An Introduction to Iterative Solvers and Preconditioning Techniques

Alison Ramage
Department of Mathematics
University of Strathclyde

ar@maths.strath.ac.uk
http://www.maths.strath.ac.uk/~caas63/

THE
UNIVERSITY OF
STRATHCLYDE
IN GLASGOW

Overview

- motivation
- Krylov subspace methods for
  - symmetric positive definite systems
  - symmetric indefinite systems
  - nonsymmetric systems
- preconditioning
- further information

PDE-based Problems

- large-scale simulations in e.g. fluid mechanics, structural engineering, medical applications, meteorology . . .

- linear algebra costs often dominate

- finite element, finite difference, finite volume discretisations

- only local connections between unknowns on the computational grid

- very large, very sparse linear systems

Model Finite Element BVP

- Laplace's equation
- d-dimensional uniform grid, discretisation parameter h
- $n = \dfrac{1}{h}$ nodes in each dimension
- coefficient matrix A is $N \times N$ where $N = O(n^d) = O(h^{-d})$
- e.g. bilinear finite elements
  - $d = 2 :$ 9 nonzeros per row
  - $d = 3 :$ 27 nonzeros per row

Direct or Iterative Solvers?

- direct methods
  - efficient for full matrices
  - good for lots of RHS vectors
  - sparse matrices may lead to fill-in
  - node ordering often important
  - storage and CPU restrictions

Direct or Iterative Solvers?

- direct methods
  - efficient for full matrices
  - good for lots of RHS vectors
  - sparse matrices may lead to fill-in
  - node ordering often important
  - storage and CPU restrictions
- iterative methods
  - data structures predetermined
  - no need for special node ordering
  - efficient for extremely large sparse problems
  - last iterate can give a good starting vector
  - some expertise needed
  - often no guarantee of success

Asymptotic Estimates

Iterative method: Conjugate Gradients

Direct method: Gaussian Elimination with band-minimising node ordering

Asymptotic Estimates

Iterative method: Conjugate Gradients

Direct method: Gaussian Elimination with band-minimising node ordering

Computational Work

	$d = 2$	$d = 3$
CG	$O(N^{\frac{3}{2}})$	$O(N^{\frac{4}{3}})$
GE factorise	$O(N^2)$	$O(N^{\frac{7}{3}})$
GE solve	$O(N^{\frac{3}{2}})$	$O(N^{\frac{5}{3}})$

Asymptotic Estimates

Iterative method: Conjugate Gradients

Direct method: Gaussian Elimination with band-minimising node ordering

Computational Work

	$d = 2$	$d = 3$
CG	$O(N^{\frac{3}{2}})$	$O(N^{\frac{4}{3}})$
GE factorise	$O(N^2)$	$O(N^{\frac{7}{3}})$
GE solve	$O(N^{\frac{3}{2}})$	$O(N^{\frac{5}{3}})$

Storage

	$d = 2$	$d = 3$
CG	$O(N)$	$O(N)$
GE factorise	$O(N^{\frac{3}{2}})$	$O(N^{\frac{5}{3}})$
GE solve	$O(N^{\frac{3}{2}})$	$O(N^{\frac{5}{3}})$

I. Symmetric Positive Definite Systems

Solve $A\mathbf{x} = \mathbf{b}$ where

- A is symmetric and positive definite
- A has s distinct (positive) eigenvalues

Minimal polynomial:

$$A^s + m_1 A^{s-1} + \cdots + m_{s-1} A + m_s I = 0$$

so

$$A^{-1} = -\frac{1}{m_s} A^{s-1} - \frac{m_1}{m_s} A^{s-2} - \cdots - \frac{m_{s-1}}{m_s} I$$

$$\hat{\mathbf{x}} = A^{-1}\mathbf{b} \in \mathcal{K}(A, \mathbf{b}, s) \equiv \text{span}\{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \ldots, A^{s-1}\mathbf{b}\}$$

Krylov Subspace

Three equivalent problems

1. solve $A\mathbf{x} = \mathbf{b}$

Three equivalent problems

1. solve $A\mathbf{x} = \mathbf{b}$

2. minimise $\Phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}$

$$\nabla\Phi(\mathbf{x}) = A\mathbf{x} - \mathbf{b} = \mathbf{0}$$

Three equivalent problems

1. solve $A\mathbf{x} = \mathbf{b}$

2. minimise $\Phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}$

$$\nabla\Phi(\mathbf{x}) = A\mathbf{x} - \mathbf{b} = \mathbf{0}$$

3. minimise $\|\mathbf{x} - \hat{\mathbf{x}}\|_A$

$$\|\mathbf{v}\|_A = \left\{\mathbf{v}^T A\mathbf{v}\right\}^{\frac{1}{2}}$$

$$(\mathbf{x} - \hat{\mathbf{x}})^T A(\mathbf{x} - \hat{\mathbf{x}}) = \mathbf{b}^T A^{-1}\mathbf{b} + 2\Phi(\mathbf{x})$$

Steepest Descent Method

$$\Phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}$$

At a point $\mathbf{x}_k$, Φ decreases most rapidly in direction

$$-\nabla\Phi(\mathbf{x}_k) = \mathbf{b} - A\mathbf{x}_k = \mathbf{r}_k$$

Value of Φ at point $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{r}_k$ minimised when

$$\alpha_k = \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{r}_k^T A\mathbf{r}_k}$$

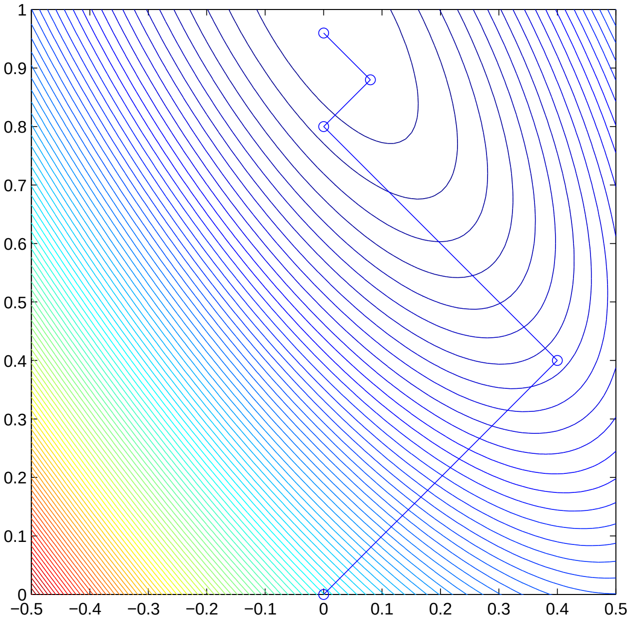
and

$$\Phi(\mathbf{x}_{k+1}) < \Phi(\mathbf{x}_k)$$

is guaranteed

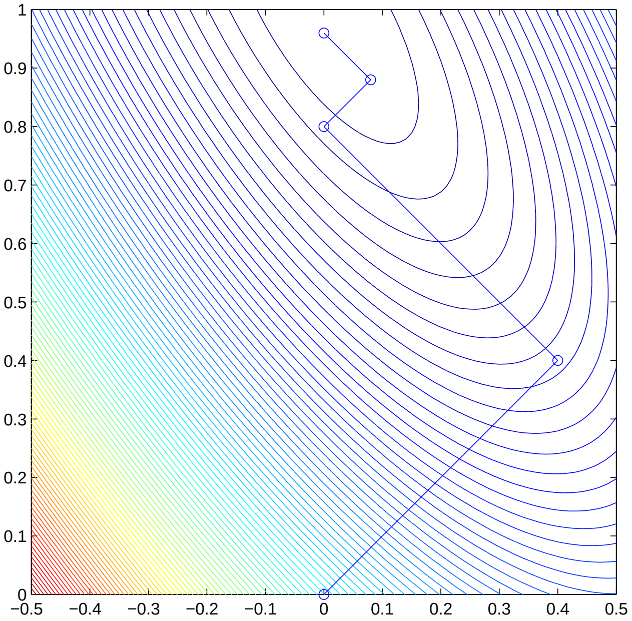
A Practical SD Example

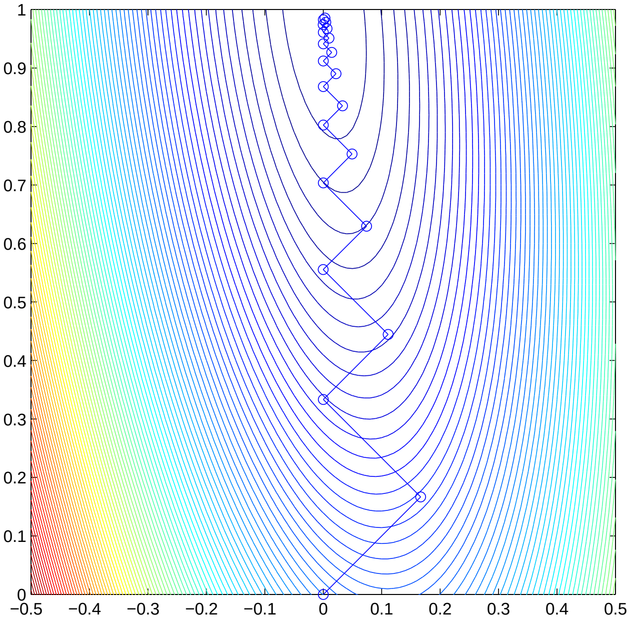
$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$



A Practical SD Example

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \qquad A = \begin{bmatrix} 9 & 1 \\ 1 & 1 \end{bmatrix}$$





Conjugate Gradient Method

Choose new search directions $\{\mathbf{p}_0, \mathbf{p}_1, \ldots\}$ with iterates

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

Writing $P_{k+1} \equiv \mathrm{span}\{\mathbf{p}_0, \ldots, \mathbf{p}_k\}$, we require

(i) $$\mathbf{x}_{k+1} = \min_{\mathbf{x} \in P_{k+1}} \Phi(\mathbf{x})$$

(ii) $$\min_{\alpha} \Phi(\mathbf{x}_k + \alpha \mathbf{p}_k)$$

QUESTION: can we choose $\mathbf{p}_k$ so that $\mathbf{x}_{k+1}$ satisfies (i) and (ii) simultaneously?

