelijk om de verzameling der reële getallen te vervangen door een geschikt gekozen eindige deelverzameling $\tilde{R}$. ($\tilde{R}$ heet de verzameling der representeerbare getallen.).
Elk reëel getal wordt afgerond tot een van de naburige elementen van $\tilde{R}$.
In de praktijk blijkt het handig te zijn een vast aantal geheugenplaatsen per getal te gebruiken.

Men onderscheidt twee systemen:

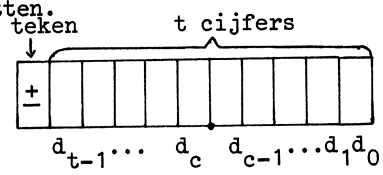
1) Fixed-point representatie.

2) Floating-point representatie.

We beschouwen een computer die rekent in het β-tallig stelsel. Voor de meeste computers geldt $\beta = 2$ of $\beta = 10$, voor sommige $\beta = 8$ of $\beta = 16$.

4.1. Fixed-point representatie

Zij t het aantal geheugenplaatsen dat beschikbaar is voor één getal (afgezien van het teken), waarbij elke geheugenplaats één van de cijfers 0, 1, ..., β-1 kan bevatten.



In fixed-point representatie moeten we ons de decimale punt op een vaste plaats denken. Wanneer het aantal cijfers na de decimale punt c bedraagt (in veel machines met fixed-point representatie geldt c = t), dan hebben alle representeerbare getallen de gedaante

$$\pm \sum_{i=0}^{t-1} d_i \, \beta^{i-c},$$

waarbij $0 \leq d_i \leq \beta-1$ en d_i geheel voor $i = 0, \ldots, t-1$.

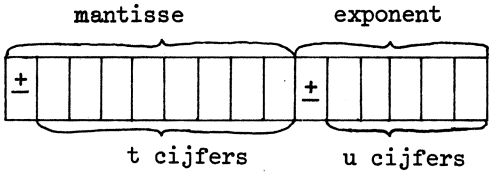
Het grootste representeerbare getal is: $M = (\beta^t - 1)\beta^{-c}$.
Daarom noemen we het interval $[-M,M]$ de <u>range</u> van deze getalrepresentatie.
Ga na dat deze getallen equidistant zijn en dat de <u>absolute</u> fout bij afronding van een willekeurig reëel getal binnen de range $\leq \frac{1}{2}\beta^{-c}$ is.

4.2. <u>Floating-point representatie</u>

Getallen in floating-point representatie hebben de gedaante

$$\text{mantisse} \times \beta^{\text{exponent}}.$$

Hierin is de exponent een geheel getal en de mantisse een fixed-point getal (in de meeste machines met $c = t$, maar in de EL X8 met $c = 0$, d.w.z. de mantisse is geheel).



De in floating-point representeerbare getallen zijn <u>niet</u> equidistant, terwijl hierbij geldt dat de <u>relatieve</u> fout bij afronding $\leq \frac{1}{2}\beta^{1-t}$, mits het gegeven getal binnen de range der floating-point getallen ligt.

<u>Standaardvoorstelling van floating-point getallen</u>.

De hierboven gegeven getalvoorstelling is niet eenduidig.
Kies bijv. $\beta = 2$. Dan kan het getal 5 op de volgende manieren worden voorgesteld:

mantisse	exponent
+ 0000101	± 000
+ 0001010	- 001
+ 0010100	- 010
+ 0101000	- 011
...	...

Vaak kiest men daarom een standaardvoorstelling (genormeerde representatie), bijvoorbeeld, met c = 0, luidt de eis $\beta^{t-1} \leq |\text{ mantisse }| \leq \beta^{t}-1$, tenzij mantisse = 0 of exponent = $-(\beta^{u}-1)$.

Voorbeeld

Zij $\beta = 2$, $t = 3$, $u = 2$, $c = 0$. Dan zijn de volgende getallen (in standaardvoorstelling) exact voorstelbaar.

mantisse	exponent	
± 111	+ 11	$\pm 7 \times 2^{3}$
± 110	+ 11	$\pm 6 \times 2^{3}$
± 101	+ 11	$\pm 5 \times 2^{3}$
± 100	+ 11	$\pm 4 \times 2^{3}$
± 111	+ 10	$\pm 7 \times 2^{2}$
± 110	+ 10	$\pm 6 \times 2^{2}$
...	...	...
...	...	...
± 101	+ 00	$\pm 5 \times 2^{0}$
± 100	+ 00	$\pm 4 \times 2^{0}$
± 111	- 01	$\pm 7 \times 2^{-1}$
± 110	- 01	$\pm 6 \times 2^{-1}$
...	...	...
...	...	...
± 100	- 11	$\pm 4 \times 2^{-3}$
± 011	- 11	$\pm 3 \times 2^{-3}$
± 010	- 11	$\pm 2 \times 2^{-3}$
± 001	- 11	$\pm 1 \times 2^{-3}$
+ 000	+ 00	0

Voor de EL X8 geldt: $\beta = 2$, $t = 40$, $u = 11$, $c = 0$, zodat voor de elementen x van $\tilde{R}$ geldt:

$$2^{-2047} \leq |x| \leq (2^{40}-1)2^{2047} \text{ of } x = 0.$$

Aangezien de voordelen van de floating-point representatie, nl. een veel grotere range van de getallen en een beperkte relatieve fout, in de meeste

gevallen belangrijker geacht worden dan de iets eenvoudiger arithmetiek en een beperkte absolute fout bij fixed-point representatie, zullen we de fouten-analyse uitsluitend voor floating-point arithmetiek behandelen.

4.3. Floating-point arithmetiek.

Laat E een expressie zijn waarin behalve representeerbare getallen ook arithmetische operatoren (bijv. +, -, ×, /) voorkomen.

4.3.1. Definitie

fl(E) is de expressie die ontstaat door de in E voorkomende arithmetische operaties te vervangen door de corresponderende floating-point operaties. Bijvoorbeeld: fl(a+b), fl(a×b+c).

4.3.2. Definitie

Een arithmetische operatie $\circ$ (bijv. +, -, × , /) heet optimaal t.o.v. $\tilde{R}$ als het resultaat van fl(a $\circ$ b) een zo dicht mogelijk bij a $\circ$ b gelegen element van $\tilde{R}$ is.
In sommige computers zijn +, -, ×, / optimaal (bijv. EL X8).

Behoudens overflow (exponent resultaat te groot) of underflow (exponent resultaat te klein) geldt voor een optimale operatie $\circ$

$$(4.3.3) \qquad fl(a \circ b) = (a \circ b)(1+\varepsilon) ,$$

waarbij $|\varepsilon| \leq \frac{1}{2} \beta^{1-t}$, m.a.w. het resultaat heeft een kleine relatieve fout (zie 4.2).

Een niet-optimale floating point optelling of aftrekking, levert niet altijd een resultaat met kleine relatieve fout. Wel geldt algemeen, behoudens overflow/underflow

$$(4.3.4) \qquad fl(a \pm b) = a(1+\varepsilon) \pm b(1+\varepsilon'),$$

met $|\varepsilon|, |\varepsilon'| \leq c\beta^{-t}$, waarbij c een constante is, die van de nauwkeurig-

heid van de arithmetiek afhangt.

Toelichting

Zij $|b| < |a|$, dus, indien a en b genormeerd zijn, $e_b \leq e_a$.
Dan moet, voordat de mantissen kunnen worden opgeteld, m_b over $e_a - e_b$ plaatsen naar rechts worden geschoven. Hierbij treedt in sommige machines een afronding of afkapping van de laatste $e_a - e_b$ cijfers van m_b op.
M.a.w. b wordt vervangen door

$$b + a\times\tilde{\varepsilon}, \text{ waarbij } |\tilde{\varepsilon}| \leq \tilde{c}\beta^{-t}.$$

Nadat dit bij a opgeteld is kan nog een tweede afronding of afkapping van het resultaat nodig zijn, zodat we krijgen

$$fl(a+b) = (a + (b+a\times\tilde{\varepsilon}))(1+\varepsilon')$$

$$= a(1+\tilde{\varepsilon})(1+\varepsilon') + b(1+\varepsilon').$$

Stellen we $(1+\tilde{\varepsilon})(1+\varepsilon') = (1+\varepsilon)$, dan hebben we bovengenoemde formule (4.3.4) verkregen.

Voor voorbeelden van de verschillende mogelijkheden zie Wilkinson (1963, p. 8-12).

In vele computers worden de floating-point optelling en aftrekking zo goed uitgevoerd, dat het resultaat altijd een kleine relatieve fout heeft, behoudens overflow/underflow. M.a.w. dan geldt (4.3.4) met $\varepsilon = \varepsilon'$.
Deze kleine relatieve fout bij optelling en aftrekking is alleen dan van belang, als de termen a en b exact zijn. Als a en b niet exact en nagenoeg tegengesteld zijn, heeft a+b altijd een grote relatieve fout, die catastrofale gevolgen kan hebben, vooral, als later door een klein getal gedeeld wordt. Dit verschijnsel heet het wegvallen van cijfers.

Floating-point vermenigvuldiging levert, eveneens behoudens overflow/underflow, altijd een resultaat met kleine relatieve fout. In formule:

(4.3.5) $\quad fl(a\times b) = (a\times b)(1+\varepsilon^*),$

met $|\varepsilon^*| \leq c\beta^{-t}$ (voor het gemak gebruiken we hier dezelfde c als bij optelling en aftrekking).

Toelichting

Zij $a = ma \times \beta^{ea}$ en $b = mb \times \beta^{eb}$. Dan geldt

$$a \times b = (ma \times mb) \times \beta^{ea+eb} .$$

Nu moet ma × mb worden genormeerd en afgerond (of afgekapt), wat een factor $1+\varepsilon^*$ oplevert, waaruit formule (4.3.5) volgt.

Indien fl(a×b) optimaal is,dan geldt c = β/2 (vgl.(4.3.3)). Voor floating-point deling geldt een analoge redenering, alleen moet bovendien voldaan zijn aan: $b \neq 0$.
Samenvattend krijgen we, mits geen overflow of underflow optreedt:

$$fl(a\pm b) = a(1+\varepsilon) \pm b(1+\varepsilon'), \text{ (ideaal geval : } \varepsilon = \varepsilon'),$$

(4.3.6) $\quad fl(a\times b) = (a\times b)(1+\varepsilon^*),$

$$fl(a/b) = (a/b)(1+\varepsilon^*),$$

met $|\varepsilon|, |\varepsilon'|, |\varepsilon^*| \leq c\beta^{-t}$, waarin c = β/2 indien fl(a o b) optimaal is.

4.4. Toepassingen

1) De floating point som van n termen: $s_n = fl(\sum_{i=1}^{n} x_i)$.

Er geldt:

$$s_1 = fl(x_1) = x_1,$$

$$s_2 = fl(s_1+x_2) = x_1(1+\varepsilon_2) + x_2(1+\varepsilon_2'),$$

$$s_3 = fl(s_2+x_3) = x_1(1+\varepsilon_2)(1+\varepsilon_3) + x_2(1+\varepsilon_2')(1+\varepsilon_3) + x_3(1+\varepsilon_3'),$$

$$\vdots$$

$$s_n = fl(s_{n-1}+x_n) = x_1 \prod_{i=2}^{n}(1+\varepsilon_i) + x_2(1+\varepsilon_2')\prod_{i=3}^{n}(1+\varepsilon_i) + \ldots +$$

$$+ x_{n-1}(1+\varepsilon_{n-1}')(1+\varepsilon_n) + x_n(1+\varepsilon_n'),$$

$$\text{met } |\varepsilon_i|, |\varepsilon_i'| \leq c\beta^{-t}.$$

Nu geldt:

$$(1-c\beta^{-t})^{n-1} \leq \prod_{i=j}^{n}(1+\varepsilon_i) \leq (1+c\beta^{-t})^{n-1} \qquad (j = 2,\ldots,n).$$

Dus

$$|s_n - \sum_{i=1}^{n} x_i| \leq (\sum_{i=1}^{n} |x_i|)\alpha,$$

waarbij $\alpha = \max\{(1+c\beta^{-t})^{n-1} - 1,\ 1 - (1-c\beta^{-t})^{n-1}\}$.

Stel $n\,c\,\beta^{-t} < 0.1$ (waar in de praktijk royaal aan voldaan is), dan geldt:

$$(1+c\beta^{-t})^n = 1 + nc\beta^{-t} + \frac{n(n-1)}{2!}(c\beta^{-t})^2 + \ldots <$$

$$< 1 + nc\beta^{-t}\left(1 + \frac{n-1}{2!}c\beta^{-t} + \frac{(n-1)(n-2)}{3!}(c\beta^{-t})^2 + \ldots\right) <$$

$$< 1 + nc\beta^{-t}\,(1 + 0.05 + 0.005 + \ldots) < 1 + 1.06\,nc\beta^{-t}.$$

Dus

(4.4.1) $\qquad (1+c\beta^{-t})^n < 1 + 1.06\,nc\beta^{-t}.$

Evenzo

(4.4.2) $\qquad (1-c\beta^{-t})^n > 1 - 1.06\,nc\beta^{-t}.$

Dit levert: $\alpha < (n-1)(1.06\,c\beta^{-t})$.

Stellen we $1.06\ c\beta^{-t} = \beta^{-t_1}$ met $t_1 = t - {}^{\beta}\log(1.06\ c)$, dan krijgen we dus

$$(4.4.3) \qquad |fl(\sum_{i=1}^{n} x_i) - \sum_{i=1}^{n} x_i| \leq (n-1)\ \beta^{-t_1} \sum_{i=1}^{n} |x_i| .$$

In het vervolg zullen we ook nog een bovengrens nodig hebben voor $(1+\eta)^{-n}$, waarbij $|\eta| \leq c\beta^{-t}$. Nemen we aan dat $(n+1)\ c\beta^{-t} < 0.1$, dan krijgen we op analoge wijze

$$(4.4.4) \qquad 1 - n\beta^{-t_1} \leq (1+c\beta^{-t})^{-n} \leq (1+\eta)^{-n} \leq (1-c\beta^{-t})^{-n} \leq 1 + n\beta^{-t_1} .$$

2) Het floating point scalair produkt: $s_n = fl(\sum_{i=1}^{n} (a_i \times b_i))$.

Dit levert:

$$\begin{aligned}
s_1 &= (a_1 \times b_1)(1+\varepsilon_1^*), \\
s_2 &= (a_1 \times b_1)(1+\varepsilon_1^*)(1+\varepsilon_2) + (a_2 \times b_2)(1+\varepsilon_2^*)(1+\varepsilon_2') \\
&\cdot \\
&\cdot \\
&\cdot \\
s_n &= (a_1 \times b_1)(1+\varepsilon_1^*) \prod_{i=2}^{n} (1+\varepsilon_i) + (a_2 \times b_2)(1+\varepsilon_2^*)(1+\varepsilon_2') \prod_{i=3}^{n} (1+\varepsilon_i) \\
&+ \ldots + (a_{n-1} \times b_{n-1})(1+\varepsilon_{n-1}^*)(1+\varepsilon_{n-1}')(1+\varepsilon_n) + (a_n \times b_n)(1+\varepsilon_n^*)(1+\varepsilon_n')
\end{aligned}$$

met $|\varepsilon_i^*|, |\varepsilon_i|, |\varepsilon_i'| \leq c\beta^{-t}$.

Op analoge wijze als boven, volgt hieruit, mits $nc\beta^{-t} < 0.1$,

$$(4.4.5) \qquad |s_n - \sum_{i=1}^{n} (a_i \times b_i)| \leq \sum_{i=1}^{n} |a_i \times b_i| \alpha ,$$

waarbij nu $\alpha < n\ (1.06\ c\beta^{-t}) = n\beta^{-t_1}$.

5. Oplossen van vierkante lineaire stelsels

Voor literatuur zie o.a. Fox(1951),Wilkinson (1961, 1963, 1965), Faddeev & Faddeeva (1963), Forsythe & Moler (1967) en Dekker (1968).

5.1. Gauss eliminatie

Gauss eliminatie is een methode om het stelsel $Ax = b$ op te lossen. Hiertoe transformeren we de gegeven matrix van de orde n in n-1 stappen tot een bovendriekhoeksmatrix, d.w.z. een matrix waarvan alle elementen onder de hoofddiagonaal nul zijn. Zij $A^1 = A$. In de k^e stap wordt matrix A^{k+1} gevormd uit A^k als volgt.
Van elke rij met een index groter dan k wordt de k^e rij vermenigvuldigd met een zekere factor afgetrokken. De vermenigvuldigingsfactoren worden zo gekozen dat alle subdiagonaalelementen in de k^e kolom nul worden.
Zij het diagonaal element in de k^e kolom van matrix A^k gelijk aan A^k_{kk}. De n-k verschillende vermenigvuldigingsfactoren in de k^e stap worden dan gegeven door $m_{ik} = A^k_{ik}/A^k_{kk}$. Nu kunnen we aangeven hoe de elementen van A^{k+1} er uit komen te zien:

$$(5.1.1)\qquad \begin{cases} A^{k+1}_{ij} = A^k_{ij} & \text{voor } i \le k,\ j \ge i, \\ A^{k+1}_{ij} = A^k_{ij} = 0 & \text{voor } j \le k-1,\ i \ge j+1, \\ A^{k+1}_{ik} = 0 & \text{voor } i \ge k+1, \\ A^{k+1}_{ij} = A^k_{ij} - m_{ik} A^k_{kj} & \text{voor } i \ge k+1,\ j \ge k+1. \end{cases}$$

We hebben nu een veelvoud van de k^e vergelijking afgetrokken van de i^e vergelijking, $i \ge k+1$.
Indien we de oplossingsvector x willen vinden die voldoet aan $A^1x = b$, zullen we ditzelfde veelvoud van b_k van b_i moeten aftrekken. Laat daartoe $b^1 = b$. Dan moeten we dus aan onze transformatieregels nog toevoegen

$$(5.1.2)\qquad \begin{cases} b^{k+1}_i = b^k_i & \text{voor } i \le k, \\ b^{k+1}_i = b^k_i - m_{ik} b^k_k & \text{voor } i \ge k+1. \end{cases}$$

Indien vóór het uitvoeren van de k^e stap blijkt dat $A^k_{kk} = 0$, zullen we eerst voor zekere $i > k$, waarvoor $A^k_{ik} \neq 0$, de k^e rij van A^k moeten verwisselen met de i^e rij. Dit komt neer op het verwisselen van twee vergelijkingen, mits daarbij de overeenkomstige elementen van b ook verwisseld worden. Hierbij blijft dus de oplossing x ongewijzigd. Dit verwisselen noemen we <u>pivoten.</u> Het element A^k_{kk} na de verwisseling noemen we de k^e <u>pivot</u> p_k. Uit het volgende voorbeeld blijkt dat we ook moeten pivoten als $|A^k_{kk}|$ "klein" is.

