Ongepreconditioneerd Bi-CG en QMR

Arthur van Dam

3 juni 1999

Inleiding

Dit verslag geeft een beschrijving van het afsluitend practicum bij het vak Numerieke Wiskunde III. Hierin zal worden besproken hoe algemene stelsels $Ax = b$ kunnen worden opgelost met (ongepreconditioneerd) Bi-CG. Ter voorkoming van de mogelijke *breakdowns* hierbij, zal ook een alternatieve methode, QMR worden bestudeerd.

1 Lanczos, of: de basis

Een eenvoudige oplosmethode voor stelsels $Ax = b$, is de zogenaamde *Richardson-iteratie*:

$$x_{i+1} = b + (I - A)x_i = x_i + r_i$$

Dit afgetrokken van de werkelijk oplossing x en uitwerken van de ontstane recursie levert een uitdrukking voor de fout:

$$x - x_{i+1} = (I - A)^{i+1}(x - x_0) = P_{i+1}(A)(x - x_0) \tag{1}$$

Als we $x_0 = 0$ kiezen, levert de Richardson-iteratie elementen $x_{i+1} \in \{r_0, Ar_0, \ldots, A^i r_0\}$, oftewel elementen uit de Krylov deelruimte $\mathcal{K}_{i+1}(A; r_0)$.

Om snel een goede oplossing te verkrijgen, willen we de x_{i+1} zó opstellen, dat $\|x_{i+1} - x\|$ minimaal is. Dit is bijvoorbeeld mogelijk door de nieuwe residu-vector r_{i+1} loodrecht te zetten op de voorgaande. Hierbij wordt het inprodukt $(x, y)_A \equiv (x, Ay)$ gebruikt, zodat de gegenereerde residu-vectoren $r_0 \ldots r_i$ een orthogonale basis vormen voor de Krylov deelruimte $\mathcal{K}_{i+1}(A; r_0)$:

$$r_{i+1} \perp \mathcal{K}_{i+1}(A; r_0)$$

Een efficiente oplosmethode zou dus een orthogonale basis kunnen genereren voor de Krylov deelruimte en de iteranden x_i hierop kunnen afbeelden.
Zoals bewezen wordt in [5] voldoen in het symmetrische geval de r_i aan de recurrente betrekking:

$$\alpha_{i+1} r_{i+1} = Ar_i - \beta_i r_i - \gamma_i r_{i-1}, \tag{2}$$

waarin

$$\beta_i = (r_i, Ar_i)/(r_i, r_i) \quad \text{en} \quad \gamma_i = (r_{i-1}, Ar_{i-1})/(r_{i-1}, r_{i-1}) \quad \text{en} \quad \alpha_{i+1} + \beta_i + \gamma_i = 0$$

Hierbij bepaalt α_i de juiste grootte van r_i. Wanneer we de vectoren $r_0 \dots r_{i-1}$ als kolommen in een matrix R_i zetten, is (2) te schrijven als

$$AR_i = R_{i+1} \begin{pmatrix} \ddots & \ddots & & & \\ \ddots & \ddots & \gamma_j & & \\ & \ddots & \beta_j & \ddots & \\ & & \alpha_{j+1} & \ddots & \ddots \\ & & & \ddots & \ddots \\ & & & & \ddots \end{pmatrix} = R_{i+1} T_{i+1,i} \tag{3}$$

Hierin is $T_{i+1,i}$ een tridiagonaalmatrix met een extra $i+1$-e rij.
Aangezien $x_i \in \mathcal{K}_i(A; r_0)$ en dus een combinatie van de basisvectoren is, geldt: $x_i = R_i y$. Tijdens het iteratieproces nemen we als nieuwe basisvector $r_i = b - Ax_i \perp \mathcal{K}_i(A; r_0)$ en vanwege deze orthogonaliteit geldt:

$$\begin{aligned} R_i^T \{b - AR_i y\} &= 0 \\ \|r_0\| e_1 - R_i^T R_{i+1} T_{i+1,i} y &= 0 \end{aligned} \tag{4}$$

