ERG, University of Nottingham, February 27, 2001

Iterative solvers for Ax=b
Efficient and smooth convergence

Universiteit Utrecht
Department of Mathematics

Gerard Sleijpen
Diederik Fokkema
Henk van der Vorst

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

APPROXIMATE SYSTEMS

$\mathbf{Ax} = \mathbf{b}$: replace by $\mathbf{Ky}^{(0)} = \mathbf{b}$
solve for $\mathbf{y}^{(0)}$. Then try to solve

$$\mathbf{A}(\mathbf{y}^{(0)} + \mathbf{z}) = \mathbf{b}$$

or

$$\mathbf{Az} = \mathbf{b} - \mathbf{Ay}^{(0)}.$$

Same trick: solve $\mathbf{Ky}^{(1)} = \mathbf{b} - \mathbf{Ay}^{(0)}$

$\Rightarrow$ ITERATIVE PROCESS

Formally $\mathbf{A} = \mathbf{K} - (\mathbf{K} - \mathbf{A})$
Hence $\mathbf{Kx} = \mathbf{b} + (\mathbf{K} - \mathbf{A})\mathbf{x}$

Iteration:

$$\begin{aligned}\mathbf{Kx}_{i+1} &= \mathbf{Kx}_i + \mathbf{b} - \mathbf{Ax}_i \\ &= \mathbf{Kx}_i + \mathbf{r}_i\end{aligned}$$

Iterative process for $\mathbf{Ax} = \mathbf{b}$:

$$\mathbf{Kx}_{i+1} = \mathbf{Kx}_i + \mathbf{r}_i$$

KRYLOV SUBSPACES

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{K}^{-1}\mathbf{r}_i, \qquad \mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{A}(\mathbf{K}^{-1}\mathbf{r}_i)$$

Starting with $\mathbf{x}_0 = \mathbf{0}$ and using $\mathbf{r}_i = \mathbf{b} - \mathbf{Ax}_i$: $\mathbf{x}_i$ is in

$\text{span}\{\mathbf{K}^{-1}\mathbf{b}, (\mathbf{K}^{-1}\mathbf{A})\mathbf{K}^{-1}\mathbf{b}, \ldots, (\mathbf{K}^{-1}\mathbf{A})^{i-1}\mathbf{K}^{-1}\mathbf{b}\}$

KRYLOV SUBSPACE $\mathcal{K}_i(\mathbf{K}^{-1}\mathbf{A}; \mathbf{K}^{-1}\mathbf{b})$ for
Preconditioned System $\mathbf{K}^{-1}\mathbf{Ax} = \mathbf{K}^{-1}\mathbf{b}$

Basic iteration corrects only with last vector

CAN WE DO BETTER?

For simplicity take $\mathbf{K} = \mathbf{I}$

$$\mathbf{Ax} = \mathbf{b}, \quad \mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$$

$$\begin{aligned}\mathbf{x}_i &= \mathbf{x}_0 + \text{span}\{\mathbf{r}_0, \mathbf{Ar}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{i-1}\mathbf{r}_0\} \\ &= \mathbf{x}_0 + \mathcal{K}_i(\mathbf{A}; \mathbf{r}_0)\end{aligned}$$

KRYLOV SUBSPACE METHODS

a *Ritz-Galerkin*:
$\mathbf{b} - \mathbf{Ax}_i \perp \mathcal{K}_i(\mathbf{A}; \mathbf{r}_0)$
CG, Lanczos, **FOM**, GENCG

b *Minimum Residual*:
$\|\mathbf{b} - \mathbf{Ax}_i\|_2$ minimal over $\mathcal{K}_i(\mathbf{A}; \mathbf{r}_0)$
MINRES, **GMRES**, GCR

c *Petrov-Galerkin*:
$\mathbf{b} - \mathbf{Ax}_i \perp \mathcal{K}_i(\mathbf{A}^T; \tilde{\mathbf{r}}_0)$
Bi-CG, QMR

HOW TO SELECT

for $\mathbf{A}\mathbf{x} = \mathbf{b}$, with $\mathbf{A} \neq \mathbf{A}^T$, $\mathbf{A} \in \mathbb{R}^{n \times n}$?