<u>Voorbeeld</u>

Stel dat we rekenen met $\beta = 10$, $t = 5$. Het stelsel

$$\begin{pmatrix} 10^{-6} & 1 & -1 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix} x = \begin{pmatrix} 10^{-6} \\ 3 \\ 4 \end{pmatrix}$$

heeft als oplossing $x_1 = x_2 = x_3 = 1$.

In onze gegeven rekenprecisie geldt dat $fl(10^6 \pm 1) = 10^6$. Indien we de bovenstaande transformatieregels zonder verwisselen toepassen, krijgen we de volgende tussenresultaten:

$$A^2 = \begin{pmatrix} 10^{-6} & 1 & -1 \\ 0 & -10^6 & 10^6 \\ 0 & -10^6 & 10^6 \end{pmatrix}, \quad b^2 = \begin{pmatrix} 10^{-6} \\ 2 \\ 3 \end{pmatrix}.$$

$$A^3 = \begin{pmatrix} 10^{-6} & 1 & -1 \\ 0 & -10^6 & 10^6 \\ 0 & 0 & 0 \end{pmatrix}, \quad b^3 = \begin{pmatrix} 10^{-6} \\ 2 \\ 1 \end{pmatrix}.$$

Nu blijkt dat voor $A^3x = b^3$ geen oplossing bestaat.
Indien we in matrix A^1 de 1^e en 2^e rij verwisselen komen we tot het volgende resultaat:

$$A^1 = \begin{pmatrix} 1 & 1 & 1 \\ 10^{-6} & 1 & -1 \\ 1 & 2 & 1 \end{pmatrix}, \quad b^1 = \begin{pmatrix} 3 \\ 10^{-6} \\ 4 \end{pmatrix}.$$

$$A^2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad b^2 = \begin{pmatrix} 3 \\ -2_{10}-6 \\ 1 \end{pmatrix},$$

daar in de gegeven precisie $fl(1+_{10}-6) = 1$.

$$A^3 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}, \quad b^3 = \begin{pmatrix} 3 \\ -2_{10}-6 \\ 1 \end{pmatrix}.$$

Nu geldt kennelijk $x_3 = 1$, $x_2 - x_3 = -2_{10}-6$, dus $x_2 = fl(1-2_{10}-6) = 1$ en $x_1 + x_2 + x_3 = 3$, dus $x_1 = 1$.
Ondanks de afrondingsfouten komen we in dit voorbeeld toch tot het correcte antwoord.

In de praktijk kennen we twee strategieën voor het kiezen van de pivots:
1) <u>partieel pivoten</u>, d.w.z. in de k^e eliminatiestap wordt als pivot gekozen een element A^k_{ik}, $i > k$, met zo groot mogelijke modulus, waarna (als $i > k$) de i^e en de k^e rij van A^k verwisseld worden en dus ook het i^e en het k^e element van b^k.
2) <u>compleet pivoten</u>, d.w.z. in de k^e eliminatiestap wordt als pivot gekozen een element A^k_{ij}, $i \geq k$, $j \geq k$, met zo groot mogelijke modulus.
Hierna worden (als $i > k$) de i^e en de k^e rij van A^k verwisseld en vervolgens ook (als $j > k$) de j^e en de k^e kolom van A^k. Hierbij moeten niet alleen het i^e en het k^e element van b^k verwisseld worden, maar ook het j^e en het k^e element van de oplossingsvector x.
In beide strategieën heeft het verwisselen tot gevolg, dat voor alle m_{ik} geldt $|m_{ik}| \leq 1$, en dat nooit een element gelijk aan 0 als pivot optreedt, tenzij A singulier is. Later zullen we zien, dat voor zeer grote matrices

compleet pivoten te verkiezen is boven partieel pivoten.

5.2. Equivalente formulering van de Gauss-eliminatie

Voor een verdere theoretische behandeling kunnen we de rij en/of kolom verwisselingen buiten beschouwing laten. We kunnen ons nl. indenken dat alle noodzakelijke verwisselingen vooraf uitgevoerd zijn. Laat in het vervolg A^1 deze getransformeerde matrix zijn, en b^1 de overeenkomstig getransformeerde rechterlidvector.
Dan geldt dus voor partieel pivoten

$$(5.2.1) \qquad A^1 = PA, \qquad b^1 = Pb,$$

waarbij P een permutatiematrix is. In de praktijk kunnen de verwisselingen niet echt vooraf uitgevoerd worden, omdat de elementen A^k_{ij} op grond waarvan de pivots gekozen worden, pas tijdens het proces bekend worden. De transformatie die we uitvoeren om A^{k+1} uit A^k te vormen kunnen we ons indenken als een voorvermenigvuldiging met een matrix M_k^{-1} als volgt:

$$M_1^{-1}A^1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -m_{21} & 1 & \cdots & 0 \\ -m_{31} & 0 & \ddots & 0 \\ \vdots & \vdots & & \vdots \\ -m_{n1} & 0 & \cdots & 1 \end{pmatrix} A^1 = A^2 ,$$

$$M_2^{-1}A^2 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \\ 0 & -m_{32} & \ddots & \\ \vdots & 0 & & 0 \\ 0 & -m_{n2} & \cdots & 1 \end{pmatrix} A^2 = A^3 \ldots,$$

$$M_{n-1}^{-1}A^{n-1} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & 0 & & \vdots \\ \vdots & & & 0 & 1 \\ 0 & 0 & \cdots & 0 & -m_{nn-1} \end{pmatrix} A^{n-1} = A^{n}.$$

Samenvattend: $A^n = M_{n-1}^{-1} M_{n-2}^{-1} \ldots M_2^{-1} M_1^{-1} A^1$, waarin de matrices M_k^{-1}, $k=1,\ldots,n-1$, <u>onderdriehoeksmatrices</u> zijn en A^n een bovendriehoeksmatrix is.

Merk op dat:

a) het produkt van twee onderdriehoeksmatrices weer een onderdriehoeksmatrix is,

b) de inverse van een onderdriehoeksmatrix weer een onderdriehoeksmatrix is.

Noem nu:

$M_{n-1}^{-1} M_{n-2}^{-1} \ldots M_2^{-1} M_1^{-1} = L^{-1}$ (lower triangular matrix)

en $A^n = U$ (upper triangular matrix)

dan geldt:

$$U = L^{-1} A^1 \text{ , m.a.w. } A^1 = LU.$$

We definiëren bovendien

$$y = L^{-1}b^1.$$

Uit een eenvoudige berekening volgt

$$L = M_1 \ldots\ldots M_{n-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ m_{n1} & \cdots\cdots & m_{nn-1} & 1 \end{pmatrix},$$

m.a.w. voor $i > k$ geldt $L_{ik} = m_{ik}$.

Uit de transformatie regels voor de Gauss-eliminatie (5.1.1 en 5.1.2) volgt

$$U_{ij} = A^{n}_{ij} = A^{i}_{ij} \qquad , \qquad j \geq i,$$

$$y_i = b^{n}_{i} = b^{i}_{i} \quad .$$

5.3. Achterwaartse fouten-analyse Gauss-eliminatie.

We onderscheiden twee soorten fouten-analyses:

1) voorwaartse (forward) analyse, waarin wordt nagegaan wat de fout in het numerieke resultaat is,

2) achterwaartse (backward) analyse, waarin wordt nagegaan hoe de gegevens gewijzigd moeten worden, om bij exact rekenen het gevonden numerieke resultaat te krijgen.

Deze laatste analyse sluit nauw aan bij de praktijk, omdat vaak de gegevens niet exact bekend zijn. We zullen daarom deze behandelen.

De achterwaartse fouten-analyse is ontwikkeld door Von Neumann & Goldstine (1947), Givens (1954) en vooral door Wilkinson (1961, 1963, 1965).

We gaan uit van het stelsel:

$$A^1 x = b^1 \ ,$$

waarbij A^1 en b^1 zijn gedefinieerd volgens (5.2.1).

Bij de Gauss-eliminatie wordt nu A^1 ontbonden in een onder- en een bovendriehoeksmatrix L resp. U. Deze ontbinding is niet exact. We gaan nu na wat de matrix is, die exact gelijk is aan LU.
We zullen bewijzen dat exact geldt:

$$LU = A^1 + E,$$

waarbij $||E|| << ||A^1||$.
Daarna zullen we bewijzen dat eveneens exact geldt:

$$(A^1+\delta A^1)x = b^1,$$

waarbij $||\delta A^1|| << ||A^1||$.

Na de k-de eliminatiestap geldt:

(5.3.1) $$A_{ij}^{k+1} = A_{ij}^{k} - m_{ik}\, U_{kj} + \varepsilon_{ij}^{k} , \qquad i,j = k+1, \ldots , n ,$$

waarbij exact geldt:

$$m_{ik} = \frac{A_{ik}^{k} + \varepsilon_{ik}^{k}}{U_{kk}} ,$$

m.a.w.

(5.3.2) $$0 = A_{ik}^{k} - m_{ik}\, U_{kk} + \varepsilon_{ik}^{k} , \qquad i = k+1, \ldots , n.$$

Dus, voor $i \leq j$ geldt:

(5.3.3) $$U_{ij} = A_{ij}^{i} = A_{ij}^{1} - m_{i1}\, U_{1j} - \ldots - m_{ii-1}\, U_{i-1j} + e_{ij},$$

waarbij $$e_{ij} = \varepsilon_{ij}^{1} + \ldots + \varepsilon_{ij}^{i-1}$$

en voor $i > j$:

(5.3.4) $$0 = A_{ij}^{1} - m_{i1}\, U_{ij} - \ldots - m_{ij-1}\, U_{j-1j} - m_{ij}\, U_{jj} + e_{ij},$$

waarbij

$$e_{ij} = \varepsilon_{ij}^{1} + \ldots + \varepsilon_{ij}^{j-1} + \varepsilon_{ij}^{j} .$$

Uit formule (4.3.6) volgt

$$\begin{aligned} A_{ij}^{k+1} &= A_{ij}^{k}(1+\varepsilon) - m_{ik}\, U_{kj}(1+\varepsilon^{*})(1+\varepsilon') \\ &= A_{ij}^{k} - m_{ik}\, U_{kj} + A_{ij}^{k}\varepsilon - m_{ik}\, U_{kj}\, [(1+\varepsilon^{*})(1+\varepsilon')-1]. \end{aligned}$$

Vergelijken we dit met (5.3.1), dan vinden we

$$|\varepsilon_{ij}^{k}| \leq |A_{ij}^{k}|\ |\varepsilon| + |m_{ik}|\ |U_{kj}|\ |(1+\varepsilon^{*})(1+\varepsilon')-1|.$$

Wegens $|\varepsilon|, |\varepsilon'|, |\varepsilon^{*}| \leq c\beta^{-t}$, geldt (vgl. 4.4)

$$|(1+\varepsilon^*)(1+\varepsilon') - 1| \leq 2\beta^{-t_1},$$

waarbij $\beta^{-t_1} = 1.06\ c\ \beta^{-t}$.

Aangezien $|m_{ik}| \leq 1$ volgt uit deze afschatting

$$|\varepsilon^k_{ij}| \leq |A^k_{ij}|\ \beta^{-t_1} + |U_{kj}|\ 2\beta^{-t_1},$$

d.w.z.

$$|\varepsilon^k_{ij}| \leq 3\beta^{-t_1} \max_{i,j,k} |A^k_{ij}|$$

We definiëren nu de <u>groeifactor</u>, g, als volgt

$$g \underset{\text{def.}}{=} \max_{i,j,k} |A^k_{ij}|. \tag{5.3.5}$$

Dan hebben we dus $|\varepsilon^k_{ij}| \leq 3\beta^{-t_1}\, g$, dus (vgl. 5.3.3)

$$|e_{ij}| \leq (i-1)\ 3\beta^{-t_1}\ g, \quad i \leq j. \tag{5.3.6}$$

Voor i > k geldt:

$$m_{ik} = \frac{A^k_{ik}}{U_{kk}}\ (1+\varepsilon^*).$$

Vergelijken we dit met (5.3.2), dan krijgen we

$$\varepsilon^k_{ik} = A^k_{ik}\ \varepsilon^*,$$

dus

$$|\varepsilon^k_{ik}| = |A^k_{ik}|\ |\varepsilon^*| \leq |A^k_{ik}|\ \beta^{-t_1},$$

dus a fortiori:

$$|\varepsilon^k_{ik}| \leq |A^k_{ik}|\ 3\beta^{-t_1}.$$

Hieruit en uit (5.3.4) volgt

$$|e_{ij}| \leq 3j\ \beta^{-t_1} g\ , \quad i > j.$$

Samenvattend geldt dus:

$$|e_{ij}| \leq 3\beta^{-t_1} \min(i-1,j)g.$$

M.a.w. de foutenmatrix E voldoet aan

$$|E| = (|e_{ij}|) \leq 3\beta^{-t_1} g \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ 1 & \cdots & \cdots & 1 \\ 1 & 2 & \cdots & 2 \\ \vdots & \vdots & & \vdots \\ 1 & 2 & \cdots & n-1 \end{pmatrix}.$$

Hieruit volgt dat exact geldt:

$$A^1 + E = LU, \tag{5.3.7}$$

waarbij (vgl. 5.3.5)

$$\begin{aligned} ||E||_1 &= \max_j \sum_i |e_{ij}| \leq 3\beta^{-t_1} g \sum_i (i-1) \\ &= \frac{3}{2} n(n-1)\, \beta^{-t_1} g. \end{aligned} \tag{5.3.8}$$

<u>Pivot-strategie</u>

Voor partieel pivot-strategie (voor definitie zie 5.1) geldt:

$$g \leq 2^{n-1} \max_{i,j} |A^1_{ij}|.$$

Dat g werkelijk zo ernstig groot kan worden is door Wilkinson aangetoond m.b.v. het volgende voorbeeld:

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & 1 & \ddots & \vdots & 1 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ \vdots & & \ddots & 1 & \vdots \\ -1 & \cdots & \cdots & -1 & 1 \end{pmatrix}.$$

Voor deze matrix geldt $U_{nn} = A^n_{nn} = 2^{n-1}$.

Uit dit voorbeeld blijkt dat partieel pivoten weleens tot grove fouten kan leiden.
Daarom zullen we in het navolgende ook een andere strategie bekijken, nl. compleet pivoten (voor definitie, zie 5.1).
Voor compleet pivot strategie geldt:

$$g = \max_{i,j,k} |A_{ij}^{k}| \leq f(n) \max_{i,j} |A_{ij}^{1}|$$

met

$$f(n) = n^{\frac{1}{2}}(2^{1}\, 3^{\frac{1}{2}} \ldots\ldots n^{\frac{1}{n-1}})^{\frac{1}{2}},$$

zie Wilkinson (1961) en (1965, p.213).

Dan gaat (5.3.8) over in

$$||E||_1 \leq \frac{3}{2}n(n-1)\, f(n)\, \beta^{-t_1} \max_{i,j} |A_{ij}^{1}|. \qquad (5.3.9)$$

Ter illustratie geven we van f het volgende tabelletje:

n	f(n)
10	19
20	67
50	530
100	3300

De bovengrens (5.3.9) is dus veel lager dan die voor partieel pivoten.

5.4. Heen- en terugsubstitutie

Afgezien van de foutenmatrix E, hebben we nu verkregen het met Ax = b equivalente stelsel

$$LUx = b^{1}.$$

We berekenen x in twee stappen, nl. we lossen op het stelsel

$$Ly = b^{1} \quad \text{(heensubstitutie)}$$

en het stelsel

$$Ux = y \quad \text{(terugsubstitutie)}.$$

Dit geldt evenwel niet exact. We zullen bewijzen dat wel exact geldt:

$$(L+\delta L)y = b^1 \text{ en } (U+\delta U)\tilde{x} = y,$$

waarbij $||\delta L|| \,/\, ||L||$ en $||\delta U|| \,/\, ||U||$ klein zijn.

Heensubstitutie

Wegens $b_k^k = y_k$ luidt de transformatie-regel voor het rechterlid:

$$b_i^{k+1} \approx b_i^k - m_{ik}\, y_k, \qquad i = k+1, \ldots, n.$$

Volgens (4.3.6) krijgen we

$$b_i^{k+1} = b_i^k(1+\varepsilon_k) - m_{ik}\, y_k(1+\varepsilon_k^*)(1+\varepsilon_k').$$

Dus

$$y_i = b_i^1 \prod_{k=1}^{i-1}(1+\varepsilon_k) - m_{i1}y_1(1+\varepsilon_1^*)(1+\varepsilon_1') \prod_{k=2}^{i-1}(1+\varepsilon_k)$$

(5.4.1)
$$-m_{i2}y_2(1+\varepsilon_2^*)(1+\varepsilon_2') \prod_{k=3}^{i-1}(1+\varepsilon_k) - \ldots - m_{ii-1}y_{i-1}(1+\varepsilon_{i-1}^*)(1+\varepsilon_{i-1}').$$

We willen y_k en b_i^1 exact houden en alleen wijziging in m_{ik} en ook in $m_{ii} = 1$ toelaten. We delen daarom door $(1+\varepsilon_1)\ldots(1+\varepsilon_{i-1})$, dus

$$\frac{y_i}{\prod_{k=1}^{i-1}(1+\varepsilon_k)} = b_i^1 - m_{i1}y_1 \frac{(1+\varepsilon_1^*)(1+\varepsilon_1')}{1+\varepsilon_1} - m_{i2}y_2 \frac{(1+\varepsilon_2^*)(1+\varepsilon_2')}{(1+\varepsilon_1)(1+\varepsilon_2)}$$

$$- \ldots - m_{ii-1}y_{i-1} \frac{(1+\varepsilon_{i-1}^*)(1+\varepsilon_{i-1}')}{\prod_{k=1}^{i-1}(1+\varepsilon_k)}.$$

Wij verkorten dit tot

$$\frac{y_i}{1+\delta L_{ii}} = b_i^1 - (L_{i1}+\delta L_{i1})y_1 - (L_{i2}+\delta L_{i2})y_2 - \ldots - (L_{ii-1}+\delta L_{ii-1})y_{i-1}$$

waarbij $L_{ij} = m_{ij}$. Hieruit volgt (vgl. 4.4), dat onder de voorwaarde $(n+2)\ c\ \beta^{-t} < .1$, geldt (wegens $L_{ii} = 1$)

$$|\delta L_{ij}| \leq |L_{ij}|\ (j+2)\ \beta^{-t_1}, \qquad 1 \leq j \leq i-1,$$

en $|\delta L_{ii}| \leq (i-1)\ \beta^{-t_1}$.

