Utrecht, 14 december 2016

Scientific Computing

Gerard Sleijpen

Universiteit Utrecht
Department of Mathematics

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

Computational Science

- Design and analysis of numerical methods that are
  robust, accurate, efficient and versatile
- Focus on a large class of problems (as linear equations, least square problems, eigenvalue problems, . . .)
- Focus on one (class of) numerical method(s)

Scientific Computing

- Design and analysis of numerical methods that are
  accurate and efficient
- specifically for one family of practical problems only (as Navier Stokes, electronics, . . .)
- Combines several methods, methods that are most suitable for specific subproblems. Find optimal parameters.

As we will learn today, problems from SC may lead to interesting new classes of problems in CS.

Program

- Multigrid (PDEs)
- Compressed Sensing (MRI)
- Model order reduction (Electronics)
- Relax to the max (QCD)

Sparse reconstruction

With $\mathbf{A}$ an $n \times n$ matrix and $\mathbf{b}$ an n-vector: $\mathbf{A}\mathbf{x}^\star = \mathbf{b}$.

Suppose for $\mathcal{J} \subset \{1, 2, \ldots, n\}$ only $\mathbf{b}(\mathcal{J})$ is available.

Can we solve $(*)$ $\mathbf{A}(\mathcal{J}, :)\mathbf{x} = \mathbf{b}(\mathcal{J})$ to obtain $\mathbf{x}^\star$?

However, in some applications it is known that

$$\|\mathbf{x}^\star\|_0 \equiv \#\{i \mid x_i^\star \neq 0\} \ll n,$$

the solution vector $\mathbf{x}^\star$ is **sparse**.

Idea. Solve $\min \|\mathbf{x}\|_0$ such that $\mathbf{A}(\mathcal{J}, :)\mathbf{x} = \mathbf{b}(\mathcal{J})$.

Is $\mathbf{x}_{\min} = \mathbf{x}^\star$?

Compressed sensing: an example

Let $n = 2^m$ and let $\mathbf{x}^\star$ be such that $\|\mathbf{x}^\star\|_0 = 3$.
Can we "sense" $\mathbf{x}^\star$ by only inspecting a few coordinates?

Select $\mathcal{J} \subset \{1, \ldots, n\}$ randomly such that $|\mathcal{J}| = 8$.

a) Take $\mathbf{x}(\mathcal{J}) \equiv \mathbf{x}^\star(\mathcal{J})$ and $\mathbf{x}$ is zero elsewhere.

b) View $\mathbf{x}^\star$ as a function on $\{1, 2, \ldots, 2^m\}$.
Let $\mathbf{A} = \mathcal{F}$ be the Fourier transform, $\mathbf{b} = \mathbf{A}\mathbf{x}^\star = \hat{\mathbf{x}}^\star$.
Find $\mathbf{x}$ such that $\|\mathbf{x}\|_0 = 3$ and $\hat{\mathbf{x}}(\mathcal{J}) = \hat{\mathbf{x}}^\star(\mathcal{J})$.

What is the probability that $\mathbf{x} = \mathbf{x}^\star$?

In, for instance MRI, $\mathbf{A}$ is a 2 (or 3) dimensional Fourier transform. The MRI scanner measures $\mathbf{b}$, the Fourier transform of the (discretized) density function $\mathbf{x}$ of water in the scanned tissue. Measuring $\mathbf{b}$ only partially would reduce the scanning time (by a factor $|\mathcal{J}|/n$). With respect to, for instance, an appropriate wavelet basis, $\mathbf{x}$ is sparse, i.e., $\|\mathbf{x}\|_0 \ll n$. However the value of $\|\mathbf{x}\|_0$ is unknown.

Sparse reconstruction

$$\mathbf{x}_{\min} \equiv \operatorname{argmin}\{\|\mathbf{x}\|_1 \mid \mathbf{x} \in \mathbb{R}^n \text{ st } \mathbf{A}(\mathcal{J},:)\mathbf{x} = \mathbf{A}(\mathcal{J},:)\mathbf{x}^\star\}$$

For $\mathbf{A} = \mathcal{F}$ (1,2, or 3-d), $d < k < n$ and random $\mathcal{J} \subset \{1, \ldots, n\}$, $|\mathcal{J}| = k$, consider the following statement.

Statement.
For all $\mathbf{x}^\star$ with $\|\mathbf{x}^\star\|_0 = d$, we have that $\mathbf{x}^\star = \mathbf{x}_{\min}$.

Theorem. The statement holds with high probability, where the probability depends on the ratios $d : k : n$.

In practice $\widetilde{\mathbf{b}} \equiv \mathbf{A}(\mathcal{J},:)\mathbf{x}^\star + \delta_b$ is measured with $\|\delta_b\|_2 \leq \delta$.
Solve $\min \|\mathbf{x}\|_1$ such that $\|\mathbf{A}(\mathcal{J},:)\mathbf{x} - \widetilde{\mathbf{b}}\|_2 \leq \delta$.
The following statement holds with high probability.

Statement. For some modest constant κ, we have
$\|\mathbf{x}^\star - \mathbf{x}_{\min}\|_2 \leq \kappa\delta$ for all $\mathbf{x}^\star$ with $\|\mathbf{x}^\star\|_0 = d$.

Sparse reconstruction

Some general observations.

- Similar results for some other classes of matrices.
- Whether minimisation resolves $\mathbf{x}^\star$ depends on $\mathbf{A}(\mathcal{J},:)$
  ($\mathbf{A} = \mathbf{I}$ not solvable, $\mathbf{A} = \mathcal{F}$ solvable)
  and the number of non-zeros of $\mathbf{x}^\star$, not on the values or the location (index) of the non-zeros.
- Some randomness (in selecting $\mathcal{J}$) is required.
- $\mathbf{A}$ (that is, $\mathbf{A}(\mathcal{J},:)$) will not be sparse
  (otherwise $\mathbf{b}$ is sparse if $\mathbf{x}$ is sparse).

  Nevertheless $\mathbf{c} = \mathbf{A}(\mathcal{J},:)\mathbf{u}$ might be efficiently computable
  (if $\mathbf{A} = \mathcal{F}$ then FFT can be used
  to compute $\mathbf{c}' \equiv \mathbf{A}\mathbf{u}$ and $\mathbf{c} = \mathbf{c}'(\mathcal{J})$).

Constrained 1-norm minimisation

Let $\mathbf{A}$ be an $k \times n$ matrix, $k < n$ and $\mathbf{b}$ a k-vector.
With $f(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2$ and $g(\mathbf{x}) \equiv \|\mathbf{x}\|_1$, solve

$$\mathbf{x}_{\min} = \operatorname{argmin}\{g(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n \text{ st } f(\mathbf{x}) = \delta^2\}$$

Standard approach. Find $\lambda \in \mathbb{R}$ (**Lagrange multiplier**) and $\mathbf{x}$ (solution) that solve the **Lagrange equation**

$$\nabla f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = \mathbf{0} \quad \& \quad f(\mathbf{x}) = \delta^2.$$

Note that, with the solution $\tau = \lambda$, the minimizer $\mathbf{x}$ of $f + \tau g$,

$$\mathbf{x}_{\min} = \operatorname{argmin}\{f(\mathbf{x}) + \tau g(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n\}$$

also solves the Lagrange equation.

