Utrecht, November 27, 1997

Hybrid methods: efficient teams for solving large sparse systems of equations

Gerard Sleijpen
Henk van der Vorst
Diederik Fokkema

Universiteit Utrecht
Department of Mathematics

http://www.math.ruu.nl/people/sleijpen

$\boxed{\mathbf{Ax = b}}$

Hybrid method: product of two methods
Orth. Res. $\times$ Auxiliary method
Ex. Aux = Min. Res.

Program

1. Introduction
2. Minimal residual methods
3. Orthogonal residual methods
4. Hybrid methods
5. Accurate results

Conclusions

1.1 Large systems of sparse linear equations can only be solved by iterative methods.
1.2 There is no best iterative method.
1.3 Krylov subspace methods subject to appropriate preconditioning are very attractive.

- *For a class of related problems for a given type of computer architecture one may find good methods by combining theoretical insight and experimental results.*
- *It is not easy to find a good method for a given situation.*

2.1 Hybrid methods as

Bi-CG, CGS, Bi-CGSTAB, BiCGstab(ℓ), ...

are very competitive.

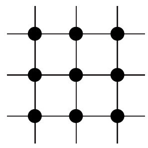
2.2 The auxiliary method can be tuned to enhance performance.

Global ocean circulation

$\Downarrow$ Oceanography

$$-fv = -g\frac{\partial h}{\partial x} - ru + A\nabla^2 u + F_1 \quad \text{(NS)}$$
...
$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$\Downarrow$ Finite differences
Finite elements
Finite volumes, ...



...
$$\frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x} + \ldots$$

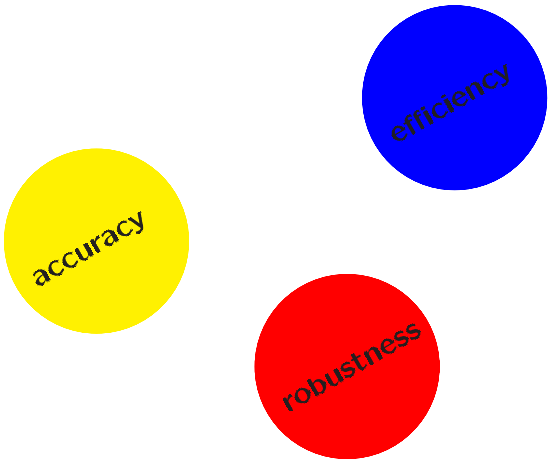
$\Downarrow$ Linearize

$\mathbf{Ax = b}$

$\Downarrow$ Numerical Linear Algebra
Computer Science

Simulation

The art of numerics



- Approximation error
- Stability: effect of (rounding) errors

- Computational costs
  - # of steps (speed of conv)
  - costs per step
- Storage

- Class of problems

Properties depend on (architecture) computer

Direct methods?

Direct method: $\mathbf{A} = \mathbf{L}\,\mathbf{U}$

Due to fill-in, $\mathbf{L}$ and $\mathbf{U}$ can not be stored for realistic problems

Comparison for $n \to \infty$

model problem	**Direct**	**Iterative**
Computational costs	$\mathcal{O}(n^\nu)$ $\nu \approx 2.0$ if $d=2$ $\nu \approx 2.3$ if $d=3$	$\mathcal{O}(n^\nu)$ $\nu \approx 1.4$ if $d=2$ $\nu \approx 1.2$ if $d=3$
Memory requirement	$\mathcal{O}(n^\nu)$ $\nu \approx 1.5$ if $d=2$ $\nu \approx 1.7$ if $d=3$	$\mathcal{O}(n)$

Iterative Methods to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$

$$\mathbf{x}_0 \to \mathbf{x}_1 \to \ldots \to \mathbf{x}_{k-1} \xrightarrow{\text{step } k} \mathbf{x}_k$$

initial guess ($\mathbf{x}_0$) $\qquad$ k-th approximate ($\mathbf{x}_k$)

- computational costs per step low
- low on memory
- $\|\mathbf{x} - \mathbf{x}_k\| \leq$ *tol* for some non-large k (hopefully)

k-th **error**
$\mathbf{x} - \mathbf{x}_k$
not computable

k-th **residual**
$\mathbf{r}_k = \mathbf{A}(\mathbf{x} - \mathbf{x}_k) = \mathbf{b} - \mathbf{A}\mathbf{x}_k$
computable

$$\begin{matrix}\mathbf{x}_0 \\ \mathbf{r}_0\end{matrix} \to \begin{matrix}\mathbf{x}_1 \\ \mathbf{r}_1\end{matrix} \to \ldots \to \begin{matrix}\mathbf{x}_{k-1} \\ \mathbf{r}_{k-1}\end{matrix} \to \begin{matrix}\mathbf{x}_k \\ \mathbf{r}_k\end{matrix}$$

stop if $\quad \|\mathbf{r}_k\| \leq$ *tol*

[Hestenes Stiefel '52]

efficient

CG $\quad A > 0,\ K > 0$

Choose $\mathbf{x}_0$
$\mathbf{x} = \mathbf{x}_0,\ \mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
$\mathbf{u} = 0,\ \rho = 1$
while $\|\mathbf{r}\| > tol$ do
$\quad \sigma \leftarrow -\rho$
$\quad$ solve $\mathbf{w}$ from $\mathbf{K}\mathbf{w} = \mathbf{r}$
$\quad \rho \leftarrow (\mathbf{r}, \mathbf{w}),\ \beta \leftarrow \rho/\sigma$
$\quad \mathbf{u} \leftarrow \mathbf{w} - \beta\,\mathbf{u}$
$\quad \mathbf{c} \leftarrow \mathbf{A}\mathbf{u}$
$\quad \sigma \leftarrow (\mathbf{c}, \mathbf{u}),\ \alpha \leftarrow \rho/\sigma$
$\quad \mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
$\quad \mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$
end while

Properties CG

accurate

Advantages

- Efficient:
  - 1 MV, 2 DOT, 3 AXPY per step
  - Storage: 5 large vectors (incl. $\mathbf{b}$)
  - Minimal residual method if $\mathbf{A}$ pos.def.: $\|\mathbf{r}_k\|_{\mathbf{A}^{-1}}$ is minimal.
- Robust: CG (for pos.def.) always converges
- No knowledge on properties of $\mathbf{A}$ is needed

Drawback

- Not robust: does not work if $\mathbf{A} \neq \mathbf{A}^*$.

robust

Is there a best iterative method?

If no further restrictions: no.

For a(ny) certain class of related problems, comp. arch.,...: yes.

efficient

Success of any iterative method depends on:

- properties of A
(size, sparsity structure, spectral properties,...),

accurate

- computer architecture
(scalar, vector, parallel, massive parallel, shared memory, distributed memory, ...)

robust

- memory space that is available,

- accuracy that is required.

Approximate as a side product

In many iterative methods:

compute **search direction** $\mathbf{u}_k$,
based on previously computed
$\mathbf{r}_k,\ \mathbf{u}_{k-1}, \mathbf{u}_{k-2}, \ldots,\ \mathbf{c}_{k-1}, \ldots$

Select $\mathbf{x}_0$, compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\vdots$
$\mathbf{c}_k = \mathbf{A}\,\mathbf{u}_k$
$\vdots$
$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\,\mathbf{u}_k$
$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\,\mathbf{c}_k$

Objective: get $\|\mathbf{r}_{k+1}\|$ “small”

Work on the residual, get the associated approximate as a side-product by updating the residual with a vector $\mathbf{c}_k$ of the form $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$ with $\mathbf{u}_k$ explicitly known.