1. If overhead no problem: **GMRES**

2. Too much overhead:
   QMR, **BiCGSTAB**,
   TFQMR, **CGS**, **Bi-CGSTAB(ℓ)**

3. Variable preconditioning:
   GMRESR, **FGMRES**

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 $\mathbb{A}$ 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 $\mathbb{A}^*$.

Properties hybrid Bi-CG

Advantages

- Converges often twice as fast as Bi-CG w.r.t. # MVs:

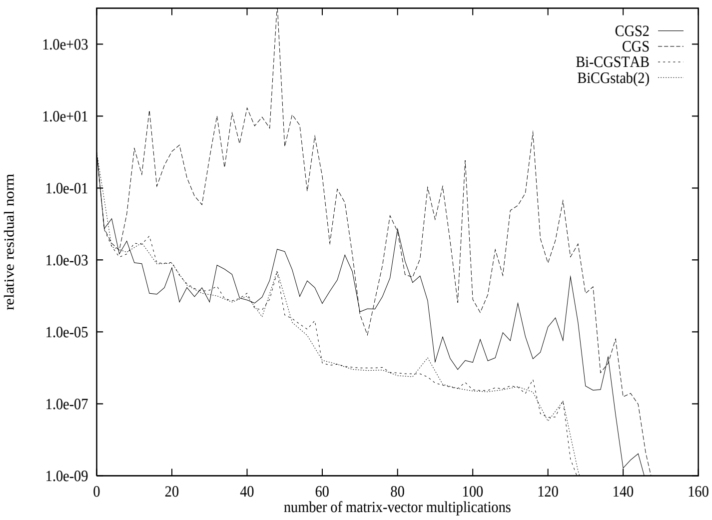
each MV expands the search subspace
Bi-CG: $\mathbf{x}_k - \mathbf{x}_0 \in \mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)$ á 2k MV.
Hybrid Bi-CG: $\mathbf{x}_k - \mathbf{x}_0 \in \mathcal{K}_{2k}(\mathbf{A}; \mathbf{r}_0)$ á 2k MV.

- 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.

Example from device simulation: dr15c
469741 non-zeros, ILU(0) preconditioner



near-breakdown may lead to

- inaccurate coefficients (ρ_k or σ_k)
- peaks in the convergence

Inaccurate coefficients

- Loss of speed of convergence

Peaks in the 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.

Solve x from $\boxed{\mathbf{A}x = b}$

Choose $x_0, \ldots$
Compute $r_0 = b - \mathbf{A}\,x_0$
$\mathbf{x} = x_0$, $\mathbf{r} = r_0$
......
While $\|\mathbf{r}\| > tol$ do
  compute $\mathbf{p}$ *and* $\mathbf{A}\mathbf{p}$
  $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{A}\mathbf{p}$
  $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{p}$
end while

Objective:

x gets a free ride

- $\|\mathbf{r}\| \to 0$,
- update $\mathbf{r}$ by $\mathbf{A}\mathbf{p}$ with $\mathbf{p}$ explicitly known.
- $\mathbf{p}$ depends on $\mathbf{x}$ only via x_0

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

Rounded arithmetic

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

$$\mathbf{r}^\star_{k+1} = \mathbf{r}^\star_k - \mathbf{A}\,\mathbf{p}^\star_k + \mathbf{f}_k$$

$\mathbf{f}_k$ is the **local** rounding error

Theorem. assuming MV main source of error

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

where $\overline{\xi}$ 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. $\|\mathbf{f}_k\| \lesssim \overline{\xi}\, n_A \,\| |\mathbf{A}| \|\, \|\mathbf{p}^\star\| \leq \overline{\xi}\,\Gamma\,\|\mathbf{A}\mathbf{p}^\star_k\|$

$\mathbf{r}$ is the chauffeur, $\mathbf{x}$ is his cargo

When it gets rough

(1) $\mathbf{r}$ may get off the track

(2) $\mathbf{r}$ may lose its cargo

Remedies

(1) Install (better) shock absorbers

(2) Tie the cargo more securely

(3) Turn the bolds more tightly

$\mathbf{r}$ is the chauffeur, $\mathbf{x}$ is his cargo

When it gets rough

(1) $\mathbf{r}$ may get off the track

(2) $\mathbf{r}$ may lose its cargo

Remedies

(1) Install (better) shock absorbers

(2) Tie the cargo more securely

(3) Turn the bolds more tightly

[Sl vdV Fokkema 94, Greenbaum 94 96]