However, g is **not** differentiable.

1-norm regularisation

Let $\mathbf{A}$ be an $k \times n$ matrix, $k < n$ and $\mathbf{b}$ a k-vector.
With $f(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2$ and $g(\mathbf{x}) \equiv \|\mathbf{x}\|_1$, solve

$$\mathbf{x}_{\min} = \operatorname{argmin}\{\rho(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n\} \quad \text{with} \quad \rho \equiv f + \tau g.$$

Example. Soft threshholding

$$s_\tau(\mathbf{b})_j \equiv \operatorname{sign}(b_j)\, \max(|b_j| - \tau, 0)$$

solves the minimisation for $\mathbf{A} = \mathbf{I}$:

$$s_\tau(\mathbf{b}) = \operatorname{argmin}\{\tfrac{1}{2}\|\mathbf{b} - \mathbf{x}\|_2^2 + \tau\|\mathbf{x}\|_1 \mid \mathbf{x} \in \mathbb{R}^n\}$$

Proof. Find $t_{\min} = \operatorname{argmin}\{\frac{1}{2}|\beta - t|^2 + \tau\,|t| \mid t \in \mathbb{R}\}$.

1-norm regularisation

Let $\mathbf{A}$ be an $k \times n$ matrix, $k < n$ and $\mathbf{b}$ a k-vector.
With $f(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2$ and $g(\mathbf{x}) \equiv \|\mathbf{x}\|_1$, solve

$$\mathbf{x}_{\min} = \operatorname{argmin}\{\rho(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n\} \quad \text{with} \quad \rho \equiv f + \tau g.$$

Towards an iterative solution method.
Let $\mathbf{x}_0$ be an approximate solution.
For updating $\mathbf{x}_0$, use Taylor expansion to bound f.
$\nabla f(\mathbf{x}) = -\mathbf{A}^*(\mathbf{b} - \mathbf{A}\mathbf{x})$. Select an $L \geq \lambda_{\max}(\mathbf{A}^*\mathbf{A})$. Then

$$\begin{aligned} f(\mathbf{x}) &\leq f(\mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0, \nabla f(\mathbf{x}_0)) + \tfrac{1}{2}L\|\mathbf{x} - \mathbf{x}_0\|_2^2 \\ &= d_0 + \tfrac{1}{2}L\|\mathbf{x} - \mathbf{x}_0'\|_2^2, \end{aligned}$$

where

$$d_0 \equiv f(\mathbf{x}_0) - \tfrac{1}{2L}\|\nabla f(\mathbf{x}_0)\|_2^2, \quad \mathbf{x}_0' \equiv \mathbf{x}_0 - \tfrac{1}{L}\nabla f(\mathbf{x}_0).$$

In particular,

$$\min \rho(\mathbf{x}) \leq d_0 + \min\{\tfrac{1}{2}L\|\mathbf{x} - \mathbf{x}_0'\|_2^2 + g(\mathbf{x})\} \leq \rho(\mathbf{x}_0).$$

The second term is minimised by $\mathbf{x}_1 \equiv s_{\tau/L}(\mathbf{x}_0')$.

Dynamical systems. The (given) quantities $(\mathbf{A}, \mathbf{E}, \mathbf{b}, \mathbf{c}, d)$ define a linear, time invariant, dynamical system:

$$\begin{cases} \mathbf{E}\,\dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{b}\,u(t) \\ y(t) &= \mathbf{c}^*\mathbf{x}(t) + d\,u(t) \end{cases}$$

$\mathbf{A}$ (**state space**) and $\mathbf{E}$ are $n \times n$ matrices
n is the number of **states** or **order** of the system.

$\mathbf{E}$ may be non-singular, but $(\mathbf{A}, \mathbf{E})$ is a regular pencil
$s \rightsquigarrow \det(s\mathbf{E} - \mathbf{A})$ not trivial on $\mathbb{C}$.

$\mathbf{b}$ (**input**), $\mathbf{c}$ (**output**) n-vectors,

$d \in \mathbb{R}$, $t \rightsquigarrow u(t)$ given real-valued (**control**) function on $\mathbb{R}$.

The function $t \rightsquigarrow y(t)$ is the function of interest:
the **output** of the system.

$$\begin{cases} \mathbf{E}\,\dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{b}\,u(t) \\ y(t) &= \mathbf{c}^*\mathbf{x}(t) + d\,u(t) \end{cases}$$

Examples

Electrical Circuits.

Characteristics.

- In electronic chips: n is huge, $10^4 \sim 10^8$.
- $\mathbf{A}$, $\mathbf{E}$ are sparse, G is not structured.
- Entries of $\mathbf{A}$ and $\mathbf{E}$ do *not* vary smoothly, neighboring entries may differ order of magnitudes.

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{b}\,u(t) \\ y(t) &= \mathbf{c}^*\mathbf{x}(t) + d\,u(t) \end{cases}$$

Examples

Power systems.

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{b}\,u(t) \\ y(t) &= \mathbf{c}^*\mathbf{x}(t) + d\,u(t) \end{cases}$$

Examples

Technical constructions.

Structural mechanics $\rightsquigarrow$

set of partial differential equation.

Discretization of the spatial part $\rightsquigarrow$

dynamical system.

Input from forcing acting on certain points,

Interested in the response at certain points (output).

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{b}\,u(t) \\ y(t) &= \mathbf{c}^*\mathbf{x}(t) + d\,u(t) \end{cases}$$

b is $n \times 1$, u is real-valued **single input**

c is $n \times 1$, y is real-valued **single outut**:

Single **I**nput **S**ingle **O**utput (SISO) system.

In practice.

b is $n \times m$, u is $\mathbb{R}^m$-valued **multiple input**

c is $n \times p$, y is $\mathbb{R}^p$-valued **multiple outut**:

Multiple **I**nput **M**ultiple **O**utput (MIMO) system.

Non-linear (apply linearization), n in the range $10^4 - 10^8$.

A, **E** are sparse, and (often) unstructured.