YES: choose the vectors $\mathbf{p}_k$ to be A-conjugate, i.e.

$$\mathbf{p}_j^T A \mathbf{p}_k = 0, \quad j < k$$

Some nice properties:

- $\mathbf{x}_k$ minimises $\Phi(\mathbf{x})$ over all $\mathbf{x} \in P_k$
- $\|\mathbf{x}_k - \hat{\mathbf{x}}\|_A \equiv$ minimum
- $\mathbf{r}_k \perp \mathcal{K}(A, \mathbf{r}_0, k)$
- $\mathbf{p}_j^T A \mathbf{p}_k = \mathbf{r}_k^T \mathbf{p}_j = \mathbf{r}_k^T \mathbf{r}_j = 0 \qquad j < k$
- $\mathrm{span}\{\mathbf{r}_0, \ldots, \mathbf{r}_{k-1}\} = \mathrm{span}\{\mathbf{p}_0, \ldots, \mathbf{p}_{k-1}\} = \mathcal{K}(A, \mathbf{r}_0, k)$

Conjugate Gradient Method

Hestenes & Stiefel (1952)

```
choose $\mathbf{x}_0$
compute $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
set $\mathbf{p}_0 = \mathbf{r}_0$
for $k = 0$ until convergence do
        $\alpha_k = \mathbf{r}_k^T \mathbf{r}_k / \mathbf{p}_k^T A\mathbf{p}_k$
        $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
        $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k$
        $\beta_k = \mathbf{r}_{k+1}^T \mathbf{r}_{k+1} / \mathbf{r}_k^T \mathbf{r}_k$
        $\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$
end do
```

Finite Termination

CG method constructs iterates

$$\mathbf{x}_k \in \mathbf{x}_0 + \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0\}$$

with properties

- $\mathbf{x}_k$ minimises $\|\mathbf{x}_k - \hat{\mathbf{x}}\|_A$
- $\mathbf{r}_k \perp \mathcal{K}(A, \mathbf{r}_0, k)$
- $\mathbf{p}_k$ can be calculated via a three-term recurrence relation

Theorem: The CG method finds $\hat{\mathbf{x}}$ in s steps.

Finite Termination

CG method constructs iterates

$$\mathbf{x}_k \in \mathbf{x}_0 + \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0\}$$

with properties

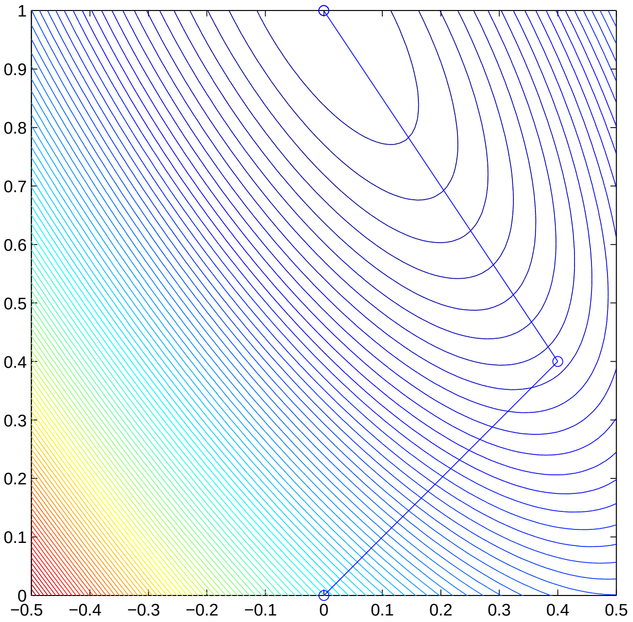
- $\mathbf{x}_k$ minimises $\|\mathbf{x}_k - \hat{\mathbf{x}}\|_A$
- $\mathbf{r}_k \perp \mathcal{K}(A, \mathbf{r}_0, k)$
- $\mathbf{p}_k$ can be calculated via a three-term recurrence relation

Theorem: The CG method finds $\hat{\mathbf{x}}$ in s steps.

Proof: $\hat{\mathbf{x}} \in \mathcal{K}(A, \mathbf{r}_0, s)$ so $\hat{\mathbf{x}} \in P_s$. But $\hat{\mathbf{x}}$ minimises $\Phi(\mathbf{x})$ over $\mathbf{x} \in \mathcal{K}(A, \mathbf{r}_0, s)$ and $\mathbf{x}_s$ minimises $\Phi(\mathbf{x})$ over $\mathbf{x} \in P_s$. Hence $\mathbf{x}_s = \hat{\mathbf{x}}$.

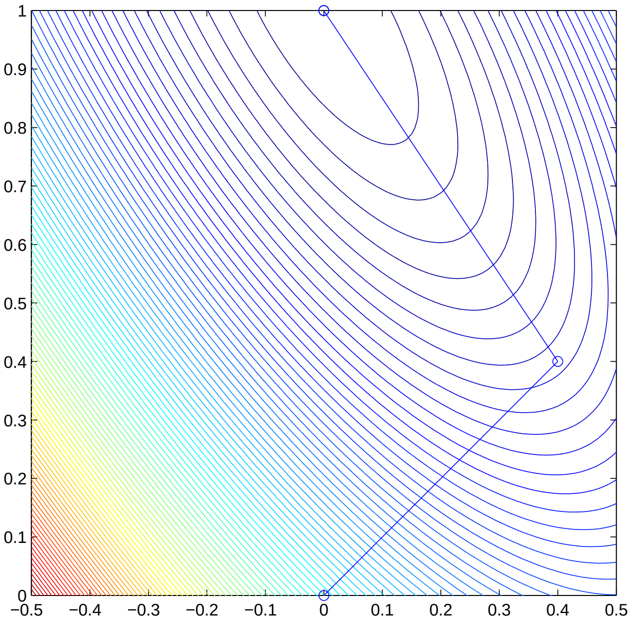
A Practical CG Example

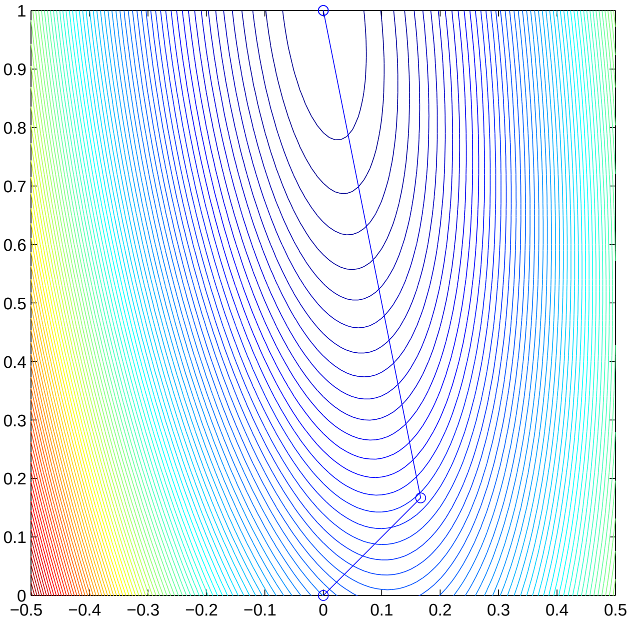
$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$



A Practical CG Example

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \qquad A = \begin{bmatrix} 9 & 1 \\ 1 & 1 \end{bmatrix}$$





CG Convergence

$$\mathbf{x}_k \in \mathbf{x}_0 + \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0\}$$

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k = \mathbf{b} - A\left(\mathbf{x}_0 + \sum_{i=1}^{k} \gamma_i A^{i-1}\mathbf{r}_0\right) = \mathbf{r}_0 - \sum_{i=1}^{k} \gamma_i A^i \mathbf{r}_0$$

i.e.

$$\mathbf{r}_k = \hat{P}_k(A)\mathbf{r}_0$$

$\hat{P}_k \in \Pi_k^1 \equiv$ polynomials of degree k with constant term 1

$$\|\mathbf{x}_k - \hat{\mathbf{x}}\|_A = \|\mathbf{r}_k\|_{A^{-1}} = \min_{P_k \in \Pi_k^1} \|P_k(A)\mathbf{r}_0\|_{A^{-1}}$$

Expand $\mathbf{r}_0$ in terms of orthonormal eigenvectors:

$$\mathbf{r}_0 = \sum_{i=1}^{n} \rho_i \mathbf{v}_i, \qquad \rho_i = \mathbf{v}_i^T \mathbf{r}_0, \qquad A\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

$$\begin{aligned}
\|\mathbf{x}_k - \hat{\mathbf{x}}\|_A &= \min_{P_k \in \Pi_k^1} \|P_k(A) \sum_{i=1}^{n} \rho_i \mathbf{v}_i\|_{A^{-1}} \\
&= \left\{ \min_{P_k \in \Pi_k^1} \sum_{i=1}^{n} P_k(\lambda_i)^2 (\rho_i \mathbf{v}_i)^T A^{-1} (\rho_i \mathbf{v}_i) \right\}^{\frac{1}{2}} \\
&\leq \min_{P_k \in \Pi_k^1} \max_i |P_k(\lambda_i)| \left\{ \mathbf{r}_0^T A^{-1} \mathbf{r}_0 \right\}^{\frac{1}{2}} \\
&= \min_{P_k \in \Pi_k^1} \max_i |P_k(\lambda_i)| \|\mathbf{r}_0\|_{A^{-1}}
\end{aligned}$$

$$\|\mathbf{x}_k - \hat{\mathbf{x}}\|_A \leq \min_{P_k \in \Pi_k^1} \max_i |P_k(\lambda_i)| \|\mathbf{x}_0 - \hat{\mathbf{x}}\|_A$$