Voor δL geldt dus elementsgewijs

$$|\delta L| \leq \beta^{-t_1} \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 3|L_{21}| & 1 & & \\ \vdots & 4|L_{32}| & & 0 \\ 3|L_{n1}| & 4|L_{n2}| & \cdots\cdots (n+1)|L_{nn-1}| & n-1 \end{pmatrix},$$

m.a.w. wegens $|L_{ij}| \leq 1$,

$$||\delta L||_1 \leq \beta^{-t_1} \max_j \left[(j+2)(n-j) + j - 1\right].$$

M.b.v. differentiëren (naar j) vinden we die j waarvoor het maximum bereikt wordt, nl. $j = \frac{n-1}{2}$, dus

$$(5.4.2) \qquad ||\delta L||_1 \leq \beta^{-t_1} \left[\frac{(n+3)(n+1)}{4} + \frac{n-3}{2}\right] = \beta^{-t_1}\ \frac{n^2+6n-3}{4}.$$

Bovendien geldt blijkbaar $||L||_1 \leq n$.

Terugsubstitutie

We lossen nu op $(U+\delta U)\tilde{x} = y$ en passen een analyse toe, analoog aan die bij de heensubstitutie; dan geldt:

$$\tilde{x}_i = fl\left(\frac{-U_{in}x_n - \ldots - U_{ii+1}x_{i+1} + y_i}{U_{ii}}\right),$$

$$i = n, n-1, \ldots, 2,1.$$

We vinden nu

$$|\delta U_{in}| \leq |U_{in}| \ (n-i+2)\ \beta^{-t_1}$$

en

$$|\delta U_{ij}| \leq |U_{ij}| \ (j-i+3)\ \beta^{-t_1}, \quad i < j < n,$$

terwijl

$$|\delta U_{ii}| \leq 2|U_{ii}|\ c\ \beta^{-t} < 2|U_{ii}|\ \beta^{-t_1}$$

en $$|\delta U_{nn}| < |U_{nn}|\beta^{-t_1}.$$

Dus voor δU geldt elementsgewijs:

$$|\delta U| \leq \beta^{-t_1}$$

$$\begin{pmatrix} 2|U_{11}| & 4|U_{12}| & \ldots\ldots & (n+1)|U_{1n-1}| & (n+1)|U_{1n}| \\ & & & \vdots & \vdots \\ & & & 4|U_{n-2n-1}| & 4|U_{n-2n}| \\ & & & 2|U_{n-1n-1}| & 3|U_{n-1n}| \\ 0 & & & & |U_{nn}| \end{pmatrix}$$

Hieruit volgt

$$(5.4.3) \qquad ||\delta U||_1 \leq \frac{n^2+3n-2}{2}\ \beta^{-t_1}\ \max|U_{ij}| \leq \frac{n^2+3n-2}{2}\ \beta^{-t_1} g$$

(voor definitie van g zie (5.3.5)).

Bovendien geldt blijkbaar: $||U||_1 \leq gn$.

5.5. Samenvatting

Uit voorgaande fouten-analyse volgt

$$A^1 + E = LU,$$

$$(L+\delta L)y = b^1,$$

$$(U+\delta U)\tilde{x} = y.$$

Dit levert ons:

$$(L+\delta L)(U+\delta U)\tilde{x} = b^1.$$

Uitwerken geeft:

$$(LU+L\delta U+\delta L\ U+\delta L\ \delta U)\tilde{x} = b^1.$$

Stellen we $\delta A^1 = E + L\delta U + \delta L\ U + \delta L\ \delta U$, dan geldt, wegens $LU = A^1 + E$,

$$(A^1 + \delta A^1)\tilde{x} = b^1. \tag{5.5.1}$$

Hieruit volgt voor multiplicatieve matrix norm:

$$||\delta A^1|| < ||E|| + ||L||\ ||\delta U|| + ||\delta L||\ ||U|| + ||\delta L||\ ||\delta U||,$$

dus, als we de boven gevonden afschattingen voor de 1-norm invullen:

$$||\delta A^1||_1 \leq \beta^{-t_1} g\ \left[\frac{3}{2}n(n-1) + \frac{n^3+3n^2-2n}{2} + \frac{n^3+6n^2-3n}{4} + \right.$$

$$\left. \frac{\beta^{-t_1}(n^2+6n-3)}{8}(n^2+3n-2)\right].$$

Mits de laatste term kleiner is dan 3n (wat een zeer redelijke aanname is) krijgen we

$$||\delta A^1||_1 \leq \beta^{-t_1} g\ (\tfrac{3}{4}n^3 + \tfrac{9}{2}n^2).$$

M.a.w. bij de compleet-pivotstrategie geldt

$$||\delta A^1||_1 \leq \beta^{-t_1}(\tfrac{3}{4}n^3 + \tfrac{9}{2}n^2)\ f(n)\ ||A^1||_1. \tag{5.5.2}$$

Deze bovengrens is zeker niet de best mogelijke en is vermoedelijk een royale overschatting.
In de praktijk geldt gewoonlijk

$$||\delta A^1||_1 \leq \beta^{-t_1}\, n\, f(n)\, ||A^1||_1.$$

Stellen we

$$\alpha = \alpha(t,n) = \beta^{-t_1}(\tfrac{3}{4} n^3 + \tfrac{9}{2} n^2)\, f(n),$$

dan gaat (5.5.2) over in

(5.5.3) $$||\delta A^1||_1 \,/\, ||A^1||_1 \leq \alpha.$$

Conclusie

Indien de gegeven matrix A (en dus ook $A^1 = PA$) niet exact bekend is (bijvoorbeeld: indien de elementen meetresultaten zijn), dan zal deze bovengrens voor $||\delta A^1||_1$ bij redelijke rekenprecisie in de praktijk vaak veel kleiner zijn dan de (meet-)fouten in de gegeven matrixelementen. Dan levert Gauss-eliminatie met compleet pitvoten (of, voor niet al te grote matrices, partieel pivoten) dus een volkomen acceptabel resultaat. In dergelijke gevallen is gebruik van een hogere rekenprecisie (bijv. dubbele lengte) dus ongegrond.

5.6. Kleinheid van het residu en conditie

Stellen we (vgl. 5.2.1)

(5.6.1) $$A^1 = PA,\ b^1 = Pb,\ \delta A^1 = P\delta A,$$

dan gaat (5.5.1) over in

(5.6.2) $$(A+\delta A)\,\tilde{x} = b.$$

Hieruit en uit formule (5.5.3) volgt

(5.6.3) $\qquad ||\delta A||_1 \,/\, ||A||_1 \leq \alpha << 1.$

Wat kunnen we hieruit concluderen omtrent de kleinheid van het residu? Definiëren we de residuvector als

(5.6.4) $\qquad \delta b = A\tilde{x} - b,$

dan volgt uit (5.6.2)

$$\delta b = (A+\delta A)\,\tilde{x} - (\delta A)\,\tilde{x} - b = -(\delta A)\,\tilde{x}.$$

Dus geldt voor vectornorm en daarmee consistente matrixnorm (vgl. 3.3 en 3.4)

(5.6.5) $$||\delta b|| \leq ||\delta A||\ ||\tilde{x}|| = \frac{||\delta A||}{||A||}\ ||A||\ ||\tilde{x}||.$$

Aangezien $\tilde{x} = A^{-1}(b+\delta b)$, volgt hieruit

$$||\delta b|| \leq \frac{||\delta A||}{||A||}\ \text{cond}\ (A)\ (||b||+||\delta b||).$$

Stellen we (vgl. 3.4.1)

$$\gamma = \frac{||\delta A||}{||A||}\ \text{cond}\ (A),$$

dan krijgen we, mits $\gamma < 1$,

(5.6.6) $\qquad ||\delta b|| \,/\, ||b|| \leq \gamma/(1-\gamma).$

Als de conditie van het stelsel niet te slecht is (d.w.z. als cond (A) niet veel groter dan 1 is), dan kunnen we uit (5.6.3) en (5.6.6) concluderen, dat de gevonden numerieke oplossing $\tilde{x}$ een klein residu heeft, d.w.z.

(5.6.7) $\qquad \text{cond}\ (A) \approx 1 \rightarrow ||\delta b|| \,/\, ||b|| << 1.$

Omgekeerd geldt: als het residu niet klein is, dan is de conditie slecht; formule (5.6.6) levert dan een ondergrens voor het conditiegetal. Uit (5.6.5) volgt, dat het residu alleen groot kan zijn, als de gevonden oplossing x groot is, d.w.z. als

$$||\tilde{x}|| >> ||b|| \;/\; ||A||.$$

Aangezien $\tilde{x} = A^{-1}(b+\delta b) = A^{-1}(b-(\delta A)\tilde{x})$, hebben we

$$\frac{||A||\;||\tilde{x}||}{||b||+||\delta A||\;||\tilde{x}||} \leq \text{cond } (A). \tag{5.6.8}$$

M.a.w. als $\tilde{x}$ groot is, dan is de conditie slecht en (5.6.8) levert dan een ondergrens voor het conditiegetal.
We kunnen (5.6.7) dus uitbreiden tot:

$$\text{cond } (A) \approx 1 \Rightarrow ||\tilde{x}|| \approx ||b|| \;/\; ||A|| \Rightarrow ||\delta b|| \;/\; ||b|| << 1. \tag{5.6.9}$$

Of, andersom gesteld:

$$||\delta b|| \;/\; ||b|| >> \beta^{-t} \Rightarrow ||\tilde{x}|| >> ||b|| \;/\; ||A|| \Rightarrow \text{cond } (A) >> 1.$$

Er is nog een derde verschijnsel dat zich alleen kan voordoen bij slechte conditie, nl. het optreden van kleine pivots. Helaas zijn er stelsels met zeer slechte conditie, waarvoor geen dezer verschijnselen optreedt, d.w.z. waarvoor geen kleine pivots optreden en noch de gevonden oplossing noch het residu verdacht groot zijn. Slechte conditie van een stelsel kan echter niet verborgen blijven wanneer een numerieke inverse van de matrix van het stelsel berekend wordt.

5.7. Numerieke matrix-inversie

Vervangen we in (5.6.2) b door de j^e eenheids-vector e_j (dat is de j^e kolom van I), en noemen we δA nu K_j en de verkregen oplossingsvector $\tilde{X}_j$, dan gaat (5.6.2) over in

(5.7.1) $\qquad (A+K_j)\,\tilde{X}_j = e_j,$

waarbij, volgens (5.5.3) en (5.6.1),

$$||K_j||_1 \,/\, ||A||_1 \leq \alpha.$$

De matrix $\tilde{X} = (\tilde{X}_1,\ldots,\tilde{X}_n)$ is dan een numerieke benadering van A^{-1}.
Voor de residumatrix $R = A\tilde{X} - I$ (vgl. 3.4) geldt dus, volgens definitie van de 1-norm (zie 2.2)

$$||R||_1 = \max_j ||-K_j \tilde{X}_j||_1 \leq \max_j ||K_j||_1 \, ||\tilde{X}_j||_1$$

$$\leq \max_j ||K_j||_1 \max_j ||\tilde{X}_j||_1 =$$

$$(\max_j ||K_j||_1)\, ||\tilde{X}||_1 \leq \alpha ||A||_1 \, ||\tilde{X}||_1.$$

Dus volgens (3.4.3) geldt, mits $\alpha||A||_1 \, ||\tilde{X}||_1 < 1$,

(5.7.2) $$\text{cond}_1(A) \leq \frac{||A||_1 ||\tilde{X}||_1}{1 - ||R||_1} \leq \frac{||A||_1 \, ||\tilde{X}||_1}{1 - \alpha||A||_1 \, ||\tilde{X}||_1}.$$

M.a.w. de numerieke inverse $\tilde{X}$ van A, berekend m.b.v. Gauss-eliminatie met compleet (of voor niet al te grote matrices: partieel) pivoten, levert een bovengrens voor het conditiegetal van A. Op analoge wijze als in (3.4) vinden we tevens de volgende ondergrens

(5.7.3) $$\text{cond}_1(A) \geq \frac{||A||_1 \, ||\tilde{X}||_1}{1 + ||R||_1} \geq \frac{||A||_1 \, ||\tilde{X}||_1}{1 + \alpha||A||_1 \, ||\tilde{X}||_1}.$$

Als $||\tilde{X}||_1$ niet al te groot is bepalen (5.7.2 en 5.7.3) de conditie vrij nauwkeurig. Uit de gevonden grenzen kan men bepalen of de gebruikte reken-precisie voldoende is.
Passen we heen- en terugsubstitutie toe voor het berekenen van de kolommen van $\tilde{X}$, dan zijn hiervoor in totaal nodig n^3 bewerkingen (vermenigvuldiging en optelling of aftrekking). Wegens speciale vorm der rechterleden kan dit

worden gereduceerd tot $\frac{2}{3}n^3$, wat samen met de $\frac{1}{3}n^3$ bewerkingen nodig voor het berekenen van L en U een totaal van n^3 bewerkingen levert.
We zien dus dat het oplossen van een lineair stelsel (met behulp van Gauss-eliminatie of LU-ontbinding) driemaal zo lang gaat duren, indien ook een numerieke inverse van de matrix wordt berekend. Dit is een hoge prijs die men moet betalen, indien men een garantie voor de nauwkeurigheid van de oplossing wenst. Men kan de bovengrens (5.7.2) verder omlaag drukken door tevens expliciet de residumatrix R te berekenen. Aangezien dit n^3 extra bewerkingen kost (dus het proces dan in totaal 6 maal zo lang gaat duren), is dit niet of slechts bij uitzondering aan te bevelen.
Wel is het nuttig de residuvector horende bij een gevonden numerieke oplossing te berekenen, zoals we in het vervolg zullen zien.
Wat betreft het geheugengebruik in een computer blijkt de matrix $\tilde{X}$ tijdens het rekenen op de plaats van A te kunnen worden opgebouwd, waarbij slechts één n-vector tijdelijk als extra werkruimte nodig is. Procedures in ALGOL 60 voor het oplossen van lineaire stelsels en het inverteren van matrices zijn o.a. gepubliceerd door Dekker (1968) en Wilkinson, e.a. (1965 & 1966).

5.8. Iteratief verbeteren van een oplossing

Indien matrix en rechterlid van een lineair stelsel exact bekend zijn heeft het zin de oplossing in een voorgeschreven precisie te vragen. Een machtig middel hiertoe is het volgende iteratieve procédé, mits de conditie van het stelsel niet al te slecht is.
Bereken eerst startvector $x^0 = \tilde{x}$ zoals boven geschetst. Dan hebben we volgens (5.6.2), als $K^0 = \delta A$,

$$(A+K^0)\, x^0 = b.$$

Vervolgens, voor i = 1, 2, 3, ...,

1) bereken de residuvector

$$r^i \approx Ax^{i-1} - b;$$

2) los op het stelsel

$$Ay^i \approx r^i;$$

3) bereken nieuwe schatting van de oplossing

$$x^i \approx x^{i-1} - y^i.$$

Het is van belang de berekening van de residu-vector in dubbele-lengte precisie uit te voeren. Daarna wordt het resultaat tot enkele lengte afgerond. Het berekenen van de correctie y^i kan gewoon gebeuren door heen- en terug-substitutie. Volgens (5.6.2) geldt dus exact, als we $K^i = \delta A$ stellen,

$$(A+K^i)\, y^i = r^i,$$

waarbij $||K^i||_1 \,/\, ||A||_1 \leq \alpha$.

Zien we even af van de fouten gemaakt in (1) en (3). (Die in (1) zijn klein vanwege het dubbele-lengte rekenen en die in (3) zijn vrijwel verwaarloosbaar aangezien we een oplossing in niet meer dan enkele-lengte precisie zoeken.) Dan hebben we

$$y^i = (A+K^i)^{-1}\, r^i = (A+K^i)^{-1}\, (Ax^{i-1}-b) = (A+K^i)^{-1}\, A(x^{i-1}-x),$$

waarbij x de exacte oplossing is. Dus

$$x^i - x = x^{i-1} - x - y^i = (I-(A+K^i)^{-1}A)\, (x^{i-1}-x).$$

Dus

$$||x-x^i|| \leq ||I-(A+K^i)^{-1}A||\; ||x-x^{i-1}||,$$

m.a.w. de vectoren x^i convergeren naar x als $||I-(A+K^i)^{-1}A|| < 1$.
Nu is

$$I - (A+K^i)^{-1}\, A = (A+K^i)^{-1}\, (A+K^i) - (A+K^i)^{-1}\, A =$$

$$(A+K^i)^{-1}\, K^i = (I+A^{-1}K^i)^{-1}\, A^{-1}\, K^i.$$

Daar $||(I+A^{-1}K^i)^{-1}||_1 \leq (1-||A^{-1}K^i||_1)^{-1}$, mits $||A^{-1}K^i||_1 < 1$, volgt hieruit

$$||I-(A+K^i)^{-1}A||_1 \leq \frac{||A^{-1}K^i||_1}{1 - ||A^{-1}K^i||_1}.$$

Aangezien

$$||A^{-1}K^i||_1 \leq ||A^{-1}||_1 \, ||K^i||_1 =$$

$$= \text{cond}_1(A) \frac{||K^i||_1}{||A||_1} \leq \alpha \, \text{cond}_1(A),$$

kunnen we concluderen, dat de vectoren x_i naar de exacte oplossing x convergeren, mits $\text{cond}_1(A)$ niet al te groot is.
Brengen we ook de fouten gemaakt in (1) en (3) in rekening, dan krijgen we dat de norm van de fout $||x^i-x||$ voor voldoend grote i niet veel groter is dan $\beta^{-t_1} ||x||$, mits cond (A) niet al te groot is.
M.a.w. dan levert bovenstaand procédé de oplossing in vrijwel de volle rekenprecisie.
Als cond (A) te groot is krijgen we geen convergentie (of bij uitzondering: "convergentie" naar een onbruikbaar resultaat), en moeten we dus in hogere precisie gaan rekenen. Als matrix en rechterlid exact bekend zijn, is het in zo'n geval zinvol hogere rekenprecisie te eisen.
Wat betreft het schatten van het conditie-getal, het volgende.
Het is hoogst onwaarschijnlijk (maar onmogelijkheid is niet bewezen), dat tijdens iteratief verbeteren van een oplossing slechte conditie niet aan het licht komt (nl. door het optreden van een grote oplossingsvector y^i en/of door langzame convergentie van de vectoren x^i). In de praktijk laat men dan ook vaak de (vrij dure) matrix-inversie achterwege.
Het aantal benodigde iteraties om de oplossing in vrijwel volle rekenprecisie te krijgen is meestal vrij klein (bv. 2 of 3), maar neemt toe met het conditie-getal. Aangezien de berekening van L en U $n^3/3$ bewerkingen kost en het aantal bewerkingen voor het berekenen van r^i, y^i en x^i ongeveer evenredig met n^2 toeneemt, is voor grote n het iteratief verbeteren van een oplossing een betrekkelijk goedkoop proces. (Stap (1) in dubbele lengte uitgevoerd weegt nog het zwaarst.)

