WISB356, Utrecht, 2 oktober 2012

Scientific Computing

Gerard Sleijpen
Rob Bisseling
Alessandro Sbrizzi

Universiteit Utrecht
Department of Mathematics

http://www.staff.science.uu.nl/∼sleij101/

WISB356, Utrecht, 2 oktober 2012

Iterative methoden voor lineaire vergelijkingen

Gerard Sleijpen

Universiteit Utrecht
Department of Mathematics

http://www.staff.science.uu.nl/∼sleij101/

Program

- Basale Lineaire Algebra Operaties
- LU-decompositie
- Blas versus LU
- Iteratieve oplosmethoden
- Local Minimal Residuals
- Generalized Conjugate Residuals
- Convergentie
- Krylov ruimte methoden

Basale Lineaire Algebra operatieS

Basale **L**ineaire **A**lgebra operatie**S**:

- **MV**: $\mathbf{c} = \mathbf{A}\mathbf{u}$
- **AXPY** of **vector update**: $\mathbf{z} = \alpha\mathbf{x} + \mathbf{y}$
- **DOT** of **inproduct**: $(\mathbf{y}, \mathbf{z}) = \mathbf{z}^*\mathbf{y} = \mathbf{z} \bullet \mathbf{y}$

BLAS operaties kunnen snel uitgevoerd worden.

Computerfabrikanten zorgen ervoor dat BLAS software voor hun machine beschikbaar met een efficiëntie die voor hun machine geoptimaliseerd is.

Opmerking. De optimale MV is ook afhankelijk van de structuur van de matrix.

Basale Lineaire Algebra operatieS

BLAS operaties kunnen snel uitgevoerd worden.

We meten de kosten voor een operatie in
floating point operaties.

Floating point. $7.9487e{-}4 \quad = 794.87e{-}2$

Een **flop** is een scalaire operatie,
vermenigvuldiging $*$, deling $/$, optelling $+$, of aftrekking $-$.

In de numerieke lineaire algebra wordt vrijwel altijd
'n $+$ (of $-$) gecombineerd met 'n $*$.

Voorbeelden. 1 AXPY: $2n$ flop, 1 DOT: $2n$ flop
1 MV voor onze matrix op een 2-d gebied: $9n$ flop
3-d gebied: $13n$ flop

Gauss eliminatie (LU-decompositie)

LU-decompositie $\mathbf{A} = \mathbf{LU}$, waarbij

$\mathbf{L}$ beneden driehoek met $\text{diag}(\mathbf{L}) = \mathbf{I}$ en
$\mathbf{U}$ bovendriehoek.

De collectie (i,j) met $\mathbf{A}_{i,j} = 0$,
en $\mathbf{L}_{i,j} \neq 0$ (als $i < j$) of $\mathbf{U}_{i,j} \neq 0$ (als $i \geq j$) heet **fill**.

Gauss eliminatie (of LU-decompositie) is inefficiënt
als $\mathbf{A}$ **ijl** is (d.w.z., per rij maar een paar $\mathbf{A}_{i,j} \neq 0$)
en er veel fill optreedt.

Gauss eliminatie (LU-decompositie)

Voor Grondwaterstroming/verspreiding van gif

Methode		Geheugen (in 8 bytes)	Werk (in flop)
MV	2d	$5(n_x n_y)$	$9(n_x n_y)$
Gauss	2d	$\approx 2n_x(n_x n_y)$	$2n_x^2(n_x n_y)$
MV	3d	$7(n_x n_y n_z)$	$13(n_x n_y n_z)$
Gauss	3d	$\approx 2(n_x n_y)(n_x n_y n_z)$	$2(n_x n_y)^2(n_x n_y n_z)$

Gauss. $+$ Maar één methode
$+$ Geeft exact antwoord
$-$ **fill**: (duur mbt geheugen, rekenkosten)

Iteratief. $-$ Methode per probleemtype te kiezen
$-$ Benaderend antwoord
$+$ **alleen MV's, axpy's, dot's**

Gauss eliminatie (LU-decompositie)

Voor Grondwaterstroming/verspreiding van gif

Methode		Geheugen (in 8 bytes)	Werk (in flop)
MV	2d	$5(n_x n_y)$	$9(n_x n_y)$
Gauss	2d	$\approx 2n_x(n_x n_y)$	$2n_x^2(n_x n_y)$
MV	3d	$7(n_x n_y n_z)$	$13(n_x n_y n_z)$
Gauss	3d	$\approx 2(n_x n_y)(n_x n_y n_z)$	$2(n_x n_y)^2(n_x n_y n_z)$

Gauss. $+$ Maar één methode (andere nummering als $n_y < n_x$?)
$+$ Geeft exact antwoord (je maakt ook afrondfouten!)
$-$ **fill**: (duur mbt geheugen, rekenkosten)