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{b}\,u(t) \\ y(t) &= \mathbf{c}^*\mathbf{x}(t) + d\,u(t) \end{cases}$$

Transfer function

Analysis strategy. Apply Laplace transform:

for $s \in \mathbb{C}$, consider $u(t) \equiv e^{st}$ $(t \in \mathbb{R})$

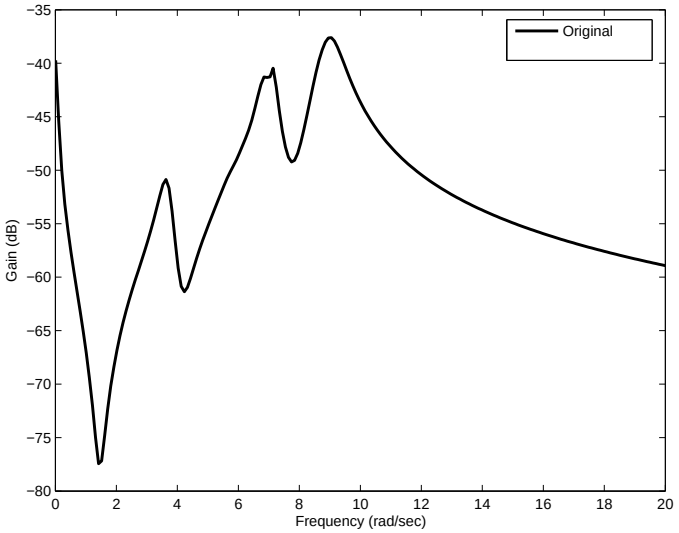
then $\quad \mathbf{x}(t) = (s\mathbf{E} - \mathbf{A})^{-1}\mathbf{b}\,e^{st}$

and $\quad y(t) = \left[\mathbf{c}^*(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} + d\right] e^{st}$

$$\boxed{H(s) \equiv \mathbf{c}^*(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} + d}$$

is the **transfer function** of the system.

New England test system



Bode plot: $\omega \rightsquigarrow |H(2\pi i\omega)|$ on decibel scale.
$n = 66$ (Solid line —).

$$H(s) \equiv \mathbf{c}^*(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} + d$$

The transfer function describes

how the system responses at the output

to an harmonic oscillation with frequency ω if $s = 2\pi i\omega$.

Computational obstacles.

- n large
- H needed for a wide range of ω (i.e., $s = 2\pi i\omega \in i\mathbb{R}$),
- preconditioners are hard to include
(preconditioner for $\mathbf{A}$ is not a preconditioner for $\mathbf{A} - s\mathbf{E}$),
- solutions required for a number of systems
(in a design stage).

Model Order Reduction

Find a kth order system $(\widetilde{A}, \widetilde{E}, \widetilde{b}, \widetilde{c}, d)$ with $k \ll n$ such that

- $\|y(t) - \widetilde{y}(t)\|$ 'small' all u
(2-norm, Hankel-norm,...)
- preservation of (physical and numerical) properties
(as, stability, passivity,...)
- computationally efficient and stable
- cheap measurement for the error
(when constructing reduced system)

- preserve structure (from 2nd order system)
- realizable
- fit in existing simulation software

Approaches

MOR

- Balanced truncation,
Hankel norm appr.
- Padé approximation,
moment matching
- Modal approximation

Eigenvalue computation

- QZ (dense systems)
(n not large)
- **Bi-Lanczos & Arnoldi**
- (Jacobi–)Davidson

Arnoldi

- $\mathbf{V}_k$ orthonormal, spans $\mathcal{K}_k((s_0\mathbf{I} - \mathbf{A})^{-1}, (s_0\mathbf{I} - \mathbf{A})^{-1}\mathbf{b})$.
- Take $\mathbf{W}_k = \mathbf{V}_k$. Project onto $\mathbf{V}_k$.

Variants: block versions,
Rational **K**rylov **S**equence,
two-sided versions **bi-Lanczos**
...

Approaches

MOR

- Balanced truncation, Hankel norm appr.
- Padé approximation, moment matching
- **Modal approximation**

Eigenvalue computation

- QZ (dense systems) (n not large)
- Bi-Lanczos & Arnoldi
- **(Jacobi–)Davidson**

Modal approximations.

Compute $\mathbf{V}_k$, $\mathbf{W}_k$ $k \times n$ matrices with columns appropriate right, left, respectively eigenvectors $\mathbf{A}$,.

$H(s) = \mathbf{c}^*(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d$

Dominant poles

λ is **pole** of H if $\lim_{s\to\lambda} |H(s)| = \infty$.

Poles are eigenvalues of $\mathbf{A}$: for non-zero $\mathbf{v}_i$, $\mathbf{w}_i$

$$\begin{array}{ll} \mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i & \text{right eigenvectors} \\ \mathbf{w}_i^*\mathbf{A} = \lambda_i \mathbf{w}_i^* & \text{left eigenvectors} \end{array}$$

$$H(s) = \sum_{i=1}^{n'} \frac{R_i}{s-\lambda_i} + d, \qquad R_i = (\mathbf{c}^*\mathbf{v}_i)(\mathbf{w}_i^*\mathbf{b})$$

A pole λ_i is '**dominant**' if $\dfrac{|R_i|}{|\mathrm{Re}(\lambda_i)|}$ is large.

(determine the high peaks in the Bode plot.)

$H(s) = \mathbf{c}^*(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d$

Dominant poles

λ is **pole** of H if $\lim_{s\to\lambda} |H(s)| = \infty$.

Poles are eigenvalues of $\mathbf{A}$: for non-zero $\mathbf{v}_i$, $\mathbf{w}_i$

$$\begin{array}{ll} \mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i & \text{right eigenvectors} \\ \mathbf{w}_i^*\mathbf{A} = \lambda_i \mathbf{w}_i^* & \text{left eigenvectors} \end{array}$$

$$H(s) = \sum_{i=1}^{n'} \frac{R_i}{s-\lambda_i} + d, \qquad R_i = (\mathbf{c}^*\mathbf{v}_i)(\mathbf{w}_i^*\mathbf{b})$$

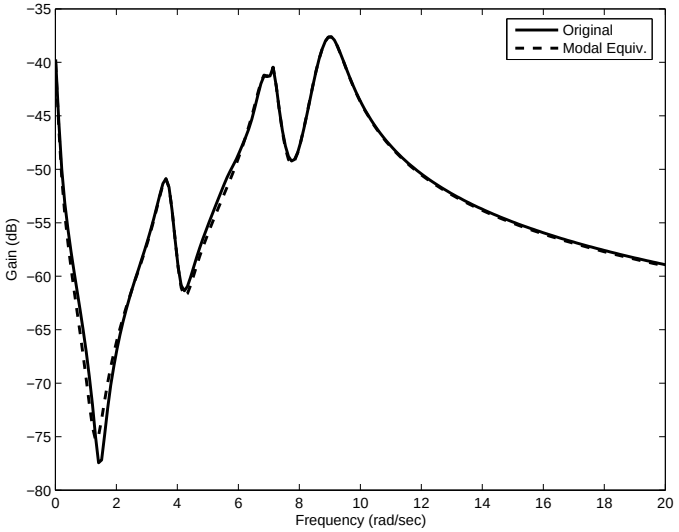
R_i are the **residuals**. **Note.** $R_i = 0$ if $\mathbf{w}_i^*\mathbf{v}_i = 0$.

Practice.

\# dominant poles $\ll$ \# poles $<$ \# eigenvalues $\leq n$

Approach. Project onto (and work in) the spaces spanned by the eigenvectors associated with dominant poles: $\mathbf{b}$ onto the space spanned by appropriate right eigenvectors, $\mathbf{c}$ onto the space of appropriate left eigenvectors.