Subspace methods

- Generate subspace with basis $\mathbf{v}_1, \ldots, \mathbf{v}_k$
- With $\mathbf{V}_k := [\mathbf{v}_1, \ldots, \mathbf{v}_k]$, find y such that

  $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k = \mathbf{r}_0 - \mathbf{A}\mathbf{V}_k\, y$ is "small"

  $\mathbf{x}_k = \mathbf{x}_0 + \mathbf{V}_k\, y$

Ex. min $\|\mathbf{r}_k\|$, $\quad \mathbf{r}_k \perp \mathbf{V}_k$, $\quad \mathbf{r}_k \perp \mathbf{W}_k$

Min. Res $\quad$ Orth. Res

In practice $\quad k \ll n$

∘ Generate orthonormal basis: $\mathbf{V}_k^*\mathbf{V}_k = I$
(or bi-orthogonal: $\mathbf{W}_k^*\mathbf{V}_k$ =diag)

∘ Solve low dimensional problem:

$$(\mathbf{W}_k^*\mathbf{A}\mathbf{V}_k)\, y = \mathbf{W}_k^*\mathbf{b}$$

∘ Find an efficient & stable implementation

Krylov subspace methods

Search subspace span$(\mathbf{V}_k)$ = **Krylov subspace**:

span$(\mathbf{V}_k) = \mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)$ $\quad$, where

$$\begin{aligned}\mathcal{K}_k(\mathbf{A}; \mathbf{r}_0) &= \text{span}\left(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\right) \\ &= \{q(\mathbf{A})\mathbf{r}_0 \mid q \text{ polynomial deg. } < k\} \\ &= \{q(\mathbf{A})\mathbf{r}_0 \mid q \in \mathcal{P}_{k-1}\}\end{aligned}$$

$$\begin{aligned}\mathbf{r}_k &= \mathbf{b} - \mathbf{A}\mathbf{x}_k = \mathbf{r}_0 - \mathbf{A}\mathbf{y} \\ &= (\mathbf{I} - \mathbf{A}q_k(\mathbf{A}))\mathbf{r}_0 \\ &= p_k(\mathbf{A})\mathbf{r}_0 \text{ where } p_k(t) = 1 - t q_k(t).\end{aligned}$$

Ex. Compute $\mathbf{v}_{k+1}$ by
orthonormalizing $\mathbf{A}\mathbf{v}_k$ against $\mathbf{V}_k$

Minimal residual methods

Optimality:

$\mathbf{y}_k \in \mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)$ solves

$$\min\{\|\mathbf{r}_0 - \mathbf{A}\mathbf{y}\| \mid \mathbf{y} \in \mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)\}$$

or, equivalently

$\mathbf{s}_k \in \mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)$ solves

$$\min\{\|\mathbf{r}_0 - \mathbf{s}\| \mid \mathbf{s} \in \mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)\}$$

$$\min\{\|\mathbf{r}_0 - \mathbf{s}\| \mid \mathbf{s} \in \mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)\}$$

$\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{k-1}$ such that

$\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1}$ is an orthogonal basis of $\mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)$

where $\quad \mathbf{c}_j = \mathbf{A}\mathbf{u}_j \quad$ for $\quad j = 0, 1, \ldots, k-1$

$$\begin{aligned}\text{Then} \quad \mathbf{r}_k &= \mathbf{r}_0 - \sum_{j=0}^{k-1} \frac{(\mathbf{r}_0, \mathbf{c}_j)}{(\mathbf{c}_j, \mathbf{c}_j)}\mathbf{c}_j \\ &= \mathbf{r}_{k-1} - \alpha_{k-1}\mathbf{c}_{k-1} \\ \text{and} \quad \mathbf{x}_k &= \mathbf{x}_{k-1} + \alpha_{k-1}\mathbf{u}_{k-1}\end{aligned}$$

$\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{k-1}$ such that

$\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1}$ is an orthogonal basis of $\mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)$

where $\quad \mathbf{c}_j = \mathbf{A}\mathbf{u}_j \quad$ for $\quad j = 0, 1, \ldots, k-1$

How to compute the $\mathbf{c}_j$? [Young Jea 80]

Note that:

$$\mathbf{r}_k \in \mathcal{K}_k(\mathbf{A}; \mathbf{r}_0) \quad \Rightarrow \quad \mathbf{A}\mathbf{r}_k \in \mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{r}_0).$$

$$\mathbf{c}_k = \mathbf{A}\mathbf{r}_k - \sum_{j=0}^{k-1} \frac{(\mathbf{A}\mathbf{r}_k, \mathbf{c}_j)}{(\mathbf{c}_j, \mathbf{c}_j)} \mathbf{c}_j \quad \text{and}$$

$$\mathbf{u}_k = \mathbf{r}_k - \sum_{j=0}^{k-1} \frac{(\mathbf{A}\mathbf{r}_k, \mathbf{c}_j)}{(\mathbf{c}_j, \mathbf{c}_j)} \mathbf{u}_j$$

Gram-Schmidt orthogonalization

Minimal residual methods

Advantages:

robust

- Krylov subspace methods.
- Optimal.
- No knowledge (in advance) on (spectral) properties of $\mathbf{A}$ is needed.

accurate

Drawbacks:

- Growing computational costs if the number of iteration steps increases.
Per step: # DOTs & # AXPYs proportional to the step number.
- Growing memory requirement if the number of iteration steps increases, # large vectors is proportional to the step number.

efficient

Towards more efficiency

If $\mathbf{A}^* = \mathbf{A}$ then $(\mathbf{A}\mathbf{r}_k, \mathbf{c}_j) = 0 \quad (j < k-1)$

leads to CG (for approp. inner product)

Only if $\mathbf{A}$ pos. def.

[Faber Manteuffel 84]

Th. Min. Res. & efficient/step $\Leftrightarrow$ $\mathbf{A} = \alpha + \beta$pos.def.

[Saad Schulz 86, Sl v.d. Vorst 93]

u_j lin. comb. $\mathbf{r}_0, \ldots, \mathbf{r}_j$

$$\begin{aligned} \mathbf{x}_k =& \mathbf{x}_0 + [\mathbf{u}_0, \ldots, \mathbf{u}_k] y = \\ & \mathbf{x}_0 + [\mathbf{r}_0, \ldots, \mathbf{r}_k] By \end{aligned}$$

No need to compute the u_j

Best if MV very expensive

Costs still proportional to k

Restart every ℓ step

Retain only last ℓ basis vectors $\mathbf{c}_j$ & $\mathbf{u}_j$

Retain combinations of $\mathbf{c}_j$ & of $\mathbf{u}_j$ [vdV Vuik 94]

[de Sturler Fokkema 93, de Sturler 96, ...97]

Loss of speed of convergence

$\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{k-1}$ such that

$\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1}$ is an orthogonal basis of ...

where $\quad \mathbf{c}_j = \mathbf{A}\mathbf{u}_j \quad$ for $\quad j = 0, 1, \ldots, k-1$