Aangezien r_i orthogonaal is ten opzichte van de kolommen van R_i, bevat de laatste kolom in $R_i^T R_{i+1}$ louter nullen, zodat deze kolom en de laatste ($i+1$-e) rij van $T_{i+1,i}$ genegeerd kan worden:

$$\|r_0\| e_1 - R_i^T R_i T_{i,i} y = 0 \quad \Longrightarrow \quad T_{i,i} y = e_1 \quad \Longrightarrow \quad y \quad \Longrightarrow \quad x_i = R_i y \tag{5}$$

Hiermee is weer een nieuwe iterand x_i bepaald. Afhankelijk van het huidige residu, kan worden gekozen of er al dan niet verder geïtereerd wordt.
Deze oplosmethode heet de Lanczos methode.

2 Oplossen van niet-symmetrische stelsels: Bi-CG

Zoals gezegd, werkt de methode van Lanczos alleen voor symmetrische stelsels. Wanneer we een asymmetrische A hebben, is de boven-Hessenbergmatrix $T_{i+1,i}$ niet meer tridiagonaal en moeten nieuwe basisvectoren op alle voorgaande loodrecht gezet worden. Om toch eenvoudig een orthogonale basis te creëren met een eenvoudige recurrente betrekking, gebruiken we orthogonaliteit van de basisvectoren op de zogenaamde *schaduw Krylov deelruimten* $\mathcal{K}_i(A^T; s_0)$, we noemen dit *bi-orthogonaliteit* ([4]).
We gebruiken nu de twee recurrente betrekkingen:

$$\alpha_{i+1} r_{i+1} = Ar_i - \beta_i r_i - \gamma_i r_{i-1}, \tag{6}$$

$$\alpha_{i+1} s_{i+1} = A^T s_i - \beta_i s_i - \gamma_i s_{i-1}. \tag{7}$$

De keuze van s_0 is vrij; vaak wordt genomen $s_0 = r_0$ of s_0 willekeurig.
Vanwege de bi-orthogonaliteit geldt: $r_j \perp \{s_0, \dots, s_{j-1}\}$, ofwel: $(r_j, s_k) = 0$ voor $j \neq k$ en dus kunnen we weer op de gebruikelijke manier verder. In plaats van (4) schrijven we nu :

$$S_i^T \{b - AR_i y\} = 0 \quad \Longrightarrow S_i^T R_i T_{i,i} y = S_i^T b \tag{8}$$

Hierbij is $S_i^T R_i$ een diagonaalmatrix met diagonaalelementen (r_j, s_j). Omdat we ook hadden gekozen $r_0 = b$, geldt weer: $T_{i,i}y = e_1$ zodat net als in (5) x_i kan worden bepaald.

Als een diagonaalelement van $S_i^T R_i$ (bijna) nul wordt (als $(r_j, s_j) = 0$), dan wordt er door (bijna) nul gedeeld. Dit noemen we een *breakdown eerste soort*. Dergelijke breakdowns kunnen voorkomen worden door in het algoritme *look-ahead* toe te passen: er wordt een stap vooruit gekeken of er een deling door nul zou plaats vinden. Is dit het geval, dan wordt nog voor die nieuwe stap herstart.

De zojuist beschreven methode heet Bi-Lanczos. Een kleine aanpassing leidt tot Bi-CG: In [5] wordt voor Lanczos bewezen dat het niet nodig is om alle vectoren r_i van de gegenereerde basis op te slaan, omdat nieuwe r_i en x_i uit r_{i-1} en x_{i-1} te bepalen zijn. Dit is mogelijk omdat $T_{i,i}$ zonder pivoting LU-gedecomposeerd kan worden. Datzelfde kunnen we op Bi-Lanczos toepassen. Deze LU-decompositie brengt echter wel extra risico's met zich mee: als $T_{i,i}$ singulier is, treedt tijdens de LU-decompositie een breakdown op.

3 Voorkomen van breakdowns: QMR

Een andere techniek, die breakdowns omzeilt is QMR([3],[5]). QMR lijkt veel op Bi-CG, er wordt weer uitgegaan van (3) met een klein verschil: ten behoeve van de stabiliteit worden de basisvectoren r_j en s_j genormaliseerd, zodat de factoren α_j, β_j en γ_j ook veranderen: $\bar{T}_{i+1,i}$. Daarnaast wordt op$\bar{T}_{i+1,i}$ nu geen LU- maar QR-decompositie toegepast, waardoor die breakdowns worden vermeden.