Ideal Bound

$P_{\min}$ is such that

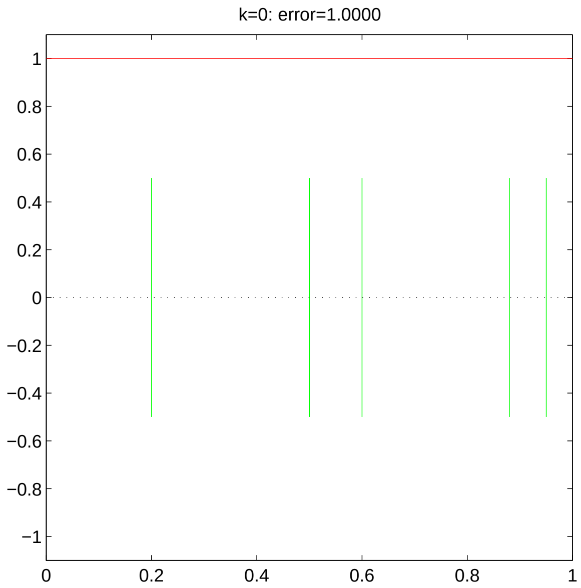
$$M = \max_i |P_{\min}(\lambda_i)| = \min_{P_k \in \Pi_k^1} \max_i |P_k(\lambda_i)|$$

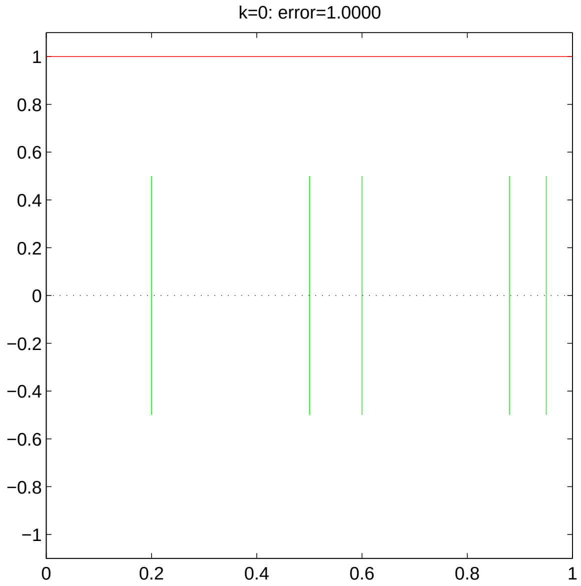
MINIMAX APPROXIMATION

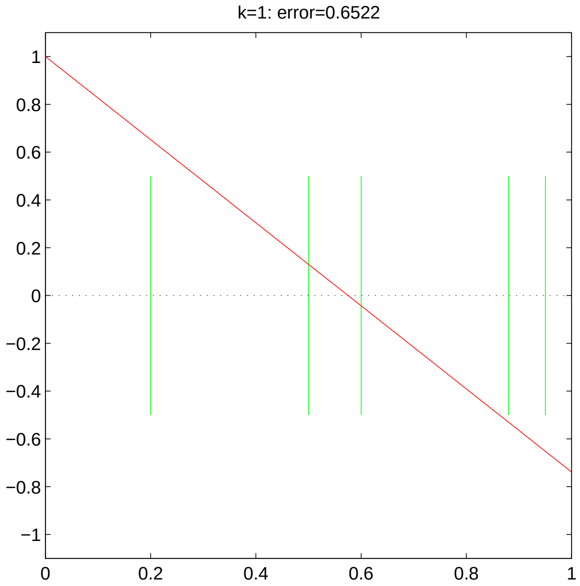
Theorem: Greenbaum (1979)
This error bound is sharp, i.e. there is always some $\mathbf{x}_0$ such that the discrete minimax bound

$$\|\mathbf{x}_k - \hat{\mathbf{x}}\|_A \leq M \|\mathbf{x}_0 - \hat{\mathbf{x}}\|_A$$

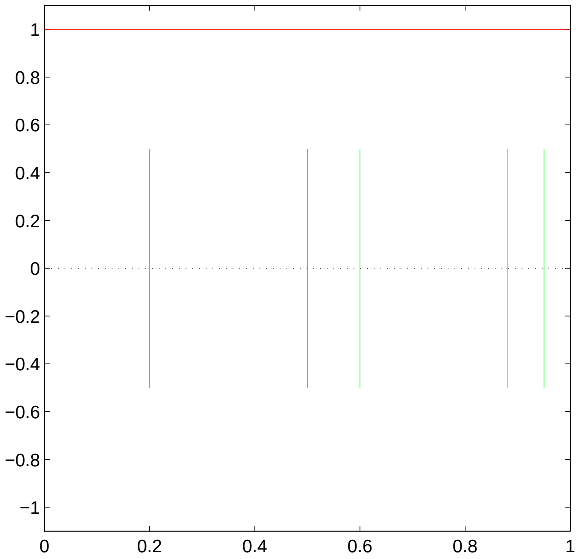
is attained.


k=0: error=1.0000
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
0.2
0.4
0.6
0.8
1

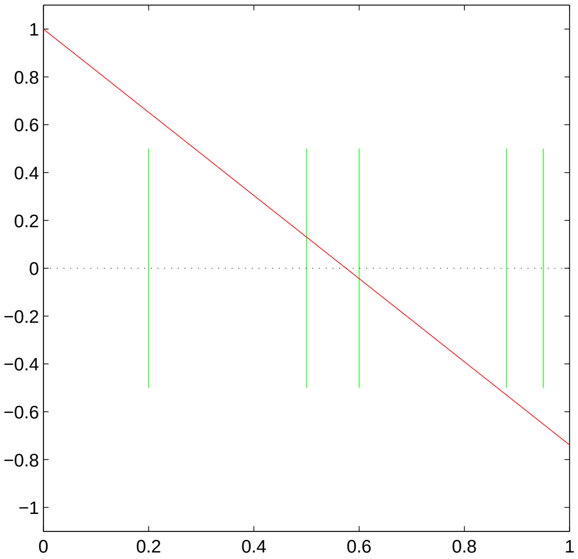

k=0: error=1.0000
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
0.2
0.4
0.6
0.8
1


k=1: error=0.6522
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
0.2
0.4
0.6
0.8
1