Mede hierdoor kan het zinvol zijn iteratief verbeteren toe te passen ook als het rechterlid en/of de matrix niet exact bekend zijn. In dit geval moet men wel goed in het oog houden, dat de nauwkeurigheid van de oplossing niet alleen afhangt van het rekenproces, maar ook van de fouten in de gegevens. Tenslotte zij nog opgemerkt, dat iteratief verbeteren van een oplossing in de praktijk vaak geen noemenswaardige verkleining van het residu oplevert (zie bijv. Wilkinson (1965, p.258). We hebben nl.

$$||r^i|| = ||A(x^{i-1}-x)|| \leq ||A||\ ||x^{i-1}-x||.$$

Dus, voor voldoend grote i is $||r^i||$ niet veel groter dan $\beta^{-t_1}\ ||A||\ ||x||$, mits cond (A) niet al te groot is. Als echter cond (A) ook niet al te klein is, kan het voorkomen dat $||x||$ veel groter is dan $||b|| / ||A||$, zodat het residu dan (net) niet klein genoeg wordt. In een dergelijke (overigens onwaarschijnlijke) situatie zou men hogere rekenprecisie nodig hebben. Voor procedures in ALGOL 60 ter oplossing van lineaire stelsels met iteratief verbeteren, zie o.a. Forsythe & Moler (1967) en Wilkinson, e.a. (1966).

6. Lineaire programmering

Voor literatuur zie Gass (1958), Zoutendijk (1960), Dantzig (1963), Beale (1968), Orchard-Hays (1968), de Leve (1969).

6.1. Inleiding

Het lineaire programmeringsprobleem luidt als volgt:

Maximaliseer de objectfunctie

$$z = \sum_{j=1}^{n} c_j x_j$$

onder de voorwaarden (constraints):

$$\sum_{j=1}^{n} A_{ij} x_j \leq b_i, \qquad i = 1,\ldots,m,$$

$$x_j \geq 0, \qquad j = 1,\ldots,n.$$

De c_j, A_{ij} en b_i zijn gegeven reële getallen, de x_j moeten worden bepaald. Voegen we m verschilvariabelen (slack variables) $x_{n+1},\ldots,x_{n+m}$ toe, dan is het gestelde probleem equivalent met:

Maximaliseer

$$z = \sum_{j=1}^{n} c_j x_j$$

onder de voorwaarden

$$\sum_{j=1}^{n} A_{ij} x_j + x_{n+i} = b_i, \qquad i = 1,\ldots,m,$$

$$x_j \geq 0, \qquad j = 1,\ldots,n+m.$$

Definiëren we

$$c_{n+i} = 0, \qquad i = 1,\ldots,m,$$

$$A_{ij} = \delta_{i,j-n}, \qquad \begin{matrix} j = n+1,\ldots,n+m, \\ i = 1,\ldots,m, \end{matrix}$$

dan luidt het probleem in matrix-notatie:

Maximaliseer

$$z = c^T x$$

onder de voorwaarden

$$Ax = b \ , \ x \geq 0.$$

Als x aan de voorwaarden voldoet, dan heet x <u>toegelaten</u> (<u>feasible</u>).
Als bovendien z maximaal is, dan heet x <u>optimaal</u>.
Indien $b \geq 0$, dan wordt een triviale toegelaten oplossing gegeven door:

$$x_j = 0 \quad , \qquad j = 1,\ldots,n \ ,$$

$$x_{n+i} = b_i \ , \qquad i = 1,\ldots,m \ .$$

Voor het gemak zullen we ons tot het geval $b \geq 0$ beperken.

Stel $A_{\nu_1},\ldots,A_{\nu_m}$ zijn m lineair onafhankelijke kolommen van A.
Indien x toegelaten oplossing is en $x_j = 0$ voor $j \notin \{\nu_1,\ldots,\nu_m\}$, dan heet x <u>toegelaten basis oplossing</u> (<u>basic feasible solution</u>).

De kolommen

$A_{\nu_1},\ldots,A_{\nu_m}$ heten <u>basisvectoren</u>.

$B = [A_{\nu_1},\ldots,A_{\nu_m}]$ heet <u>basismatrix</u>.

Als $x_{\nu_i} = 0$ voor zekere ν_i dan heet x <u>ontaard</u> (<u>degenerate</u>).

<u>Stellingen</u>.

1. De verzameling F van alle toegelaten oplossingen is convex en vormt een niet-leeg (eindig of oneindig) polyeder in de (n+m)-dimensionale ruimte.
2. De verzameling van alle toegelaten basisoplossingen is gelijk aan de verzameling der hoekpunten van F.
3. ($\exists$ toegelaten oplossing) $\rightarrow$ ($\exists$ toegelaten basisoplossing).

4. Indien z naar boven begrensd is op F, dan is minstens één toegelaten basisoplossing optimaal. (Optimale oplossing hoeft niet eenduidig te zijn.)

6.2. Simplex-methode

Voor het oplossen van lineaire programmeringsproblemen worden technieken gebruikt die zijn gebaseerd op de simplex-methode, uitgevonden door G.B. Dantzig in 1947, zie o.a. Dantzig (1963).
De simplex-methode gaat uit van een toegelaten basisoplossing en berekent stapsgewijs betere (of althans niet slechtere) oplossingen totdat een optimale oplossing is bereikt. Een simplex-stap bestaat uit het uitwisselen van een basisvector A_{ν_r} tegen een niet-basisvector A_s (m.a.w. $s \notin \{\nu_1,\ldots,\nu_r\}$). Dit komt overeen met de overgang van een hoekpunt van F naar een aanliggend hoekpunt van F. Normaliter neemt in elke stap de waarde van de objectfunctie toe. Alleen in geval van ontaarding kan de objectfunctie constant blijven.
Zij

$$(6.2.1) \qquad B = [A_{\nu_1},\ldots,A_{\nu_m}],\ \gamma^T = (c_{\nu_1},\ldots,c_{\nu_m}) \text{ en } u^T = (x_{\nu_1},\ldots,x_{\nu_m}).$$

Dan geldt: $Bu = b$ en $z = \gamma^T u$.
We gaan nu A_{ν_r} uitwisselen tegen A_s, $s \notin \{\nu_1,\ldots,\nu_m\}$ en geven de daarbij ontstane nieuwe waarden aan met accenten. Dus

$$(6.2.2) \qquad B' = [A_{\nu_1},\ldots,A_{\nu_{r-1}}, A_s, A_{\nu_{r+1}},\ldots,A_{\nu_m}],$$

m.a.w. $B' = B + (A_s - B_r)e_r^T$ met $B_r = A_{\nu_r}$ en $e_r^T = (0,\ldots,0,1,0,\ldots,0)$ met de 1 op de r-de plaats. Evenzo: $\gamma' = \gamma + (c_s - \gamma_r)e_r$ met $\gamma_r = c_{\nu_r}$.

Definiëren we vector π door $\pi^T B = \gamma^T$, dan geldt $z = \gamma^T u = \gamma^T B^{-1} b = \pi^T b$.
Voor u' geldt

$$B'u' = b, \text{ m.a.w. } \{B + (A_s - B_r)e_r^T\}u' = b.$$

Voorvermenigvuldigen met π^T geeft:

$$\{\pi^T B + (\pi^T A_s - \pi^T B_r)e_r^T\}u' = \pi^T b = z.$$

Aangezien $\pi^T = \gamma^T B^{-1}$ en $\pi^T B_r = \gamma^T B^{-1} B_r = \gamma^T e_r = \gamma_r$, krijgen we:

$$\gamma^T u' + (\pi^T A_s - \gamma_r) e_r^T u' = z.$$

Wegens $e_r^T u' = u'_r$, volgt hieruit

$$\gamma^T u' + (\pi^T A_s - \gamma_r) u'_r = z.$$

Voor de nieuwe waarde z' van de objectfunctie geldt dan

$$z' = \gamma'^T u' = \gamma^T u' + (c_s - \gamma_r) u'_r.$$

Hieruit volgt: $z' - z = -u'_r(\pi^T A_s - c_s)$.
Definiëren we $\bar{c}_j = \pi^T A_j - c_j$, $j = 1,\ldots,n+m$, dan geldt $z' - z = -u'_r \bar{c}_s$.
Omdat moet gelden $z' \geq z$ en $u'_r \geq 0$, kiezen we s zó, dat $\bar{c}_s < 0$.
Als echter voor alle j geldt: $\bar{c}_j \geq 0$ dan is de oplossing x optimaal.

Stel nu dat een niet-basisvector A_s met $\bar{c}_s < 0$ is gekozen. Welke basisvector B_r moet dan worden uitgewisseld tegen A_s. Ga hiertoe uit van
$[B + (A_s - B_r)e_r^T]u' = b$.
Vermenigvuldigen met B^{-1} levert:

$$\{I + (B^{-1}A_s - B^{-1}B_r)e_r^T\}u' = B^{-1}b = u.$$

Stel $y = B^{-1}A_s$. Wegens $ye_r^T = y_r$, krijgen we dan:

$$u' + (y_r - e_r)u'_r = u .$$

Hieruit volgt:

$$u'_r = u_r / y_r$$

en, voor alle $k \neq r$,

$$u'_k = u_k - y_k u'_r.$$

Aangezien moet gelden $u' \geq 0$, moet r zó gekozen worden dat $y_r > 0$ en, voor

alle $k \neq r$, hetzij $y_k \leq 0$, hetzij $u_k/y_k \geq u_r/y_r$.
M.a.w. r moet zó gekozen worden dat voldaan is aan

$$y_r > 0 \ , \ \frac{u_r}{y_r} = \min_{y_k>0} \frac{u_k}{y_k} \ .$$

Als echter voor alle r geldt $y_r \leq 0$, dan kunnen we een oplossing met willekeurig grote waarde van de objectfunctie vinden. Immers, de oplossing x' gedefinieerd door

$$x'_{\nu_i} = u_i - \lambda y_i, \ i = 1,\ldots,m, \ x'_s = \lambda, \ x'_j = 0 \text{ voor } j \notin \{\nu_1,\ldots,\nu_m,s\}$$

voldoet voor willekeurige $\lambda > 0$ aan de voorwaarden, terwijl $z' = z - \lambda\bar{c}_s$ de bijbehorende waarde van de objectfunctie is, die dus willekeurig groot kan worden.

Samenvatting

Een simplex-stap bestaat dus uit de volgende onderdelen:

1) Los op: $Bu = b$ en bereken $z = \gamma^T u$.

2) Los op: $B^T\pi = \gamma$.

3) Bereken $\bar{c}_j = \pi^T A_j - c_j, \ \forall_j \notin \{\nu_1,\ldots,\nu_m\}$.
   Als $(\forall_j)\ \bar{c}_j \geq 0$ dan is x optimaal en de berekening is klaar.

4) Kies s zo dat $\bar{c}_s < 0$ (bijv. $\bar{c}_s$ minimaal).

5) Los op: $By = A_s$. Als alle $y_k \leq 0$ dan is er geen eindige optimale oplossing en de berekening is klaar.

6) Bepaal r zo, dat $y_r > 0$ en $\frac{u_r}{y_r} = \min_{y_k>0} \frac{u_k}{y_k}$.

7) Verwijder B_r uit de basis en voeg A_s aan de basis toe.

8) Ga naar (1).

Voorbeeld van Fröberg (1965, p. 313).

Maximaliseer $z = -5x_1 + 8x_2 + 3x_3$, waarbij $x \geq 0$ en

$$\begin{pmatrix} 2 & 5 & -1 \\ -3 & -8 & 2 \\ -2 & -12 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 4 \\ 9 \end{pmatrix} .$$

Van deze ongelijkheid maken we een gelijkheid door het toevoegen van de verschilvariabelen x_4, x_5, x_6. Dan hebben we

$$A = \begin{pmatrix} 2 & 5 & -1 & 1 & 0 & 0 \\ -3 & -8 & 2 & 0 & 1 & 0 \\ -2 & 12 & 3 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 4 \\ 9 \end{pmatrix} .$$

De triviale toegelaten basisoplossing luidt

$$x^T = (0,0,0,1,4,9).$$

Verder geldt: $\bar{c}^T = -c^T = (5,-8,-3,0,0,0)$.
Bovendien $\nu_1 = 4$, $\nu_2 = 5$, $\nu_3 = 6$, dus

$$\gamma^T = \pi^T = (0,0,0),$$

$$u^T = (1,4,9) \text{ en } z = 0.$$

We kiezen $s = 2$, want $\bar{c}_2^T = -8$ is negatief en minimaal (zie 4). Wegens $y = B^{-1}A_s$ en $B = I$ geldt $y = A_2 = \begin{pmatrix} 5 \\ -8 \\ -12 \end{pmatrix}$ (zie 5). Dus $r = 1$ (zie 6).
Dit leidt tot:

$$u_1' = \frac{u_1}{y_1} = \frac{1}{5}, \quad u_2' = u_2 - y_2 u_1' = \frac{28}{5}, \quad u_3' = u_3 - y_3 u_1' = \frac{57}{5} .$$

Daar $r = 1$ en $s = 2$, verwijderen we B_1 uit de basis en voegen A_2 toe.
Dit geeft: $B = \begin{pmatrix} 5 & 0 & 0 \\ -8 & 1 & 0 \\ -12 & 0 & 1 \end{pmatrix}$ (zie 7). We gaan nu opnieuw itereren

(zie 8) en de nieuwe waarden worden

$$\nu_1 = 2,\ \nu_2 = 5,\ \nu_3 = 6,\ u = \begin{pmatrix} \frac{1}{5} \\ \frac{28}{5} \\ \frac{57}{5} \end{pmatrix},$$

$$\gamma^T = (8,0,0),\ z = (8,0,0)\begin{pmatrix} \frac{1}{5} \\ \frac{28}{5} \\ \frac{57}{5} \end{pmatrix} = \frac{8}{5} \quad \text{(zie 1)}.$$

Vervolgens lossen we op $B^T\pi = \gamma$:

$$\begin{pmatrix} 5 & -8 & -12 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 0 \\ 0 \end{pmatrix},$$

dus $\pi^T = (\frac{8}{5},0,0)$, (zie 2).

Daarna berekenen we $\bar{c}_j = \pi^T A_j - c_j$; het resultaat is

$$\bar{c}^T = (\tfrac{41}{5},\ 0,\ -\tfrac{23}{5},\ \tfrac{8}{5},\ 0,\ 0) \text{ (zie 3)}$$

dus $s = 3$.

Oplossing van het stelsel $By = A_s$ levert:

$$y = \begin{pmatrix} -\frac{1}{5} \\ \frac{2}{5} \\ \frac{3}{5} \end{pmatrix} \quad \text{(zie 5)}.$$

Dus $r = 2$ (zie 6). We verwijderen nu B_2 uit de basis en voegen A_3 toe (zie 7). Nu keren we terug naar (1) en krijgen de volgende nieuwe waarden:

$\nu_1 = 2,\ \nu_2 = 3,\ \nu_3 = 6,\ B = \begin{pmatrix} 5 & -1 & 0 \\ -8 & 2 & 0 \\ -12 & 3 & 1 \end{pmatrix}, u^T = (3,14,3),\ \gamma^T = (+8,+3,0),$

dus $z = \gamma^T u = +66$ (zie 1).

Oplossen van $B^T\pi = \gamma$ levert $\pi^T = (20, \frac{23}{2}, 0)$ (zie 2). Dus (zie 3)

$$\bar{c}^T = (\tfrac{21}{2}, 0, 0, 20, \tfrac{23}{2}, 0) .$$

Aangezien nu $\bar{c}_j \geq 0$ voor alle j, hebben we een maximale oplossing bereikt. Deze luidt dus

$$x^T = (0,3,14,0,0,3)$$

en de bijbehorende maximale waarde van de objectfunctie is 66.