How to compute the $\mathbf{c}_j$? [Saad 93, vdV Vuik 94]

Note that:

$$\mathbf{r}_k \in \mathcal{K}_k(\mathbf{A}; \mathbf{r}_0) \quad \Rightarrow \quad \mathbf{A}\mathbf{r}_k \in \mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{r}_0).$$

$$\mathbf{c}_k = \mathbf{A}\mathbf{w}_k - \sum_{j=0}^{k-1} \frac{(\mathbf{A}\mathbf{w}_k, \mathbf{c}_j)}{(\mathbf{c}_j, \mathbf{c}_j)} \mathbf{c}_j \quad \text{and}$$

$$\mathbf{u}_k = \mathbf{w}_k - \sum_{j=0}^{k-1} \frac{(\mathbf{A}\mathbf{w}_k, \mathbf{c}_j)}{(\mathbf{c}_j, \mathbf{c}_j)} \mathbf{u}_j$$

Examples

$\mathbf{w}_k = \mathbf{r}_k, \quad \mathbf{w}_k = \mathbf{c}_k$ [Vinsome 76]

$\mathbf{w}_k$ solves approx. $\mathbf{A}\mathbf{w} = \mathbf{r}_k$

$\mathbf{w}_k = \mathbf{K}^{-1}\mathbf{r}_k$ with $\mathbf{K} \approx \mathbf{A}$

$\mathbf{w}_k \approx$ (nearly sing.) eigenvector [Morgan 95]

(Bi-)Orthogonal residual methods

Find $\mathbf{y}_k \in \mathcal{K}_k(\mathbf{A};\mathbf{r}_0)$ such that

$$\mathbf{r}_k = \mathbf{r}_0 - \mathbf{A}\mathbf{y}_k \perp \mathbf{W}_k$$

With test subspace span$(\mathbf{W}_k) =$

shadow Krylov subspace $\mathcal{K}_k(\mathbf{A}^*;\tilde{\mathbf{r}}_0)$

short recurrences (= efficient/step) possible.

efficient

(Bi-)Orthogonal residual methods

Advantages:

- Krylov, Efficient per step, low on memory
- Often nearly optimal
- No knowledge on prop. $\mathbf{A}$ needed

robust

Drawbacks:

- Not optimal
- Sensitive to effects of rounding errors

accurate

efficient

Conjugate Gradients

$$\boxed{\begin{aligned} \mathbf{u}_k &= \mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1} \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k \end{aligned}}$$

$$\mathbf{r}_k,\ \mathbf{u}_k \in \mathcal{K}_{k+1}(\mathbf{A};\mathbf{r}_0)$$

Theorem. If $\mathbf{A}^* = \mathbf{A}$ and

$\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \mathbf{r}_{k-1}$ then $\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \mathcal{K}_k(\mathbf{A};\mathbf{r}_0)$

Proof.

$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_{k-1}\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \mathbf{r}_{k-1}$ by construction

$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_{k-1}\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \mathcal{K}_{k-1}(\mathbf{A};\mathbf{r}_0)$ by induction

$\mathbf{A}\,\mathbf{u}_k = \mathbf{A}\,\mathbf{r}_k - \beta_k\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \mathbf{r}_k$ by construction

$\mathbf{A}\,\mathbf{u}_k = \mathbf{A}\,\mathbf{r}_k - \beta_k\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \mathcal{K}_{k-1}(\mathbf{A};\mathbf{r}_0)$ by induction:

$\mathbf{A}\mathbf{r}_k \perp \mathcal{K}_{k-1}(\mathbf{A};\mathbf{r}_0) \quad \Leftrightarrow \quad \mathbf{r}_k \perp \mathbf{A}\mathcal{K}_{k-1}(\mathbf{A};\mathbf{r}_0) \subset \mathcal{K}_k(\mathbf{A};\mathbf{r}_0)$

Bi-Conjugate Gradients

$$\boxed{\begin{aligned} \mathbf{u}_k &= \mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1} \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k \end{aligned}}$$

$$\mathbf{r}_k,\ \mathbf{u}_k \in \mathcal{K}_{k+1}(\mathbf{A};\mathbf{r}_0)$$

Theorem. $\tilde{\mathbf{r}}_0,\ldots,\tilde{\mathbf{r}}_{k-1}$ Krylov basis $\mathcal{K}_k(\mathbf{A}^*;\tilde{\mathbf{r}}_0)$

$\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \tilde{\mathbf{r}}_{k-1}$ then $\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \mathcal{K}_k(\mathbf{A}^*;\tilde{\mathbf{r}}_0)$

Proof.

$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_{k-1}\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \tilde{\mathbf{r}}_{k-1}$ by construction

$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_{k-1}\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \mathcal{K}_{k-1}(\mathbf{A}^*;\tilde{\mathbf{r}}_0)$ by induction

$\mathbf{A}\,\mathbf{u}_k = \mathbf{A}\,\mathbf{r}_k - \beta_k\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \tilde{\mathbf{r}}_{k-1}$ by construction

$\mathbf{A}\,\mathbf{u}_k = \mathbf{A}\,\mathbf{r}_k - \beta_k\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \mathcal{K}_{k-1}(\mathbf{A}^*;\tilde{\mathbf{r}}_0)$ by induction:

$\mathbf{A}\,\mathbf{r}_k \perp \mathcal{K}_{k-1}(\mathbf{A}^*;\tilde{\mathbf{r}}_0) \ \Leftrightarrow\ \mathbf{r}_k \perp \mathbf{A}^*\,\mathcal{K}_{k-1}(\mathbf{A}^*;\tilde{\mathbf{r}}_0) \subset \mathcal{K}_k(\mathbf{A}^*;\tilde{\mathbf{r}}_0)$

Bi-Conjugate Gradients

$$\boxed{\begin{aligned}\mathbf{u}_k &= \mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1}\\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k\end{aligned}}$$

$$\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \bar{q}_{k-1}(\mathbf{A}^*)\tilde{\mathbf{r}}_0$$

with q_{k-1} a polynomial of exact degree $k-1$.

With

$\rho_k := (\mathbf{r}_k, \bar{q}_k(\mathbf{A}^*)\tilde{\mathbf{r}}_0)$ & $\sigma_k := (\mathbf{A}\mathbf{u}_k, \bar{q}_k(\mathbf{A}^*)\tilde{\mathbf{r}}_0)$

and θ_k such that

$$q_k(\zeta) - \theta_k\,\zeta\,q_{k-1}(\zeta) \in \mathcal{P}_{k-1}$$

we have

$$\alpha_k = \frac{\rho_k}{\sigma_k} \qquad \& \qquad \beta_k = \frac{\rho_k}{\theta_k\,\sigma_{k-1}}$$

[Fletcher '76]

Bi-CG

Choose $\mathbf{x}_0$ and $\tilde{\mathbf{r}}$
$\mathbf{x} = \mathbf{x}_0,\ \mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
$\mathbf{u} = 0,\ \rho = 1,$ $\tilde{\mathbf{c}} = 0$
while $\|\mathbf{r}\| > tol$ do
  $\sigma \leftarrow -\rho$
  solve $\mathbf{w}$ from $\mathbf{K}\mathbf{w} = \mathbf{r}$
  $\rho \leftarrow (\mathbf{r}, \tilde{\mathbf{r}}),\ \beta \leftarrow \rho/\sigma$
  $\mathbf{u} \leftarrow \mathbf{w} - \beta\,\mathbf{u}$
  $\mathbf{c} \leftarrow \mathbf{A}\mathbf{u}$
  $\sigma \leftarrow (\mathbf{c}, \tilde{\mathbf{r}}),\ \alpha \leftarrow \rho/\sigma$
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$
    $\tilde{\mathbf{c}} \leftarrow \mathbf{A}^*\,\tilde{\mathbf{r}} - \bar{\beta}\,\tilde{\mathbf{c}}$
    $\tilde{\mathbf{r}} \leftarrow \tilde{\mathbf{r}} - \bar{\alpha}\,\tilde{\mathbf{c}}$
end while

Properties Bi-CG

Advantages

- Usually selects good approximations from the search subspaces (Krylov subspaces).
- 2 DOT, 5 AXPY per step.
- Storage: 8 large vectors.
- No knowledge on properties of $\mathbf{A}$ is needed.