Iteratief. $-$ Methode per probleemtype te kiezen
$-$ Benaderend antwoord (Kan heel nauwkeurig!)
$+$ **alleen MV's, axpy's, dot's**

Gauss eliminatie (LU-decompositie)

Voor Grondwaterstroming/verspreiding van gif

Methode		Geheugen (in 8 bytes)	Werk (in flop)
MV	2d	$5(n_x n_y)$	$9(n_x n_y)$
Gauss	2d	$\approx 2n_x(n_x n_y)$	$2n_x^2(n_x n_y)$
MV	3d	$7(n_x n_y n_z)$	$13(n_x n_y n_z)$
Gauss	3d	$\approx 2(n_x n_y)(n_x n_y n_z)$	$2(n_x n_y)^2(n_x n_y n_z)$

$\mathbf{A}$ is $n \times n$.

Vuistregel: (beste) Gauss is sneller dan (beste) iteratief

in 2 d als $n < 60\,000$ $\quad n_x = n_y = 240$

in 3 d als $n < 40\,000$ $\quad n_x = n_y = 65,\ n_z = 10$

Vuistregel geldt voor "algemene" praktische problemen

Iteratief

Observatie. We zijn niet echt geïnteresseerd in de **exacte** oplossing $\mathbf{x}$. We zijn al tevreden als de berekende oplossing goed is in pakweg 5 decimalen.

Idee. Als we een **benaderende** oplossing $\mathbf{x}_k$ hebben, construeer dan een beter benaderende oplossing uit $\mathbf{x}_k$ (en eventueel uit $\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \ldots$ of andere vectoren die we al berekend hebben). Herhaal dit proces tot dat

de **fout** $\mathbf{e}_k \equiv \mathbf{x} - \mathbf{x}_k$ klein genoeg is.

Terminologie. **Residu:** $\mathbf{r}_k \equiv \mathbf{A}(\mathbf{x} - \mathbf{x}_k) = \mathbf{b} - \mathbf{A}\mathbf{x}_k$.

Meet de fout of residu in de (geschaalde) 2-norm:

$$\|\mathbf{r}\|_2 \equiv \sqrt{(\mathbf{r}, \mathbf{r})} \text{ met } (\mathbf{y}, \mathbf{z}) = \sum_{j=1}^{n} \tfrac{1}{n} \overline{z_j} y_j = \tfrac{1}{n} \mathbf{z}^* \mathbf{y}.$$

Iteratief

Kies α $\qquad \mathbf{x}_{\text{opl}} = \mathbf{x}_{\text{opl}} + \alpha(\mathbf{b} - \mathbf{A}\mathbf{x}_{\text{opl}})$

Probleem van de vorm $\mathbf{x} = \phi(\mathbf{x})$.

Probeer op te lossen met successieve substitutie:

$$\mathbf{x}_{k+1} = \phi(\mathbf{x}_k) = \mathbf{x}_k + \alpha\,\mathbf{r}_k$$

$$\mathbf{e}_{k+1} = \mathbf{e}_k - \alpha\,\mathbf{r}_k = (\mathbf{I} - \alpha\mathbf{A})\mathbf{e}_k.$$

Voorbeeld. $n = 1$, $\mathbf{A} = [\lambda]$.

Dan $\qquad \mathbf{e}_{k+1} = (1 - \alpha\,\lambda)\,\mathbf{e}_k = (1 - \alpha\,\lambda)^{k+1}\mathbf{e}_0 \to 0$

$$\Leftrightarrow \qquad |1 - \alpha\,\lambda| < 1.$$

Iteratief

Richardson iteration

```
Choose tol > 0, x_0, k_max, α
Compute r_0 = b − A x_0
For k = 0, 1, 2, ..., k_max
   Stop if ‖r_k‖_2 ≤ tol ‖b‖_2
   u_k = r_k
   c_k = A u_k
   Determine α
   x_{k+1} = x_k + α u_k
   r_{k+1} = r_k − α c_k
end for
```

Iteratief

Stelling. Richardson convergeert

als $\quad \alpha\,\lambda \in \{\zeta \in \mathbb{C} \mid |1-\zeta| < 1\} \qquad (*)$
voor alle eigenwaarden λ van $\mathbf{A}$

Stelling. Als $\mathrm{Re}(\lambda) > 0$ voor alle eigenwaarden van $\mathbf{A}$, dan is er een $\alpha > 0$ zodat $(*)$ [*Bewijs: zie animatie*]

Voorbeeld. $\mathrm{Re}(\lambda) > 0$ voor alle eigw. van $\mathbf{A}$:
1) Grondwaterstroming (geschikte discretisatie)
2) Verspreiding gif, $c > 0$, en h_x, h_y voldoende klein

Idee. Kies α zo dat $\|\mathbf{r}_k - \alpha\,\mathbf{c}_k\|_2$ minimaal.