Het uitgangspunt is echter nog wel steeds het minimaliseren van het residu:

$$\begin{aligned} \|Ax_i - b\|_2 &= \|AR_i\bar{y} - b\|_2 = \|R_{i+1}\bar{T}_{i+1,i}\bar{y} - b\|_2 \\ &= \|R_{i+1}D_{i+1}^{-1}\{D_{i+1}\bar{T}_{i+1,i} - \|r_0\|_2 e_1\}\|_2 \end{aligned} \tag{9}$$

De orthogonaliteit tussen de kolommen van R_{i+1} gaat hier echter niet meer op. De laatste kolom van $\bar{T}_{i+1,i}$ valt dus niet weg en – belangrijker nog – het bepalen van het minimum van (9) zou veel arbeid kosten. In QMR wordt dan ook niet dit residu geminmaliseerd, maar:

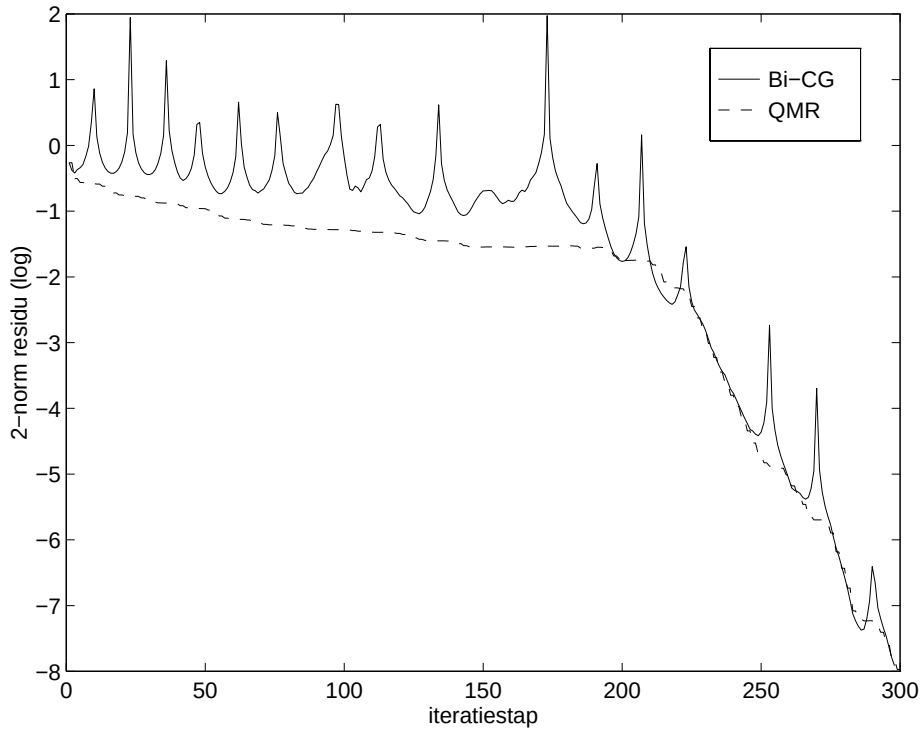
$$\min_{y \in \mathbb{R}^i} \|D_{i+1}\bar{T}_{i+1,i}y - \|r_0\|_2 e_1\|_2. \tag{10}$$

Omdat hier niet de echte residuen worden geminimaliseerd, wordt gesproken van *Quasi Minimale Residuen (QMR)*.

4 Numerieke experimenten

Alle experimenten zijn uitgevoerd op een Sun SPARC5-station met 32 MB RAM. De algoritmen zijn geimplementeerd in C++ en zijn kleine aanpassingen van, en toevoegingen op een bestaand programma uit [2]. De implementatie van Bi-CG en QMR is afkomstig van `www.netlib.org/templates` ([1]) en zijn gebaseerd op de algoritmen uit het *SIAM Templates book*.