Drawbacks

- ◦ Non-optimal Krylov subspace method.
- Not robust: Bi-CG may break down.
- Bi-CG is sensitive to rounding errors
- ◦ Convergence depends on shadow residual $\tilde{\mathbf{r}}_0$.
- 2 MV needed to expand search subspace.
- 1 MV is by $\mathbf{A}^*$.

Near Breakdowns

Bi-CG breaks down if:

- $\rho_k = 0$ – $\mathbf{r}_k = 0$ **lucky breakdown**
  - $\tilde{\mathbf{r}}_k = 0$
  - $\mathbf{r}_k \neq 0$, $\tilde{\mathbf{r}}_k \neq 0$ and $\mathbf{r}_k \perp \tilde{\mathbf{r}}_k$ **Lanczos breakdown**
- $\sigma_k = 0$ **pivot breakdown**

Breakdown hardly ever occurs in practice.
Cure: restart with another $\tilde{\mathbf{r}}_0$.

However, **near-breakdown** may lead to
- inaccurate coefficients (ρ_k or σ_k)
- irregular convergence

Cure [Freund Nachtigal 91, --- Gutknecht --- 93]
- $\rho_k \approx 0$: look-ahead Lanczos [...]
- $\sigma_k \approx 0$: composite step Bi-CG, QMR

near-breakdown may lead to
- inaccurate coefficients (ρ_k or σ_k)
- irregular convergence

Inaccurate coefficients

- Loss of speed of convergence

Irregular convergence

- Loss of speed of convergence
- Loss of accuracy
  i.e., for computed $\mathbf{r}_k$ and computed $\mathbf{x}_k$
  $$\frac{\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|}{\|\mathbf{r}_k\|} \not\approx 1$$

Short recurrences

- Loss of bi-orth.: loss of speed of conv.

$$\mathbf{u}_k = \mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1}$$
$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k$$

Rounded arithmetic

In rounded arithmetic $\cdot^\star$ computed quantity

$$\mathbf{r}^\star_{k+1} = \mathbf{r}^\star_k - \alpha_k\,\mathbf{A}\,\mathbf{u}^\star_k + \mathbf{f}_k$$
$\mathbf{f}_k$ is the **local** rounding error

Theorem. assuming MV main source of error

$$\boxed{\|\mathbf{f}_k\| \lesssim \bar{\xi}\,\Gamma\left(\|\mathbf{r}^\star_{k+1}\| + \|\mathbf{r}^\star_k\|\right)}$$

where $\bar{\xi}$ is the **relative machine precision**

and $\Gamma := n_A\,\left\|\,|\mathbf{A}|\,\right\|\,\left\|\,\mathbf{A}^{-1}\,\right\|$

with $n_A :=$ max # non-zeros per row of $\mathbf{A}$

Proof. $\|\mathbf{f}_k\| \lesssim \bar{\xi}\, n_A\,\|\,|\mathbf{A}|\,\|\,\|\alpha_k\mathbf{u}^\star\| \leq \bar{\xi}\,\Gamma\,\|\mathbf{A}(\alpha_k\mathbf{u}^\star_k)\|$

$$\mathbf{u}_k = \mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1}$$
$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k$$

In rounded arithmetic
$$\mathbf{r}^\star_{k+1} = \mathbf{r}^\star_k - \alpha_k\,\mathbf{A}\,\mathbf{u}^\star_k + \mathbf{f}_k$$
$$\|\mathbf{f}_k\| \lesssim \bar{\xi}\,\Gamma\left(\|\mathbf{r}^\star_{k+1}\| + \|\mathbf{r}^\star_k\|\right)$$

Perturbed convergence

Experimental result

Convergence depends very sensitively on $\mathbf{f}_k$.

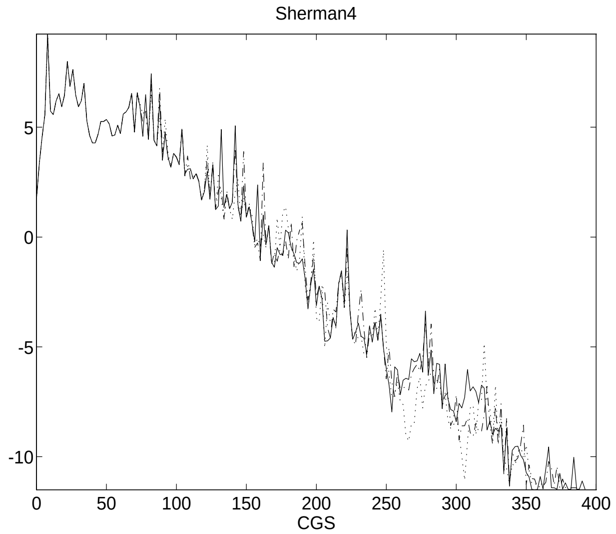
Speed of convergence does **not** depend heavily on $\mathbf{f}_k$ if $\|\mathbf{f}_k\| \ll \|\mathbf{r}^\star_{k+1}\|$

Speed of convergence may be disturbed seriously if $\|\mathbf{f}_k\| \not\ll \|\mathbf{r}^\star_{k+1}\|$

Therefore,
irregularities harmless if $\bar{\xi}\,\Gamma\,\|\mathbf{r}^\star_k\| \ll \|\mathbf{r}^\star_{k+1}\|$

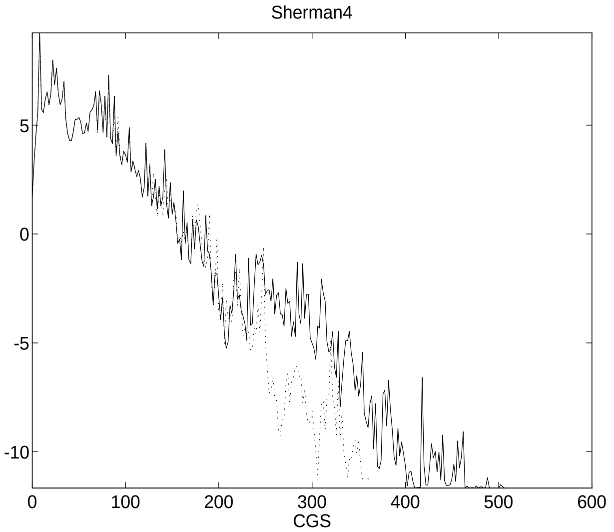
$$\mathbf{r}^\star_{k+1} = \mathbf{r}^\star_k - \alpha_k\,\mathbf{A}\,\mathbf{u}^\star_k + \mathbf{f}_k, \qquad \|\mathbf{f}_k\| \ll \|\mathbf{r}^\star_{k+1}\|$$