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

In rounded arithmetic

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

Accuracy

True residual

$$\mathbf{b} - \mathbf{A}\mathbf{x}^\star_{j+1} = \mathbf{b} - \mathbf{A}\mathbf{x}^\star_j - \mathbf{A}\mathbf{p}^\star_j$$

and recursively updated residual

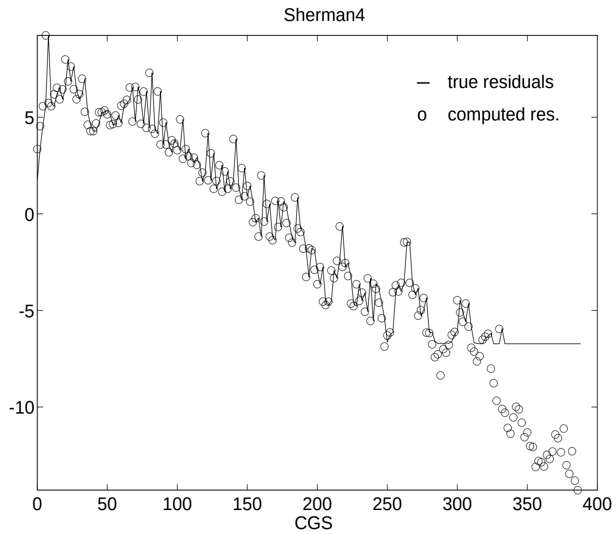
$$\mathbf{r}^\star_{j+1} \quad = \quad \mathbf{r}^\star_j \quad - \mathbf{A}\mathbf{p}^\star_j + \mathbf{f}_j$$

drift apart.

Theorem.

$$\boxed{\|(\mathbf{b} - \mathbf{A}\mathbf{x}^\star_k) - \mathbf{r}^\star_k\| \leq 2\,\overline{\xi}\,\Gamma \sum_{j \leq k} \|\mathbf{r}^\star_j\| \qquad \text{where} \quad \Gamma := n_A \,\Big\|\,|\mathbf{A}|\,\Big\|\,\Big\|\,\mathbf{A}^{-1}\Big\|}$$

Large residuals in (hybrid) Bi-CG



[Sl vdV 96, vdV Ye 99]

Avoiding inaccurate solutions

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

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

Additional recursion error:

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

Disadvantages:

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

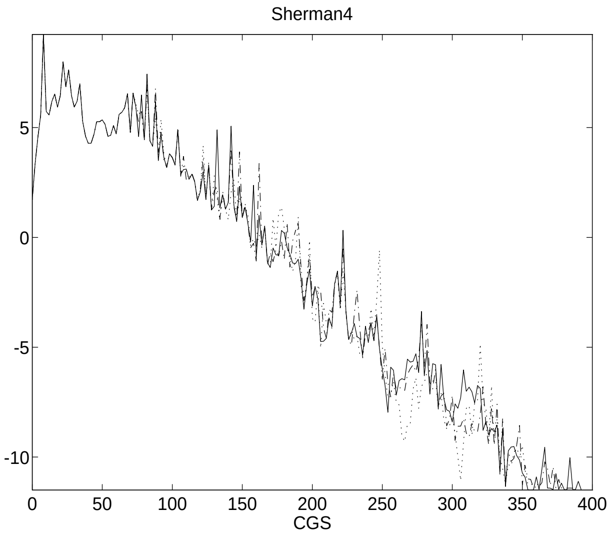
Use $\quad \mathbf{r}_{k+1} \leftarrow \mathbf{b} - \mathbf{A}\,\mathbf{x}_{k+1} \quad$ occasionally!

Additional recursion error:

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

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

Use $\mathbf{r}_{k+1} \leftarrow \mathbf{b} - \mathbf{A}\,\mathbf{x}_{k+1}$ only if
the residual-norm decreases significantly
and is not (much) smaller than $\|\,\mathbf{b}\,\|$.