New England test system



Bode plot: $\omega \rightsquigarrow |H(2\pi i\omega)|$ on decibel scale.
$n = 66$ (Solid line —).
Modal approximation of order $k = 11$ (dashed $- -$).

Dominant Pole Algorithm

```
Select s_0 ∈ ℂ and tol > 0.
Set ν = 1, s = s_0
While ν > tol repeat
    Solve (sI − A)x = b for x
    Solve (sI − A)*y = c for y
    s = y*Ax / y*x
    ν = max(‖Ax − sx‖_2, ‖y*A − sy*‖_2)
end while
```

Note. For the moment, assume
exact LU-decomposition $s\mathbf{I} - \mathbf{A}$ is feasible.

$$H(s) = \sum_{i=1}^{n'} \frac{R_i}{s - \lambda_i} + d, \qquad R_i = (\mathbf{c}^*\mathbf{v}_i)(\mathbf{w}_i^*\mathbf{b})$$

[Aguirre 93, Varga 95, Green Limebeer 95]

A pole λ_i is '**dominant**' if $\dfrac{|R_i|}{|\mathrm{Re}(\lambda_i)|}$ is large.

[Hamdan Nayfeh 89]

In our convergence **analysis**:

A pole λ_i is **dominant** if $|R_i| > |R_j|$ for all j.

Definition. (two-sided) **Rayleigh quotient**

$$\rho(\mathbf{x}, \mathbf{y}) = \rho(\mathbf{A}, \mathbf{x}, \mathbf{y}) \equiv \frac{\mathbf{y}^*\mathbf{A}\mathbf{x}}{\mathbf{y}^*\mathbf{x}} \quad \text{provided} \quad \mathbf{y}^*\mathbf{x} \neq 0$$

$$\rho(\mathbf{x}) \equiv \rho(\mathbf{x}, \mathbf{x})$$

DPA is Newton $\Rightarrow$

if $s_k \to \lambda_i$, then convergence is quadratic

Theorem. $\mathbf{A}\mathbf{v} = \mathbf{v}\lambda$ and $\mathbf{w}^*\mathbf{A} = \lambda\mathbf{w}^*$, $\mathbf{w}^*\mathbf{v} = 1$.

Apply DPA. Then

$$\mathbf{x}_k \to \mathbf{v} \quad \Leftrightarrow \quad \mathbf{y}_k \to \mathbf{w} \quad \Leftrightarrow \quad s_{k+1} = \rho(\mathbf{x}_k, \mathbf{y}_k) \to \lambda.$$

If convergence, then quadratic convergence:

$$\|\mathbf{v} - \mathbf{x}_{k+1}\| \leq \kappa \, \|\mathbf{v} - \mathbf{x}_k\|_2 \, \|\mathbf{w} - \mathbf{y}_k\|_2$$

$$\|\mathbf{w} - \mathbf{y}_{k+1}\| \leq \kappa \, \|\mathbf{v} - \mathbf{x}_k\|_2 \, \|\mathbf{w} - \mathbf{y}_k\|_2$$

[Ostrowski 59, Parlett 74]

Two-sided Rayleigh quotient iteration:

$\mathbf{x}$ and $\mathbf{y}$ right, left, respectively, eigenvector approximations

use, not only

the best available eigenvalue approximation $s \equiv \rho(\mathbf{x}, \mathbf{y})$

but also the best available eigenvector approximation

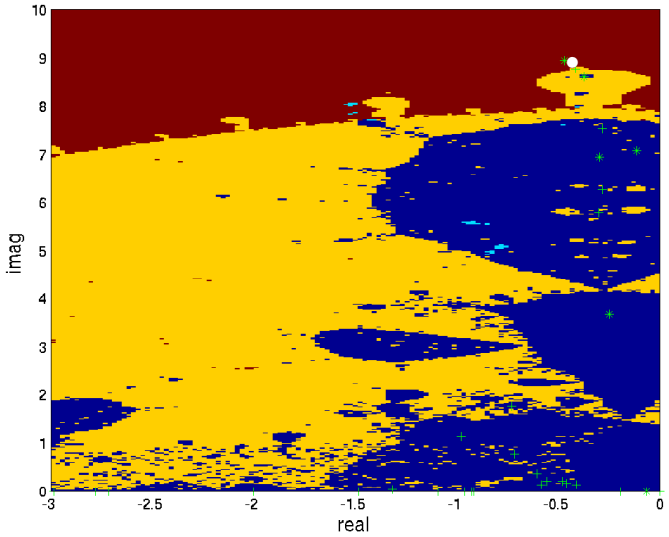
$$\mathbf{x} \leftarrow (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x} \quad \text{and} \quad \mathbf{y}^* \leftarrow \mathbf{y}^*(s\mathbf{I} - \mathbf{A})^{-1},$$

$$s = \frac{\mathbf{y}^*\mathbf{A}\mathbf{x}}{\mathbf{y}^*\mathbf{x}}$$

[Ostrowski 59, Parlett 74]

Theorem. Cubic convergence.

Part of the complex plane



Dominant pole $-0.456 \pm 8.96i$ (white •).
DPA converges for s_0 in red and yellow
RQI converges for s_0 in red and light blue
Dark blue convergence to less dominant poles.

Selecting initial shift s_0

DPA: $s_0 = \dfrac{\mathbf{c}^*\mathbf{A}\mathbf{b}}{\mathbf{c}^*\mathbf{b}}$. Reasonable? What if $\mathbf{c}^*\mathbf{b} = 0$?

RQI: $\mathbf{x}_0 = \mathbf{b}$, $\mathbf{y}_0 = \mathbf{c}$. Reasonable? What if $\mathbf{c}^*\mathbf{b} = 0$?

Recall that $\mathbf{b}$ represents input, $\mathbf{c}$ represents output.

Therefore,

Select s_0.

In RQI: $\mathbf{x}_0 = (s_0\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}$, $\mathbf{y}_0^* = \mathbf{c}^*(s_0\mathbf{I} - \mathbf{A})^{-1}$.

We stop if $\nu \leq 10^{-8}$.

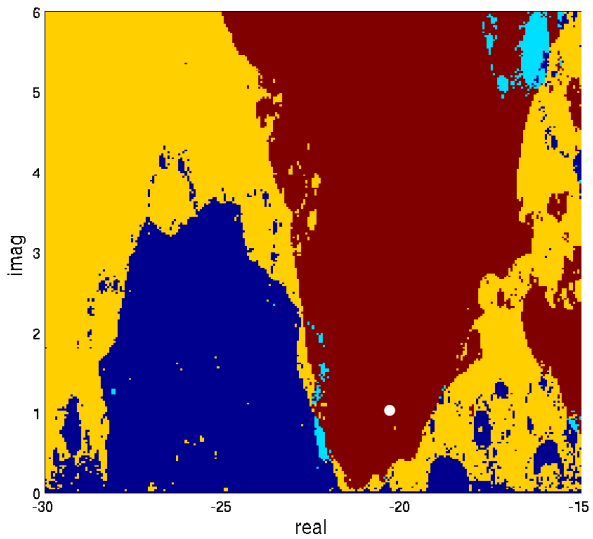
[Rommes Martins 06]

Example. Brazilian Interconnect Power System.