Small pert. in (hybrid) Bi-CG



$$\mathbf{r}^\star_{k+1} = \mathbf{r}^\star_k - \alpha_k\,\mathbf{A}\,\mathbf{u}^\star_k + \mathbf{f}_k, \qquad \|\mathbf{f}_k\| \not\ll \|\mathbf{r}^\star_{k+1}\|$$

Large pert. in (hybrid) Bi-CG



[Sl vdV Fokkema 94, Greenbaum 94 96]

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\,\mathbf{u}_k$$

In rounded arithmetic

$$\mathbf{r}^\star_{k+1} = \mathbf{r}^\star_k - \alpha_k\,\mathbf{A}\,\mathbf{u}^\star_k + \mathbf{f}_k$$
$$\|\mathbf{f}_k\| \lesssim \bar{\xi}\,\Gamma\left(\|\mathbf{r}^\star_{k+1}\| + \|\mathbf{r}^\star_k\|\right)$$

Accuracy

True residual

$$\mathbf{b} - \mathbf{A}\mathbf{x}^\star_{j+1} = \mathbf{b} - \mathbf{A}\mathbf{x}^\star_j - \mathbf{A}(\alpha_j\,\mathbf{u}^\star_j)$$

and recursively updated residual

$$\mathbf{r}^\star_{j+1} \quad = \quad \mathbf{r}^\star_j \quad - \alpha_j\,\mathbf{A}\mathbf{u}^\star_j + \mathbf{f}_j$$

drift apart.

Theorem.

$$\|(\mathbf{b} - \mathbf{A}\mathbf{x}^\star_k) - \mathbf{r}^\star_k\| \le 2\,\bar{\xi}\,\Gamma \sum_{j\le k} \|\,\mathbf{r}^\star_j\|$$

where $\Gamma := n_A\,\left\|\,|\mathbf{A}|\,\right\|\,\left\|\,\mathbf{A}^{-1}\right\|$

[Sl vdV 96]

Avoiding inaccurate solutions

Use $\quad \mathbf{r}_{k+1} \leftarrow \mathbf{b} - \mathbf{A}\,\mathbf{x}_{k+1} \quad$ in each step?

$$\begin{aligned}\mathbf{r}^\star_{k+1} &= \mathbf{b} - (\mathbf{A}\mathbf{x}^\star_{k+1})^\star = \mathbf{b} - \mathbf{A}\mathbf{x}^\star_{k+1} + \mathbf{f}'_k \\ &= \mathbf{b} - \mathbf{A}\mathbf{x}^\star_k - \mathbf{A}(\alpha_k\mathbf{u}^\star_k) + \mathbf{f}'_k \\ &= \mathbf{b} - (\mathbf{A}\mathbf{x}^\star_k)^\star - \mathbf{A}(\alpha_k\mathbf{u}^\star_k) + \mathbf{f}''_k + \mathbf{f}'_k\end{aligned}$$

Additional recursion error:

$$\|\mathbf{f}_k\| \le \bar{\xi}\,\Gamma\,\left(2\,\|\,\mathbf{b}\,\| + \|\,\mathbf{r}^\star_k\,\| + \|\,\mathbf{r}^\star_{k+1}\,\|\right)$$

Disadvantages:

- disturbs convergence if $\|\,\mathbf{r}^\star_{k+1}\,\| \ll \|\,\mathbf{b}\,\|$,
- (may) cost (many) additional MVs.

Use $\quad \mathbf{r}_{k+1} \leftarrow \mathbf{b} - \mathbf{A}\,\mathbf{x}_{k+1} \quad$ occasionally!

Additional recursion error:

$$\|\mathbf{f}\| \le \bar{\xi}\,\Gamma\,\left(2\,\|\,\mathbf{b}\,\| + \sum \|\,\mathbf{r}^\star_j\,\|\right),$$

summing over all res. since previous use of true res.

[Sonneveld '89]

Transpose-free Bi-CG

$$\rho_k = (\mathbf{r}_k, \bar{q}_k(\mathbf{A}^*)\,\tilde{\mathbf{r}}_0) = (q_k(\mathbf{A})\mathbf{r}_k, \tilde{\mathbf{r}}_0),$$
$$\sigma_k = (\mathbf{A}\mathbf{u}_k, \bar{q}_k(\mathbf{A}^*)\,\tilde{\mathbf{r}}_0) = (\mathbf{A}q_k(\mathbf{A})\mathbf{u}_k, \tilde{\mathbf{r}}_0)$$

$$\mathbf{Q}_k := q_k(\mathbf{A})$$

$$(\text{Bi-CG}) \quad \left\{\begin{array}{ll}\rho_k, & \mathbf{Q}_k\mathbf{u}_k = \mathbf{Q}_k\mathbf{r}_k - \beta_k\,\mathbf{Q}_k\mathbf{u}_{k-1}, \\ \sigma_k, & \mathbf{Q}_k\mathbf{r}_{k+1} = \mathbf{Q}_k\mathbf{r}_k - \alpha_k\,\mathbf{A}\mathbf{Q}_k\mathbf{u}_k,\end{array}\right.$$

(Pol) Compute q_{k+1} of degree $k+1$

such that $q_{k+1}(0) = q_k(0) = 1$.

Compute

$$\hat{\hat{\mathbf{u}}}_k := \mathbf{Q}_{k+1}\mathbf{u}_k, \quad \hat{\hat{\mathbf{r}}}_{k+1} := \mathbf{Q}_{k+1}\mathbf{r}_{k+1}$$

Transpose-free Bi-CG; Practice

Work with $\widehat{\mathbf{u}}_k = \mathbf{Q}_k \mathbf{u}_k$ and $\widehat{\mathbf{r}}_k = \mathbf{Q}_k \mathbf{r}_{k+1}$

Write $\mathbf{u}_k$, $\mathbf{r}_k$, $\mathbf{x}_k$ instead of $\widehat{\widehat{\mathbf{u}}}_k$, $\widehat{\widehat{\mathbf{r}}}_k$, $\widehat{\widehat{\mathbf{x}}}_k$, resp.