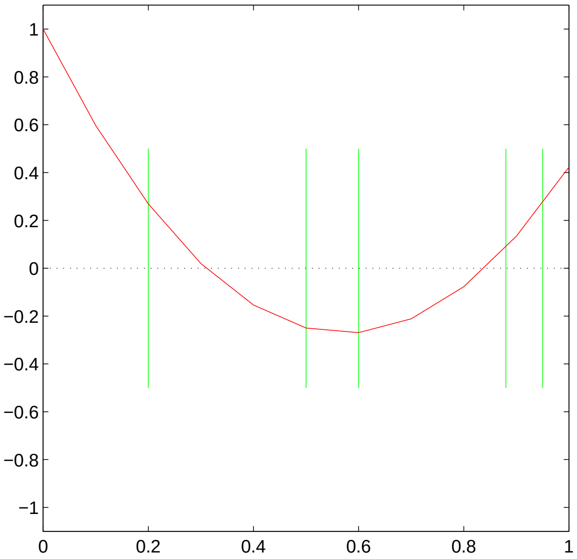
k=0: error=1.0000

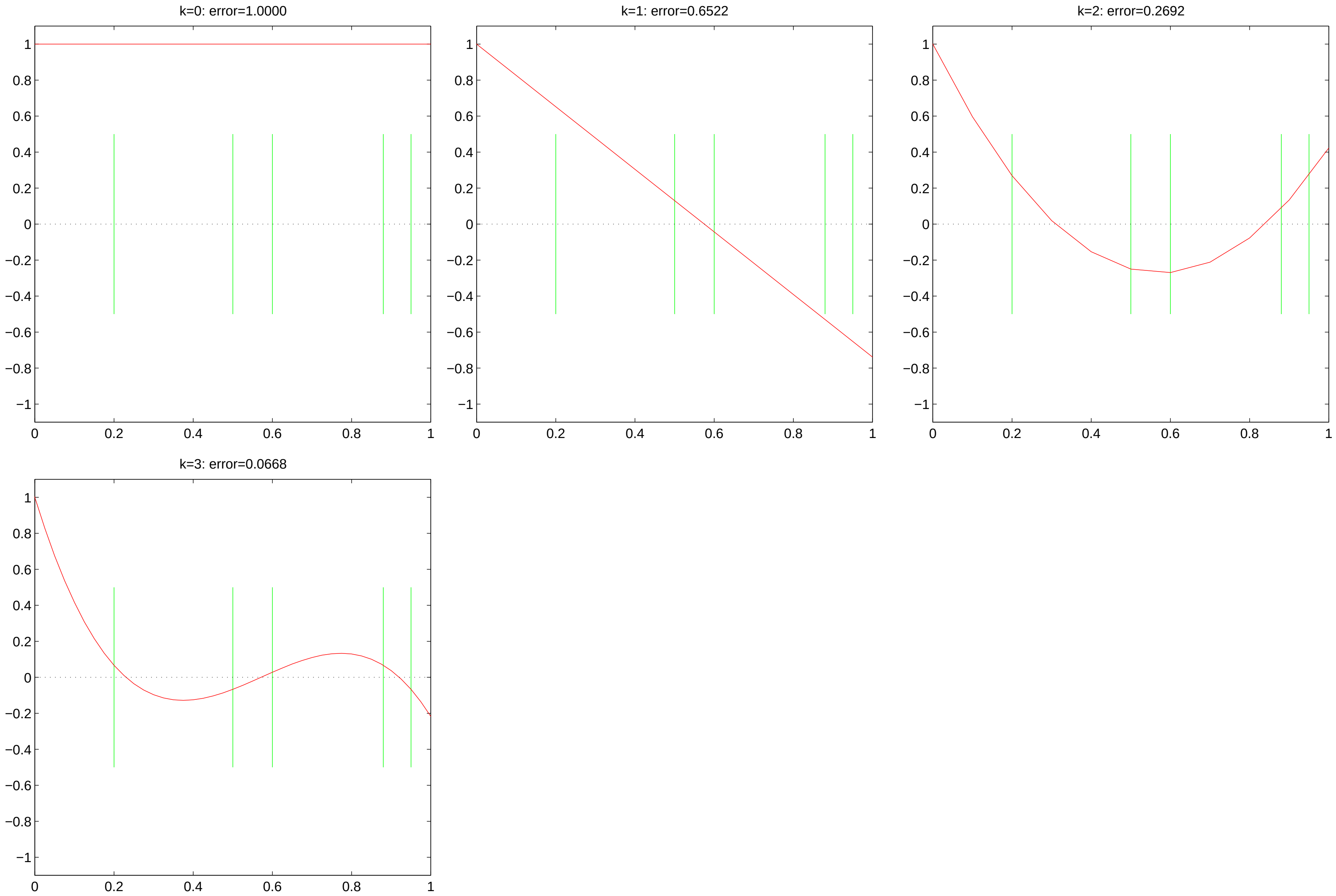


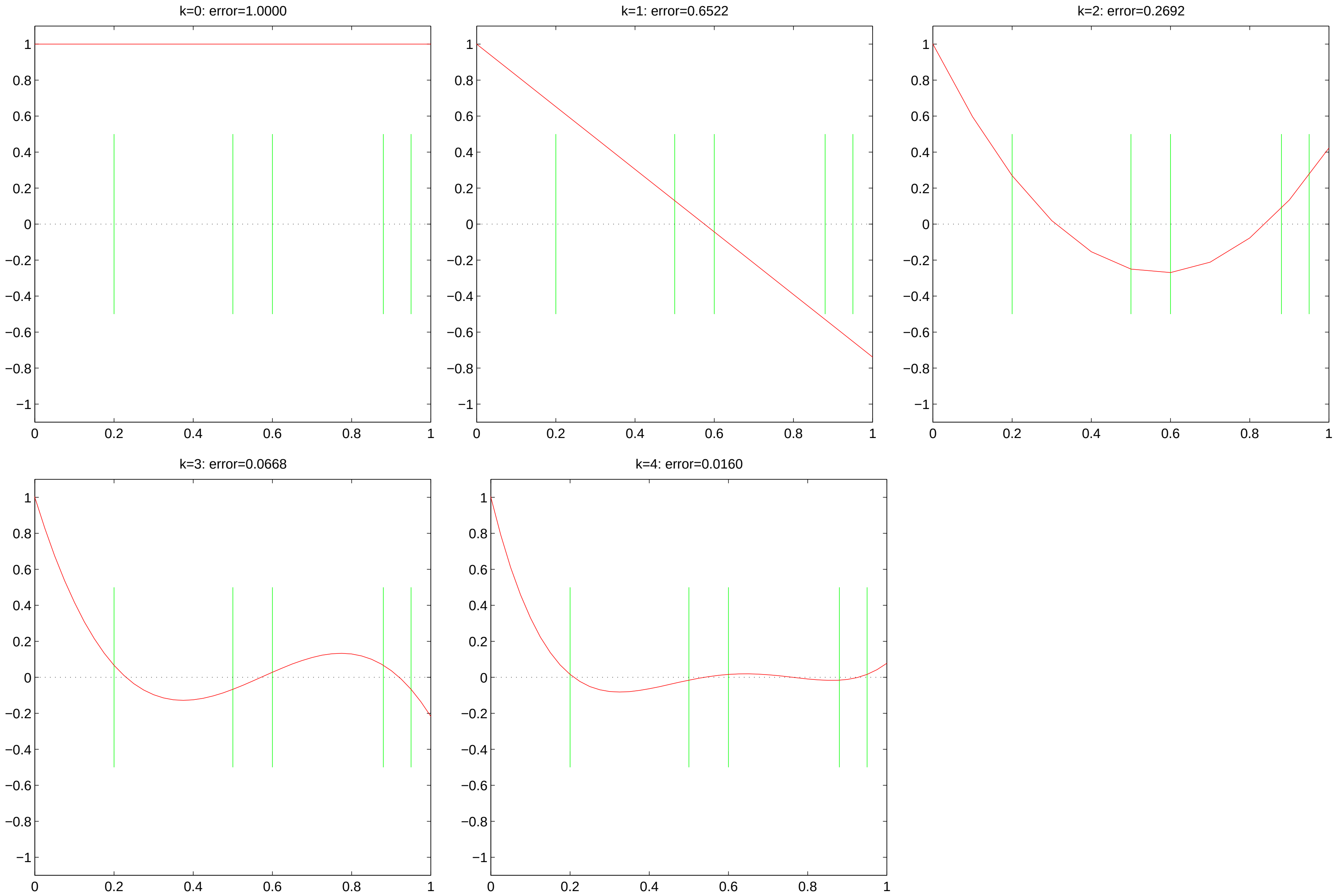
k=1: error=0.6522



k=2: error=0.2692




k=0: error=1.0000
k=1: error=0.6522
k=2: error=0.2692
k=3: error=0.0668
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
0.2
0.4
0.6
0.8
1

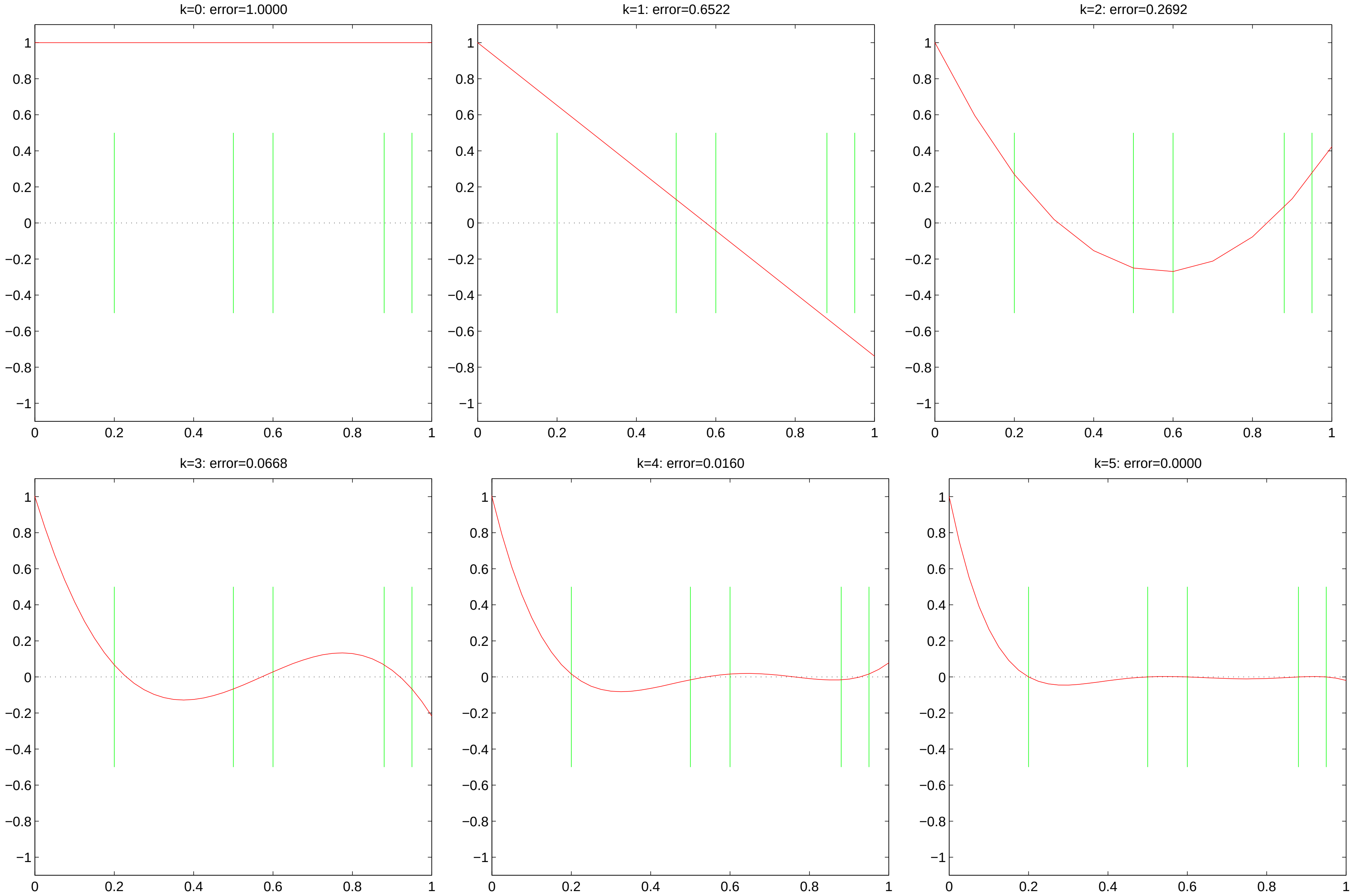

k=0: error=1.0000

k=1: error=0.6522

k=2: error=0.2692

k=3: error=0.0668

k=4: error=0.0160



k=0: error=1.0000

k=1: error=0.6522

k=2: error=0.2692

k=3: error=0.0668

k=4: error=0.0160

k=5: error=0.0000

Practical Bound

Based on knowledge of $\lambda_{\max}$ and $\lambda_{\min}$ alone, bound involves

$$\hat{T}_k(\lambda) = \frac{T_k\left[\frac{\lambda_{\max}+\lambda_{\min}-2\lambda}{\lambda_{\max}-\lambda_{\min}}\right]}{T_k\left[\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right]}, \qquad \text{condition number} \quad \kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$$

$$M = \max_i |\hat{T}_k(\lambda_i)| = \frac{1}{T_k\left[\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right]} = \frac{1}{T_k\left[\frac{\kappa+1}{\kappa-1}\right]}$$