$\mathbf{A}$ and $\mathbf{E}$ are of dimension $n = 13,251$, $\mathbf{E}$ is singular.

Both $\mathbf{b}$ and $\mathbf{c}$ have only one non-zero entry, $\mathbf{c}^*\mathbf{E}\mathbf{b} = 0$.

Part of the complex plane



Dominant pole $-20.5 \pm 1.1i$ (white •).
DPA converges for s_0 in red and yellow
RQI converges for s_0 in red and light blue
Dark blue convergence to less dominant poles.

Average nu
of steps
DPA: 7.2
RQI: 6.0

In red regio
DPA: 6.1
RQI: 5.9

Problem

$$\mathbf{Ax=b}$$

Compute efficiently a $\widetilde{\mathbf{x}}$ with residual accuracy ϵ (i.e., $\|\mathbf{b} - \mathbf{A}\widetilde{\mathbf{x}}\| \leq \epsilon$)

Properties of the square matrix $\mathbf{A}$:

- The matrix $\mathbf{A}$ is expensive to store (dimension, density)

but

- we have a device that approximates $\mathbf{Au}$ by $\mathcal{A}_\eta(\mathbf{u})$ s.t.

$$\mathcal{A}_\eta(\mathbf{u}) = \mathbf{Au} + \mathbf{f} \quad \text{with} \quad \|\mathbf{f}\| \leq \eta \, \|\mathbf{A}\| \, \|\mathbf{u}\|,$$

η is the relative accuracy of the matrix-vector mult.,

- is more costly for higher 'rel. accuracy' (i.e., smaller η).

Conclusions

- DPA has better global convergence than RQI to dominant poles for a large class of dynamical systems:

DPA has (much) larger convergence areas for dominant poles than RQI, becoming larger with increasing dominance.

- The local cubic convergence of RQI versus the local quadratic convergence of DPA leads to a small advantage for RQI in iteration steps (typically, 10%–20%)

- The computational costs per step are $\approx$ the same (DPA slightly more efficient).

$$\mathcal{A}_\eta(\mathbf{u}) = \mathbf{Au} + \mathbf{f} \quad \text{with} \quad \|\mathbf{f}\| \leq \eta \, \|\mathbf{A}\| \, \|\mathbf{u}\|$$

Computation $\mathcal{A}_\eta(\mathbf{u})$ more costly for smaller η

Examples.

- In floating point arithmetic: $\eta = \mathcal{O}$(rel. machine prec.)

- Schurcomplement systems

- Matrix sign functions

⋮

Schurcomplement system

Fields of applications.

- Domain decomposition [Bouras Fraysse 00, Bouras Fraysse Giraud 00]
- Oceanography [van den Eshof S. van Gijzen 03]
- Optimisation
- CFD
- Electronic circuit simulation

⋮

Matrix sign functions

Fields of applications.

- Quantum Chromodynamics (QCD) [Cundy van den Eshof Frommer Krieg Lippert Schäfer 04]

Computational challenge.

- Monte Carlo simulations:
  very high dimensional linear systems are repeatedly solved
  b random, **A** some randomness.

Schurcomplement system

$$\begin{bmatrix} \mathbf{H} & \mathbf{B}_1^* \\ \mathbf{B}_2 & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix}$$

is equivalent to the **Schurcomplement** system (i.e., eliminate **y**, solve for **z**, as in Lecture 11):

Schurcomplement

$$\underbrace{(\,\mathbf{C} + \mathbf{B}_2 \mathbf{H}^{-1} \mathbf{B}_1^*\,)}_{\mathbf{A}}\ \underbrace{\mathbf{z}}_{\mathbf{x}} = \underbrace{\mathbf{B}_2 \mathbf{H}^{-1} \mathbf{g}}_{\mathbf{b}}\ .$$

To compute $\mathbf{c} = \mathbf{A}\mathbf{u}$ with relative accuracy η, use sufficiently many steps of an iterative meth. to solve $\mathbf{H}\tilde{\mathbf{u}} = \mathbf{B}_1^*\mathbf{u}$

higher accuracy (=smaller η) requires more iterative steps.

[Neuberger 98]

Lattice QCD and the overlap operator

$$\mathbf{A} = \rho\,\Gamma_5 + \mathrm{sign}(\mathbf{H}),\ \text{where } \rho \geq 1,\ \Gamma_5 \equiv \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix}$$

Properties H

- **H** is explicitly available (coeff. from stochastic process)
- **H** is sparse and Hermitian ($\mathbf{H}^* \equiv \bar{\mathbf{H}}^\top = \mathbf{H}$)
- **H** is high-dimensional ($16^4 \cdot 12 \approx 0.79\,10^6$, $32^4 \cdot 12 = 12.5\,10^6$)
- $\mathrm{sign}(\mathbf{H}) \equiv \mathbf{V}\,\mathrm{sign}(\Lambda)\,\mathbf{V}^*$, where $\mathbf{H} = \mathbf{V}\Lambda\mathbf{V}^*$, Λ diagonal and $\mathrm{sign}(\lambda) \equiv \lambda/|\lambda|$

Properties A.

- $\Gamma_5\,\mathbf{H} \neq \mathbf{H}\,\Gamma_5$,
- $\mathbf{A}^* = \mathbf{A} \not> 0$,
- No preconditioner for **A**.

No problem computing **Hu**. **What about Au**?
Via $\mathrm{sign}(\mathbf{H})\mathbf{u}$? Computing eigensystem **H** is not feasable.

[van den Eshof Frommer Lippert Schilling van den Vorst 02]

Computing the sign of a matrix

$\text{sign}(\lambda) = \frac{\lambda}{\sqrt{\lambda^2}}$

$$\text{sign}(\mathbf{H}) = \mathbf{H}(\mathbf{H}^2)^{-\frac{1}{2}} = \mathbf{H}f(\mathbf{H}^2), \quad \text{where} \quad f(\lambda) \equiv 1/\sqrt{\lambda}$$

Determine scalars ω_i, τ_i (explicit solutions are available)

such that $$f(\lambda) \approx \sum_{i=1}^{m} \omega_i \frac{1}{\lambda + \tau_i} \quad \Rightarrow$$

$$\text{sign}(\mathbf{H})\mathbf{u} \approx \sum_{i=0}^{m} \omega_i \mathbf{H} \underbrace{(\mathbf{H}^2 + \tau_i \mathbf{I})^{-1}\mathbf{u}}$$

Solve with CG (multishift CG)

Computation **Au** to high accuracy is very costly.
Costs are higher if higher accuracy is required.