$\rho_k = (\mathbf{r}_k, \widetilde{\mathbf{r}}_0)$,
$\sigma_k = (\mathbf{A}\widehat{\mathbf{u}}_k, \widetilde{\mathbf{r}}_0)$

$$\text{(Bi-CG)} \left\{ \begin{array}{ll} \rho_k = (\mathbf{r}_k, \widetilde{\mathbf{r}}_0), & \widehat{\mathbf{u}}_k = \mathbf{r}_k - \beta_k \mathbf{u}_{k-1}, \\ \sigma_k = (\mathbf{A}\widehat{\mathbf{u}}_k, \widetilde{\mathbf{r}}_0), & \widehat{\mathbf{r}}_k = \mathbf{r}_k - \alpha_k \mathbf{A}\widehat{\mathbf{u}}_k, \end{array} \right. \\ \qquad\qquad\qquad\qquad \widehat{\mathbf{x}} = \mathbf{x}_k + \alpha_k \widehat{\mathbf{u}}_k$$

(Pol) Compute updating coefficients for q_{k+1}.
Compute

$\mathbf{u}_k$, $\quad \mathbf{r}_{k+1}$, $\quad \mathbf{x}_{k+1}$

Hybrid Bi-CG

$$\boxed{\mathbf{r}_k = q_k(\mathbf{A})\, \phi_k^{\text{Bi-CG}}(\mathbf{A})\, \mathbf{r}_0}$$

$r_k := \mathbf{r}_k^{\text{Bi-CG}} := \phi_k^{\text{Bi-CG}}(\mathbf{A})\mathbf{r}_0$:
$\phi_k^{\text{Bi-CG}}$ is the "Bi-CG polynomial"
of degree k

q_k??

q_k for efficient steps & fast convergence.

Fast convergence by

- reducing the residual
- stabilizing the Bi-CG part
- other — when used as linear solver for the Jacobian system in a Newton scheme for non-linear equations, by reducing the number of Newton steps

Properties hybrid Bi-CG

Advantages

- Converges often twice as fast as Bi-CG

w.r.t. # MVs:

each MV expands the search subspace
Bi-CG: $\mathbf{x}_k - \mathbf{x}_0 \in \mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)$ á 2k MV.
Hybrid Bi-CG: $\mathbf{x}_k - \mathbf{x}_0 \in \mathcal{K}_{2k}(\mathbf{A}; \mathbf{r}_0)$ á 2k MV.

- Work/MV and storage similar to Bi-CG.
- Transpose free.
- ∘ Explicit computation of Bi-CG scalars.

Drawbacks

- Non-optimal Krylov subspace method.
- Peaks in the convergence history.
- Large intermediate residuals.
- ∘ Breakdown possibilities.

[Sonneveld '89]

Conjugate Gradients Squared

$$q_k = \phi_k^{\text{Bi-CG}}, \qquad \mathbf{r}_k = \phi_k^{\text{Bi-CG}}(\mathbf{A})\, \phi_k^{\text{Bi-CG}}(\mathbf{A})\, \mathbf{r}_0$$

CGS exploits recurrence relations for the Bi-CG polynomials to design a very efficient algorithm.

Properties

+ Hybrid Bi-CG.
+ A very efficient algorithm:
  1 DOT/MV, 3.25 AXPY/MV;
  storage: 7 large vectors.
− Often high peaks in its convergence history
− Often large intermediate residuals
+ Seems to do well as linear solver in a Newton scheme

[Sonneveld '89]

CGS

Choose $\mathbf{x}_0$ and $\tilde{\mathbf{r}}_0$.

$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$.

$\mathbf{u} = \mathbf{w} = 0$, $\rho = 1$.

While $\|\mathbf{r}\| > tol$ do

$\sigma \leftarrow -\rho$

$\rho \leftarrow (\mathbf{r}, \tilde{\mathbf{r}}_0)$, $\quad \beta \leftarrow \rho/\sigma$

$\mathbf{w} \leftarrow \mathbf{u} - \beta\,\mathbf{w}$

$\mathbf{v} \leftarrow \mathbf{r} - \beta\,\mathbf{u}$

$\mathbf{w} \leftarrow \mathbf{v} - \beta\,\mathbf{w}$

$\mathbf{c} \leftarrow \mathbf{A}\mathbf{w}$

$\sigma \leftarrow (\mathbf{c}, \tilde{\mathbf{r}}_0)$, $\quad \alpha \leftarrow \rho/\sigma$

$\mathbf{u} \leftarrow \mathbf{v} - \alpha\,\mathbf{c}$

$\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{A}(\mathbf{v} + \mathbf{u})$

$\mathbf{x} \leftarrow \mathbf{x} + \alpha\,(\mathbf{v} + \mathbf{u})$

end while

How

to preserve the reducing effect of squaring

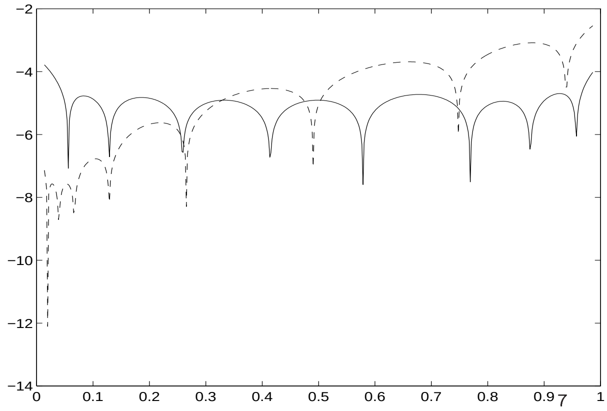
and to get rid of the amplifying effects?

Idea. [Fokkema Sl vdV 96]

Take, for instance, $q_k = \tilde{\phi}_k$ with

$\tilde{\phi}_k$ Bi-CG polynomial for $\mathbf{A}$,

but with another $\tilde{\mathbf{r}}_0$, say, $\tilde{\mathbf{s}}_0 \neq \tilde{\mathbf{r}}_0$.



[van der Vorst 92]

Smoothing the convergence:

Bi-CG STABilized

Put $\quad r_k := \mathbf{r}_k^{\text{Bi-CG}}, \quad u_k := \mathbf{u}_k^{\text{Bi-CG}}$

$$q_{k+1}(t) = (1 - \omega_k t) q_k$$

$$\mathbf{Q}_{k+1}\, r_{k+1} = (\mathbf{I} - \omega_k \mathbf{A})\, \mathbf{Q}_k\, r_{k+1},$$

$$\mathbf{x}_{k+1} = \hat{\mathbf{x}} + \omega_k\, \mathbf{Q}_k\, r_{k+1}$$

$$\mathbf{Q}_{k+1}\, u_k = (\mathbf{I} - \omega_k \mathbf{A})\, \mathbf{Q}_k\, u_k$$

where ω_k minimizes $\|(\mathbf{I} - \omega\mathbf{A})\, \underbrace{\mathbf{Q}_k\, r_{k+1}}_{\substack{\| \\ \hat{\mathbf{r}}}}\, \|$

$$\omega_k = \frac{(\hat{\mathbf{r}}, \mathbf{A}\hat{\mathbf{r}})}{(\mathbf{A}\hat{\mathbf{r}}, \mathbf{A}\hat{\mathbf{r}})} \quad \text{and} \quad \theta_k = -\omega_{k-1}$$

[van der Vorst '92]

Bi-CGSTAB

Choose $\mathbf{x}_0$ and $\tilde{\mathbf{r}}_0$

$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$

$\mathbf{u} = 0$, $\quad \omega = \sigma = 1$

while $\|\mathbf{r}\| > tol$ do

$\sigma \leftarrow -\omega\sigma$

$\rho \leftarrow (\mathbf{r}, \tilde{\mathbf{r}}_0)$, $\quad \beta \leftarrow \rho/\sigma$

$\mathbf{u} \leftarrow \mathbf{r} - \beta\,\mathbf{u}$, $\quad \mathbf{c} \leftarrow \mathbf{A}\mathbf{u}$

$\sigma \leftarrow (\mathbf{c}, \tilde{\mathbf{r}}_0)$, $\quad \alpha \leftarrow \rho/\sigma$

$\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$

$\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$, $\quad \mathbf{s} \leftarrow \mathbf{A}\mathbf{r}$

$\omega \leftarrow (\mathbf{r}, \mathbf{s})/(\mathbf{s}, \mathbf{s})$

$\mathbf{u} \leftarrow \mathbf{u} - \omega\,\mathbf{c}$

$\mathbf{x} \leftarrow \mathbf{x} + \omega\,\mathbf{r}$

$\mathbf{r} \leftarrow \mathbf{r} - \omega\,\mathbf{s}$

end while

Properties Bi-CGSTAB

Advantages

- Hybrid Bi-CG.
- Converges faster (& smoother) than CGS.
- More accurate than CGS.
- 2 DOT/MV, 3 AXPY/MV.
- Storage: 6 large vectors.