6.3. Praktische uitvoeringen

Er bestaan verscheidene praktische uitvoeringen van de simplex methode, d.w.z. verschillende programma's.
Een belangrijk verschilpunt is de wijze waarop de lineaire stelsels opgelost worden.

a) Expliciete inverse . In elke simplex-stap wordt B^{-1} expliciet berekend en in het geheugen bewaard. Een standaard matrix-inversie kost m^3 bewerkingen (zie 5.7). De inverse van B' kan evenwel op een veel snellere wijze uit die van B worden berekend. We merken daartoe op dat B' slechts in één kolom van B verschilt (vgl. 6.2.2.):

$$B' = B(I + (y - e_r)\, e_r^T). \tag{6.3.1}$$

Stel

$$E_i = I + (y - e_r)\, e_r^T . \tag{6.3.2}$$

Dan geldt

$$E_i^{-1} = I - \frac{1}{y_r} (y - e_r)\, e_r^T , \tag{6.3.3}$$

zodat

$$(B')^{-1} = (E_i)^{-1}B^{-1}. \qquad (6.3.4)$$

Daar E_i^{-1} in één kolom van de eenheidsmatrix verschilt, vergt de berekening van $(B')^{-1}$ uit B^{-1} slechts m^2 bewerkingen.

b) Productvorm van de inverse. Indien we stellen $B^i = B'$ en $B^{i-1} = B$, dan volgt uit (6.3.4)

$$(B^i)^{-1} = E_i^{-1} \ldots E_1^{-1} , \qquad (6.3.5)$$

met $B^0 = I$ (vgl. 6.1).
Formule (6.3.5) is de productvorm van de inverse.
In de gelijknamige variant van de simplex-methode slaat men niet $(B^i)^{-1}$,maar (de relevante kolommen van) E_1^{-1} , ... , E_i^{-1} in het geheugen op.
Nu worden de vectoren π' en y' berekend volgens (vgl. 6.2):

$$(\pi')^T = (\gamma'^T E_i^{-1} \ldots E_1^{-1}) , \qquad (6.3.6)$$

$$y' = (E_i^{-1} \ldots E_1^{-1} A_s). \qquad (6.3.7)$$

Wegens de speciale vorm van E_1^{-1} , ... , E_i^{-1}, vergen deze formules elk i×m bewerkingen. Een groot voordeel van de productvorm is, dat, indien de gegeven matrix A schaars is, dan ook de relevante kolommen van E_1^{-1} , ... , E_i^{-1} in de praktijk meestal schaars blijken te zijn.
Dit leidt tot grote geheugen- en rekentijdbesparing. (Dan is nl. slechts een fractie van i×m bewerkingen nodig).

Bovendien schijnt de productvorm-variant voordeel te bieden bij gebruik van machines met magneetbanden, zie Zoutendijk (1960). Zoutendijk stelde voor de berekening van $\bar{c}_j'$ te baseren op de formule (vgl. 6.2)

$$\bar{c}_j' = (\pi'^T - \pi^T) A_j + \bar{c}_j .$$

Dit leidt tot een besparing van rekentijd, omdat, voor schaarse A vaak $\bar{c}_j' = \bar{c}_j$ blijkt te zijn.

Numerieke overwegingen.

Wanneer we de methodes a en b uit het voorgaande nader beschouwen, blijkt dat ze numeriek equivalent zijn aan Gauss-Jordan eliminatie, waarbij de pivotkeuze niet gericht is op numerieke stabiliteit, maar op het niet-negatief houden van de vector u.
De pivot wordt nl. niet gekozen als een maximum-element in zekere sub-kolom of sub-matrix, maar wordt bepaald door de voorwaarde in onderdeel (6) van een simplex-stap. We kunnen hierdoor geen algemeen geldende uitspraak omtrent de stabiliteit van methodes a en b doen.

Men krijgt een stabiel proces door af en toe B^{-1} (expliciet of in productvorm) opnieuw te bepalen; we noemen dit herinversie.
Deze herinversie geschiedt in (hoogstens) m stappen, waarin de verlangde kolommen in de basis worden opgenomen in een volgorde, die niet wordt bepaald door voorwaarden (4) en (6) van de simplex-stap, maar uitsluitend op grond van stabiliteitsoverwegingen. Voor methode b is dit nodig als $i >> m$ mede om geheugen en rekentijd te besparen. Herinversie is ook aan te bevelen als de pivot in absolute waarde kleiner dan een zekere tolerantie is, vgl. Zoutendijk (1960, p.43).Door herinversie kan men bereiken dat de uiteindelijke oplossing even nauwkeurig is als met Gauss-eliminatie met partieel pivoten. Desgewenst kan de oplossing iteratief verbeterd worden.

6.4. Methodes van Bartels

In plaats van B^{-1} (expliciet of in productvorm), kan men de LU-ontbinding van B bijhouden, zie Bartels (1968, 1969a & 1969b).

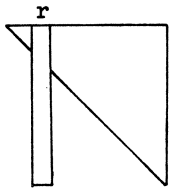
Volgens onderzoekingen van Bartels blijkt deze methode (alsmede een variant waarin een enigszins gewijzigde L en U in product-vorm worden bijgehouden) numeriek stabiel te zijn.

We kunnen B schrijven als : PB = LU, waarin L en U resp. een onder- en een bovendriehoeksmatrix zijn.

Volgens (6.2.2) hebben we

$$B' = B + (A_s - B_r)\, e_r^T .$$

Nu heeft $L^{-1}PB'$ de volgende vorm:



De r^e kolom van U blijkt vervangen te zijn door $v = L^{-1}PA_s$. (Dit hebben we toch nodig om $y = B^{-1}A_s = U^{-1}v$ te berekenen). Nu zetten we in B' de r^e kolom achteraan. We krijgen nu $\tilde{B}'$, waarvoor geldt:

$$P\tilde{B}' = LH, \text{waarin } H = [U_1,\ldots,U_{r-1},U_{r+1},\ldots,U_m,v].$$

H is een zogenaamde <u>boven-Hessenberg-matrix</u>, dat is een matrix, waarvan alleen de bovendriehoek en de aanliggende sub-diagonaal niet nul zijn.
Vervolgens brengen we H in driehoeks-vorm door middel van Gauss-eliminatie met partieel pivoten zodat $\tilde{P}H = \tilde{L}\tilde{U}$.
Wegens de speciale vorm van H gaat dit relatief snel. Het aantal bewerkingen is afhankelijk van r (omdat de eerste r-1 elementen van de subdiagonaal gelijk aan 0 zijn) en gemiddeld $m^2/6$.

We hebben nu:

$$\tilde{B}' = P^{-1}\, L\, \tilde{P}^{-1}\, \tilde{L}\, \tilde{U}$$

en na i stappen:

$$B^i = P_1^{-1}\, L_1 P_2^{-1} L_2 \ldots P_i^{-1} L_i U_i.$$

Deze vorm heet de gewijzigde LU-ontbinding.
Men kan ook $\tilde{B}' = P^{-1}LH$ omvormen tot $(P')^{-1}L'U'$ zodat we dan hebben:

$$P'\tilde{B}' = L'U'.$$

Dit is de expliciete LU-vorm.
Hiervoor blijken gemiddeld $m^3/6$ bewerkingen nodig te zijn.

6.5. Overzicht

Nu volgt een overzicht van de verschillende methodes waarin voor de onderdelen per simplex-stap het aantal bewerkingen wordt opgegeven.

	Klassieke simplex	expliciete inverse	productvorm inverse	expliciete LU	gewijzigde LU
$B^{-1}A$	$m(n-m)$	-	-	-	-
B^{-1} resp.LU	-	m^2	m	$\frac{1}{6}m^3$	$\frac{1}{6}m^2$
$\bar{c}_j = \pi^T A_j - c_j$	$n-m$	$m(n-m)$	$m(n-m)$	$m(n-m)$	$m(n-m)$
$By = A_s$	-	m^2	im	m^2	$m^2+\frac{1}{2}im$
$Bu = b$	m	m	m	m^2	m
$B^T\pi = \gamma$	-	m	im	m^2	$\frac{5}{6}m^2+\frac{1}{2}im$
Totaal = $m(n-m)+$	n	$2m^2+2m$	$2m+2im$	$\frac{1}{6}m^3+3m^2$	$2m^2+im$
stabiel	-	-	-	+	+
schaars	-	-	+	+	+

In de laatste twee regels is aangegeven of de methode al dan niet stabiel is en of de methode, uitgaande van een schaarse matrix, tijdens het proces al dan niet schaarsheid handhaaft.

Opmerking.

Op het Mathematisch Centrum (afdeling Mathematische Besliskunde), is een Lineair-programmeringsprogramma met product-vorm van de inverse en her-inversie ontwikkeld.

7. Lineaire kleinste-kwadratenproblemen

Voor literatuur zie o.a. Businger & Golub (1965), Björck (1967a, 1967b & 1968), Björck & Golub (1967) en Dekker (1968).

7.1. Beschouw het lineaire stelsel

$$Ax = b,$$

waarbij A een reële, m × n matrix, met m > n (d.w.z. aantal vergelijkingen m > aantal onbekenden n) en b een reële m-vector is. In het algemeen is er geen oplossing. Zij de residuvector r gedefinieerd door

$$r = Ax - b. \tag{7.1.1}$$

Het kleinste-kwadratenprobleem (least-squares problem) luidt als volgt: bepaal vector x zó, dat $R \underset{\text{def}}{=} ||r||_2^2 = r^T r$ minimaal is.
Uit (7.1.1) volgt meteen:

$$R = x^T A^T A x - 2b^T A x + b^T b.$$

Voor R minimaal moet gelden: $\frac{\partial R}{\partial x_i} = 0$, i = 1, ..., n, maw.

$$(x^T A^T A)_i + (A^T A x)_i - 2(b^T A)_i = 0, \quad i = 1,\ldots,n.$$

Hieruit volgt direct, dat x gelijk is aan de oplossingsvector van het stelsel

$$A^T A x = A^T b. \tag{7.1.2}$$

Dit is de z.g. normaal-vergelijking.
Het kleinste-kwadratenprobleem is hiermee herleid tot een vierkant stelsel van de orde n, waarvan de matrix $A^T A$ symmetrisch en positief (semi-) definiet is.

Opmerkingen

1) Als A complex is, dan luidt de normaalvergelijking voor de kleinste-kwadratenoplossing

$$A^*Ax = A^*b.$$

De afleiding is evenwel iets anders.

2) Vaak worden aan de vergelijkingen verschillende gewichten $w_1, \ldots, w_m$ toegekend, m.a.w. de te minimalizeren functie is dan

$$||Wr||,$$

waarbij $W = \text{diag}(w_1, \ldots, w_m)$, dat is de diagonaalmatrix met $w_1, \ldots, w_m$ op de hoofddiagonaal.
Dit probleem kan tot bovengenoemd probleem herleid worden door A en b voor te vermenigvuldigen met W.

3) In een kleinste-kwadratenprobleem kunnen extra voorwaarden voorkomen, zoals ongelijkheden en gelijkheden, waaraan niet in de zin van kleinste kwadraten, maar exact moet worden voldaan.
Lineaire ongelijkheidsvoorwaarden leiden tot kwadratische-programmerings-problemen, waarop we niet nader ingaan.
Op het toevoegen van lineaire gelijkheidsvoorwaarden komen we nog terug. (zie 7.4).

7.2. Slechte conditie normaal-vergelijking

De vorm van de normaalvergelijking, maakt op het eerste gezicht het gebruik van de Gauss-eliminatie-methode aantrekkelijk. Immers, wegens het positief definiet zijn van A^TA kan rij- of kolomverwisseling achterwege blijven en kunnen de pivots op de hoofddiagonaal gekozen worden. Nog beter is de hierna beschreven wortel-methode (square-root method) van Cholesky, die de symmetrie bewaart, en daardoor 2 × sneller is (aantal bewerkingen ca. $\frac{1}{6}n^3$ i.p.v. $\frac{1}{3}n^3$ voor het algemene geval).
Zij $M = A^TA$.
Bereken $M = U^TU$, waarbij U een bovendriehoek is, met:

$$U_{ii} = \sqrt{M_{ii} - \sum_{k=1}^{i-1} U_{ki}^2}, \quad i = 1, \ldots, n.$$

Merk op, dat $U_{ii} > 0$, als M positief definiet is.

Verder:

$$U_{ij} = (M_{ij} - \sum_{k=1}^{i-1} U_{ki}U_{kj})/U_{ii},\ i < j.$$

Omdat M positief definiet is, voldoet de groeifactor g (zie 5.3.5) bij de methode van Cholesky aan

$$g \leq \max_{i,j} |A_{ij}|.$$

Deze methode is in de praktijk helaas niet bruikbaar vanwege de slechte conditie van A^TA voor grote n. Het onheil geschiedt niet bij het oplossen van de normaal-vergelijking, maar bij het vormen van A^TA. Voor een vierkante matrix A geldt nl.

(7.2.1) $$\text{cond}_2(A^TA) = [\text{cond}_2(A)]^2.$$

Bewijs:

$$\text{cond}_2(A^TA) = ||A^TA||_2\ ||(A^TA)^{-1}||_2 = \frac{\lambda_{max}(A^TA)}{\lambda_{min}(A^TA)},$$

terwijl

$$\text{cond}_2(A) = ||A||_2\ ||A^{-1}||_2 = \frac{\sqrt{\lambda_{max}(A^TA)}}{\sqrt{\lambda_{min}(A^TA)}}$$

(zie 2.2).

Deze laatste formule kunnen we uitbreiden tot rechthoekige A door de volgende definitie

(7.2.2) $$\text{cond}_a(A) = \sqrt{\frac{\lambda_{max}(A^TA)}{\lambda_{min}(A^TA)}}$$

als rang (A) = n, anders $\text{cond}_2(A) = \infty$.

Algemeen hebben we dan dus:

$$\text{cond}_2(A^TA) = [\text{cond}_2(A)]^2.$$

Hieruit volgt dus, dat de normaalvergelijking aanmerkelijk slechter geconditioneerd is dan het gegeven stelsel.
Om een bruikbare oplossing te vinden, moet men vaak overgaan op dubbelelengte rekenprecisie, d.w.z. zowel de berekening van A^TA als de oplossing van de normaalvergelijking met behulp van Cholesky geschiedt in dubbelelengte. Een andere doelmatige aanpak is orthogonalisatie in enkele-lengte precisie. Daarbij wordt vorming van A^TA, en dus ook kwadratering van het conditiegetal vermeden.

7.3. Orthogonalisatie

We schrijven de m × n-matrix A als produkt van een orthogonale m × n-matrix Q en een bovendriehoeksmatrix R van de orde n:

$$A = QR.$$

Hoe we dit realiseren, zal later blijken.
De normaalvergelijking gaat over in:

$$R^TQ^TQRx = R^TQ^Tb = R^Ts,$$

als $s = Q^Tb$.
Omdat $Q^TQ = I_n$, hebben we

$$R^TRx = R^Ts.$$

Als we aannemen dat rang (A) = n, dan is R niet-singulier. We mogen dus voorvermenigvuldigen met $(R^T)^{-1}$ en krijgen dan het equivalente stelsel:

(7.3.1) $$Rx = s.$$

Omdat R bovendriehoeks is, is dit stelsel eenvoudig door terugsubstitutie op te lossen.
Nu geldt $\text{cond}_2(A^TA) = \text{cond}_2(R^TR) = [\text{cond}_2(R)]^2$. M.a.w. we hebben de normaalvergelijking (7.1.2) herleid tot het veel beter geconditioneerde stelsel (7.3.1), dat zonder moeite kan worden opgelost.

Klassieke Gram-Schmidt orthogonalisatie

Laat A_k de k-de kolom van A zijn voor k = 1, ..., n.
Dan is:

$$A_1 = Q_1 R_{11},\ R_{11} = ||A_1||_2 = \sqrt{\sum A_{i1}^2},$$

$$A_2 = Q_1 R_{12} + Q_2 R_{22}.$$

Voorvermenigvuldiging met Q_1^T levert

$$Q_1^T A_2 = Q_1^T Q_1 R_{12} + Q_1^T Q_2 R_{22} = R_{12},$$

want $Q_1^T Q_1 = 1$ en $Q_1^T Q_2 = 0$.
Vervolgens wordt berekend

$$\tilde{Q}_2 = A_2 - Q_1 R_{12},\ R_{22} = ||\tilde{Q}_2||_2.$$

Algemener, voor de k-de stap, k = 1, ..., n, hebben we

$$A_k = Q_1 R_{1k} + \ldots + Q_k R_{kk}.$$

Voorvermenigvuldiging met Q_i^T levert

$$R_{ik} = Q_i^T A_k,\ i = 1, \ldots, k-1,$$

dus

$$(7.3.2) \quad \begin{cases} \tilde{Q}_k = A_k - Q_1 R_{1k} - \ldots - Q_{k-1} R_{k-1k}, \\ R_{kk} = ||\tilde{Q}_k||_2,\ Q_k = \frac{1}{R_{kk}} \tilde{Q}_k. \end{cases}$$

Als rang (A) = n, dan $R_{kk} \neq 0$ en na n stappen heeft men de ontbinding A = QR verkregen. Helaas is dit proces numeriek niet stabiel. Het volgende, hiermee mathematisch equivalente, proces is dat wel.

Gewijzigde Gram-Schmidt orthogonalisatie

Zie Björck, (1967a).

Zij

$$A^1 = A.$$

We berekenen A_j^2 voor $j = 2, \ldots, n$, volgens

$$A_j^2 = A_j^1 - Q_1 R_{1j}, \text{ waarbij } R_{1j} = Q_1^T A_j^1.$$

Algemeen, voor $k = 1, \ldots, n-1$, berekenen we A_j^{k+1} voor $j > k$ volgens

$$A_j^{k+1} = A_j^k - Q_k R_{kj}, \text{ waarbij } R_{kj} = Q_k^T A_j^k.$$

M.a.w. de volledige formule voor de k-de gewijzigde Gram-Schmidt-stap luidt voor $k = 1, \ldots, n$:

$$(7.3.3) \qquad \begin{cases} R_{kk} = \|A_k^k\|_2 , \quad Q_k = \dfrac{1}{R_{kk}} A_k^k , \\ \\ R_{kj} = Q_k^T A_j^k, \; A_j^{k+1} = A_j^k - Q_k R_{kj}, \; j = k+1, \ldots, n. \end{cases}$$

Ga na, dat (7.3.3) mathematisch equivalent is met (7.3.2).
Bij berekening m.b.v. een rekenautomaat wordt Q_k geschreven over A_k^k en A_j^{k+1} over A_j^k voor $j > k$, zodat vóór de k-de stap in het geheugen staat de matrix

$$A^k \underset{\text{def}}{=} (Q_1, \ldots, Q_{k-1}, A_k^k, \ldots, A_n^k),$$

met $A^1 = A$ en $A^{n+1} = Q$.

Het rechterlid $b = b^1$ wordt op analoge wijze getransformeerd volgens de formule, voor $k = 1, \ldots, n$:

$$(7.3.4) \qquad s_k = Q_k^T b^k, \; b^{k+1} = b^k - Q_k s_k.$$

Dus $Q_b^T = s$ evenals in de klassieke Gram-Schmidt (ga dit na) en $Q^T b^{n+1} = \underline{0}$.