Om de twee methoden te vergelijken, gebruiken we ijle matrices, zodat de experimenten niet al te duur zijn. De stelsels zijn afkomstig uit een groter programma, waarin de stroming van grondwater en de gifconcentratie werd gemodelleerd, op een gediscretiseerd rechthoekig



Figuur 1: Convergentie bij een asymmetrische, ijle, reele matrix met reeele eigenwaarden

gebied ([2]). Hierin hangen waterdruk of gifconcentratie in een roosterpunt alleen van de directe buren af, waardoor de marix die deze samenhang beschrijft, ijl is. Dit grondwaterprobleem wordt gemodelleerd door de partiële differentiaalvergelijking:

$$-\nabla \cdot (B\nabla\psi) + \nabla \cdot (V\psi) + c\psi = f, \tag{11}$$

Gediscretiseerd, levert dit probleem een stelsel $Ax = b$, waarin B de symmetrische component van A bepaalt en V de antisymetrische component.

4.1 Verband tussen Bi-CG en QMR

Voor een asymmetrisch stelsel $(A-200I)x = b$ met A en b uit (11), waarbij $B = (1+y, 1+x)^T$, $V = (x, y)^T$, $c = 4$ en $f = 8xy$ in (11), gediscretiseerd op een 30×30-rooster (A is een 900×900-matrix), is in figuur 1 voor zowel Bi-CG als QMR $\log \|r_i\|_2/\|r_0\|_2$ tegen de iteratiestap geplot. Het aftrekken $-200I$ is een toepassing van een vederop beschreven techniek om de convergentie te beinvloeden door verschuiving van de eigenwaarden. Duidelijk is hier het effect van QMR te zien: in de punten waar Bi-CG 'opblaast', blijft QMR constant of dalend. In de meeste gevallen kan het verband tussen QMR en Bi-CG als volgt worden beschreven:

$$r_j^{QMR} \approx \sum_{k=1}^{j} (\sigma_k r_k^{Bi-CG}) \quad \text{met} \quad \sum_{k=1}^{j} \sigma_k = 1$$

In woorden: QMR volgt zo'n beetje de tendens van Bi-CG. Het verband blijft een beetje vaag, maar bij QMR worden immers hele andere (quasi!) residuen bepaald dan bij Bi-CG.

Wat echter wel opvalt, is dat beide methoden uiteindelijk vrijwel evenveel iteraties hebben doorlopen. Toch zal het echte residu $\|Ax_k - b\|$ bij Bi-CG uiteindelijk nog groter zijn dan bij QMR; na de *near breakdowns* is het benaderde residu wel weer gaan dalen, maar de oplossing x_i heeft bij iedere piek wel een verstoring meegekregen. Ook zorgt het zeer grote verschil tussen de opeenvolgende residuen voor verlies van significante cijfers. In de praktijk wordt vaak een maximale verschilfactor van $\mathcal{O}(1/\sqrt{\mu})$ toegestaan, met μ gelijk aan de machineprecisie. Het aantal en de hoogte van de pieken geeft een indicatie van de totale verstoring.

Om een sterkere *near-breakdown* af te dwingen,vervangen we A door $A - \sigma I$ en laten σ variëren. De reden waarom dit werkt is het feit dat in Bi-CG de fout beschreven wordt door een polynoom $P(A)$ zoals in (1). Dit polynoom moet geminimaliseerd worden over A, wat kleiner dan of gelijk is aan het minimum over de eigenwaarden λ_j van A:

$$\min_{\lambda_j} \|P(\lambda)\| \|r_0\|$$

Wanneer de eigenwaarden rondom de oorsprong liggen, is het minimaliseren moeilijk, omdat het polynoom in de oorsprong ongelijk nul is. Dit polynoom wordt genormaliseerd in de oorsprong en wanneer het polynoom oorspronkelijk bijna door de oorsprong liep, kan bij het herschalen het polynoom behoorlijk opblazen. Door nu σI van A af te trekken. worden de eigenwaarden verschoven; zo kan de minimalisering moeilijker of makkelijker worden gemaakt. In figuur 1 was $\sigma = 200$, in figuur 2 is dezelfde A gebruikt, maar nu met $\sigma = 2000$. Bi-CG verloopt nu een stuk grilliger. De hoogte van de uitschieters is weliswaar niet zo heel veel hoger, maar wel zijn ze veel frequenter. QMR blijft zich over het algemeen heel constant gedragen: echte uitschieters zijn niet waarneembaar.

