

Henk van der Vorst

Mathematisch Instituut, Universiteit Utrecht
Postbus 80010, 3508 TA Utrecht
vorst@math.uu.nl

Vakantiecursus 2001

Snel oplossen is een experiment waard

Jaarlijks organiseert het Centrum voor Wiskunde en Informatica (CWI) onder auspiciën van de Nederlandse Vereniging van Wiskundeleraars een vakantiecursus voor wiskundeleraars en andere belangstellenden. Bij die gelegenheid verschijnt steeds een syllabus met teksten bij de voordrachten. Die syllabi zijn ook afzonderlijk bij het CWI verkrijgbaar. Het NAW start een nieuwe serie waarin geselecteerde teksten uit recente syllabi worden geplaatst. Het eerste artikel is afkomstig uit de syllabus bij de Vakantiecursus 2001, die als thema had: 'Experimentele wiskunde'. Het onderwerp is het oplossen van grote stelsels lineaire vergelijkingen. Henk van der Vorst is hoogleraar numerieke wiskunde aan de Universiteit Utrecht. Het internationaal meest geciteerde artikel van de jaren negentig is door hem geschreven; het had hetzelfde onderwerp.

Bij wetenschappelijk rekenwerk wordt de meeste rekentijd verstoekt aan het oplossen van stelsels lineaire vergelijkingen. Die stelsels kunnen behoorlijk groot van omvang zijn, zoals bijvoorbeeld bij stromingsproblemen waarbij elke vergelijking weergeeft hoe de waarde van een lokale (onbekende) grootte (zoals de stroomsnelheid) afhangt van de waarden in de directe omgeving. Men beperkt zich daarbij tot waarden over een van te voren vastgelegd rooster en het aantal roosterpunten bepaalt de omvang van het lineaire stelsel. In figuur 1 zien we zo'n rooster schematisch aangegeven voor het doorrekenen van oceaanstromingen. Bij elk knooppunt hoort een onbekende en een vergelijking waarin deze onbekende gekoppeld wordt aan onbekenden uit de naaste omgeving in het rooster. Omdat we meestal veel roosterpunten nodig hebben om een redelijke nauwkeurigheid te halen, hebben we dus ook veel vergelijkingen. Een prettige bijkomstigheid is wel dat er in elke vergelijking maar een paar onbekenden voorkomen. De matrix van het stelsel staat dus grotendeels vol met nullen. We zullen daar later in dit verhaal nog dankbaar gebruik van maken.

We zien dat het rooster in de figuur uit een viertal verschillende aangegeven deelroosters bestaat. Dat komt omdat dit model in onze Utrechtse groep daadwerkelijk gebruikt is om oceaanstromingen door te rekenen, om te zien hoe onze oplosschema's zich

zouden gedragen. Het probleem was voor een enkele computer echter te groot en daarom werd het rekenprobleem over vier samenwerkende computers verdeeld. De rekenschema's die we zullen introduceren lenen zich redelijk voor een dergelijke parallelle verwerking.

Eliminatie volgens Gauss

Zoals aanstonds duidelijk zal worden leidt het bepalen van de onbekenden uit grote stelsels vergelijkingen tot zeer omvangrijk rekenwerk. De aanpak via directe (Gauss-)eliminatie van onbekenden is vaak onaantrekkelijk, dat zag Gauss in 1823 ook al in, zij het om andere redenen dan de pure rekensnelheid [3]. Hij stelde in dat jaar een iteratieve methode voor om stelsels van vier vergelijkingen met vier onbekenden, die voortkwamen uit driehoeksmetingen, nauwkeurig op te lossen.

Laten we eerst maar eens kijken hoe dat elimineren ook al weer in zijn werk gaat. We bekijken als voorbeeld het stelsel

$$\begin{bmatrix} 10 & 0 & 1 \\ \frac{1}{2} & 7 & 1 \\ 1 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 21 \\ 9 \\ 8 \end{bmatrix}.$$

Het veegproces verloopt als volgt. Men trekt $\frac{1}{20}$ keer de eerste rij van de tweede rij af en daarna nog eens $\frac{1}{10}$ keer de eerste rij van de derde. Daardoor zijn er in de eerste kolom nullen gekomen en ziet het stelsel er als volgt uit

$$\begin{bmatrix} 10 & 0 & 1 \\ 0 & 7 & 1 - \frac{1}{20} \\ 0 & 0 & 6 - \frac{1}{10} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 21 \\ 9 - \frac{21}{20} \\ 8 - \frac{21}{10} \end{bmatrix}.$$