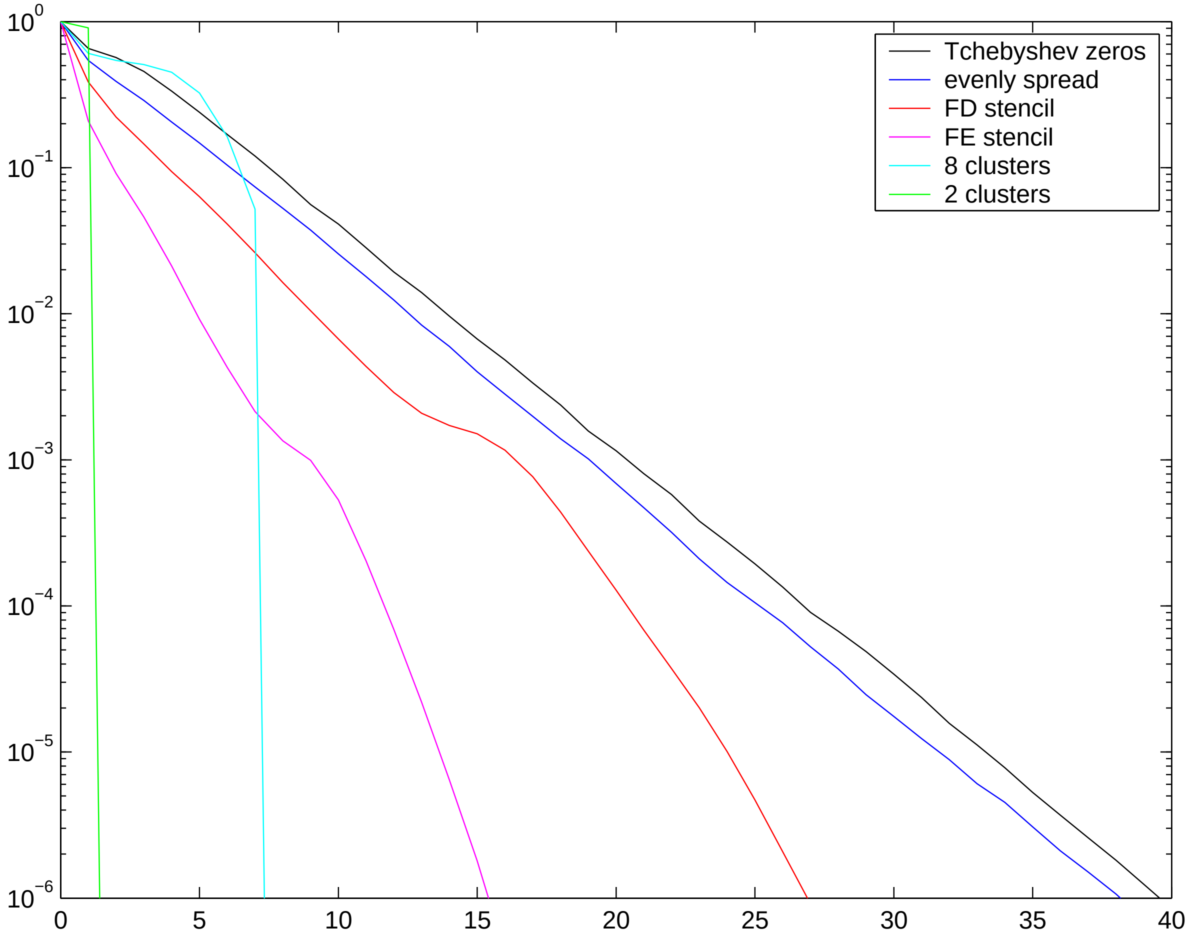
TCHEBYSHEV APPROXIMATION

Number of iterations required for convergence is

$$k \simeq \frac{1}{2}\ln\frac{2}{\epsilon}\sqrt{\kappa}$$

CG residual reduction

512×512 matrices, zero initial guess, random RHS
different eigenvalue spectra, same condition number



PDE examples

3D uniform grid with n nodes per dimension

$$r, s, t = 1, \ldots, n$$

7 point Finite Difference Stencil

$$\lambda^{rst} = 1 - \frac{1}{3}\cos\frac{r\pi}{n+1} - \frac{1}{3}\cos\frac{s\pi}{n+1} - \frac{1}{3}\cos\frac{t\pi}{n+1}$$

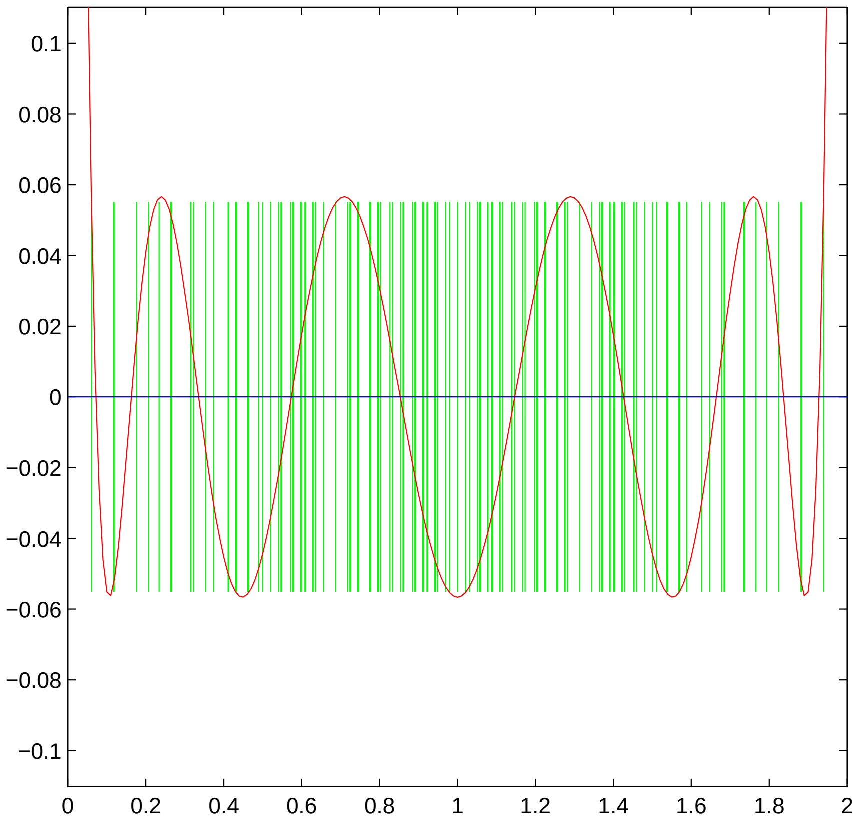
27 point Finite Element Stencil

$$\lambda^{rst} = 1 - \frac{1}{4}\cos\frac{r\pi}{n+1}\cos\frac{s\pi}{n+1} - \frac{1}{4}\cos\frac{r\pi}{n+1}\cos\frac{t\pi}{n+1}$$

$$-\frac{1}{4}\cos\frac{s\pi}{n+1}\cos\frac{t\pi}{n+1} - \frac{1}{4}\cos\frac{r\pi}{n+1}\cos\frac{s\pi}{n+1}\cos\frac{t\pi}{n+1}$$