Voorbeeld van P. Läuchli, zie Björck (1967a, pag.4), dat het voordeel van de gewijzigde boven de klassieke vorm aantoont:

$$A = \begin{pmatrix} 1 & 1 & 1 \\ \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{pmatrix} ,$$

waarbij $\varepsilon^2 + 1 \approx 1$ in de machine-precisie.
Dan

$$Q_1 = \begin{pmatrix} 1 \\ \varepsilon \\ 0 \\ 0 \end{pmatrix} , \quad Q_2 = \begin{pmatrix} 0 \\ -1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{pmatrix} , \quad Q_3 = \begin{pmatrix} 0 \\ -1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{pmatrix} ,$$

$$Q_1^T A_2 = Q_1^T A_3 = 1, \quad Q_2^T A_3 = 0.$$

Dus

$$Q_1^T Q_2 = -\frac{\varepsilon}{\sqrt{2}} , \quad Q_1^T Q_3 = -\frac{\varepsilon}{\sqrt{2}} ,$$

echter:

$$Q_2^T Q_3 = \frac{1}{2},$$

d.w.z. Q is verre van orthogonaal.
Nu volgens de gewijzigde Gram-Schmidt:

Q_1 en Q_2 als boven.

$$A_3^2 = \begin{pmatrix} 0 \\ -\varepsilon \\ 0 \\ \varepsilon \end{pmatrix} , \quad Q_2^T A_3^2 = \frac{\varepsilon}{\sqrt{2}} = R_{23} ,$$

$$A_3^3 = \begin{pmatrix} 0 \\ -\varepsilon/2 \\ -\varepsilon/2 \\ \varepsilon \end{pmatrix}, \quad Q_3 = \frac{1}{\sqrt{6}} \begin{pmatrix} 0 \\ -1 \\ -1 \\ 2 \end{pmatrix} .$$

Dan $Q_1^T Q_2 = -\frac{\varepsilon}{\sqrt{2}}$, $Q_1^T Q_3 = -\frac{\varepsilon}{\sqrt{6}}$, $Q_2^T Q_3 = 0$.

D.w.z. Q is redelijk orthogonaal.

Björck (1967a) heeft een fouten-analyse uitgevoerd en bewezen dat de gewijzigde Gram-Schmidt orthogonalisatie stabiel is.

In het bijzonder geldt volgens hem

$$||A - QR|| \approx \beta^{-t}||A||,$$

$$||b - Qs - b^{n+1}|| \approx \beta^{-t}||b||.$$

Opmerking. De gewijzigde Gram-Schmidt orthogonalisatie kan geschieden met kolomverwisseling, d.w.z. in de k-de stap wordt j zo gekozen dat $||A_j^k||$ maximaal is en daarna worden j^e en k^e kolom verwisseld. Dit helpt voor het bepalen van de rang van A, doch maakt voor niet-singuliere A vrijwel geen verschil wat betreft numerieke stabiliteit.

Householder-orthogonalisatie

De matrix A wordt hier geschreven als $A = (Q,\tilde{Q})\begin{pmatrix} R \\ 0 \end{pmatrix}$, waarbij $(Q,\tilde{Q})$ een vierkante orthogonale matrix van de orde m is. R is weer een bovendriehoeksmatrix van orde n en 0 de (m-n) × n nulmatrix. Q en R hebben hier dezelfde betekenis als in beide Gram-Schmidt processen.
Het blijkt, dat $\tilde{Q}$ zo gekozen kan worden, dat

$$(Q,\tilde{Q}) = P_1 \ \ldots \ P_n,$$

waarbij

$$P_k = I - 2u_k u_k^T, \ k = 1, \ \ldots, \ n.$$

Hierbij is u_k een kolom-vector met Euclidische norm 1, dus

$$P_k^T P_k = I - 2u_k u_k^T - 2u_k u_k^T + 4u_k u_k^T u_k u_k^T = I,$$

m.a.w. P_k is een symmetrische orthogonale matrix (vaak Householder-matrix genoemd).
We stellen

$$\tilde{A}^1 = A, \quad \tilde{A}^{k+1} = P_k \tilde{A}^k, \quad k = 1, \ldots, n.$$

Hierbij wordt P_k zo gekozen, dat de eerste k-1 elementen van u_k nul zijn en in de k-de kolom van $\tilde{A}^{k+1}$ de elementen onder de hoofddiagonaal nul worden. Daar de eerste k-1 elementen van u_k nul zijn, zijn de eerste k-1 kolommen van $\tilde{A}^{k+1}$ gelijk aan die van $\tilde{A}^k$, zodat we na n stappen krijgen

$$\tilde{A}^{n+1} = \begin{pmatrix} R \\ 0 \end{pmatrix}.$$

Hieruit volgt

$$\begin{pmatrix} R \\ 0 \end{pmatrix} = \tilde{A}^{n+1} = P_n \ldots P_1 A = (Q, \tilde{Q})^T A = \begin{pmatrix} Q^T A \\ \tilde{Q}^T A \end{pmatrix},$$

m.a.w. $Q^T A = R$, $\tilde{Q}^T A = 0$.
Het rechterlid $b = \tilde{b}^1$ wordt op analoge wijze getransformeerd.
Stellen we

$$\tilde{b}^{k+1} = P_k \tilde{b}^k, \quad k = 1, \ldots, n,$$

dan vinden we

$$\tilde{b}^{n+1} = P_n \ldots P_1 b = (Q, \tilde{Q})^T b = \begin{pmatrix} Q^T b \\ \tilde{Q}^T b \end{pmatrix}.$$

Wegens $Q^T b = s$, mogen we stellen $\tilde{b}^{n+1} = \begin{pmatrix} s \\ \tilde{s} \end{pmatrix}$, waarbij $\tilde{s} = \tilde{Q}^T b$.
We vinden tenslotte de kleinste-kwadratenoplossing x uit

$$Rx = s$$

en het residu is gelijk aan $||\tilde{s}||$.

Opmerkingen. Daar voor niet-singuliere A de ontbinding A = QR met $R_{jj} > 0$ eenduidig is, zijn r, Q en s hier dezelfde als in beide Gram-Schmidt processen. De Householder-orthogonalisatie is dus mathematisch daarmee equivalent. Bovendien is Householder-orthogonalisatie even stabiel als de gewijzigde Gram-Schmidt.
Householder's methode is echter iets sneller en vergt minder ruimte, omdat de vectoren u_k en matrix R op de plaats van A opgebouwd kunnen worden, terwijl bij Gram-Schmidt aparte ruimte (nl. $\frac{1}{2}n(n+1)$) voor R nodig is.

7.4. Iteratief verbeteren

Evenals bij het oplossen van vierkante lineaire stelsels, kan men proberen een gevonden oplossing ($\tilde{x}$) van een kleinste-kwadratenprobleem door iteratie te verbeteren.
Dit kan geschieden na de orthogonalisatie, door de residu-vector in dubbele-lengte-precisie te berekenen, de correctie door heen- en terugsubstitutie te bepalen en $\tilde{x}$ hiermee te corrigeren, Businger & Golub (1965). Dat deze methode evenwel niet effectief is, komt omdat het residu niet naar 0 convergeert. Afronding tot enkele-lengte precisie is in dit geval funest.
Als x de exacte oplossing is, geldt:

$$||x - \tilde{x}|| \leq \varepsilon \text{ cond } (A)(||x|| + 1) + \varepsilon[\text{cond}(A)]^2||r|| + O(\varepsilon^2),$$

zie Golub & Wilkinson (1966).
Een andere aanpak is die van Björck & Golub (1967) en Björck (1967b en 1968). Deze aanpak is ook geschikt voor iets algemenere problemen, waarin de kleinste-kwadratenoplossing ook nog aan een lineair stelsel gelijkheidsvoorwaarden, Cx = d, moet voldoen (vgl. 7.1 opm. 3).
Het probleem luidt dan:
Bepaal min $||r||$ als

$$r = b - Ax \ , \ Cx = d.$$

Met multiplicatoren-methode van Lagrange kan dit probleem worden herleid

tot

$$(7.4.1)\qquad \begin{pmatrix} 0 & 0 & C \\ 0 & I & A \\ C^T & A^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ r \\ x \end{pmatrix} = \begin{pmatrix} d \\ b \\ \underline{0} \end{pmatrix} .$$

Als het stelsel voorwaarden Cx = d ontbreekt (d.w.z. dan hebben we een gewoon kleinste-kwadratenprobleem), gaat dit over in

$$\begin{pmatrix} I & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ \underline{0} \end{pmatrix},$$

m.a.w.

$$r + Ax = b ,$$

$$A^T r = \underline{0},$$

welke laatste vergelijking equivalent is met de normaalvergelijking (7.1.2). Het stelsel (7.4.1) kan m.b.v. Gauss-eliminatie opgelost worden, eventueel gevolgd door iteratief verbeteren. De norm van de residuvector van dit stelsel gaat wel naar 0, zodat de oplossing in vrijwel volle precisie verkregen kan worden mits de conditie niet te slecht is.
Vanwege de speciale gedaante van de matrix van stelsel (7.4.1) kan aanzienlijk bezuinigd worden in geheugenruimte en rekentijd.

8. Eigenwaarden en eigenvectoren

Voor literatuur zie o.a. Faddeev & Faddeeva (1963), Householder (1964), Wilkinson (1965), Dekker & Hoffmann (1968) en Dekker (1969).

8.1. Inleiding

Ter inleiding zullen we enige grotendeels recente stellingen vermelden. Voor de bewijzen wordt de lezer verwezen naar de literatuur. Beschouw een matrix A van de orde n. We zoeken naar de eigenwaarden λ met eigenvectoren $x \neq \underline{0}$ zodat geldt:

$$Ax = \lambda x.$$

Dit stelsel heeft dan en alleen dan een oplossing $x \neq \underline{0}$ als $\det(A-\lambda I) = 0$. Een veel gebruikte methode voor symmetrische zowel als niet-symmetrische matrices is de QR-iteratie van Francis (1961) en Kublanovskaya (1961). Deze verloopt als volgt. Zij

$$(8.1.1) \qquad \left\{ \begin{array}{l} A^{(0)} = A, \\ A^{(i)} = Q^{(i)}R^{(i)} \\ A^{(i+1)} = R^{(i)}Q^{(i)}, \end{array} \right\} \quad i = 0,1,2,\ldots .$$

Hier stelt $Q^{(i)}$ een orthogonale matrix en $R^{(i)}$ een bovendriehoeksmatrix voor. Deze ontbinding is altijd mogelijk (b.v. met Gram-Schmidt, vgl. (7.3), numeriek doet men dit echter anders). Als A niet-singulier is, kan men voorts $R^{(i)}_{jj} > 0$ kiezen en de ontbinding is dan eenduidig.
Uit (8.1.1) volgt meteen $A^{(i+1)} = (Q^{(i)})^{-1} A^{(i)} Q^{(i)}$, d.w.z. $A^{(i+1)}$ is gelijkvormig met $A^{(i)}$ (en dus ook met A). Dan heeft $A^{(i+1)}$ dus dezelfde eigenwaarden als $A^{(i)}$ en als A.
De volgende stelling is afkomstig van Francis (1961), zie ook Wilkinson (1965, p. 517).

(8.1.2) Stelling. Als A een niet-singuliere matrix is, waarvan alle eigenwaarden verschillende modulus hebben, dan convergeert de rij $\{A^{(i)}_{kj}\}_i$ naar 0 voor $k > j$ en naar een eigenwaarde voor $k = j$, terwijl $\{|A^{(i)}_{kj}|\}_i$ naar een constante convergeert voor $k < j$.

(8.1.3) Definitie. Een matrix A heet normaal als $A^*A = AA^*$.
Hierbij stelt A^* de toegevoegd-complexe getransponeerde van A voor.
Eenvoudig is in te zien dat een Hermitische (in het reële geval: symmetrische) matrix normaal is.

(8.1.4) Stelling. Als A normaal is dan is ook $A^{(i)}$ normaal.

Een gevolg van (8.1.2) en (8.1.4) is dat voor een normale, niet-singuliere matrix A, met eigenwaarden van verschillende modulus, $A^{(i)}$ convergeert naar de diagonaalmatrix Λ der eigenwaarden $\lambda_1, \ldots, \lambda_n$.

Als voor zekere $m \geq 0$ geldt $A_{kj}^{(i)} = 0$ voor $k > j+m$, dan volgt daaruit $Q_{kj}^{(i)} = 0$ en dus ook $A_{kj}^{(i+1)} = 0$. Dit is direct in te zien aan de hand van het Gram-Schmidt-orthogonalisatie proces. Hieruit volgt meteen:

(8.1.5) Stelling. Als voor zekere $m \geq 0$ geldt $A_{kj} = 0$ voor $k > j+m$, dan geldt $A_{kj}^{(i)} = 0$ voor $k > j+m$.

Enige toepassingen vindt men in de volgende stellingen.

(8.1.6) Stelling. Als A een Hermitische bandmatrix is met bandbreedte $b = 2m+1$ (m.a.w. $A_{kj} = 0$ voor $|k-j| > m$) dan is $A^{(i)}$ dat ook.

(8.1.7) Definitie. Een matrix A heet (boven-) Hessenberg-matrix of bijna (boven-) driehoeksmatrix als $A_{kj} = 0$ voor $k > j+1$.

Belangrijke toepassingen van (8.1.5) en (8.1.6) zijn:
Als A een reële, symmetrische tridiagonale matrix is (d.w.z. bandbreedte = 3) dan is $A^{(i)}$ dat ook.
Merk op: de bandstructuur van een niet-Hermitische matrix blijft niet bewaard.

(8.1.8) Stelling. Als A een Hessenberg-matrix is, dan is $A^{(i)}$ dat ook.

(8.1.9) Definitie. Een matrix A heet diagonaliseerbaar als er een niet singuliere matrix X en een diagonaalmatrix Λ bestaan zodat $A = X\Lambda X^{-1}$.

(8.1.10) Stelling. Als A een diagonaliseerbare Hessenberg-matrix is, waarvan alle sub-diagonaal-elementen $A_{j+1,j}$ ongelijk aan 0 zijn, dan zijn de eigenwaarden van A enkelvoudig.

De volgende stelling is afkomstig van Parlett (1968).

(8.1.11) Stelling. Als A een Hessenbergmatrix is waarvan alle $A_{j+1,j}$ ongelijk aan 0 zijn dan nadert $A^{(i)}$ tot blokdriehoeksvorm met 1×1 en 2×2 blokjes langs de hoofddiagonaal d.e.s.d. als elke verzameling van eigenwaarden met gelijke modulus hoogstens 2 evenvoudige en 2 onevenvoudige eigenwaarden bevat.

8.2. Convergentie, shift

De hierboven beschreven iteratie convergeert slechts langzaam. Als alle $|\lambda_k|$ verschillend en niet nul zijn dan is de convergentie lineair met de factor λ_n/λ_{n-1}, waarbij λ_n en λ_{n-1} de -in modulus- twee kleinste eigenwaarden voorstellen, d.w.z.

$$A_{nn}^{(i+1)} - \lambda_n \approx \frac{\lambda_n}{\lambda_{n-1}} (A_{nn}^{(i)} - \lambda_n). \tag{8.2.1}$$

Om de convergentie te versnellen wijzigen we het iteratie-proces (8.1.1) als volgt:

$$\left\{ \begin{array}{l} A^{(i)} - \sigma^{(i)}I = Q^{(i)}R^{(i)} \\ A^{(i+1)} \qquad\quad = R^{(i)}Q^{(i)} + \sigma^{(i)}I \end{array} \right\} , \; i = 0,1,2,\ldots . \tag{8.2.2}$$

Hierbij is $\sigma^{(i)}$ een schatting van λ_n en heet shift (verschuiving).
De convergentie-factor wordt nu $(\lambda_n-\sigma^{(i)})/(\lambda_{n-1}-\sigma^{(i)})$, zodat bij goede keuze van $\sigma^{(i)}$ de convergentie veel sneller is.
We merken is dit verband op, dat de eerder genoemde Hessenberg-vorm (in het reële geval: symmetrische tridiagonale vorm) als voordelen heeft:

1) per QR-iteratie-stap is veel minder rekenwerk ($4n^2$ vermenigvuldigingen) vereist;
2) de shift $\sigma^{(i)}$ kan op eenvoudige wijze zo gekozen worden dat $\sigma^{(i)} \to \lambda_n$.

In het navolgende zullen we twee shiftstrategieën bekijken:

1) $\sigma^{(i)} = A_{nn}^{(i)}$;
2) $\sigma^{(i)}$ = dichtst bij $A_{nn}^{(i)}$ gelegen eigenwaarde van

$$M = \begin{pmatrix} A^{(i)}_{n-1n-1} & A^{(i)}_{n-1n} \\ A^{(i)}_{nn-1} & A^{(i)}_{nn} \end{pmatrix} .$$

In de praktijk blijkt deze laatste shift de beste te zijn.
De waarde van deze strategie komt tot uitdrukking in de volgende stelling van Wilkinson (1968):

(8.2.3) Stelling. Als A een reële symmetrische tridiagonale matrix is, dan is de QR-iteratie met shift-strategie (2) convergent, d.w.z. $A^{(i)}_{nn-1} \to 0$, dus $A^{(i)}_{nn}$ convergeert naar een (niet altijd kleinste) eigenwaarde van A.
Bovendien is deze convergentie kwadratisch, vaak zelfs kubisch.
Als A een Hessenberg matrix is, dan leidt strategie (2) in de praktijk bijna altijd tot convergentie en als A diagonaliseerbaar is, dan is de convergentie meestal kwadratisch.

Tegenvoorbeeld. Zij A de volgende permutatiematrix:

$$A = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 1 & & & 0 \\ 0 & & & \vdots \\ 0 & \cdots & 1 & 0 \end{pmatrix} .$$

Dan geldt: $\sigma^{(0)} = 0$ en $A = A^{(0)}$ is unitair, dus $Q^{(0)} = A$ en $R^{(0)} = I$.
m.a.w. $A^{(1)} = A^{(0)}$. Dus $A^{(i)}$ blijft invariant en we vinden geen eigenwaarden.