Toevallig staat er nu ook in de tweede kolom een 0 onder het diagonaal element en kan het nieuwe stelsel zonder veel moeite opgelost worden. Dat leidt tot de oplossing $x_3 = 1$, $x_2 = 1$, en $x_1 = 2$. Merk op dat we exact gerekend hebben. Dat is in dit geval niet zo lastig omdat er 'mooie' getallen uitkomen. Echter wanneer er, zoals in meer realistische situaties, getallen met een paar decimalen achter de punt optreden, dan kan het exact rekenen tot zeer gecompliceerde uitdrukkingen aanleiding geven. De

kans op rekenfoutjes is niet denkbeeldig en het is na controle door invullen in de gegeven vergelijkingen lastig om op te sporen waar de rekenfout zit. Gauss had daar ook veelvuldig last van.

Gauss had een behoorlijk fysisch inzicht en wist dat de gewenste oplossingen voor zijn stelsels (voortvloeiend uit driehoeksmeting) componenten van dezelfde grootte-orde hadden. Omdat in zijn stelsels ook de diagonaalelementen sterk overheersten, komt de grootste bijdrage van de oplossing in het rechterlid van de componenten die met een diagonaal element vermenigvuldigd zijn; in ons voorbeeld dus $10x_1$, $7x_2$, en $6x_3$. Dat betekent dat als we in het stelsel de buitendiagonaal coëfficiënten verwaarlozen, dus

$$\begin{bmatrix} 10 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 21 \\ 9 \\ 8 \end{bmatrix},$$

we van het verstoorde stelsel een redelijke benadering voor de oplossing van het ongestoorde stelsel mogen verwachten, in ons geval: $x_1 = 2.1$, $x_2 = \frac{9}{7}$, en $x_3 = \frac{8}{6}$. Dat is inderdaad een, zij het wat grove, benadering voor de gewenste oplossing. Deze vorm van benadering staat tegenwoordig ook wel bekend als de Gauss-Jacobi benadering, omdat de wiskundige-astronoom Jacobi er ook gebruik van maakte bij het berekenen van baanafwijkingen voor de planeten in ons zonnestelsel.

Gauss pakte het nog wat slimmer aan. We kunnen het oorspronkelijke stelsel wat meer intact laten door alleen de coëfficiënten in de rechter bovendrehoek door nullen te vervangen. Dat leidt tot het stelsel

$$\begin{bmatrix} 10 & 0 & 0 \\ \frac{1}{2} & 7 & 0 \\ 1 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 21 \\ 9 \\ 8 \end{bmatrix},$$

en dit stelsel heeft als oplossing $x_1 = 2.1$, $x_2 = \frac{7.95}{7}$, en $x_3 = \frac{5.9}{6}$. Dit is inderdaad nauwkeuriger. Dit hoeft niet altijd te gelden; er zijn situaties waarbij deze aanpak geen verbetering geeft.

Goedkoper en beter?

Al met al hebben we voor ons kleine probleempje tegen een slechts geringe rekenbesparing een beroerdere oplossing gekregen. Voordat we verder gaan, past daarom een beschouwing over de kosten van deze verschillende aanpakken. Voor een stelsel met n vergelijkingen en n onbekenden zijn we bij het vegen van de eerste kolom $2(n-1)^2$ rekenoperaties kwijt (als er geen operaties wegvallen door eventuele nullen die in het stelsel zelf al voorkomen). Daarna, voor de tweede kolom, bedragen de rekenkosten $2(n-2)^2$ operaties. Hieruit volgt dat voor het vegen van de gehele onderdriehoek de rekenkosten ongeveer $\frac{2}{3}n^3$ bedragen. Het oplossen van het resterende bovendrehoekstelsel kost dan nog ruwweg n^2 operaties, en voor grotere n valt dat relatief in het niet. De kosten voor het verkrijgen van de exacte oplossing lopen dus op met de derde macht van n . Men gaat gemakkelijk na dat de benaderingen die met het benedendrehoeksgedeelte van het stelsel berekend worden, verkregen worden tegen kosten die evenredig zijn met de tweede macht van n . Voor grote problemen ligt hier dus een enorm voordeel voor het grijpen.