Subspace method

Repeat until convergence, i.e., $\|\mathbf{r}_k\| \leq \texttt{tol}$

1) **Expand** the subspace $\mathcal{U}_k = \text{span}(\mathbf{U}_k)$ with the vector $\mathbf{A}\mathbf{u}_k$. Here, $\mathbf{U}_k \equiv [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$.
   Note. The vector $\mathbf{A}\mathbf{u}_k$ may be modified (orthogonalised against ...) to form $\mathbf{u}_{k+1}$
2) **Extract** some suitable approximate solution $\mathbf{x}_k \in \mathcal{U}_k$

The basis vectors $\mathbf{u}_j$ of $\mathcal{U}_k$ that are being multiplied by $\mathbf{A}$ to expand the search subspace form the **basis for expansion**.
The basis vectors $\mathbf{w}_j$ of $\mathcal{U}_k$ that are actually used to update $\mathbf{x}_k$, $\mathbf{x}_k = \sum_{j \leq k} \mathbf{w}_j \alpha_j$, form the **basis for extraction**.

Methods as GMRES use the same basis for expansion and extraction, methods as GCR, CG use different basis.

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

Approach

- **Use a Krylov subspace method**
  only MVs and basic linear algebra operations (AXPYs, DOTs)
- **Relax the MV**: replace $\mathbf{A}\mathbf{u}_k$ by $\mathcal{A}_{\eta_k}(\mathbf{u}_k)$
- **Relax to the max,**
  i.e., apply a relaxation strategy that selects η_k 'as large as possible' (η_k step dependent) without
  - **disturbing the speed of convergence**
  - **spoiling the residual accuracy (i.e. $\|\mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}\| \lesssim \epsilon$)**

Note. If the MV changes per step
$\Rightarrow$ **not** a Krylov subspace method

$\mathcal{A}_\eta(\mathbf{u}) = \mathbf{A}\mathbf{u} + \mathbf{f}$ with $\|\mathbf{f}\| \leq \eta \|\mathbf{A}\| \|\mathbf{u}\|$
Computation $\mathcal{A}_\eta(\mathbf{u})$ more costly for smaller η

Are highly accurate MVs required?

Question: How to **"relax to the max"**

that is, how to select η 'as large as possible' without

- disturbing the speed of convergence
- spoiling the residual accuracy (i.e. $\|\mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}\| \lesssim \epsilon$)

Analysis strategy

Our analysis is based on estimating the **residual gap**:

$$\text{res.-gap}_k \equiv \Big|\, \underbrace{\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|}_{\text{true residual}} - \rho_k \,\Big|, \quad \text{where} \quad \rho_k = \|\mathbf{r}_k\|_2$$

with $\mathbf{x}_k$ and ρ_k as **computed** by the method.
On convergence (i.e., $\rho_k \le \epsilon$), the residual gap determines the **residual accuracy**, i.e, the size of $\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|_2$.

- Computed residual norms ρ_k are available and in many methods the computed residual $\mathbf{r}_k$ as well.
- Computation of true residuals would require additional MVs. Expensive!

We focus on the **accuracy** (residual gap). Strategies that allow high accuracy (i.e., small res. gap) also appear not to hamper convergence (experimental evidence only).

If the convergence does not exhibit peaks, is the choice

$$\boxed{\eta_j = \frac{\epsilon}{\rho_j} \quad \text{with} \quad \rho_j = \|\mathbf{r}_j\|}$$

a good relaxation strategy (also for methods that are not of the type as on the previous transparancy)? Here, ρ_j is the norm of the residual as computed by the method.

A simple example. In example

$$\mathbf{A} = \text{diag}(1:3:300) - 13.6156\,\mathbf{I}, \qquad \mathcal{C}(\mathbf{A}) \approx 4.6\,10^2.$$

$\mathcal{A}_\eta(\mathbf{u}) \equiv \mathbf{A}\mathbf{u} + \mathbf{f}$ with $\mathbf{f}$ random such that $\|\mathbf{f}\| = \eta\,\|\mathbf{A}\|\,\|\mathbf{u}\|$

Expansion at step j by $\mathcal{A}_{\eta_j}(\mathbf{v}_j)$ with η_j as above

($\mathbf{v}_j$ is the expansion vector as selected by the method).

$$\boxed{\begin{aligned} &\mathbf{A}\mathbf{U}_k + \mathbf{F}_k \quad \text{with} \quad \|\mathbf{f}_j\| \le \eta_j \|\mathbf{A}\| \|\mathbf{u}_j\|, \quad \mathbf{x}_k = \mathbf{U}_k y_k, \\ &\|(\mathbf{b}-\mathbf{A}\mathbf{x}_k) - \mathbf{r}_k\| = \|\mathbf{F}_k y_k\|_2 \le \sum_{j\le k} \eta_j \,\|\mathbf{A}\|\,\|\mathbf{u}_j\|\,|e_j^* y_k| \end{aligned}}$$

Observations.

- $\mathbf{x}_k = \mathbf{U}\,y_k = \sum_{j\le k} \mathbf{u}_j (e_j^* y_k)$:
  $\mathbf{u}_j(e_j^* y_k)$ is the component of $\mathbf{x}_k$ in the direction $\mathbf{u}_j$.
- In all methods:
  - $\mathbf{A}\mathbf{U}_k + \mathbf{F}_k$ copied from algorithm
  - y_k requires some manipulation (without $\mathbf{A}$)
  - Analysis assumes that the MV is the only source of errors
- Estimate is 'sharp' (no specific directions in perturbs).
- Growth in η_j ($j \uparrow$) to be compensated by small $\|\mathbf{u}_j(e_j^* y_k)\|$.

$$\boxed{\begin{aligned} &\mathbf{A}\mathbf{U}_k + \mathbf{F}_k \quad \text{with} \quad \|\mathbf{f}_j\| \le \eta_j \|\mathbf{A}\| \|\mathbf{u}_j\|, \quad \mathbf{x}_k = \mathbf{U}_k y_k, \\ &\|(\mathbf{b}-\mathbf{A}\mathbf{x}_k) - \mathbf{r}_k\| = \|\mathbf{F}_k y_k\|_2 \le \sum_{j\le k} \eta_j \,\|\mathbf{A}\|\,\|\mathbf{u}_j\|\,|e_j^* y_k| \end{aligned}}$$

Observations (cont.).

- $\mathbf{x}_k$ depends on the extraction method, not on basis $\mathbf{U}_k$

However, on termination $\mathbf{x}_k \approx \mathbf{x}$ $\Rightarrow$ no essential difference between Galerkin $\mathbf{x}_k^{\text{Gal}}$ and minimal residual $\mathbf{x}_k^{\text{mr}}$.