7 true residual computations

$\mathbf{r}$ is the chauffeur, $\mathbf{x}$ is his cargo

When it gets rough
(1) $\mathbf{r}$ may get off the track
(2) $\mathbf{r}$ may lose its cargo

Remedies

(1) Install (better) shock absorbers
(2) Tie the cargo more securely

(3) Turn the bolds more tightly

Smoothing the convergence

Cosmetic approaches:

$$\begin{array}{cccc} \mathbf{r}_0, & \mathbf{r}_1, & \mathbf{r}_2, & \ldots \\ \downarrow & \downarrow & \downarrow & \\ \tilde{\mathbf{r}}_0, & \tilde{\mathbf{r}}_1, & \tilde{\mathbf{r}}_2, & \ldots \end{array}$$

Ex.: $\tilde{\mathbf{r}}_j \equiv \mathbf{r}_{k_j}$ where $\|\mathbf{r}_{k_j}\|_2 = \min_{k\leq j} \|\mathbf{r}_j\|_2$
$\tilde{\mathbf{x}}_j \equiv \mathbf{x}_{k_j}$

More elaborated approaches by
Weis 90 (MR), Zhou & Walker 94
"Quasi-variants" by
Zhou & Walker 94, Freund & Nacht. 91 (QMR)

For an analysis, see Gutknecht & Rozložník 99

Effective smoothing:

$$\begin{array}{ccccccc} \mathbf{r}_0, & & \mathbf{r}_1, & & \mathbf{r}_2, & & \ldots \\ \downarrow & \nearrow & \downarrow & \nearrow & \downarrow & \nearrow & \\ \tilde{\mathbf{r}}_0, & & \tilde{\mathbf{r}}_1, & & \tilde{\mathbf{r}}_2, & & \ldots \end{array}$$

Bi-CGSTAB, Bi-CGstab(ℓ), ... [V 92, F & Sl 93]

[Sonneveld '89]

Transpose-free Bi-CG

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

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

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

(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 $\hat{\mathbf{u}}_k = \mathbf{Q}_k \mathbf{u}_k$ and $\hat{\mathbf{r}}_k = \mathbf{Q}_k \mathbf{r}_{k+1}$

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

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

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

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

Compute

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

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

[van der Vorst 92]

Smoothing the convergence: Bi-CG STABilized

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

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

$$\begin{aligned} \mathbf{Q}_{k+1}\, r_{k+1} &= (\mathbf{I} - \omega_k \mathbf{A})\, \mathbf{Q}_k\, r_{k+1}, \\ & \mathbf{x}_{k+1} = \hat{\mathbf{x}} + \omega_k\, \mathbf{Q}_k\, r_{k+1} \\ \mathbf{Q}_{k+1}\, u_k &= (\mathbf{I} - \omega_k \mathbf{A})\, \mathbf{Q}_k\, u_k \end{aligned}$$

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

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

VARIANTS OF BiCG (2)

BiCGSTAB

[van der Vorst '92]

- 2 MV's in BiCG can be used to combine BiCG with repeated GMRES(1)
- same costs as BiCG
- often much faster than BiCG
- much smoother than CGS
- often faster than CGS
- breakdown when GMRES(1) stagnates
- poor when GMRES(1) is very poor