Echter: hoe nu deze benaderde oplossing goedkoop te verbeteren? Ook daar had Gauss natuurlijk aan gedacht. Om zijn aanpak te beschrijven, gaan we over op matrixnotatie. Deze was in Gauss' tijd nog niet bekend en dat maakte de beschrijving van zijn reken-



Figuur 1 Het rekenrooster voor een oceaanstroming

aanpak wat omslachtig. We schrijven het stelsel als $Ax = b$, met

$$A = \begin{bmatrix} 10 & 0 & 1 \\ \frac{1}{2} & 7 & 1 \\ 1 & 0 & 6 \end{bmatrix},$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \text{en} \quad \begin{bmatrix} 21 \\ 9 \\ 8 \end{bmatrix}.$$

Het benedendrehoeksgedeelte van A noteren we met L :

$$L = \begin{bmatrix} 10 & 0 & 0 \\ \frac{1}{2} & 7 & 0 \\ 1 & 0 & 6 \end{bmatrix}$$

en dan is de benaderde oplossing dus verkregen uit het oplossen van $L\tilde{x} = b$. Voor verbetering van deze oplossing zoeken we naar het ontbrekende deel Δx :

$$A(\tilde{x} + \Delta x) = b,$$

en hiervoor geldt dus

$$A\Delta x = b - A\tilde{x} \equiv r.$$

Het ligt nu voor de hand om Δx ook weer met een Gauss benadering te berekenen, dus we bepalen $\tilde{\Delta x}$ uit

$$L\tilde{\Delta x} = r,$$

en we corrigeren met deze benadering onze eerste schatting \tilde{x} :

$$\tilde{\tilde{x}} = \tilde{x} + \tilde{\Delta x}.$$

We kunnen deze truc natuurlijk blijven herhalen en dat levert ons de volgende iteratieformule op

$$x^{(i+1)} = x^{(i)} + L^{-1}(b - Ax^{(i)}),$$

waarbij de vector $y = L^{-1}(b - Ax^{(i)})$ berekend wordt door

$$Ly = b - Ax^{(i)}$$

op te lossen.

We voeren dit proces nu uit voor ons kleine stelseltje. Bij gebrek aan betere informatie starten we maar met $x^{(0)} = 0$. In Tabel 1 staan de resultaten voor de eerste drie iteratiestappen weergegeven. Zoals we zien, neemt het aantal correcte decimalen met ongeveer twee per stap toe. Dat is natuurlijk niet altijd het geval: het hangt af van de mate waarin de diagonaal in A overheerst.

De rekenkosten per iteratiestap bedragen ruwweg $2n^2$ operaties (optellingen, aftrekkingen, vermenigvuldigingen) voor het berekenen van $Ax^{(i)}$, plus nog eens n^2 voor het oplossen van het benedendriehoekstelsel met L , dus totaal $\approx 3n^2$ operaties per stap. De oplossing via de directe veegmethode vergt $\approx \frac{2}{3}n^3$ operaties, dus als je met het benaderde resultaten na minder dan

$$\left(\frac{2}{3}n^3\right)/(3n^2) = \frac{2}{9}n$$

slagen tevreden bent, dan heb je rekenwinst geboekt.

Voor Gauss had het rekenen met deze iteratiemethode ook enorme voordelen: het was weliswaar voor $n = 4$ niet zo efficiënt in termen van rekenoperaties als het directe oplossen, maar het grote voordeel voor hem was dat je de benaderingen niet nauwkeurig hoeft uit te rekenen. Reken- en afrondfouten corrigeren zich zelf als het ware in het verdere verloop van de iteratie. Aan het residu $b - Ax^{(i)}$, dat je toch moet uitrekenen, kan je in iedere stap zien hoe goed de gevonden benadering bij het gegeven stelsel past. Hij schreef dan ook opgetogen in een brief aan zijn collega Gerling dat je dit half in je slaap kon uitvoeren, danwel dat je onderwijl aan andere zaken kon denken.