Hence,

$\mathbf{U}_k$ ill-conditioned $\Rightarrow$ some $\|\mathbf{u}_j\|\,|e_j^* y_k|$ large

Basis used for expansion exact MV

- Orthogonal basis $\mathbf{u}_k \perp \mathbf{u}_j$

Particular implementations:

FOM	**ORTHORES**	Galerkin
GMRES	MINRES	Min. res.
general **A**	symmetric **A**	

- '**A**-Orthogonal' basis $\mathbf{A}\mathbf{u}_k \perp \mathbf{u}_j$

Particular implementations:

	CG	Galerkin
GCR	CR	Min. res.
general **A**	symmetric **A**	

Note. '**A**-Orthogonal' basis guaranteed to be a (well-conditioned) basis only if **A** positive definite.

$$\text{res-gap}_k \le \sum_{j \le k} \mu_j, \qquad \mu_j \equiv \eta_j \, \|\mathbf{A}\| \, \|\mathbf{u}_j\| \, |e_j^* y_k|$$

Estimates for exact MVs & A-orth. $\mathbf{U}_k$

Galerkin: $\boxed{\mu_j \le \eta_j \, \mathcal{C}(\mathbf{A}) \left(\|\mathbf{r}_j^{\text{Gal}}\| + \|\mathbf{r}_{j+1}^{\text{Gal}}\| \right)}$ CG

Min. res.: $\boxed{\mu_j \le \eta_j \, \mathcal{C}(\mathbf{A}) \left(\|\mathbf{r}_j^{\text{Gal}}\| + \|\mathbf{r}_{j+1}^{\text{Gal}}\| \frac{\|\mathbf{r}_{j+1}^{\text{mr}}\|}{\|\mathbf{r}_j^{\text{mr}}\|} \right)}$ GCR

If **A** is positive

Galerkin:
min. res.: $\boxed{\mu_j \le \eta_j \, \mathcal{C}(\mathbf{A}) \, \|\mathbf{r}_j^{\text{mr}}\|_2}$

Theoretical results are sharp (experimental evidence)

$$\text{res-gap}_k \le \sum_{j \le k} \mu_j, \qquad \mu_j \equiv \eta_j \, \|\mathbf{A}\| \, \|\mathbf{u}_j\| \, |e_j^* y_k|$$

Estimates for exact MVs & orthogonal $\mathbf{U}_k$

Galerkin: $\boxed{\mu_j \le \eta_j \, \mathcal{C}(\mathbf{A}) \left(\|\mathbf{r}_j^{\text{mr}}\| + \|\mathbf{r}_k^{\text{Gal}}\| \right)}$ FOM ORTHORES

Min. res.: $\boxed{\mu_j \le \eta_j \, \mathcal{C}(\mathbf{A}) \, \|\mathbf{r}_j^{\text{mr}}\|}$ GMRES

If **A** is positive definite

Galerkin:
min. res.: $\boxed{\mu_j \le \eta_j \, \mathcal{C}(\mathbf{A}) \, \|\mathbf{r}_j^{\text{mr}}\|}$

Theoretical results are sharp (experimental evidence)

Alternative analysis: [Simoncini Szyld, 2003]

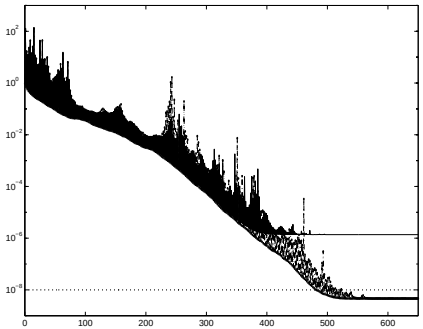
But

MV is not exact . . .

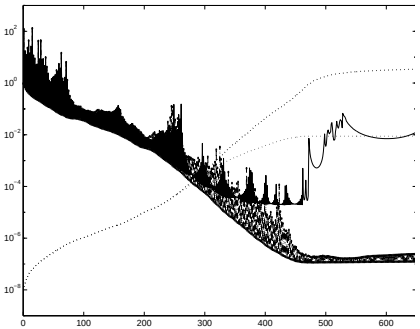
and rounding errors play a role

Example from QCD

$\mathbf{D}_W =$ `CONF6.0-0.0014x4.2000` (Matrix market, $n = 3072$)



$\epsilon = 10^{-8}, \quad \eta_k = \epsilon$



$\epsilon = 10^{-8}, \quad \eta_k = \epsilon/\rho_k$

k versus $\|$true residual$\|$
CG —
ORTHORES $- \cdot -$
η_k $\cdots$

$\mathbf{A} \equiv \Gamma_5 \mathbf{D}_W$, $\mathbf{b}$ normalized random

In case convergence accelerates:

- \+ reduces number of iteration steps :-)
- − limitted profit from relaxing MVs :-(

Alternatives: nest & relax MVs

Basic scheme:

for $k = 0, 1, \ldots$

Compute $\mathbf{u}_k$ (e.g., s.t. $\mathbf{A}\mathbf{u}_k \approx \mathbf{r}_k$)

$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_k$

either $\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{A}\mathbf{u}_k$

In "Compute $\mathbf{u}_k$", relax:

$\mathcal{A}_{\eta_k^{(i)}}(\ldots)$ with $\eta_k^{(i)} = \xi_k \frac{\|\mathbf{r}_k\|}{\|\mathbf{r}_k^{(i)}\|}$.

In "update $\mathbf{r}_k$", relax:

$\mathcal{A}_{\eta_j}(\mathbf{u}_k)$ with $\eta_k = \epsilon \frac{\|\mathbf{r}_0\|}{\|\mathbf{r}_k\|}$

Total costs of MVs

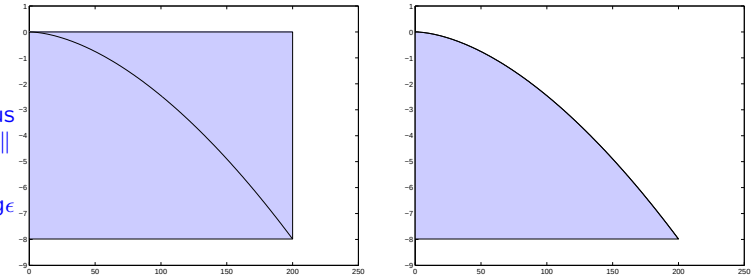
Assume: costs $\mathcal{A}_\eta(\mathbf{u})$ are $\sim -\log \eta$.

Graphical interpretation total costs MVs if convergence unaltered

$\eta_j = \epsilon$ $\qquad$ $\eta_j = \epsilon/\|\mathbf{r}_j\|$

j versus $\log\|\mathbf{r}_j\|$

$\log\epsilon$



Total costs MVs (TC) $\sim$ size blue area

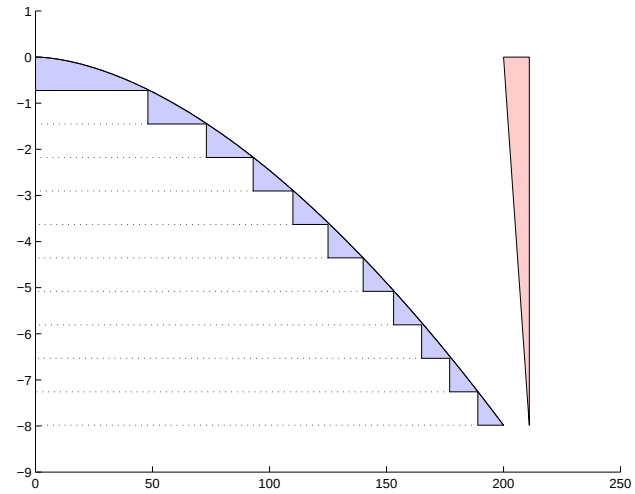
$$\text{TC} \sim -\sum_{j \leq k} \log \epsilon \qquad \text{TC} \sim -\sum_{j \leq k} (\log \epsilon - \log \|\mathbf{r}_j\|)$$

$\mathcal{A}_\eta(\mathbf{u}) = \mathbf{A}\mathbf{u} + \mathbf{f}$ $\|\mathbf{f}\| \leq \eta \|\mathbf{A}\| \|\mathbf{u}\|$

Estimated costs with nesting & relaxed MV

$\xi_\ell \updownarrow$

j versus $\log\|\mathbf{r}_j\|$



TC $\sim$ size blue area (inner loop)
\+ size red area (outer loop)

Alternative: nest & relax MVs

Inner iteration with (another) inexact solver.