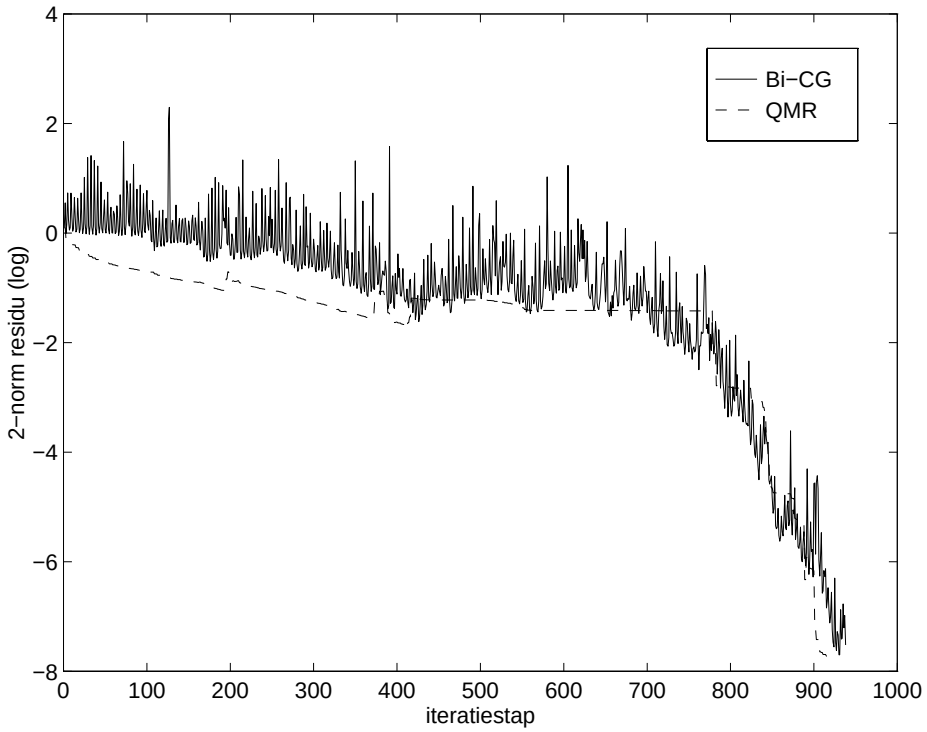
4.2 Complexe eigenwaarden

In het voorgaande waren de eigenwaarden van A allen reëel, laten we nu eens reële matrices met complexe eigenwaarden bekijken. Dergelijke matrices zijn te construeren door bij een symmetrische matrix een (met voldoende grote voorfactor) asymetrische matrix op te tellen. Preciezer: $B = (5, 5)^T$, $V = (10, 10)^T$, $c = 0$ en $f = 0$ in (11), weer met een 30×30-discretisatie. Ook hierop hebben we Bi-CG en QMR losgelaten, de resultaten staan in figuur 3.

Ook hierbij vertoont Bi-CG een grillig verloop en zijn er behoorlijk veel stappen nodig. QMR heeft zoals altijd ongeveer evenveel stappen nodig, maar vertoont nu ook wat vreemde, vrijwel horizontale trajecten. Om de convergentie een beetje te 'helpen' passen we de beschreven 'sigmatruuk' toe met $\sigma = 0.05$. Het rsultaat staat in figuur 4. Het verschil met figuur 3 is duidelijk waarneembaar: Bi-CG is al minder grillig en vooral QMR verloopt weer regelmatig. Voor nog grotere σ werd de convergentie steeds sneller, tot slechts drie iteratiestappen bij $\sigma = 100$. Dit is op zich niet zo verwonderlijk: de matrix is diagonaal dominant geworden, dus de eerste iteranden staan vrijwel meteen voor alle coordinaten heel goed.