8.3. Reële matrix met complexe eigenwaarden

In dit geval kan men complex rekenen, maar dit is duur. Aanzienlijk sneller is de volgende "dubbele" QR-methode: Kies in 2 opeenvolgende QR-iteratiestappen beide eigenwaarden van M als shifts. Als $\sigma^{(i)}$ niet reëel is dan geldt dus $\sigma^{(i+1)} = \bar{\sigma}^{(i)}$ en $A^{(i+2)}$ is weer reëel. D.w.z. met reële arithmetiek kan $A^{(i+2)}$ uit $A^{(i)}$ berekend worden, hetgeen aanzienlijk goedkoper blijkt te zijn dan een complexe berekening van $A^{(i+1)}$ en $A^{(i+2)}$.

Ter vergelijking:

Hessenbergmatrix	shifts	aantal vermenigvuldigingen
reëel	reëel	$4n^2$
complex	complex	$16n^2$
reëel	complex	$12n^2$
reëel	toegevoegd complex	$24n^2$ (voor 2 stappen)
reëel	dubbele QR	$5n^2$.

Als $|A_{j+1,j}| < \varepsilon\ ||A||$ voor voldoend kleine ε, dan kan $A_{j+1,j} = 0$ gesteld worden en de matrix worden onderverdeeld:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} .$$

Dit heet deflatie van A.
Aangezien de eigenwaarden van 1×1 en 2×2 blokjes direct kunnen worden berekend, is het proces voltooid als A gebracht is op blokdriehoeksvorm met 1×1 en 2×2 blokjes langs de hoofddiagonaal. Deflatie leidt dus tot een aanzienlijke versnelling van het proces.
Het benodigde aantal iteraties is in de praktijk ongeveer gemiddeld 3 per eigenwaarde. Voor ALGOL 60 procedures, zie o.a. Francis (1961), Dekker & Hoffmann (1968) en Wilkinson, e.a. (1968 & 1970).

8.4. Conditie van eigenwaarde-probleem

De volgende stelling is van Bauer & Fike, zie Wilkinson (1965, p. 87).

(8.4.1) Stelling. Zij A diagonaliseerbaar, zodat $A = X\Lambda X^{-1}$ en μ een eigenwaarde van $A + \delta A$, dan heeft A een eigenwaarde λ_i, zodat

$$|\lambda_i - \mu| \leq ||X||\ ||X^{-1}||\ ||\delta A|| = \mathrm{cond}(X)||\delta A|| .$$

Bewijs: $X^{-1}AX = \mathrm{diag}(\lambda_i)$.
Laat μ een eigenwaarde zijn van $A + \delta A$, dan is de matrix $A + \delta A - \mu I$ singulier. Dan is $X^{-1}(A+\delta A-\mu I)X = X^{-1}(A-\mu I)X + X^{-1}(\delta A)X = \mathrm{diag}(\lambda_i-\mu) + X^{-1}\delta AX$

ook singulier.

Er zijn nu twee mogelijkheden:

1) $\mu = \lambda_i$, dan zijn we klaar;

2) $\mu \neq \lambda_i$, dan hebben we:

$$\text{diag}(\lambda_i-\mu) + X^{-1}(\delta A)X = \text{diag}(\lambda_i-\mu)(I + \text{diag}(\lambda_i-\mu)^{-1}X^{-1}(\delta A)X)$$

is singulier. Voor elke norm die consistent is met de vectornorm geldt:

$$\text{als } I + X \text{ singulier is dan } ||X|| \geq 1.$$

Hieruit volgt

$$|| \text{ diag}(\lambda_i-\mu)^{-1} X^{-1}(\delta A)X|| \geq 1,$$

dus, in het bijzonder voor de spectrale norm:

$$\max_i |(\lambda_i-\mu)^{-1}|||X^{-1}||_2 \, ||\delta A||_2 \, ||X||_2 \geq 1,$$

waaruit volgt:

$$\min_i |\lambda_i-\mu| \leq ||X^{-1}||_2 \, ||\delta A||_2 \, ||X||_2 = \text{cond}_2(X) \, ||\delta A||_2,$$

q.e.d.

Omdat de eigenvectoren-matrix X niet eenduidig is definieren we:

(8.4.2) Definitie. Het spectrale conditie-getal t.o.v. het eigenwaarde-probleem van A is

$$\gamma_2 = \min_{X\Lambda X^{-1}=A} (||X||_2 \, ||X^{-1}||_2),$$

waarbij X een eigenvectoren-matrix van A voorstelt.

Stelling (8.4.1) zegt dus dat er een λ_i van A is, zó dat

$$|\lambda_i-\mu| \leq \gamma_2||\delta A||.$$

Speciale gevallen zijn:

1) A normaal, dan is $\gamma_2 = 1$, dus een wel-geconditioneerd eigenwaardeprobleem;

2) A niet diagonaliseerbaar, dan is $\gamma_2 = \infty$.

We beschouwen nu de conditie van eigenvectoren. Als de eigenwaarden wel-geconditioneerd zijn, dan volgt daar nog niet uit dat de eigenvectoren eveneens wel-geconditioneerd zijn. Als 2 eigenwaarden dicht bij elkaar liggen, dan zijn de eigenvectoren slecht geconditioneerd en in het geval 2 eigenwaarden samenvallen is de richting van de eigenvectoren zelfs onbepaald, zie ook Wilkinson (1965, p.63-72).

(8.4.3) Stelling. Een unitaire transformatie laat het spectrale conditiegetal t.o.v. het eigenwaardeprobleem invariant.

Bewijs: Zij $A = S\,B\,S^{-1}$, waarbij S unitair is.

Daar $A = X \wedge X^{-1}$, geldt $A = S(S^{-1}X \wedge X^{-1}S)S^{-1}$,

dus $B = (S^{-1}X) \wedge (S^{-1}X)^{-1}$,

m.a.w. de eigenvectorenmatrix X gaat over in $S^{-1}X$. Daar S unitair is geldt: $||X||_2 = ||S^{-1}X||_2$, waaruit de stelling onmiddellijk volgt.
Uit deze stelling volgt dat elke unitaire transformatie, en in het bijzonder de QR-iteratie, numeriek stabiel is.

8.5. Householder-transformatie

Literatuur: Householder (1964), Wilkinson (1965, p.290 sqq) en Wilkinson (1960, p.23-27).
De transformatie van Householder voert een matrix over in een bijna-driehoeksmatrix (Hessenbergmatrix). We schrijven $A = A^{(1)}$ als

$$A = S\,H\,S^{-1}, \tag{8.5.1}$$

waarbij H de verlangde (met A gelijkvormige) boven-Hessenbergmatrix is en de orthogonale matrix S het product is van n-1 orthogonale symmetrische Householder-matrices:

$$(8.5.2)\qquad \begin{cases} S = P_1 P_2 \ldots P_{n-1}, \\ P_r = I + k_r u_r u_r^T, \; r = 1, \ldots, n-1, \\ u_r^T = (0, \ldots, 0, u_{r+1,r}, \ldots, u_{nr}), \end{cases}$$

m.a.w de eerste r elementen van u_r zijn 0.
P_r is blijkbaar symmetrisch en de scalar k_r wordt zo gekozen dat P_r orthogonaal is, dus

$$P_r^2 = I + k_r(2 + k_r u_r^T u_r) u_r u_r^T$$

moet gelijk aan I zijn, d.w.z. $k_r u_r^T u_r = 0$ (m.a.w. $P_r = I$) of het niet-triviale geval

$$(8.5.3)\qquad k_r u_r^T u_r = -2.$$

We kiezen de elementen van u_r zó, dat de gewenste nullen ontstaan in de r^e kolom onder de subdiagonaal in

$$(8.5.4)\qquad A^{(r+1)} = P_r A^{(r)} P_r.$$

Beschouw $A^{(r)}P_r = A^{(r)} + k_r A^{(r)} u_r u_r^T = A^{(r)} + p\, u_r^T$ waarbij $p = k_r A^{(r)} u_r$
(merk op dat p een volle vector is).
De eerste r kolommen van $A^{(r)}$ zijn dus invariant bij rechtsvermenigvuldiging met P_r.
$P_r A^{(r)} = A^{(r)} + k_r u_r u_r^T A^{(r)} = A^{(r)} + u_r q^T$, waarbij $q^T = k_r u_r^T A^{(r)}$
(merk op dat $q = (0, \ldots 0, q_r, \ldots, q_n)$), d.w.z. bij linksvermenigvuldiging van $A^{(r)}$ met P_r zijn invariant de eerste r rijen en de eerste (r-1) kolommen.
Samenvattend:
links-vermenigvuldiging: $A_{ir}^{(r+1)} = A_{ir}^{(r)} + u_{ir} q_r$, $i > r$

$$\text{met } q_r = k_r \sum_{i=r+1}^{n} u_{ir} A_{ir}^{(r)};$$

rechts-vermenigvuldiging: $A_{ij}^{(r+1)} = A_{ij}^{(r)} + p_i u_{jr}^T$, $j > r$, $i \leq r$.

De transformatie (8.5.2) geeft dus:

$$\begin{aligned} A^{(r+1)} &= A^{(r)} + u_r q^T + p u_r^T + k_r^2 (u_r^T A^{(r)} u_r) u_r u_r^T \\ &= A^{(r)} + u_r q^T + p u_r^T + k_r \alpha u_r u_r^T, \end{aligned} \tag{8.5.5}$$

waarbij $\alpha = k_r u_r^T A^{(r)} u_r = u_r^T p = q^T u_r$.

Als we verder stellen: $\tilde{q}^T = q^T + k_r \alpha u_r^T = k_r (u_r^T A^{(r)} + \alpha u_r^T)$ dan gaat (8.5.5) over in:

$$A^{(r+1)} = A^{(r)} + u_r \tilde{q}^T + p u_r^T. \tag{8.5.6}$$

Om de gewenste nullen te krijgen moet gelden (met weglating van de bovenindex r):

$$A_{ir} + u_{ir} q_r = 0, \quad i = r+2, \ldots, n.$$

Verder geldt:

$$A_{r+1,r} + u_{r+1,r} q_r = A_{r+1,r}^{(r+1)} = b_r (\text{zeg}).$$

Kwadrateren en sommeren van deze twee vergelijkingen geeft

$$b_r^2 = \sum_{i=r+1}^{n} A_{ir}^2 + 2 q_r \sum_{i=r+1}^{n} u_{ir} A_{ir} + q_r^2 \sum_{i=r+1}^{n} u_{ir}^2.$$

Stellen we $\sum_{i=r+1}^{n} A_{ir}^2 = s$ dan is dit te vereenvoudigen tot:

$$b_r^2 = s + q_r (2 + k_r u_r^T u_r) \sum_{i=r+1}^{n} u_{ir} A_{ir}.$$

Volgens (8.5.3) geldt: $2 + k_r u_r^T u_r = 0$, dus:

$$b_r^2 = s, \text{ m.a.w. } b_r = \pm \sqrt{s}.$$

Indien s = 0, moeten we niet transformeren, m.a.w. P = I kiezen, anders normeren we u zó, dat $q_r = -1$.
Dan hebben we:

$$u_{ir} = A_{ir}, \qquad i = r+2, \ldots, n,$$

$$u_{r+1,r} = A_{r+1,r} - b_r = A_{r+1,r} \pm \sqrt{s},$$

$$-2 = k_r u_r^T u_r = k_r \cdot \sum_{i=r+1}^{n} u_{ir} A_{ir} - k_r u_{r+1,r} b_r$$

$$= q_r - k_r u_{r+1,r} b_r$$

$$= -1 - k_r u_{r+1,r} b_r.$$

Hieruit volgt:

$$k_r u_{r+1,r} b_r = 1, \text{ m.a.w.}$$

$$k_r = 1/(u_{r+1,r} b_r) = -1/(s \pm A_{r+1,r} \sqrt{s}).$$

Om wegvallen van cijfers te voorkomen, kiezen we het teken van $b_r = \pm \sqrt{s}$ tegengesteld aan dat van $A_{r+1,r}$, zodat we krijgen (sign(x) = 1 als $x \geq 0$, anders -1).

$$(8.5.7) \qquad \begin{cases} b_r = -\text{sign}(A_{r+1,r}) \sqrt{s}. \\ u_{r+1,r} = A_{r+1,r} - b_r = A_{r+1,r} + \text{sign}(A_{r+1,r}) \sqrt{s} \\ k_r = 1/(u_{r+1,r} b_r) = -1/(s + |A_{r+1,r}| \sqrt{s}). \end{cases}$$

Hierdoor is de transformatie volledig bepaald.
Het aantal benodigde bewerkingen (vermenigvuldigingen en optellingen of aftrekkingen) voor de Householder-transformatie is voor grote n ongeveer $\frac{5}{3} n^3$.

Terugtransformatie

Laat v een eigenkolom van $H = A^{(n)}$ en x een eigenkolom van $A = A^{(1)}$ zijn, dan geldt:

$$x = P_1 \ldots P_{n-2} v.$$

Zij nu $x_{n-1} = v$, $x_r = P_r\, x_{r+1}$, $r = n-2, \ldots, 1$ dan is $x = x_1$, dus:

$$x_r = P_r\, x_{r+1} = x_{r+1} + k_r(u_r^T x_{r+1})u_r \tag{8.5.8}$$

en evenzo voor de eigenrij w^T van $A^{(n-1)}$ en de eigenrij y^T van $A^{(1)}$

$$y_r^T = y_{r+1}^T P_r = y_{r+1}^T + k_r(y_{r+1}^T u_r)u_r^T. \tag{8.5.9}$$

Symmetrische matrix

Als A een (reele) symmetrische matrix is, zijn alle matrices $A^{(r)}$, wegens het orthogonaal zijn der Householder-matrices P_r, eveneens symmetrisch. De resulterende Hessenberg matrix H is dus ook symmetrisch, m.a.w. H is dan een symmetrische tridiagonale matrix. Het aantal benodigde bewerkingen (vermenigvuldigingen en optellingen of aftrekkingen) voor de Householder-transformatie is nu voor grote n ongeveer $\frac{2}{3}n^3$.

8.6. Berekening van eigenvectoren

We zullen twee methoden behandelen nl. inverse iteratie en berekening mbv. QR-iteratie.

Inverse iteratie

Zij λ een benaderde eigenwaarde en $x^{(0)}$ een vector ongelijk aan de nulvector. Met behulp van Gauss-eliminatie met partieel pivoten lossen we op

$$\left.\begin{aligned} (H-\lambda I)\, y^{(i)} &= x^{(i)} \\ x^{(i+1)} &= y^{(i)}/||y^{(i)}|| \end{aligned}\right\}, \quad i = 0, 1, 2, \ldots . \tag{8.6.1}$$

Als H diagonaliseerbaar is, dan kunnen we schrijven

$$x^{(i)} = \alpha_1 x_1 + \ldots + \alpha_n x_n,$$

waarbij $x_1, \ldots, x_n$ n lineair onafhankelijke eigenvectoren van H zijn horende bij de eigenwaarden $\lambda_1, \ldots, \lambda_n$, dus

$$y^{(i)} = \frac{\alpha_1}{\lambda_1-\lambda} x_1 + \ldots + \frac{\alpha_n}{\lambda_n-\lambda} x_n.$$

Als: 1) λ_1 niet te dicht bij $\lambda_2, \ldots, \lambda_n$ ligt,

2) λ een goede schatting van λ_1 is,

3) α_1 niet ongeveer 0 is,

dan convergeert de rij $\{x^{(i)}\}_i$ naar x_1.
In de praktijk heeft men meestal niet meer dan 2 iteratie-slagen nodig. Wanneer $\lambda_1 \approx \lambda_2$, dan moet men natuurlijk in de iteratie niet twee keer dezelfde benadering λ gebruiken. Door bij de 2e iteratie-slag λ kunstmatig te verstoren met $\delta\lambda$, verkrijgt men als H diagonaliseerbaar is meestal wel lineair onafhankelijke eigenvectoren; zeker is dit echter niet. Als H symmetrisch en reëel is (tridiagonaal), dan is de enige veilige aanpak voor ongeveer gelijke eigenwaarden de methode van expliciete orthogonalisatie (gewijzigde Gram-Schmidt). Voor ALGOL 60 procedures zie Dekker & Hoffmann (1968 , "vecsymtri" voor symmetrische matrices, "reaveches" en "comveches" voor de bepaling van reele resp. complexe eigenvectoren van asymmetrische matrices) en Wilkinson, e.a. (1962).

Eigenvectoren-berekening m.b.v. QR-iteratie

Het resultaat van de QR-iteratie is

$$H = Q\,U\,Q^{-1} \qquad (8.6.2)$$

waarbij Q het product is van alle orthogonale matrices $Q^{(i)}$ van de QR-iteratie en U een blok-(boven)-driehoeksmatrix met 1×1 en 2×2 blokjes op de hoofddiagonaal. Door extra rotaties kunnen 2×2 blokjes met reele eigenwaarden herleid worden tot twee 1×1 blokjes. We nemen aan dat deze extra rotaties in Q opgenomen zijn. Op deze wijze verkrijgen we een matrix U, waarvan alle 2×2 blokjes op de hoofddiagonaal uitsluitend niet-reele eigenwaarden hebben.
We moeten nu de gevallen onderscheiden dat H symmetrisch is en dat H asymmetrisch is.

a) H symmetrisch reëel (en dus tridiagonaal)

In dit geval geldt $U = \Lambda$ is de diagonaal-matrix der eigenwaarden en Q is de (orthogonale) matrix der eiegnvectoren. We moeten dan alleen nog de kolommen van Q terug transformeren.
Volgens (8.5.1) en (8.6.2) geldt:

$$A = S\ H\ S^{-1} = S\ Q\ \Lambda\ (SQ)^{-1},$$

waarin SQ de matrix der eigenvectoren van A is.

b) H asymmetrisch.

Dit geval is te splitsen in:
1) H heeft uitsluitend reele eigenwaarden.
2) H heeft niet uitsluitend reele eigenwaarden.

1) In dit geval is U bovendriehoeks. Als H diagonaliseerbaar is, dan is U dit ook. De eigenvectoren van U zijn dan eenvoudig te berekenen uit:

$$U = T\ \Lambda\ T^{-1},$$

dus

$$H = QT\ \Lambda\ (QT)^{-1},$$

m.a.w. QT is matrix der eigenvectoren van H. Terugtransformatie levert daarna de matrix SQT der eigenvectoren van A.