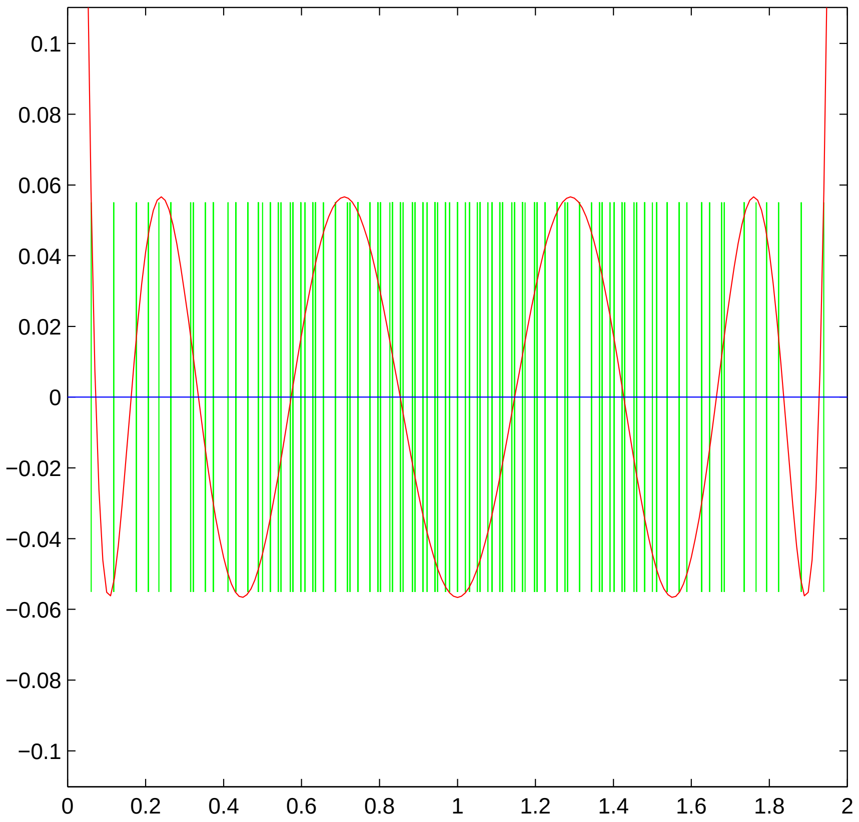
7 point Finite Difference Stencil

Tchebyshev error: 0.5662e-1

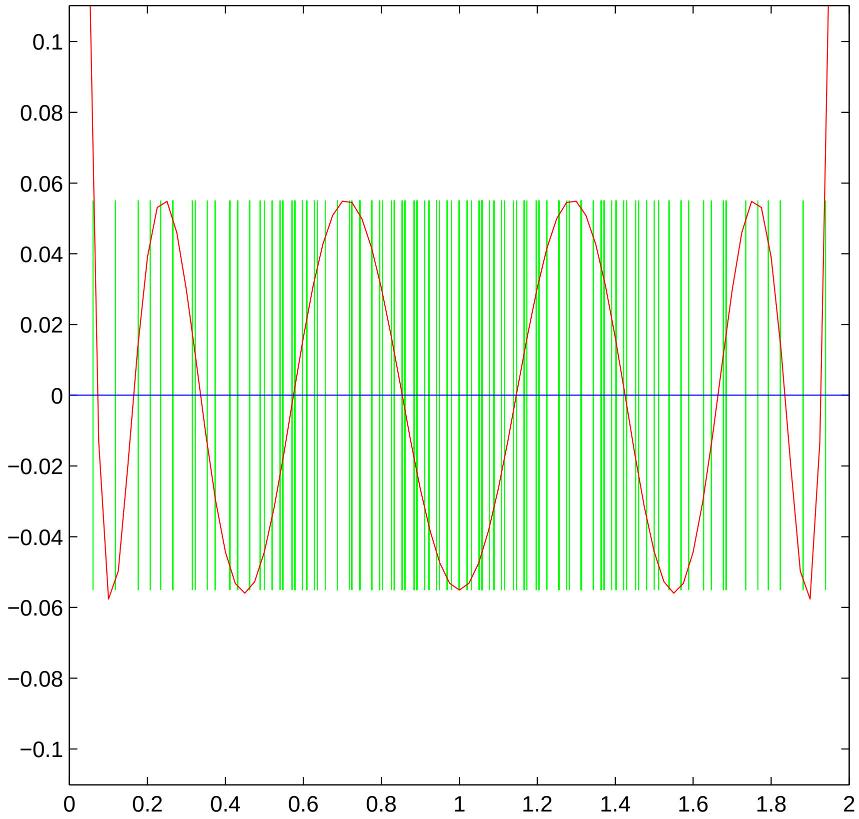


7 point Finite Difference Stencil

Tchebyshev error: 0.5662e-1

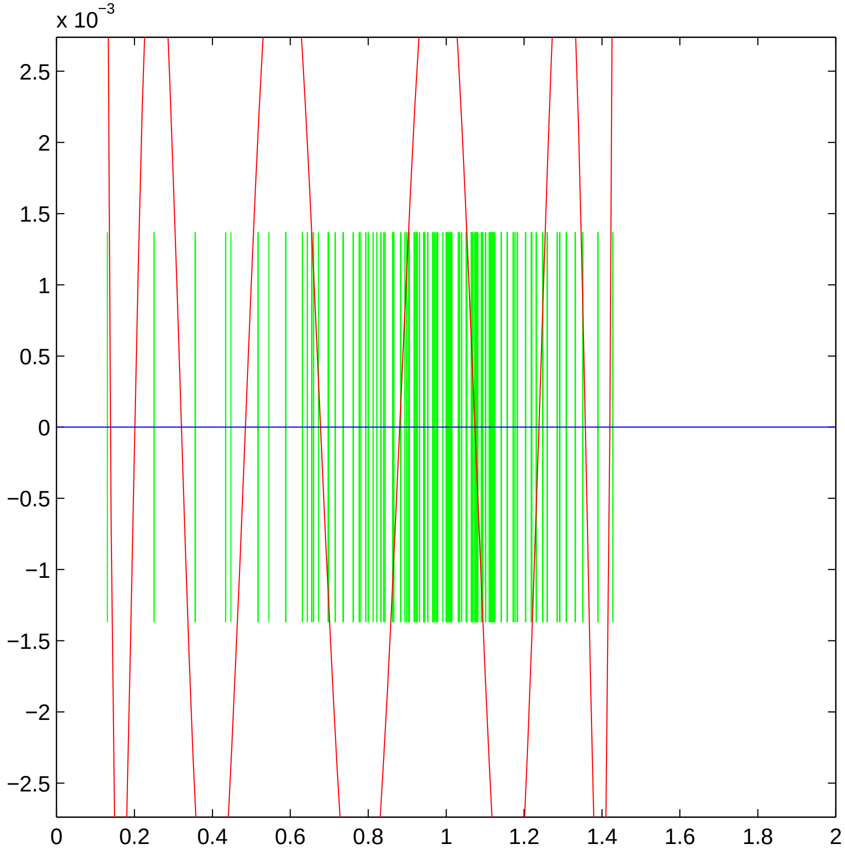


Minimax error: 0.5507e-1



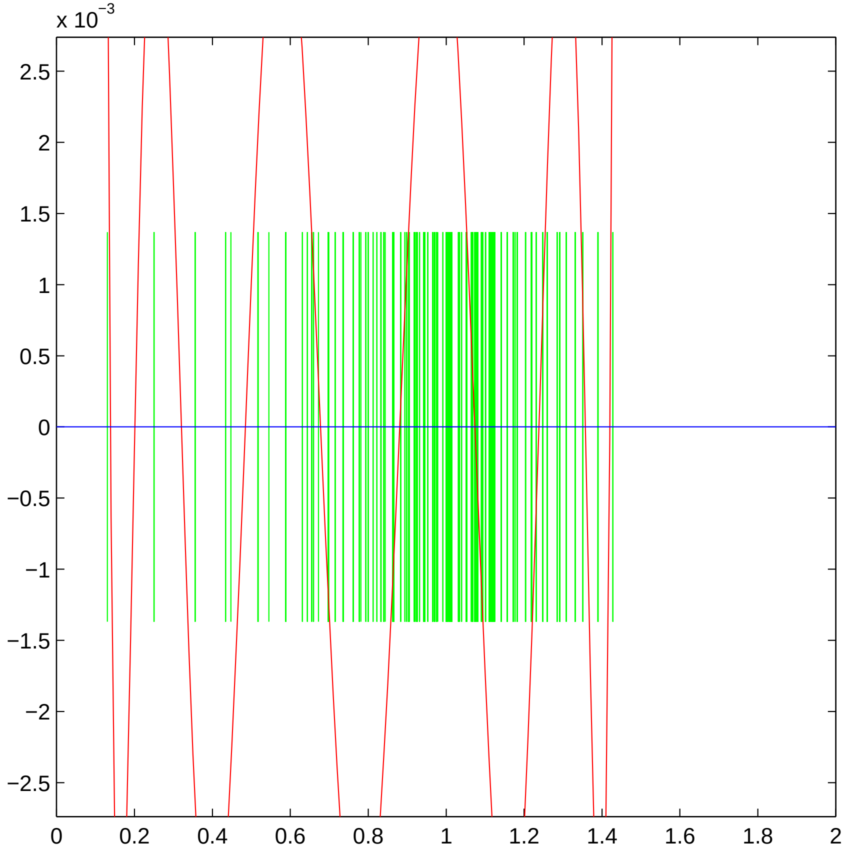
27 point Finite Element Stencil

Tchebyshev error: 0.3918e-2

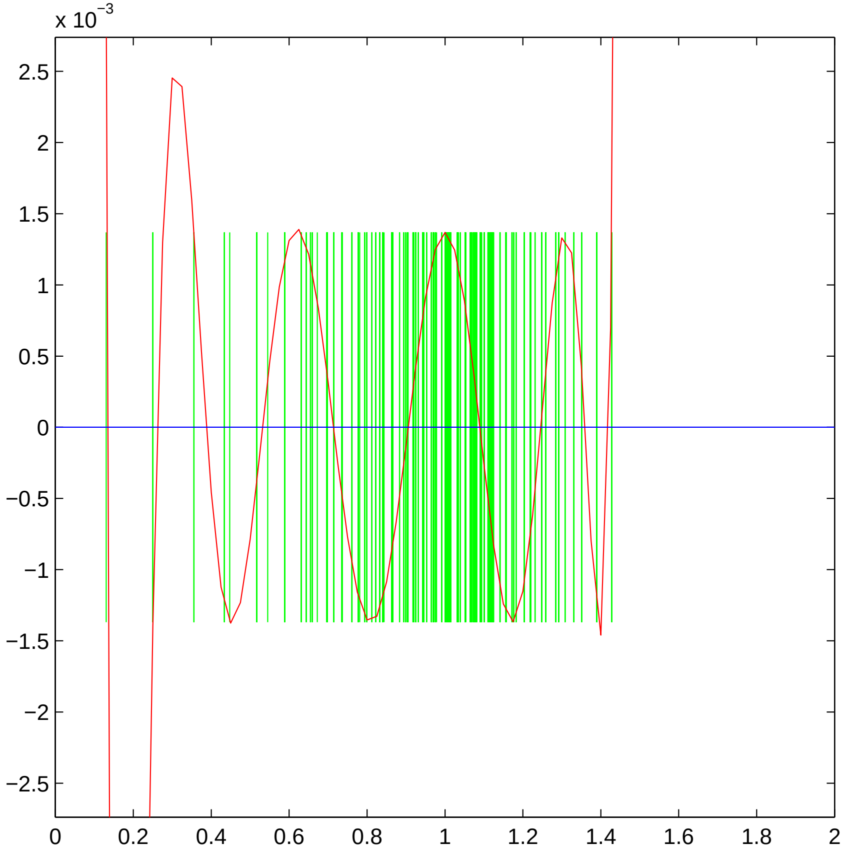


27 point Finite Element Stencil

Tchebyshev error: 0.3918e-2



Minimax error: 0.1369e-2



CG Method In Practice

- Advantages:
  - involves only matrix-vector and dot products
  - exact solution obtained in at most s iterations

CG Method In Practice

- Advantages:
  - involves only matrix-vector and dot products
  - exact solution obtained in at most s iterations
- Problems:
  - s may be very large
  - rounding error means theoretical properties lost

CG Method In Practice

- Advantages:
  - involves only matrix-vector and dot products
  - exact solution obtained in at most s iterations
- Problems:
  - s may be very large
  - rounding error means theoretical properties lost
- Solution:
  - reduce the number of CG steps required by applying PRECONDITIONING (more later . . .)

2. Symmetric Indefinite Systems

If A is symmetric indefinite:

- A has both positive and negative (nonzero) eigenvalues
- $\mathbf{v}^T A \mathbf{v}$ may equal zero for some n-vector $\mathbf{v} \neq 0$

Potential problems with CG:

- A can no longer be used to define a norm
- breakdown may occur: denominator of α_k could be zero (or close to zero)

Conjugate Residual Method

Stiefel (1955)

Solve $A^2\mathbf{x} = A\mathbf{b}$ by CG method

CR method constructs iterates

$$\mathbf{x}_k \in \mathbf{x}_0 + \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0\}$$

with properties

- $\mathbf{x}_k$ minimises $\|\mathbf{x}_k - \hat{\mathbf{x}}\|_{A^2} = \|\mathbf{r}_k\|_2$
- $\mathbf{p}_k$ can be calculated via a three-term recurrence relation