$\mathbf{r}$ is the chauffeur, $\mathbf{x}$ is his cargo

When it gets rough

(1) $\mathbf{r}$ may get off the track

(2) $\mathbf{r}$ may lose its cargo

Remedies

(1) Install better shock absorbers

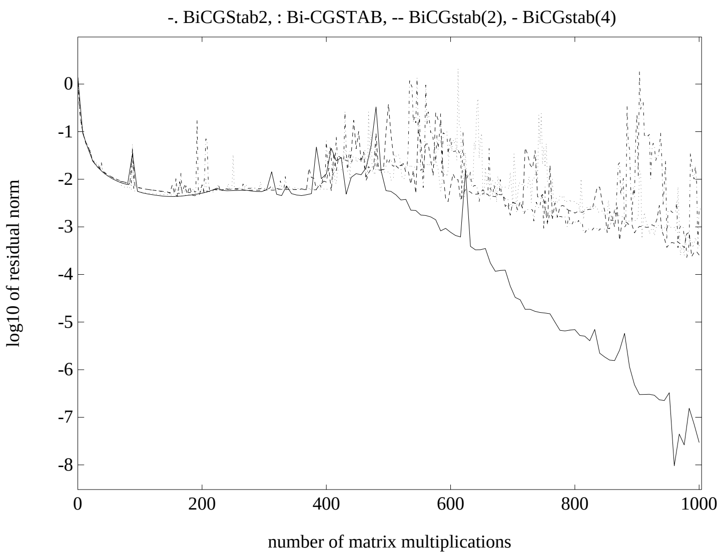
(2) Tie the cargo more securely

(3) Turn the bolds more tightly

VARIANTS OF BiCG (3)

BiCGSTAB(2) [Sleijpen, Fokkema '93]

- after each two BiCG steps: GMRES(2)
- often faster than BiCGSTAB
- also for near skew-symm. matrices
- $\approx$ same costs as BiCG
- easy to generalize: BiCGSTAB(ℓ)
- BiCGSTAB(4): fast and robust
- breakdown when GMRES(ℓ) stagnates



$$-\varepsilon(u_{xx}+u_{yy})+a(x,y)u_x+b(x,y)u_y=0,$$

on $[0,1]\times[0,1]$, Dirichlet BC $\varepsilon=10^{-1}$

$$a(x,y)=4x(x-1)(1-2y),$$
$$b(x,y)=4y(1-y)(1-2x),$$

$u(x,y)=\sin(\pi x)+\sin(13\pi x)+\sin(\pi y)+\sin(13\pi y)$

(201×201) volumes, no preconditioning

$\mathbf{r}$ is the chauffeur, $\mathbf{x}$ is his cargo

When it gets rough

(1) $\mathbf{r}$ may get off the track

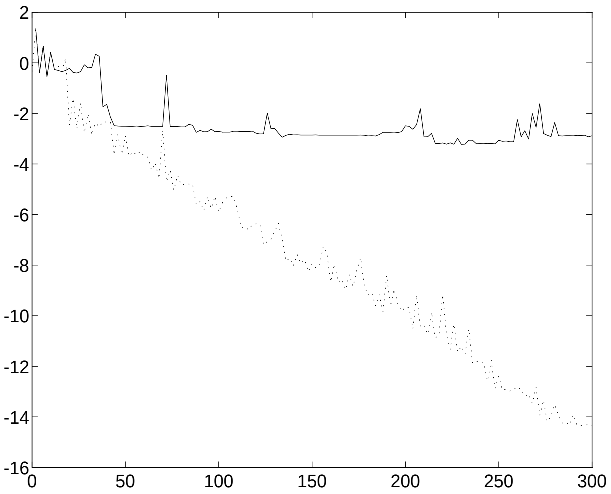
(2) $\mathbf{r}$ may lose its cargo

Remedies

(1) Install (better) shock absorbers

(2) Tie the cargo more securely

(3) Turn the bolds more tightly



$$u_{xx} + u_{yy} + u_{zz} + 1000\, u_x = f.$$

f is defined by the solution
$u(x, y, z) = \exp(xyz) \sin(\pi x) \sin(\pi y) \sin(\pi z)$.

Discretized using $(10 \times 10 \times 10)$ volumes.
No preconditioning was used.