Drawbacks Danger of

(A) Lanczos breakdown $(\rho_k = 0)$,

(B) pivot breakdown $(\sigma_k = 0)$,

(C) breakdown minimization $(\omega_k = 0)$.

Breakdown of the minimization

Exact arithmetic, $\omega_k = 0$:

– No reduction of residual by

$$\mathbf{Q}_{k+1}\, r_{k+1} = (\mathbf{I} - \omega_k \mathbf{A}\,)\,\mathbf{Q}_k\, r_{k+1}. \qquad (\star)$$

– q_{k+1} is of degree k:
Bi-CG scalars can not be computed;
breakdown of incorporated Bi-CG.

Finite precision arithmetic, $\omega_k \approx 0$:

– Poor reduction of residual by $(\star)$

– Bi-CG scalars are seriously affected by evaluation errors: drop of speed of convergence.

$\omega_k \approx 0$ to be expected if

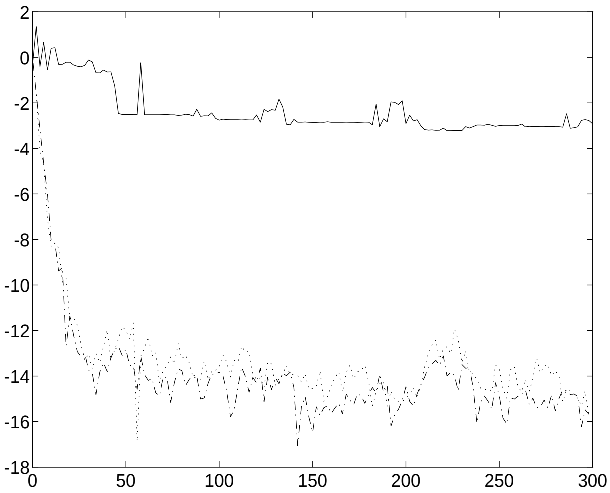
$\mathbf{A}$ has eigenvalues with rel. large imaginary part

ω_k is real!

$$\rho_k = (\mathbf{r}_k, \widetilde{\mathbf{r}}_0), \qquad \rho_k^\star = \rho_k(1+\epsilon)$$

Accurate Bi-CG coefficients

$$|\epsilon| \le n\,\overline{\xi}\,\frac{\|\,\mathbf{r}_k\,\|\,\|\,\widetilde{\mathbf{r}}_0\|}{|(\mathbf{r}_k, \widetilde{\mathbf{r}}_0)|} = \frac{n\,\overline{\xi}}{\widehat{\rho}_k} \quad \text{where } \widehat{\rho}_k := \frac{|(\mathbf{r}_k, \widetilde{\mathbf{r}}_0)|}{\|\,\mathbf{r}_k\|\,\|\widetilde{\mathbf{r}}_0\|}$$



For optimal accurate Bi-CG coefficient:

minimize $q(t) = \theta_k t^k + \ldots$

$$\frac{\|\,q_k(\mathbf{A})\,r_k\|\,\|\widetilde{\mathbf{r}}_0\|}{|(q_k(\mathbf{A})\,r_k, \widetilde{\mathbf{r}}_0)|} = \frac{\|\,q_k(\mathbf{A})\,r_k\|}{|\theta_k|} \cdot \frac{\|\widetilde{\mathbf{r}}_0\|}{|(r_k, (\mathbf{A}^*)^k\,\widetilde{\mathbf{r}}_0)|}$$

$$\begin{aligned}(q_k(\mathbf{A})\,r_k, \widetilde{\mathbf{r}}_0) &= (r_k, q_k(\mathbf{A}^*)\,\widetilde{\mathbf{r}}_0)\\ &= (r_k, \theta_k \cdot (\mathbf{A}^*)^k\,\widetilde{\mathbf{r}}_0)\end{aligned}$$

$$r_k \perp \mathcal{K}_k(\mathbf{A}^*; \widetilde{\mathbf{r}}_0)$$

Solution

$$\left.\begin{aligned} &q_k(\mathbf{A})\,r_k \perp \mathcal{K}_k(\mathbf{A}; r_k)\\ &q_k(0) = 1\end{aligned}\right\} \quad \mathbf{r}^{\mathrm{OR}} := q_k(\mathbf{A})\,r_k \text{ is an orthogonal residual}$$

For minimal residuals:

minimize

$$\| q_k(\mathbf{A})\, r_k \|$$

Solution

$$\left.\begin{array}{l} q_k(\mathbf{A})\, r_k \perp \mathbf{A}\mathcal{K}_k(\mathbf{A}; r_k) \\ q_k(0) = 1 \end{array}\right\}$$ $\mathbf{r}^{\mathrm{MR}} := q_k(\mathbf{A})\, r_k$ is a minimal residual

[Sleijpen, van der Vorst '95]

$$q_{k+1}(t) = (1 - \omega_k t) q_k(t)$$

$$\widehat{\mathbf{r}} = q_k(\mathbf{A}) r_{k+1}, \qquad \widehat{\omega} := \frac{(\,\widehat{\mathbf{r}}\,, \mathbf{A}\,\widehat{\mathbf{r}}\,)}{\|\,\widehat{\mathbf{r}}\,\|\,\|\,\mathbf{A}\,\widehat{\mathbf{r}}\,\|}$$

Bi-CGSTAB, I

For minimal residuals:

$$\|\,\mathbf{r}^{\mathrm{MR}}\| = \|(\mathbf{I} - \omega\mathbf{A})\,\widehat{\mathbf{r}}\,\| \quad \text{minimal for} \quad \omega = \omega^{\mathrm{MR}} = \widehat{\omega}\,\frac{\|\,\widehat{\mathbf{r}}\,\|}{\|\,\mathbf{A}\,\widehat{\mathbf{r}}\,\|}$$

For optimal accurate Bi-CG coefficient:

$$\frac{\|\,\mathbf{r}^{\mathrm{OR}}\|}{|\,\omega\,|} = \frac{\|(\mathbf{I} - \omega\mathbf{A})\,\widehat{\mathbf{r}}\,\|}{|\,\omega\,|} \quad \text{minimal for} \quad \omega = \omega^{\mathrm{OR}} = \frac{1}{\widehat{\omega}}\,\frac{\|\,\widehat{\mathbf{r}}\,\|}{\|\,\mathbf{A}\,\widehat{\mathbf{r}}\,\|}$$