CR Algorithm

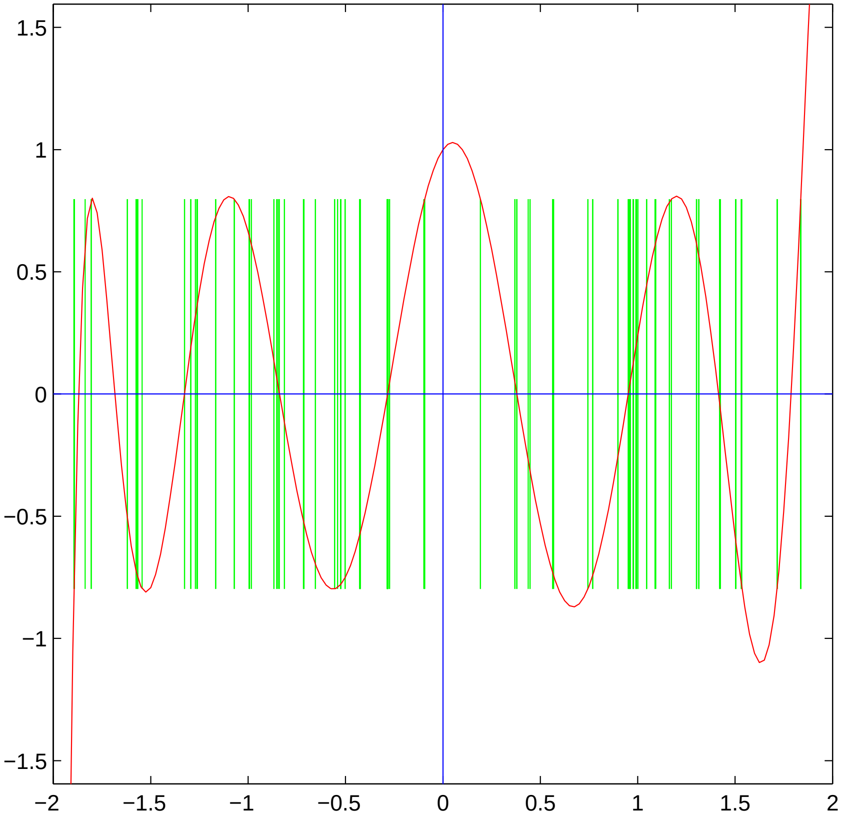
```
choose $\mathbf{x}_0$
compute $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
set $\mathbf{p}_0 = \mathbf{r}_0$
compute $A\mathbf{p}_0$
for $k = 0$ until convergence do
        $\alpha_k = \mathbf{r}_k^T \mathbf{r}_k / (A\mathbf{p}_k)^T A\mathbf{p}_k$
        $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
        $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k$
        $\beta_k = \mathbf{r}_{k+1}^T \mathbf{r}_{k+1} / \mathbf{r}_k^T A\mathbf{r}_k$
        $\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$
        $A\mathbf{p}_{k+1} = A\mathbf{r}_{k+1} + \beta_k A\mathbf{p}_k$
end do
```

can be implemented with one MVM per iteration

Convergence of CR method

$$\|\mathbf{x}_k - \hat{\mathbf{x}}\|_{A^2} \leq \min_{P \in \Pi_k^1} \max_i |P(\lambda_i)| \|\mathbf{x}_0 - \hat{\mathbf{x}}\|_{A^2}$$

For symmetric eigenvalue intervals, Tchebyshev bound implies number of iterations required to achieve convergence is $k \propto \kappa$



Alternative Methods

Potential problems with CR:

- breakdown may occur: denominator of α_k could be zero (or close to zero)
- CR algorithm is unstable in this form

Possible solution:

- generate an orthonormal basis for $\kappa(A, \mathbf{r}_0, k)$ in a more stable way, retaining the cheap three-term recurrence

$\Rightarrow$ mathematically equivalent but stable method . . .

MINRES

Paige and Saunders (1975)

Construct iterates

$$\mathbf{x}_k = \mathbf{x}_0 + V_k \mathbf{y}_k$$

satisfying

- $\mathbf{x}_k$ minimises $\|\mathbf{r}_k\|_2$
- $\mathbf{r}_k \perp \mathtt{span}\{A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^k\mathbf{r}_0\}$

$$V_k = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$$

$\mathbf{v}_k$ form an orthonormal basis for $\kappa(A, \mathbf{r}_0, k)$

Use the Lanczos method to find $\mathbf{v}_k$.

Lanczos Method

Lanczos (1950)

Three-term recurrence:

$$\beta_{j+1}\mathbf{v}_{j+1} = A\mathbf{v}_j - \alpha_j\mathbf{v}_j - \beta_j\mathbf{v}_{j-1}, \quad \alpha_j = \mathbf{v}_j^T A\mathbf{v}_j, \;\; \mathbf{v}_1 = \frac{\mathbf{r}_0}{\beta_1}, \;\; \beta_1 = \|\mathbf{r}_0\|_2$$

$$T_k \equiv \mathtt{tri}[\beta_j, \alpha_j, \beta_{j+1}]$$

extreme eigenvalues of T_k → extreme eigenvalues of A

$$\hat{T}_k \;\equiv\; \begin{array}{cl} T_k & \text{first } k \text{ rows} \\ [0, \ldots, 0, \beta_{k+1}] & \text{last row} \end{array}$$

$$AV_k = V_{k+1}\hat{T}_k$$

MINRES ideas

- use Lanczos method to produce tridiagonal matrix $\hat{T}_k$

$$\begin{aligned} \mathbf{r}_k &= \mathbf{b} - A(\mathbf{x}_0 + V_k \mathbf{y}_k) \\ &= \mathbf{r}_0 - V_{k+1}\hat{T}_k \mathbf{y}_k \\ &= V_{k+1}(\beta_1 \mathbf{e}_1 - \hat{T}_k \mathbf{y}_k) \end{aligned}$$

minimise $\|\mathbf{r}_k\|_2 \equiv$ minimise $\|\beta_1 \mathbf{e}_1 - \hat{T}_k \mathbf{y}_k\|_2$

- least squares problem for $\mathbf{y}_k$:

Givens rotations, QR factorisation of $\hat{T}_k$

MINRES Algorithm

Fischer (1994)

```
choose $\mathbf{x}_0$
compute $\hat{\mathbf{v}}_0 = \mathbf{b} - A\mathbf{x}_0$                initialise
set $\beta_0 = \|\hat{\mathbf{v}}_0\|_2,\ \eta_0 = \beta_0$
set $c_0 = 1,\ c_{-1} = 1,\ s_0 = 0,\ s_{-1} = 0$

for $k = 0$ until convergence do

    $\mathbf{v}_{k+1} = \hat{\mathbf{v}}_k / \beta_k$
    $\alpha_{k+1} = \mathbf{v}_{k+1}^T A \mathbf{v}_{k+1}$                Lanczos
    $\hat{\mathbf{v}}_{k+1} = (A - \alpha_{k+1} I)\mathbf{v}_{k+1} - \beta_k \mathbf{v}_k$
    $\beta_{k+1} = \|\hat{\mathbf{v}}_0\|_2$
```

$$
\begin{aligned}
\hat{r}_1 &= c_k \alpha_{k+1} - c_{k-1} s_k \beta_k \\
r_1 &= \sqrt{\hat{r}_1{}^2 + \beta_{k+1}^2} \\
r_2 &= s_k \alpha_{k+1} + c_{k-1} c_k \beta_k \\
r_3 &= s_{k-1} \beta_k
\end{aligned}
$$

QR

$$
\begin{aligned}
c_{k+1} &= \hat{r}_1 / r_1 \\
s_{k+1} &= \beta_{k+1} / r_1
\end{aligned}
$$

Givens

$$
\begin{aligned}
\mathbf{w}_{k+1} &= (\mathbf{v}_{k+1} - r_2 \mathbf{w}_k - r_3 \mathbf{w}_{k-1}) / r_1 \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + c \eta_k \mathbf{w} \\
\eta_{k+1} &= -s \eta_k
\end{aligned}
$$

update

end do

Alternative methods

SYMMLQ

Paige and Saunders (1975)

Also has a strong Lanczos connection, but minimises the 2-norm of the error rather than the residual

ORTHODIR

Fletcher (1976)

ORTHOMIN/ORTHODIR

Chandra et al. (1977)

Equivalent to MINRES, closer in implementation to CG/CR

3. Nonsymmetric Systems

Faber and Manteuffel (1984 & 1987): there is no Krylov type method which retains both

(i) minimisation property

(ii) fixed length recurrence

3. Nonsymmetric Systems

Faber and Manteuffel (1984 & 1987): there is no Krylov type method which retains both

(i) minimisation property

(ii) fixed length recurrence

- Normal Equations
  solve $A^TA\mathbf{x} = A^T\mathbf{b}$ using CG

3. Nonsymmetric Systems

Faber and Manteuffel (1984 & 1987): there is no Krylov type method which retains both

(i) minimisation property

(ii) fixed length recurrence

- Normal Equations
  solve $A^T A\mathbf{x} = A^T\mathbf{b}$ using CG

- Minimum Residual Methods
  retain (i), sacrifice (ii)

3. Nonsymmetric Systems

Faber and Manteuffel (1984 & 1987): there is no Krylov type method which retains both

(i) minimisation property

(ii) fixed length recurrence

- Normal Equations
  solve $A^T A\mathbf{x} = A^T\mathbf{b}$ using CG
- Minimum Residual Methods
  retain (i), sacrifice (ii)
- Biorthogonalisation Methods
  retain (ii), sacrifice (i)

CGNR

Hestenes and Stiefel (1952)

Apply CG to normal equations

$$A^T A\mathbf{x} = A^T\mathbf{b}$$

Construct iterates

$$\mathbf{x}_k \in \mathbf{x}_0 + \mathtt{span}\{A^T\mathbf{r}_0, (A^TA)A^T\mathbf{r}_0, \ldots, (A^TA)^{k-1}A^T\mathbf{r}_0\}$$

satisfying

- $\mathbf{x}_k$ minimises $\|\mathbf{x}_k - \hat{\mathbf{x}}\|_{A^TA} = \|\mathbf{r}_k\|_2$
- $\mathbf{r}_k \perp \{AA^T\mathbf{r}_0, (AA^T)^2\mathbf{r}_0, \ldots, (AA^T)^k\mathbf{r}_0\}$

CGNR Convergence

$$\|\mathbf{r}_k\|_2 \leq \min_{p\in\Pi_k^1} \max_{z\in\Sigma^2} |p(z)| \|\mathbf{r}_0\|$$

where Σ contains the singular values of A

number of iterations required for convergence is

$$k \propto \kappa$$

where $\kappa = \kappa(A^T A) = [\kappa(A)]^2$

BUT . . . recall that this may be very pessimistic

Generalised Minimal Residual Method (GMRES)

Saad and Schultz (1986)