Stelling

$$\begin{aligned}
&\alpha = \mathrm{argmin}_{\tilde\alpha} \|\mathbf{r}_k - \tilde\alpha \mathbf{c}_k\|_2 && \Leftrightarrow\\
&\mathbf{r}_k - \alpha\,\mathbf{c}_k \perp \mathbf{c}_k && \Leftrightarrow\\
&\mathbf{c}_k^*\mathbf{r}_k - \alpha\,\mathbf{c}_k^*\mathbf{c}_k = 0 && \Leftrightarrow\\
&\mathbf{c}_k^*\mathbf{r}_k - \alpha\,\mathbf{c}_k^*\mathbf{c}_k = 0 && \Leftrightarrow
\end{aligned}$$

$$\alpha = \frac{\mathbf{c}_k^*\mathbf{r}_k}{\mathbf{c}_k^*\mathbf{c}_k}$$

[zie ook plaatjes]

Local Minimal Residuals

Choose $tol > 0$, $\mathbf{x}_0$, $k_{\max}$,
Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
For $k = 0, 1, 2, \ldots, k_{\max}$
  Stop if $\|\mathbf{r}_k\|_2 \le tol\|\mathbf{b}\|_2$
  $\mathbf{u}_k = \mathbf{r}_k$
  $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$
  $\sigma_k = \mathbf{c}_k^*\mathbf{c}_k$, $\alpha_k = \mathbf{c}_k^*\mathbf{r}_k/\sigma_k$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\,\mathbf{u}_k$
  $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\,\mathbf{c}_k$
end for

LMR (geheugen vriendelijk)

Choose $tol > 0$, $\mathbf{x}$, $k_{\max}$,
Compute $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
For $k = 0, 1, 2, \ldots, k_{\max}$
  Stop if $\|\mathbf{r}\|_2 \le tol\|\mathbf{b}\|_2$
  $\mathbf{u} \leftarrow \mathbf{r}$
  $\mathbf{c} \leftarrow \mathbf{A}\mathbf{u}$
  $\sigma \leftarrow \mathbf{c}^*\mathbf{c}$, $\alpha \leftarrow \mathbf{c}^*\mathbf{r}/\sigma$
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$
end for

LMR heeft de ingrediënten die typisch zijn voor iteratieve methodes:

- Loop (iteratie stap).
- Paar BLAS operaties (MV, AXPYs, DOTs) per stap.
- Stopcriterium:
  stop als – residu voldoende klein is (success) of
  – als het aantal stappen te groot wordt (fail).
- Geheugen vriendelijk.

Belangrijke vragen (voor iedere iteratieve methode):

- Hoe snel convergeert het proces
  (d.w.z., wat is de kleinste k waarvoor $\|\mathbf{r}_k\|_2 \leq tol\|\mathbf{b}\|_2$)?
- Wat zijn de kosten per stap?

(de totale kosten volgen dan)

Generalized Conjugate Residuals

Idee.

In LMR is $\mathbf{r}_k$ alleen geminimaliseerd in de richting $\mathbf{c}_k$.
Echter de vectoren $\mathbf{c}_{k-1}, \mathbf{c}_{k-2}, \ldots$ zijn ook berekend.
Levert minimaliseren in al deze richtingen een kleiner residu?

Stelling. (α_j) zo dat $\|\mathbf{r}_k - \sum_{j=0}^{k} \alpha_j \mathbf{c}_j\|_2$ minimaal

$\Leftrightarrow \quad \mathbf{r}_{k+1} \equiv \mathbf{r}_k - \sum_{j=0}^{k} \alpha_j \mathbf{c}_j \perp \mathbf{c}_i$ alle $i = 0, \ldots, k$.

Stelling. (α_j) zo dat $\|\mathbf{r}_k - \sum_{j=0}^{k} \alpha_j \mathbf{c}_j\|_2$ minimaal.

$\mathbf{r}_k, \mathbf{c}_k \perp \text{Span}(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1}) \;\Rightarrow \mathbf{r}_{k+1} \equiv \mathbf{r}_k - \alpha_k \mathbf{c}_k \perp \mathbf{c}_k$

Dus $\alpha_0 = \ldots = \alpha_{k-1} = 0$ en $\alpha_k = \dfrac{\mathbf{c}_k^* \mathbf{r}_k}{\mathbf{c}_k^* \mathbf{c}_k}$

$\mathbf{r}_k, \mathbf{c}_k \perp \text{Span}(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1})$

Gebruik Gram-Schmidt om, voor iedere $k = 1, 2, \ldots$, de vector $\mathbf{c}_k$ loodrecht $\text{Span}(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1})$ te krijgen