Gauss had het geluk dat zijn stelsels sterk diagonaal dominant waren waardoor bij hem het iteratieproces redelijk snel convergeerde, ongeveer zoals in het voorbeeld dat wij bekeken hebben. Bij de zeer grote lineaire stelsels, zoals die bijvoorbeeld in oceaanstromingen optreden, is dat vrijwel nooit het geval en dan convergeert de Gauss-methode akelig langzaam of zelfs helemaal niet. Men heeft dus al heel lang uitgekeken naar snellere alternatieven.

stap	1	2	3
x_1	2.1000	2.0017	2.000028
x_2	1.1357	1.0023	1.000038
x_3	0.9833	0.9997	0.999995

Tabel 1 Resultaten voor drie Gauss-iteraties

Krylov-ruimten

We keren terug naar de basis iteratieformule

$$\begin{aligned} x^{(i+1)} &= x^{(i)} + L^{-1}(b - Ax^{(i)}) \\ &= x^{(i)} + L^{-1}r^{(i)}, \end{aligned} \quad (1)$$

met $r^{(i)} = b - Ax^{(i)}$. We zullen verder, voor het gemak, maar aannemen dat de startvector x_0 gelijk is aan de vector met allemaal nullen.

Door vergelijking (1) te vermenigvuldigen met $-A$ en er links en rechts b bij op te tellen, krijgen we

$$\begin{aligned} b - Ax^{(i+1)} &= b - Ax^{(i)} - AL^{-1}r^{(i)} \\ r^{(i+1)} &= r^{(i)} - AL^{-1}r^{(i)} \\ &= (I - AL^{-1})r^{(i)}. \end{aligned} \quad (2)$$

Vergelijking (2) geeft een relatie tussen twee opeenvolgende residu vectoren en we kunnen deze formule herhaald toepassen. We krijgen dan

$$\begin{aligned} r^{(i+1)} &= (I - AL^{-1})^{i+1}r^{(0)} \\ &= (I - AL^{-1})^{i+1}(b - Ax^{(0)}) \\ &= (I - AL^{-1})^{i+1}b. \end{aligned}$$

Kennelijk kan het residu in de $i + 1$ -ste stap geschreven worden als allerlei machten van AL^{-1} maal de vector b . Merk op dat we nooit, maar dan ook echt nooit, de matrix L^{-1} zelf uitrekenen. Dat zou veel te inefficiënt zijn. Omdat deze matrix steeds voorkomt in een product met een vector, komt het er dus op neer dat we in staat moeten zijn de vector $y = L^{-1}z$ uit te rekenen voor zekere z . Dat gaat op de inmiddels bekende manier, door y op te lossen uit

$$Ly = z.$$

We kijken nu nog eens naar de oorspronkelijke iteratie en we zien dat we deze herhaald kunnen toepassen, zodat we de volgende vorm overhouden

$$\begin{aligned} x^{(i+1)} &= x^{(i)} + L^{-1}r^{(i)} \\ &= x^{(i-1)} + L^{-1}r^{(i-1)} + L^{-1}r^{(i)} \\ &\quad \vdots \\ &= x^{(0)} + \sum_{j=0}^i L^{-1}r^{(j)} \\ &= \sum_{j=0}^i L^{-1}r^{(j)}. \end{aligned} \quad (3)$$

We kunnen dat nog wat handiger herschrijven:

$$\begin{aligned} L^{-1}r^{(j)} &= L^{-1}(I - AL^{-1})^j b \\ &= (I - L^{-1}A)^j L^{-1}b. \end{aligned}$$

Uit vergelijking (3) leren we dat $x^{(i+1)}$ geschreven kan worden als een combinatie van de vectoren

$$L^{-1}b, L^{-1}AL^{-1}b, (L^{-1}A)^2L^{-1}b, \dots, (L^{-1}A)^iL^{-1}b.$$

Zo'n ruimte, die opgespannen wordt door opeenvolgende machten van een matrix (in ons geval $L^{-1}A$), losgelaten op een vaste vector (in ons geval $L^{-1}b$), heet een Krylov-ruimte, naar de Russische wiskundige *A.N. Krylov*. Krylov-maakte voor zijn berekeningen in de dertiger jaren van de vorige eeuw gebruik van deze ruimten, maar het duurde tot rond 1950 voordat dat voor lineaire stelsels een beetje succesvol begon te worden.