Leads to a small number of outer iterations.

Advantages

- Only a few expensive MVs
  - Modest accumulation of 'errors' MV

Drawback

- Loss of optimality Krylov solver $\Rightarrow$ more MVs in total

Solvers for the 'outer iteration' must be '**flexible**' (i.e. must cope with variable preconditioners), e.g.:

- **GMRES Repeated**
- **Flexible GMRES**

...

(See also Hernández et al 00, Golub et al 00, Carpentieri 02)

[van den Eshof S van Gijzen 03]

Schurcomplement system

Example from Oceanography (barotropic flow)

$$\begin{bmatrix} r\mathbf{L}-\mathbf{C} & \alpha\tilde{\mathbf{L}} \\ -\tilde{\mathbf{L}}^* & \mathbf{M} \end{bmatrix} \begin{bmatrix} \psi \\ \zeta \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix}.$$

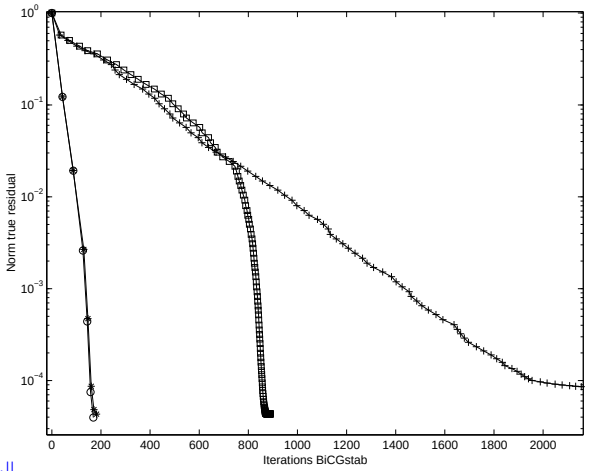
is equivalent to the Schur complement system

$$\underbrace{\left(\mathbf{M}+\alpha\tilde{\mathbf{L}}^*(r\mathbf{L}-\mathbf{C})^{-1}\tilde{\mathbf{L}}\right)}_{\mathbf{A}}\underbrace{\zeta}_{\mathbf{x}} = \underbrace{\tilde{\mathbf{L}}^*(r\mathbf{L}-\mathbf{C})^{-1}\mathbf{g}}_{\mathbf{b}}.$$

To get $\mathbf{Au}$ with relative accuracy η, use sufficiently many steps of Bi-CGSTAB to solve $(r\mathbf{L}-\mathbf{C})\tilde{\mathbf{u}} = \tilde{\mathbf{L}}\mathbf{u}$

In example $n = 26\,455$. Costs $\sim$ # Bi-CGSTAB steps

[van den Eshof S van Gijzen 03]



$\epsilon_7 = \epsilon$, logϵ

Costs versus log$\|\mathbf{r}_j\|$

$+++$ GMRES, MV fixed prec. $\epsilon = 10^{-6}$

$\square\square$ GMRES, relaxed MV $\epsilon = 10^{-7}$

$**$ FGMRES(GMRES) & relaxed MV

$\circ\circ$ GMRESR(GMRES) & relaxed MV

[Cundy van den Eshof Frommer Krieg Lippert Schäfer 04]

Matrix sign function

$$(\rho\mathbf{I} + \Gamma_5\,\mathrm{sign}(\mathbf{H}))\,\mathbf{x} = \mathbf{b}$$

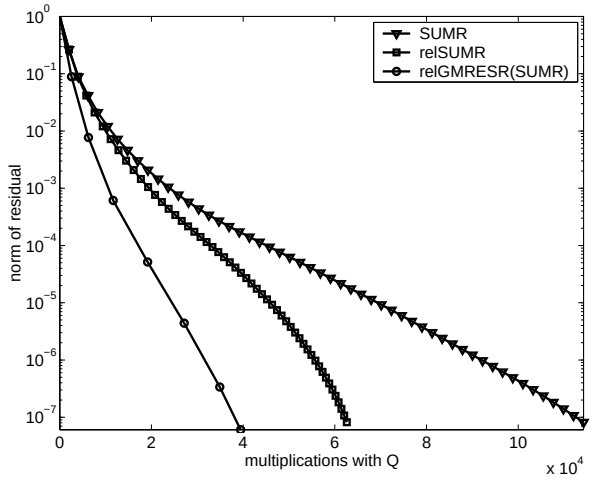
[Jagels Reichel 94]

Solve with **Shifted Unitary Minimal Residuals** (efficient variant of GMRES for shifted unitary matrices) and variants:

SUMR (`SUMR`),
relaxed SUMR (`relSUMR`),
relaxed GMRESR(relaxed SUMR) (`relGMRESR(SUMR)`)

In example, on a 8^4 lattice ($n = 49\,152$), $\rho = 1.22$.

[Cundy van den Eshof Frommer Krieg Lippert Schäfer 04]



On 16 processors of ALiCE (cluster computer Wuppertal)
Time in seconds, improvement factor in brackets

`SUMR` **2510** `relSUMR` **1500 (1.67)** `relGMRESR(SUMR)` **576 (4.36)**

[Cundy van den Eshof Frommer Krieg Lippert Schäfer 04]

Matrix sign function

$$(\rho \mathbf{I} + \Gamma_5 \operatorname{sign}(\mathbf{H}))\,\mathbf{x} = \mathbf{b}$$

[Jagels Reichel 94]

Solve with **Shifted Unitary Minimal Residuals**
(efficient variant of GMRES for shifted unitary matrices)
and variants:

SUMR (`SUMR`),
relaxed SUMR (`relSUMR`),
relaxed GMRESR(relaxed SUMR) (`relGMRESR(SUMR)`)

In example, on a 16^4 lattice ($n = 786\,432$), $\rho = 1.06$.

On 16 processors of ALiCE (cluster computer Wuppertal)
Time in seconds, improvement factor in brackets

`SUMR` **31550** `relSUMR` **18840 (1.67)** `relGMRESR(SUMR)` **5974 (5.28)**