4.3 Verschil tussen Bi-CG en QMR; what's in a row?

Zoals al besproken in sectie 2 en 3 werkt Bi-CG met $T_{i,i}$, terwijl bij QMR de $i + 1$-e rij van $T_{i+1,i}$ niet wegvalt. Tijdens de experimenten is niet echt duidelijk naar voren gekomen, dat deze extra rij de convergentie ook echt beïnvloedt.



Figuur 2: Convergentie bij $A - 2000I$

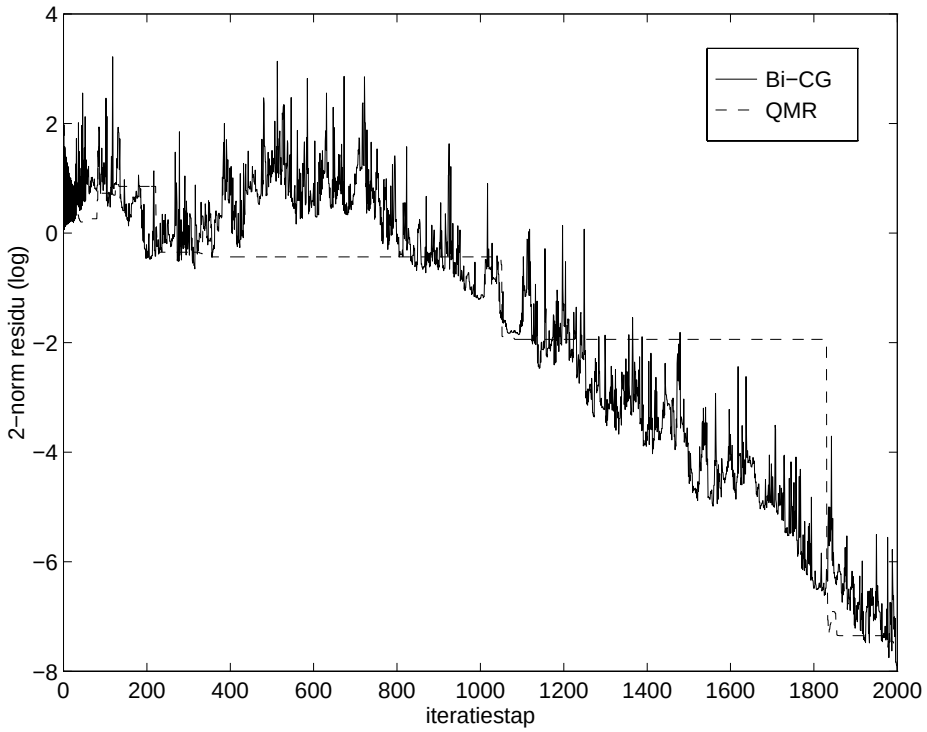
Wat nog wel vermeld kan worden is dat naarmate de iteratie vordert, de eigenwaarden van $T_{i+1,i}$ op die van A gaan lijken. In het exacte geval:

$$\begin{aligned} AV &= VT \\ A &= VTV^{-1} \\ Ax = \lambda x \quad &\Longrightarrow \quad VTV^{-1}x = \lambda x \\ T(V^{-1}x) &= \lambda(V^{-1}x) \end{aligned}$$

De eigenvectoren veranderen dus, maar de eigenwaarden van A en T zijn gelijk. Met een wat uitgebreider bewijs is aan te tonen dat tijdens deze numerieke experimenten de eigenwaarden van T steeds meer op die van A gaan lijken. Wanneer deze rond nul liggen, krijgen zowel Bi-CG als QMR problemen met de convergentie, zoals besproken in sectie 4.1.