Lanczos kwam in 1952 op het idee om te onderzoeken of de Krylov ruimte betere benaderingen bevatte dan de benadering die door het Gauss-proces wordt opgeleverd [5]. Hij merkte op dat het werk dat gedaan moest worden om de basis vectoren te berekenen ongeveer net zo veel was als het werk dat in het Gauss-iteratie proces verstookt moest worden: namelijk voor uitbreiding van de basis is een operatie met A nodig en een keer oplossen met L . De vraag rijst wel meteen of deze basis geschikt is om mee te rekenen en Lanczos had al snel in de gaten dat de 'Krylov'-basisvectoren niet erg orthogonaal waren; erger nog, ze maken na een aantal stappen meestal een zeer kleine hoek met elkaar. Hij kwam daarom op het idee om van meet af aan een orthogonale

basis voor de Krylov-ruimte te construeren. Dat gaat als volgt. De eerste vector is $L^{-1}b$ en deze delen we door zijn lengte:

$$v_1 = L^{-1}b / \|L^{-1}b\|_2.$$

Vervolgens laten we hier de operatie $L^{-1}A$ op los $w_1 = L^{-1}Av_1$. We hebben nu twee vectoren en we trekken van w de projectie op v_1 af

$$z_1 = w_1 - (w_1, v_1)v_1.$$

We normeren dit resultaat en dat levert ons de tweede orthogonale basisvector

$$v_2 = z_1 / \|z_1\|_2.$$

Dit proces zetten we voort. Stel dat we j stappen hebben uitgevoerd, dan hebben we dus de onderling orthogonale vectoren v_1, v_2, \dots, v_j . De volgende vector krijgen we door de operatie $L^{-1}A$ op v_j los te laten en er vervolgens de projecties op alle voorgaande vectoren vanaf te trekken:

$$\begin{aligned} w_j &= L^{-1}Av_j \\ z_j &= w_j - \sum_{k=1}^j (w_j, v_k)v_k \\ v_{j+1} &= z_j / \|z_j\|_2 \end{aligned} \quad (4)$$

De set vectoren v_1, v_2, \dots, v_j vormt een set van onderling loodrechte basisvectoren (met lengte 1) voor de Krylov-ruimte van dimensie j . We moeten nu nog in deze ruimte de beste benadering voor de oplossing bepalen. Dit deed Lanczos op een zeer handige manier. We vatten de basis vectoren op als de kolommen van een matrix:

$$V_j = [v_1 | v_2 | \dots | v_j],$$

dan kunnen we het resultaat van het orthogonalisatie proces (4) schrijven als

$$L^{-1}AV_j = V_{j+1}H_{j+1,j}, \quad (5)$$

met $H_{j+1,j}$ een $j+1$ bij j dimensionale matrix met de volgende elementen

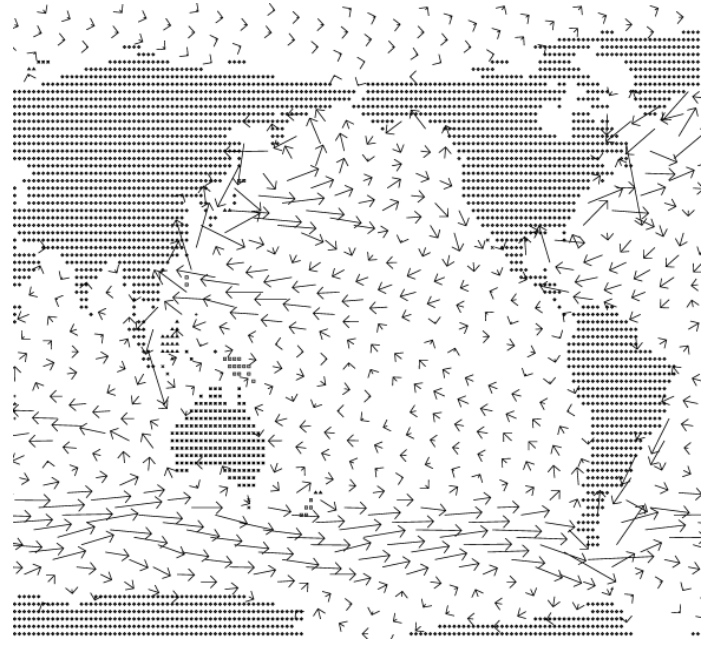
$$H_{j+1,j} = \begin{bmatrix} (w_1, v_1) & (w_2, v_1) & (w_3, v_1) & \dots & (w_j, v_1) \\ \|z_1\|_2 & (w_2, v_2) & (w_3, v_2) & \dots & (w_j, v_2) \\ 0 & \|z_2\|_2 & (w_3, v_3) & \dots & (w_j, v_3) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & (w_j, v_j) \\ 0 & 0 & 0 & 0 & \|z_j\|_2 \end{bmatrix}.$$