2) H is asymmetrisch, terwijl niet alle eigenwaarden reëel zijn. Vanwege de blok-driekhoeksvorm van U moeten nu lineaire stelsels van de orde 4, 2 of 1 worden opgelost. Dit geval is daardoor iets gecompliceerder dan (b1), echter practisch even stabiel.

De QR-methode is stabieler dan de inverse-iteratie, terwijl tevens minder geheugenruimte nodig is. Voorts is te bewijzen dat voor een diagonaliseerbare A altijd lineair onafhankelijke eigenvectoren gevonden kunnen worden. Voor ALGOL 60 procedures, zie Dekker & Hoffmann (1968,"reaqri" voor geval b1) , Hoffmann (1970, "comqri" voor geval b2) en Wilkinson, e.a. (1970).

8.7. Equilibratie

(8.7.1) Definitie. Een matrix A heet geëquilibreerd (met betrekking tot zijn eigenwaardeprobleem) als de (Euclidische) norm van elke rij gelijk is aan die van de overeenkomstige kolom.

Uit deze definitie volgt meteen dat een symmetrische matrix en een normale matrix geëquilibreerd zijn.
Als een asymmetrische matrix M sterk afwijkt van de geëquilibreerde vorm, dan blijkt de QR-methode niet zo goed te werken.
Osborne (1960) ontwikkelde een procédé om een matrix M iteratief vrij snel in een nagenoeg geëquilibreerde vorm te brengen. De matrix M wordt d.m.v. een diagonale matrix D = diag (d_i) getransformeerd:

$$M = D\,A\,D^{-1}. \tag{8.7.2}$$

Om M en A exact gelijkvormig te krijgen, moeten we d_i als een macht van β kiezen als β het grondtal is van de gebruikte arithmetiek. (Voor een binaire computer wordt d_i dus als een macht van 2 gekozen.)
Zij $\hat{M}$ = M-diag(M_{ii}) en zij r_i de i^e rij en c_i de i^e kolom van $\hat{M}$ voor $i = 1,\ldots,n$. In elke stap van het equilibratie-proces worden voor zekere i de Euclidische normen $||r_i||_2$ en $||c_i||_2$ berekend.
Als deze allebei niet nul zijn, wordt de normeringsfactor

$$\alpha_i = \sqrt{||r_i||_2 \,/\, ||c_i||_2}$$

bepaald en "afgerond" naar de dichtstbijzijnde macht van β. Vervolgens wordt de rij r_i door deze afgeronde factor gedeeld en de kolom c_i ermee vermenigvuldigd. Is daarentegen $||r_i||_2$ of $||c_i||_2$ gelijk aan nul, dan wordt een verwisseling van rijen en overeenkomstige kolommen uitgevoerd zó dat de nulrij of -kolom de eerste of de laatste van de matrix is, en het equilibratie-proces wordt voortgezet voor het resterende deel van de matrix. Op deze wijze worden alle rijen en kolommen in cyclische volgorde afgehandeld, totdat de matrix bij benadering gelijk is aan een geëquilibreerde matrix.
Het diagonaalelement d_i van D is dan voor elke i gelijk aan het product

van de afgeronde normeringsfactoren α_i.
Na berekening van de eigenvectoren van A moeten deze nog voorvermenigvuldigd worden met D om de eigenvectoren van M te krijgen.
Als matrix M sterk van de geëquilibreerde vorm afwijkt, kan het voorkomen dat de eigenvectoren van M minder goed lineair onafhankelijk zijn dan die van A. Voor een ALGOL 60 procedure zie Dekker & Hoffmann (1968, "eqilbr") of Parlett & Reinsch (1969).

9. Singuliere waarden en numerieke rang van een matrix

Voor literatuur zie Forsythe & Moler (1967), Businger & Golub (1969) en Golub & Reinsch (1970).

(9.1.1) Definitie. Onder de rang van een matrix A verstaan we het aantal lineair onafhankelijk rijen (of kolommen) van A.

(9.1.2) Stelling. Iedere m×n matrix A($m \geq n$) van de rang r is te schrijven als $A = U \sum V^*$, waarin U een m×n matrix is, V^* een n×n matrix en $\sum$ een diagonaalmatrix is, zodat

$$U^*U = V^*V = I_n \text{ (= eenheidsmatrix van de orde n)}$$
$$\text{en } \sum = \text{diag}(\sigma_1,\ldots,\sigma_n) \text{ met}$$
$$\sigma_i > 0 \text{ voor } i \leq r \text{ en } \sigma_i = 0 \text{ voor } i = r+1,\ldots,n.$$

We noemen $\sigma_1,\ldots,\sigma_n$ de singuliere waarden van A.
Bewijs: Stel dat A in deze vorm te schrijven is. Dan geldt:

$$A^*A = V\sum^* U^*U\sum V^* = V|\sum|^2 V^*,$$

dus $|\sigma_i|^2$ zijn de eigenwaarden van A^*A en de kolommen van V de bijbehorende eigenvectoren.
Evenzo geldt: $A\,A^* = U|\sum|^2 U^*$.
Dus $|\sigma_i|^2$ zijn de n grootste eigenwaarden van $A\,A^*$ en de kolommen van U de daarbij behorende eigenvectoren. (Merk op: minstens m-n eigenwaarden van AA^* zijn 0).
Zij nu A een willekeurige m×n matrix ($m \geq n$) van de rang r. We lossen het eigenwaarde-probleem op:

$$A^*AV = V\Lambda, \text{ waarbij}$$

$\Lambda = \text{diag}(\lambda_1,\ldots,\lambda_n)$ met $\lambda_1,\ldots,\lambda_r > 0$ en $\lambda_{r+1} = \ldots = \lambda_n = 0$, terwijl V unitair is van de orde n.

Dan geldt $AA^*AV = AV\Lambda$, dus als $\lambda \neq 0$ en v een eigenvector van A^*A is horende bij λ, dan geldt $Av \neq \underline{0}$. Dit betekent dat de eerste r kolommen van AV, op een factor na, gelijk zijn aan de eerste r eigenvectoren van AA^*, terwijl de andere kolommen van AV gelijk aan $\underline{0}$ zijn.

We kunnen dus een diagonaalmatrix D vinden zó dat $d_i > 0$ voor $i = 1,\ldots,r$ en $d_i = 0$ voor $i = r+1,\ldots,n$ en

$$UD = AV,$$

waarbij U de unitaire matrix der eerste n eigenvectoren van AA^* is. Hieruit volgt $D^*U^*UD = V^*A^*AV$, m.a.w. $|D|^2 = \Lambda$, daar $U^*U = I_n$ en $V^*A^*AV = V^*V\Lambda = \Lambda$.
Hieruit volgt $D = \sum$, m.a.w.: $U\sum = AV$, waaruit meteen blijkt dat $U\sum V^* = A$

q.e.d.

Voor een FORTRAN subroutine ter berekening van $\sum$, U en V voor gegeven complexe matrix A, zie Businger & Golub (1969); voor een ALGOL 60 procedure voor reele matrix A, zie Golub & Reinsch (1970).

(9.1.3) Stelling. Als σ_n de kleinste singuliere waarde van A is, dan is er een E met $||E||_2 = \sigma_n$, waarvoor A + E singulier is.
Bewijs. Wegens (9.1.2) voldoet blijkbaar $E = -U_n\sigma_n(V_n)^*$, q.e.d.

(9.1.4) Stelling. Als A + E singulier is en σ_n de kleinste singuliere waarde van A is dan geldt

$$||E||_2 \geq \sigma_n.$$

Bewijs. Zij B = A + E singulier. Dan bestaat er dus een x met $||x||_2 = 1$ waarvoor geldt $Bx = \underline{0}$.
Wegens $A^*A = (B^*-E^*)(B-E) = B^*B - E^*B - B^*E + E^*E$ geldt voor deze x dan:

$$\sigma_n^2(A) \leq x^*A^*Ax = x^*E^*Ex = ||Ex||_2^2 \leq ||E||_2^2\ ||x||_2^2 = ||E||_2^2$$

q.e.d.

Uit (9.1.3) en (9.1.4) volgt dus dat de kleinste singuliere waarde σ_n gelijk is $\min_{A+E \text{ singulier}} ||E||_2$.

Uit het voorafgaande volgt ook, dat, indien de singuliere waarden $\sigma_1,\ldots,\sigma_n$ van A voldoen aan

$$\sigma_1 \geq \ldots \geq \sigma_r > \varepsilon \geq \sigma_{r+1} \geq \ldots \geq \sigma_n$$

voor zekere $\varepsilon > 0$, dan $r = \min_{||E||_2 \leq \varepsilon} \text{rang } (A+E)$.

(9.1.5) Definitie. Onder de numerieke rang van een matrix A met tolerantie ε (notatie "rank(A,ε)") verstaan we

$$\text{rank}(A,\varepsilon) = \min_{||E||_2 \leq \varepsilon} (\text{rang } (A+E)).$$

Merk op dat de kleinste singuliere waarde van een matrix A veel kleiner kan zijn dan de -in modulus- kleinste eigenwaarde van A.

Voorbeelden.

1) Beschouw:

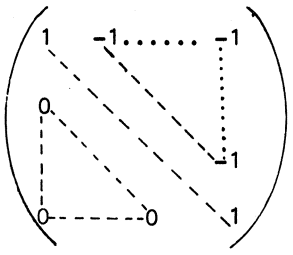
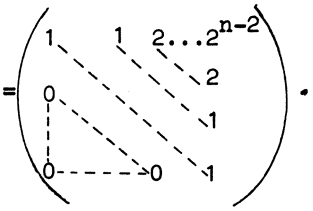
$$A = \begin{pmatrix} \varepsilon & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \varepsilon \end{pmatrix} \qquad \text{dan is} \qquad \begin{pmatrix} \varepsilon & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ (-\varepsilon)^n & & & \varepsilon \end{pmatrix} \qquad \text{singulier,}$$

dus $\sigma_n \leq |\varepsilon|^n$ (tevens $\approx |\varepsilon|^n$) en $\sigma_1 \approx 1$.
De samenhang met het spectrale conditiegetal vinden we uit:

$$\text{cond}_2(A) = \sigma_1/\sigma_n \approx |\varepsilon|^{-n}, \text{ immers}$$

$$A^{-1} = \begin{pmatrix} \varepsilon^{-1} & -\varepsilon^{-2} & \ldots & -(-\varepsilon)^{-n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & -\varepsilon^{-1} \end{pmatrix}, \text{terwijl } \sigma_1 = ||A||_2 \text{ en}$$

$\sigma_n = ||A^{-1}||_2^{-1}$. (zie ook (7.2.2).

2) Als $A =$  dan is $A^{-1} =$  .

D.w.z. $||A||_\infty = n$, terwijl $||A^{-1}||_\infty = 2^{n-1}$, dus $\text{cond}_\infty(A) = n2^{n-1}$. Alle eigenwaarden van A zijn 1, terwijl $\sigma_n \approx 2^{-(n-1)}$.
Als A normaal is geldt blijkbaar

$$\sigma_i(A) = |\lambda_i(A)|, \; i = 1,\ldots,n.$$

Opmerking. Omdat de(numerieke) rang van een matrix discontinu en daarom niet overal berekenbaar is (volgens de theorie van berekenbare getallen) is het begrip singuliere waarde, of eigenlijk de verzameling singuliere waarden $(\sigma_1,\ldots,\sigma_n)$, belangrijker. De singuliere waarden hangen nl. wel continu van de elementen van A af.

Volgens stelling (9.1.2) is iedere m×n matrix A te schrijven in de vorm $A = U\Sigma V^*$. We generaliseren nu het begrip inverse van een matrix.

9.1.6. Definitie. De gegeneraliseerde inverse ($\Sigma^\dagger$) van een diagonaalmatrix $\Sigma = \text{diag}(\sigma_1,\ldots,\sigma_n)$ is de diagonaalmatrix die uit Σ ontstaat door de niet-nul elementen te inverteren; de gegeneraliseerde inverse van een matrix $A = U\Sigma V^*$ (zie stelling (9.1.2)) is de matrix $A^\dagger = V\Sigma^\dagger U^*$.

Eigenschappen van $A^\dagger$:

1) Als A vierkant, niet-singulier is, dan geldt $A^\dagger = A^{-1}$;
2) $A\,A^\dagger A = A$ en $A^\dagger A A^\dagger = A^\dagger$;
3) $(A^\dagger A)^* = A^\dagger A$ en $(AA^\dagger)^* = AA^\dagger$;
4) $A^\dagger$ hangt niet continu van A af: ihb. als $\{A_i\}_i$ een rij vierkante, niet-singuliere matrices is, die naar een singuliere matrix A convergeert, dan convergeert $\{A_i^{-1}\}_i$ niet naar $A^\dagger$;
5) In lineaire kleinste kwadratenproblemen (zie 7) is de oplossing van

$A^TAx = A^Tb$ <u>niet</u> eenduidig als A singulier is. We stellen dan, terwille van de eenduidigheid de extra voorwaarde $||x||_2$ minimaal. Dan geldt $x = A^+b$. Het probleem ligt hier ook weer in de bepaling (keuze) van de rang van A. Dit is vaak een kwestie van smaak of moet op grond van andere overwegingen gemaakt worden, m.a.w. dit is essentieel een taak van de probleemsteller.

<u>Opmerking</u>. Als A Hermitisch, positief semidefiniet is, geldt $\sigma_i = \lambda_i$. Hier vallen dus singuliere waarden en de U, V-matrices samen met het eigensysteem. Dit wordt toegepast in factoranalyse, waar een numerieke rang en de eigenruimte, opgespannen door de niet-verwaarloosbare eigenwaarden gevraagd worden.

10. Literatuur

R.H. Bartels — A numerical investigation of the simplex method, Techn.rep. C.S. 104, Stanford University (1968).

R.H. Bartels — The simplex method of linear programming using LU decomposition, Comm. ACM 12(1969a), p. 266-268.

R.H. Bartels — Simplex method procedure employing LU decomposition, Algorithm 350, Comm. ACM 12(1969b), p. 275-278.

E.M.L. Beale — Mathematical programming in practice, Pitman and sons (1968).

Å. Björck — Solving linear least squares problems by Gram-Schmidt orthogonalization, BIT 7 (1967a), p. 1-21.

Å. Björck & G.H. Golub — Iterative refinements of linear least squares solutions by Householder transformations, BIT 7(1967), p. 322-337.

Å. Björck — Iterative refinement of linear least squares solutions I, BIT 7(1967b), p. 257-278.

Å. Björck — Iterative refinement of linear least squares solutions II, BIT 8(1968), p. 8-30.

P.A. Businger & G.H. Golub — Linear least squares solution by Householder transformations, Num. Mathematik 7(1965), p. 206-216 & 269-276.

P.A. Businger & G.H. Golub — Singular value decomposition of a complex matrix, Algorithm 358 (FORTRAN), Comm. ACM 12(1969), p. 564-565.

G.B. Dantzig — Linear programming and extensions, Princeton Univ. Press (1963).

T.J. Dekker — Algol 60 procedures in numerical algebra, part 1, M.C. Tracts 22, (1968).

T.J. Dekker & W. Hoffmann — Algol 60 procedures in numerical algebra, part 2, M.C. Tracts 23, (1968).

T.J. Dekker — Calculation of eigenvalues and eigenvectors, Informatie 3(1969), p. 118-124.

D.K. Faddeev & V.N. Faddeeva — Computational methods of linear algebra, Freeman and Co. (1963).

G.E. Forsythe & C.B. Moler — Computer solution of linear algebraic systems, Prentice-Hall (1967).

L. Fox — Practical solution of linear equations and inversion of matrices, Nat. Bur. Stand. AMS 39 (ed. O. Taussky), p. 1-54 (1951).

J.G.F. Francis — The QR Transformation, parts 1 and 2, Comp. J. 4(1961), p. 265-271 &332-345.

C.E. Fröberg — Introduction to numerical analysis, Addison-Wesley (1965).

S.J. Gass — Linear programming, Mc Graw-Hill (1968).

W. Givens	Numerical computation of the characteristic values of a real symmetric matrix, ORNL-1574, Oak Ridge National Laboratory (1954).
G.H. Golub & C. Reinsch	Singular value decomposition and least squares solutions, Num. Mathematik 14(1970), p. 403-420.
G.H. Golub & J.H. Wilkinson	Note on the iterative refinement of least squares solution, Num. Mathematik 9(1966), p. 139-148.
W. Hoffmann	Bepaling van eigenwaarden en eigenvectoren door middel van QR-iteratie, (doctoraalscriptie), Amsterdam (1970).
A.S. Householder	Theory of matrices in numerical analysis, Blaisdell Publishing Co. (1964).
V.N. Kublanovskaya	On some algorithms for the solution of the complete eigenvalue problem, Zh. Vych. Mat. 1(1961), p. 555-570.
G. de Leve	Syllabus Mathematische Besliskunde en aanvullingen (1967, 1969), Inst. voor Act. en Ec., Amsterdam.
J. von Neumann & H.H. Goldstine	Numerical inverting of matrices of high order, Bull. AMS 53(1947), p. 1021-1099.
W. Orchard-Hays	Advanced linear-programming computing techniques, Mc Graw-Hill (1968).
E.E. Osborne	On preconditioning of matrices, J. ACM 7(1960), p. 338-354.

B.N. Parlett — Global convergence of the basic QR algorithm on Hessenberg matrices, Math.Comp. 22(1968), p. 803-817.

B.N. Parlett & C. Reinsch — Balancing a matrix for calculation of eigenvalues and eigenvectors, Num. Mathematik 13 (1969), p. 293-304.

J.H. Wilkinson — Householder's method for the solution of the algebraic eigenproblem, The Computer Journal 3 (1960), p. 23-27.

J.H. Wilkinson — Error analysis of direct methods of matrix inversion, J. ACM 8 (1961), p. 281-330.

J.H. Wilkinson — Rounding errors in algebraic processes, Her Majesty's Stationary Office (1963).

J.H. Wilkinson — The algebraic eigenvalue problem, Clarendon-Press (1965).

J.H. Wilkinson — Global convergence of tridiagonal QR algorithm with origin shifts, Lin. Alg. and its applications 1 (1968), p.409-420.

J.H. Wilkinson, H.J. Bowdler, R.S. Martin, G. Peters & C. Reinsch — Handbook series linear algebra, Num. Mathematik, vanaf 4 (1962).

G. Zoutendijk — Methods of feasible directions, (dissertatie), Elsevier (1960).