[Sleijpen, van der Vorst '95]

$$q_{k+1}(t) = (1 - \omega_k t) q_k(t)$$

$$\widehat{\mathbf{r}} = q_k(\mathbf{A}) r_{k+1}, \qquad \widehat{\omega} := \frac{(\,\widehat{\mathbf{r}}\,, \mathbf{A}\,\widehat{\mathbf{r}}\,)}{\|\,\widehat{\mathbf{r}}\,\|\,\|\,\mathbf{A}\,\widehat{\mathbf{r}}\,\|}$$

Bi-CGSTAB, II

$$\|\,\mathbf{r}^{\mathrm{OR}}\| = \frac{1}{\widehat{\omega}}\,\|\mathbf{r}^{\mathrm{MR}}\|, \qquad \frac{\|\,\mathbf{r}^{\mathrm{MR}}\|}{|\,\omega^{\mathrm{MR}}\,|} = \frac{1}{\widehat{\omega}}\,\frac{\|\,\mathbf{r}^{\mathrm{OR}}\|}{|\,\omega^{\mathrm{OR}}\,|}$$

Compromise:

take

$$\mathbf{r} = \mathbf{r}_{k+1} = (\mathbf{I} - \omega\mathbf{A})\,\widehat{\mathbf{r}}$$

with

$$\omega = \mathrm{sign}(\widehat{\omega})\, \max(0.7, |\,\widehat{\omega}|)\,\frac{\|\,\widehat{\mathbf{r}}\,\|}{\|\,\mathbf{A}\,\widehat{\mathbf{r}}\,\|}$$

[Sleijpen, van der Vorst '95]

Updating by pols. of dg. $\ell \geq 2$

Consider $k = m\ell$.

$$q_{m\ell+\ell} = p_m q_{m\ell} = p_m \cdot p_{m-1} \cdot \ldots \cdot p_0,$$

where $\gamma_\ell \neq 0$

- p_m of exact degree ℓ, $p_m(\zeta) = \gamma_\ell\,\zeta^\ell + \ldots$
- $p_m(0) = 1$ and, with $\widehat{\mathbf{r}} := q_{m\ell}(\mathbf{A}) r_{m\ell+\ell}$,

---

For minimal residuals: [Sl., Fokkema '93]

- p_m minimizes $\|\, p_m(\mathbf{A})\ \widehat{\mathbf{r}}\,\|$ BiCGstab(ℓ)

For accurate ρ's: [Sl., van der Vorst '95]

- p_m minimizes $\dfrac{\|\, p_m(\mathbf{A})\,\widehat{\mathbf{r}}\,\|}{|\gamma_\ell|}$

Compromise can efficiently be computed

vanilla BiCGstab(ℓ)

[Sl Fokkema 93, Sl vdV Fokkema 94]

Updating by pol. dg. $\ell \geq 2$

Consider $\quad k = m\ell.$

$$q_{m\ell+\ell} = p_m q_{m\ell} = p_m \cdot p_{m-1} \cdot \ldots \cdot p_0,$$

where p_m of exact degree ℓ, $p_m(0) = 1$ and

p_m minimizes $\quad \|p_m(\mathbf{A})\, \underbrace{q_{m\ell}(\mathbf{A})\, r_{m\ell+\ell}}_{\widehat{\mathbf{r}}}\|.$

BiCGstab(ℓ) $\quad \ell \geq 2.$

With respect to the γ_i, minimize

$$\|\widehat{\mathbf{r}} - \sum_{j=1}^{\ell} \gamma_j\, \mathbf{A}^j\, \widehat{\mathbf{r}}\|, \quad \left(p_m(t) = 1 - \sum_{j=1}^{\ell} \gamma_j\, t^j \right)$$

Compute $\quad \mathbf{A}\widehat{\mathbf{r}},\ \mathbf{A}^2\widehat{\mathbf{r}},\ \ldots,\ \mathbf{A}^\ell\widehat{\mathbf{r}} \quad$ explicitly.

With $\mathbf{V} := \left[\mathbf{A}\widehat{\mathbf{r}} \,|\, \ldots \,|\, \mathbf{A}^\ell\widehat{\mathbf{r}}\right], \quad \vec{\gamma} := (\gamma_1, \ldots, \gamma_\ell)^T$

we have [Normal Equations, use Choleski]

$$(\mathbf{V}^*\mathbf{V})\vec{\gamma} = \mathbf{V}^*\widehat{\mathbf{r}}$$

Why using polynomial factors of degree ≥ 2?

Hybrid Bi-CG, faster than Bi-CGSTAB

1 sweep BiCGstab(ℓ) versus ℓ steps Bi-CGSTAB:

- Reduction with MR-polynomial of degree ℓ is better than $\ell\times$ MR-pol. of degr. 1.
- MR-polynomial of degree ℓ contributes only once to an increase of $\widehat{\rho}_k$

Why not?

- Efficiency:
  $1.75 + 0.25 \cdot \ell$ DOT/MV, $\quad 2.5 + 0.5 \cdot \ell$ AXPY/MV
  Storage: $\quad 2\ell + 5$ large vector.
- Loss of accuracy:
  $$\left|\|\mathbf{r}_k\| - \|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|\right| \leq \ldots + c\,\overline{\xi}\, \max\left(|\gamma_i|\, \left\|\, |\mathbf{A}|\, |\mathbf{A}^{i-1}\,\widehat{\mathbf{r}}|\, \right\|\right)$$
- break-downs are possible

Conclusions

Hybrid Bi-CG meths are efficient linear solvers.

The auxiliary polynomial can be used

- to gain efficiency and
- to improve residual reduction

but also for improving other properties

- to obtain more accurate Bi-CG coefficients
- to improve inexact Newton schemes

Effects of erratic convergence can be reduced:

- Inaccurate results due to large intermediate residuals can inexpensively be avoided.
- Large residual peaks may deteriorate convergence, but can be avoided (BiCGstab(ℓ), GCGS).

Publications

1. G.L.G. Sleijpen and D.R. Fokkema, *BiCGstab(ℓ) for Linear Equations involving Unsymmetric Matrices with Complex Spectrum*, ETNA, 1 (1993), pp. 11–32.
2. G.L.G. Sleijpen, H.A. van der Vorst, and D.R. Fokkema, *BiCGstab(ℓ) and other Hybrid Bi-CG Methods*, Numerical Algor., 7 (1994), pp. 75–109.
3. G.L.G. Sleijpen and H.A. van der Vorst, *Reliable updated residuals in hybrid Bi-CG methods*, Computing, 56 (1996), pp. 141–163.
4. D.R. Fokkema, G.L.G. Sleijpen, and H.A. van der Vorst, *Generalized Conjugate Gradient Squared*, J. Comput. Appl. Math., 71 (1996), pp. 125–146.
5. G.L.G. Sleijpen and H.A. van der Vorst, *Maintaining convergence properties of BiCGstab methods in finite precision arithmetic*, Numerical Algor., 10 (1995), pp. 203–223.
6. G.L.G. Sleijpen and H.A. van der Vorst, *Hybrid Bi-Conjugate Gradient Methods for CFD Problems*, pp. 457–476, in *Computational Fluid Dynamics Review 1995*, M. Hafez and K. Oshima, ed., John Wiley & Sons Ltd, Chichester, 1995.

http://www.math.ruu.nl/people/sleijpen