Nu merken we op dat elke vector, en dus ook de gezochte benaderde oplossing, in de j dimensionale Krylov-ruimte geschreven kan worden als $x^{(j)} = V_j y$, waarbij y een vector is met j coördinaten. We moeten nu nog aangeven wat we met beste benadering zullen bedoelen. Lanczos vond het prima als het residu dat bij die benadering past loodrecht op de Krylov-ruimte stond. Om die Krylov-ruimte bij het stelsel te laten passen, keek hij naar het residu voor de vergelijking

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

Dan eisen we dus dat

$$L^{-1}b - L^{-1}Ax^{(j)} \perp \{v_1, v_2, \dots, v_j\},$$



Figuur 2 Met behulp van de geconjugeerde gradiëntenmethode kost de berekening van de oceaanstroming slechts enkele seconden rekentijd op een workstation.

ofwel: $L^{-1}b - L^{-1}AV_j y \perp \{v_1, v_2, \dots, v_j\}$.

Door nu keurig de loodrechtheid voor alle vectoren uit te schrijven en gebruik te maken van de observatie dat

$$L^{-1}b = \|L^{-1}b\|_2 v_1$$

en van vergelijking (5), komen we tot het verrassende resultaat dat de vector y oplossing is van het j bij j stelsel

$$\begin{bmatrix} (w_1, v_1) & (w_2, v_1) & (w_3, v_1) & \dots & (w_j, v_1) \\ \|z_1\|_2 & (w_2, v_2) & (w_3, v_2) & \dots & (w_j, v_2) \\ 0 & \|z_2\|_2 & (w_3, v_3) & \dots & (w_j, v_3) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \|z_{j-1}\|_2 & (w_j, v_j) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_j \end{bmatrix} = \begin{bmatrix} \|L^{-1}b\|_2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Met de gevonden oplossing y construeren we de benaderde oplossing x_j simpelweg als $x_j = V_j y$.

Oceaanstromingen

Een aantal opmerkingen is hier op zijn plaats. Lanczos bedacht deze methode oorspronkelijk voor het geval dat $L^{-1}A$ symmetrisch, of liever symmetriseerbaar, was. In dat geval kunnen er enorme vereenvoudigingen worden aangebracht vanwege het feit dat dan $(w_k, v_j) = 0$ voor alle $j < k - 1$. Dat betekent een grote winst in efficiency. De resulterende methode is in zijn uiteindelijke vorm gepresenteerd in 1954 door Hestenes en Stiefel, onder de naam *geconjugeerde gradiënten* methode [4]. Pas veel later werd de niet-symmetrische variant populair. De meest bekende en gebruikte variant voor niet-symmetrische matrices is *GMRES*, in 1986 voorgesteld door Saad en Schultz [6].

```

Bereken  $r_0 = b - Ax_0$  voor start vector  $x_0$ 
for  $i = 1, 2, \dots$ 
  Los op  $z_{i-1}$  uit  $Kz_{i-1} = r_{i-1}$ 
   $\rho_{i-1} = r_{i-1}^* z_{i-1}$ 
  if  $i = 1$ 
     $p_1 = z_0$ 
  else
     $\beta_{i-1} = \frac{\rho_{i-1}}{\rho_{i-2}}$ ;
     $p_i = z_{i-1} + \beta_{i-1} p_{i-1}$ 
  endif
   $q_i = A p_i$ 
   $\alpha_i = \frac{\rho_{i-1}}{p_i^* q_i}$ 
   $x_i = x_{i-1} + \alpha_i p_i$ 
   $r_i = r_{i-1} - \alpha_i q_i$ 
  stop indien residu klein genoeg
end;
```