5 Conclusie

Uit de experimenten is gebleken dat de aanpak die Bi-CG volgt, goed werkt: de residuen worden met een eenvoudige recursie bepaald en door de orthogonaliteit worden de residuen verkleind. Door de goedkope stappen is de convergentie vrij snel. Natuurlijk was dit al lang bekend uit de vele theorie en experimenten die reeds gedaan zijn. Wel verrassend was het, dat QMR de breakdowns zo mooi vermijdt en verder Bi-CG redelijk volgt. Er wordt natuurlijk wel van hetzelfde principe uitgegaan, alleen worden de (quasi)residuen anders bepaald. Dit

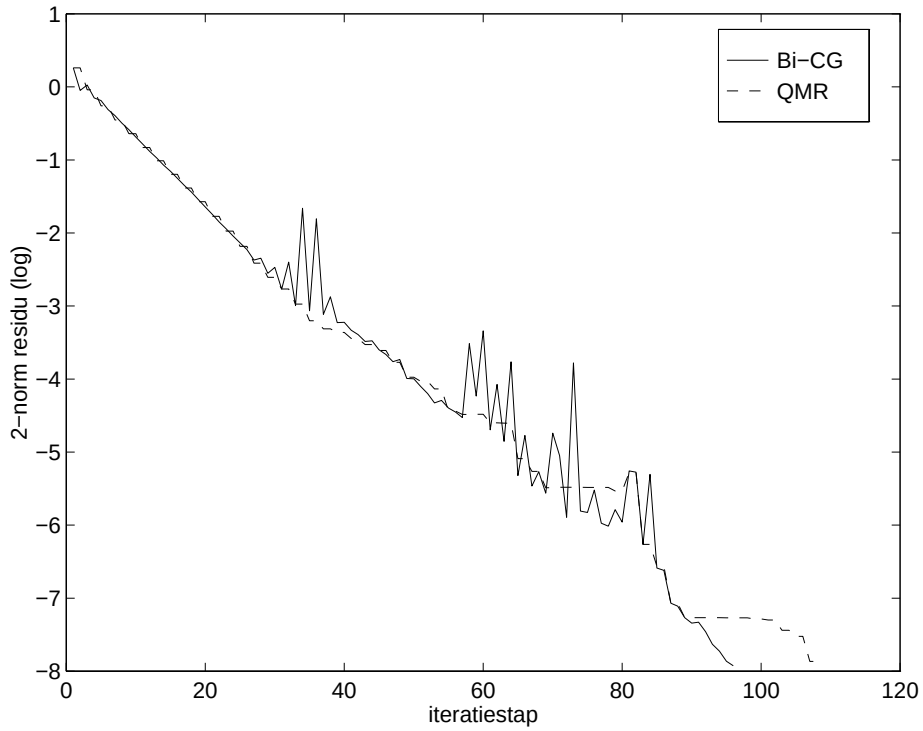


Figuur 3: Convergentie bij reële matrix met complexe eigenwaarden

blijkt ook wel uit het resultaat dat uiteindelijk QMR en Bi-CG vrijwel evenveel iteratiestappen hebben doorlopen.

Toch moet nog worden opgepast dat bij QMR geen andere breakdowns optreden. Hiervoor zou de look-ahead strategie kunnen worden toegevoegd, zoals beschreven in [3].

QMR levert, in tegenstelling tot Bi-CG, geen problemen op als $T_{i+1,i}$ nullen op de diagonaal bevat. Wel zijn beide methoden gevoelig voor eigenwaarden rond de oorsprong, zeker wanneer deze ook nog complex zijn.



Figuur 4: Convergentie bij matrix uit figuur 3 met $\sigma = 0.05$

Referenties

[1] R. Barrett, M. Berry, T. Chan, J.Demmel, J. Donato, J.Dongarra, V. Eijkhout, R. Pozo, C.Romine, en H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. `http://www.netlib.org/templates/`

[2] A. van Dam. Verslag CS Practicum 2-I, Iteratieve Lineaire Oplosmethoden, April 1999.

[3] R.W. Freund, N.M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Num. Math.*, 60:315-339, 1991.

[4] G.L.G. Sleijpen, Computational Science Praktikum*; Iteratieve Lineaire Oplosmethoden*, Universiteit Utrecht, Wiskundig Instituut, Utrecht, 3de herziene druk februari 1998.

[5] H.A. van der Vorst. Lecture notes on iterative methods. Universiteit Utrecht, Wiskundig Instituut, Utrecht, June 4, 1994.