Stelling [Modified Gram-Schmidt].
Gegeven vectoren $\tilde{\mathbf{c}}_0, \tilde{\mathbf{c}}_1, \ldots$.
Voor $k = 1, 2, \ldots$, bereken

$\mathbf{c}_k = \tilde{\mathbf{c}}_k$
For $j = 0, 1, \ldots, k-1$
  $\beta_j = \mathbf{c}_j^* \tilde{\mathbf{c}}_k / \sigma_j$
  $\mathbf{c}_k \leftarrow \mathbf{c}_k - \beta_j \mathbf{c}_j$
end for
$\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$

Dan $\text{Span}(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_k) = \text{Span}(\tilde{\mathbf{c}}_0, \tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_k)$ alle k
en $\mathbf{c}_i \perp \mathbf{c}_j$ alle $i, j, i \neq j$.

GCR (geheugen vriendelijk)

```
Choose tol > 0, x, k_max,
Compute r = b − Ax
For k = 0, 1, 2, ..., k_max
   Stop if ||r||_2 ≤ tol||b||_2          2n flop
   u_k = r
   c_k = A u_k                           13n flop (3d)
   For j = 0, 1, 2, ..., k − 1
      β ← c_j^* c_k / σ_j                    2n flop
      u_k ← u_k − β u_j                      2n flop
      c_k ← c_k − β c_j                      2n flop
   end for
   σ_k = c_k^* c_k,  α ← c_k^* r / σ_k    4n flop
   x ← x + α u_k                         2n flop
   r ← r − α c_k                         2n flop
end for
```

totaal 3-d: 23n + 6nk flop
totaal 2-d: 19n + 6nk flop

Convergentie

Stelling. Met dezelfde $\mathbf{x}_0$:

$$\|\mathbf{r}_k^{\text{GCR}}\|_2 \leq \|\mathbf{r}_k^{\text{LMR}}\|_2 \leq \|\mathbf{r}_k^{\text{Richardson}}\|_2$$

Grondwatervergl., met, voor Richardson, $\alpha = \alpha^{\text{opt}}$

$$\rho_k^{\text{LMR}} \leq \rho_k^{\text{Richardson}} \equiv \frac{\|\mathbf{r}_k^{\text{Richardson}}\|_2}{\|\mathbf{r}_0\|_2} \leq \exp\left(-2k\,\frac{\lambda_1}{\lambda_n}\right)$$

$$\rho_k^{\text{GCR}} \leq \exp\left(-2k\sqrt{\frac{\lambda_1}{\lambda_n}}\right)$$

Grondwatervergl. Bij λ_1/λ_n moet je denken aan $h_x^2 + h_y^2$.

Opdracht. Probeer deze theoretische resultaten terug te zien in de praktijk.
Plot in je experimenten
de **residu reductie** of **convergentie historie**,
d.w.z., de grafiek van $k \rightsquigarrow \rho_k$ (op $\log_{10}$ schaal)

Krylov ruimtes

De **Krylov ruimte** $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$ van orde k voorgebracht door $\mathbf{A}$ en $\mathbf{r}_0$ is

$$\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) \;=\; \text{Span}(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0)$$

Stelling. Richardson, LMR en GCR vinden hun benadering $\mathbf{x}_k$ in $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$.

GCR vindt de benadering met kleinste residu:

$$\|\mathbf{r}_k^{\text{GCR}}\|_2 \leq \|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2 \text{ voor alle } \mathbf{y} \in \mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) \quad .$$

GCR is een **minimale residu methode**.

Stagnatie en afbreken

Als in GCR $\mathbf{c}_k \perp \mathbf{r}_k$, dan
stagneert de methode, d.w.z., $\mathbf{r}_{k+1} = \mathbf{r}_k$ (waarom?)

dan is $\mathbf{c}_{k+1}$ (na orth.) $= \mathbf{0}$ (waarom?) en
breekt de methode **af**, d.w.z., er wordt gedeeld door 0 (nl., door $\sigma_{k+1} = 0$).

Stelling. Als $\text{Re}(\lambda) > 0$ voor alle eigenwaarden λ van $\mathbf{A}$, dan stagneert GCR niet.

Conclusies

Richardson, LMR, GCR zijn Krylov ruimten methoden (iteratief, alleen MVs, AXPYs & DOTs).

GCR is, wat betreft het aantal MVs, in zekere zin optimaal

Rich.: + Geen DOTs, een paar AXPYs per MV
− Langzame convergentie (in termen van MVs)
− Geschikte α van te voren zelf bepalen (kan divergeren bij "verkeerde" α)

LMR: + Een paar AXPYs en DOTs/MV
+ Bepaalt automatisch geschikte α's
− Langzame convergentie (in termen van MVs)

GCR: − # AXPYs en DOTs/MV evenredig met k
+ Snelste convergentie (in termen van MVs)