Figuur 3 De geconjugeerde gradiënten methode

Laten we nog eens op een rijtje zetten wat we gedaan hebben:

- We zijn uitgegaan van de simpele Gauss-iteratie (ook wel Gauss-Seidel-iteratie genoemd).
- Die iteratie levert benaderingen in Krylov-ruimten op.
- Met vrijwel net zoveel werk kan ook meteen een orthogonale basis voor die Krylov-ruimten geconstrueerd worden.
- De constructie van de orthogonale basis levert meteen de coëfficiënten van een klein stelsel op.
- Oplossing van dat kleine stelsel leidt tot een *optimale* benadering van de oplossing in de Krylov-ruimte van de gegeven dimensie.

Dit lijkt tot ingewikkelde code aanleiding te zullen geven, maar we zien in figuur 3 dat dit voor de geconjugeerde gradiënten methode reusachtig meevalt.

Is deze aanpak nu werkelijk zoveel goedkoper dan directe oplosmethoden of dan de oorspronkelijke Gauss-methode? Dat hangt af van het aantal iteratieslagen dat nodig is. Dat aantal hangt weer af van de gekozen basis iteratie. In plaats van L kan je natuurlijk ook andere delen K van de matrix A afsplitsen, zolang dat maar een goedkoop te berekenen benadering oplevert. Dat gebeurt tegenwoordig vaak met onvolledig uitgevoerde veegpro-

cessen. Hoe onvolledig je dat nog mag doen om toch een efficiënte methode te krijgen is niet bekend, het hangt op een onbekende manier van het probleem af. Dit aspect vereist veel ervaring en veel geëxperimenteer, maar het levert dan ook vaak groot succes op. De oceaanstromingen kunnen meer dan een factor n sneller opgelost worden in vergelijking met slim uitgevoerde veegmethoden (die gebruik maken van de vele nullen in het stelsel). Bij NASA rekende men ooit uit dat de krachtverdeling, die optreedt bij de lancering van de ruimteshuttle, met het Gauss-veegproces meer dan 520,000 jaar rekentijd op een supercomputer zou vereisen. Met de standaard Gauss-iteratie treedt vrijwel geen convergentie op en die is dus onbruikbaar, maar de geconjugeerde gradiënten methode, in combinatie met een onvolledig veegproces, vroeg slechts een tiental minuten rekentijd.

Voor het oceaanprobleem uit figuur 1 kon het stromingsprofiel uiteindelijk in luttele seconden op een normaal werkstation berekend worden: de grootte van de pijlen in figuur 2 geeft de sterkte van de stroming aan.

De zogenaamde Krylov-iteratiemethoden worden op uiteenlopende gebieden met succes toegepast: oceaanstromingen, olie-reservoir simulatie, elektrische circuits, mechanische constructies, tunnelboringen, vliegtuigontwerp, medische toepassingen, etc.

Ondanks het feit dat de methode steeds toegesneden moet worden op een bepaalde klasse van problemen bieden ze zoveel snelheidswinst dat men niet meer zonder kan. Er vindt nog steeds zeer veel onderzoek plaats op dit gebied. Er worden hele congressen gewijd aan het ontwerp van bruikbare afsplitsingen K van de matrix van het stelsel, en de publikaties op dit gebied behoren tot de meest geciteerde artikelen uit de wiskunde. Een artikel over een efficiënte Krylov methode voor niet-symmetrische stelsels (in feite een combinatie van het geconjugeerde gradiënten idee met de GMRES methode) was het meest geciteerde wiskundeartikel uit de jaren negentig [7].

Voor een goed overzicht van iteratieve methode en verdere verwijzingen zie het zogenaamde *Templates* boek [1].

Dankwoord

De redactie dankt Miente Bakker van het CWI voor zijn medewerking aan de totstandkoming van deze serie. De syllabi behorende bij de vakantiecurcussussen zijn te vinden op internetpagina [2].

Referenties

- 1 R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, PA, 1994.
- 2 www.cwi.nl/static/publications.bibl/cwi.syllabi.html
- 3 C. F. Gauss. *Werke, Band IX*. Teubner, Leipzig, 1903.
- 4 M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.*, 49:409–436, 1954.
- 5 C. Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Natl. Bur. Stand*, 49:33–53, 1952.
- 6 Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.
- 7 H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 13:631–644, 1992.