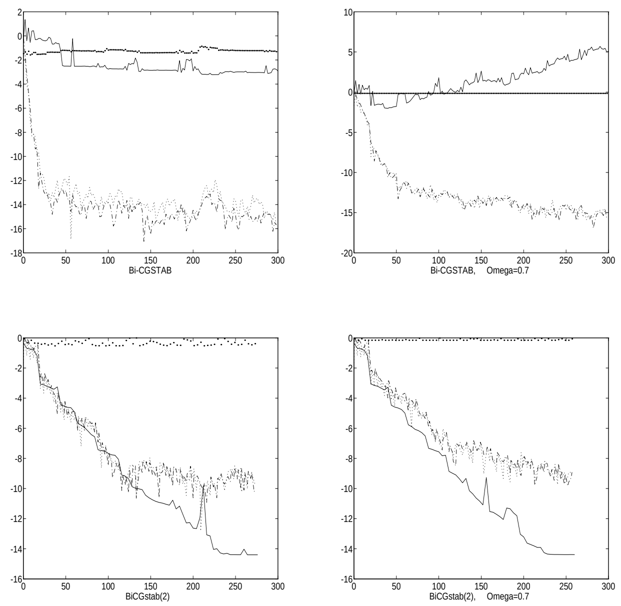
AVOIDING BREAKDOWN

two reasons for breakdown
in Bi-CGSTAB methods

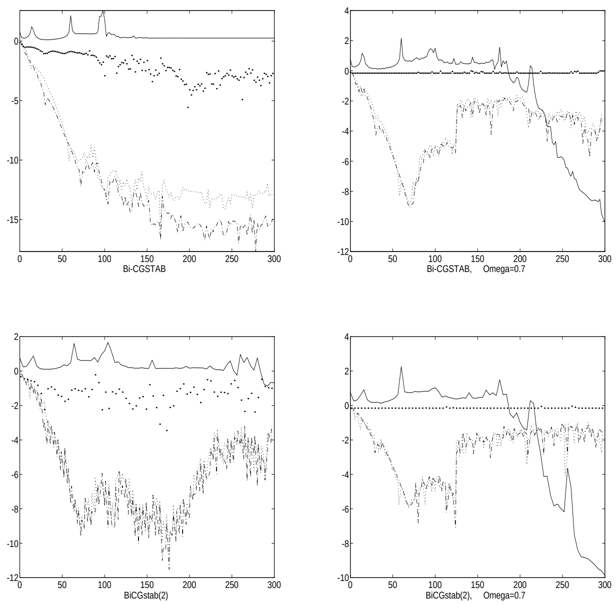
(1) Bi-CG part may break-down:
Look-ahead techniques (complicated)

(2) GMRES-part gives no reduction
no expansion of Krylov subspace

Sleijpen and van der Vorst '95:
Combination of GMRES and FOM



$u_{xx}+u_{yy}+u_{zz}+1000\, u_x = f$, $\quad (10\times10\times10)$ vol.



$$-u_{xx} - u_{yy} + \gamma(x u_x + y u_y) + \beta u = f$$

$\gamma = 100 \quad \& \quad \beta = -200, \quad (63 \times 63)$ vol.

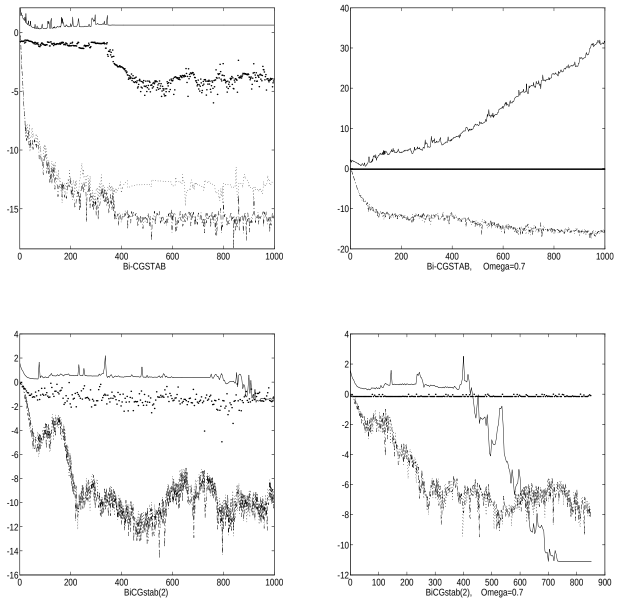
$$-u_{xx} - u_{yy} + \gamma(xu_x + yu_y) + \beta u = f$$

on unit-square, Dirichlet BC,

with discretized solution $x = (1, 1, \ldots, 1)^T$,

$\gamma = 1000$ and $\beta = 10$.

(66×66) volumes. No preconditioning.



$$-u_{xx} - u_{yy} + \gamma(xu_x + yu_y) + \beta u = f$$

$\gamma = 1000$ & $\beta = 10$, (66×66) vol

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.