Construct iterates

$$\mathbf{x}_k = \mathbf{x}_0 + V_k \mathbf{y}_k$$

satisfying

- $\mathbf{x}_k$ minimises $\|\mathbf{r}_k\|_2$
- $\mathbf{r}_k \perp \mathrm{span}\{A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^k\mathbf{r}_0\}$ *shifted Krylov space*

$$V_k = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$$

$\mathbf{v}_k$ form an orthonormal basis for $\kappa(A, \mathbf{r}_0, k)$

Use the Arnoldi method to find $\mathbf{v}_k$

Arnoldi Method

Arnoldi (1951)

recurrence relation:

$$\mathbf{v}_{j+1} = \frac{\mathbf{w}_{j+1}}{\|\mathbf{w}_{j+1}\|_2}, \quad \mathbf{w}_{j+1} = A\mathbf{v}_j - \sum_{i=1}^{j} h_{i,j}\mathbf{v}_j, \quad h_{i,j} = \mathbf{v}_i^T A\mathbf{v}_j$$

$H_k \equiv$ upper Hessenberg, $(h_{i,j} = 0,\ i > j+1)$

extreme eigenvalues of $H_k \rightarrow$ extreme eigenvalues of A

$$\hat{H}_k \equiv \begin{array}{cl} H_k & \text{first } k \text{ rows} \\ [0,\ldots,0,h_{k,k+1}] & \text{last row} \end{array}$$

$$AV_k = V_{k+1}\hat{H}_k$$

GMRES

- use Arnoldi method to produce upper Hessenberg matrix $\hat{H}_k$

$$\text{minimise } \|\mathbf{r}_k\|_2 \equiv \text{minimise } \|\beta_1\mathbf{e}_1 - \hat{H}_k\mathbf{y}_k\|_2$$

- least squares problem for y_k:

Givens rotations, QR factorisation of $\hat{H}_k$
(modified Gram-Schmidt)

- alternative implementations available e.g. based on Householder orthogonalisation: extra work but better numerical properties

Convergence Behaviour of GMRES

- Residual norms satisfy

$$\|\mathbf{r}_k\|_2 = \min_{p_k \in \Pi_k^1} \|p_k(A)\mathbf{r}_0\|_2.$$

Convergence Behaviour of GMRES

- Residual norms satisfy

$$\|\mathbf{r}_k\|_2 = \min_{p_k \in \Pi_k^1} \|p_k(A)\mathbf{r}_0\|_2.$$

- Exact solution $\hat{\mathbf{x}}$ obtained in at most n steps.

Convergence Behaviour of GMRES

- Residual norms satisfy

$$\|\mathbf{r}_k\|_2 = \min_{p_k\in\Pi_k^1} \|p_k(A)\mathbf{r}_0\|_2.$$

- Exact solution $\hat{\mathbf{x}}$ obtained in at most n steps.
- If A is diagonalisable (i.e. $A = X\Lambda X^{-1}$) then

$$\|\mathbf{r}_k\|_2 \le \|X\|_2\|X^{-1}\|_2 \min_{p_k\in\Pi_k^1} \max_{\lambda_j} |p_k(\lambda_j)| \|\mathbf{r}_0\|_2.$$

Convergence Behaviour of GMRES

- Residual norms satisfy

$$\|\mathbf{r}_k\|_2 = \min_{p_k \in \Pi_k^1} \|p_k(A)\mathbf{r}_0\|_2.$$

- Exact solution $\hat{\mathbf{x}}$ obtained in at most n steps.
- If A is diagonalisable (i.e. $A = X\Lambda X^{-1}$) then

$$\|\mathbf{r}_k\|_2 \leq \|X\|_2 \|X^{-1}\|_2 \min_{p_k \in \Pi_k^1} \max_{\lambda_j} |p_k(\lambda_j)| \|\mathbf{r}_0\|_2.$$

- If S and R are the symmetric and skew-symmetric parts of A, and S is positive definite, then

$$\|\mathbf{r}_k\|_2 \leq \left(1 - \frac{\lambda_{\min}(S)^2}{\lambda_{\min}(S)\lambda_{\max}(S) + \rho(R)^2}\right)^{\frac{k}{2}} \|\mathbf{r}_0\|_2.$$

Some Variations . . .

Restarted GMRES

- restart GMRES every m steps
- no simple rule for choosing m: convergence speed may vary drastically with different values
- some convergence analysis available

Simpler GMRES

Walker and Zhou (1994)

- calculate orthonormal basis for $A\kappa(A, \mathbf{b}, k)$ directly
- may be useful for restarting with small m

Biorthogonalisation Methods

Nonsymmetric Lanczos Method

Generates two sets of biorthogonal vectors $\mathbf{v}_i$, $\mathbf{w}_i$

$$\hat{\mathbf{v}}_{j+1} = A\mathbf{v}_j - \delta_j \mathbf{v}_j - \gamma_j \mathbf{v}_{j-1}, \quad \hat{\mathbf{w}}_{j+1} = A^T\mathbf{w}_j - \delta_j \mathbf{w}_j - \eta_j \mathbf{w}_{j-1}$$

$$\mathbf{v}_{j+1} = \eta_{j+1}\hat{\mathbf{v}}_j, \quad \mathbf{w}_{j+1} = \gamma_{j+1}\hat{\mathbf{w}}_{j+1}$$

$$\begin{aligned} \mathtt{span}\{\mathbf{v}_i\} &= \mathtt{span}\{\mathbf{v}_0, A\mathbf{v}_0, \ldots, A^{k-1}\mathbf{v}_0\} \\ \mathtt{span}\{\mathbf{w}_i\} &= \mathtt{span}\{\mathbf{w}_0, A^T\mathbf{w}_0, \ldots, (A^T)^{k-1}\mathbf{w}_0\} \end{aligned}$$

$$AV_k = V_k G_k + [0 \ldots 0\ \eta_{k+1}\mathbf{v}_k], \quad A^T W_k = W_k G_k^T + [0 \ldots 0\ \gamma_{k+1}\mathbf{w}_k]$$

$$W_k^T V_k = I_k, \quad W_k^T A V_k = G_k, \quad V_k^T A^T W_k = G_k^T$$

where

$$G_k \equiv \mathtt{tri}[\gamma_j, \delta_j, \eta_{j+1}]$$

Biconjugate Gradient Method (BiCG)

Lanczos (1952), Fletcher (1976)

```
choose $\mathbf{x}_0$, compute $\mathbf{p}_0 = \mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
choose $\hat{\mathbf{r}}_0$
set $\hat{\mathbf{p}}_0 = \hat{\mathbf{r}}_0, \rho_0 = \hat{\mathbf{r}}_0^T \mathbf{r}_0$
for $k = 1, 2, \ldots$ until convergence do
        $\sigma_{k-1} = \hat{\mathbf{p}}_{k-1}^T A \mathbf{p}_{k-1}$
        $\alpha_{k-1} = \rho_{k-1} / \sigma_{k-1}$
        $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_{k-1} \mathbf{p}_{k-1}$
        $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_{k-1} A \mathbf{p}_{k-1}$
        $\hat{\mathbf{r}}_k = \hat{\mathbf{r}}_{k-1} - \alpha_{k-1} A^T \hat{\mathbf{p}}_{k-1}$
        $\rho_k = \hat{\mathbf{r}}_k^T \mathbf{r}_k$
        $\beta_{k-1} = \rho_k / \rho_{k-1}$
        $\mathbf{p}_k = \mathbf{r}_k + \beta_k \mathbf{p}_{k-1}$
        $\hat{\mathbf{p}}_k = \hat{\mathbf{r}}_k + \beta_k \hat{\mathbf{p}}_{k-1}$
end
```

BiCG Theory

Construct iterates

$$\mathbf{x}_k = \mathbf{x}_0 + \mathtt{span}\{\mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0\}$$

satisfying

- $\mathbf{r}_k \perp \mathtt{span}\{\hat{\mathbf{r}}_0, A\hat{\mathbf{r}}_0, \ldots, A^{k-1}\hat{\mathbf{r}}_0\}$
- three-term recurrence

Potential problems:

- wild oscillations in $\|\mathbf{r}_k\|_2$
- possible breakdowns:

$$\hat{\mathbf{p}}_{k-1}^T A\mathbf{p}_{k-1} = 0, \quad \hat{\mathbf{r}}_{k-1}^T \mathbf{r}_{k-1} = 0$$

when $\hat{\mathbf{r}}_{k-1} \neq 0$, $\mathbf{r}_{k-1} \neq 0$.

Look-ahead Lanczos

Possible breakdown: $\mathbf{v}_k^T \mathbf{w}_k = 0,\ \ \mathbf{v}_k \neq 0,\ \ \mathbf{w}_k \neq 0$

Problem: cannot scale to find Lanczos vectors corresponding to basis vectors $A^k \mathbf{v}_0$ and $(A^T)^k \mathbf{w}_0$

Solution: relax biorthogonality condition for l steps (may be fulfilled again for higher powers of A and A^T)

i.e. still satisfy

$$AV_k = V_k G_k + [0 \ldots 0\ \eta_{k+1} \mathbf{v}_k], \quad A^T W_k = W_k G_k^T + [0 \ldots 0\ \gamma_{k+1} \mathbf{w}_k]$$

but not $W_k^T V_k = I_k$ for l steps

G_k is now upper Hessenberg block tridiagonal

Quasi-Minimal Residual Method (QMR)

Freund and Nachtigal (1991)

Construct iterates

$$\mathbf{x}_k = \mathbf{x}_0 + V_k\mathbf{y}_k$$

where

$$V_k = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$$

$\mathbf{v}_k$ form a basis for $\kappa(A, \mathbf{r}_0, k)$

- nonsymmetric Lanczos: V_k is not unitary
- too expensive to minimise $\|\mathbf{r}_k\|_2 = \|V_{k+1}(\beta_1\mathbf{e}_1 - G_k\mathbf{y}_k)\|_2$
- quasi-minimal: minimise $\|\beta_1\mathbf{e}_1 - G_k\mathbf{y}_k\|_2$
- avoid Lanczos breakdown: do l steps of “look-ahead” Lanczos

- G_k is upper Hessenberg: QR factorisation via Givens rotations
- if A is symmetric, QMR $\equiv$ MINRES
- incurable breakdown (unlikely due to round-off)
- convergence results (X is a matrix of eigenvectors of G_k):

$$\|\mathbf{r}_k\|_2 \leq \|V_{k+1}\|_2 \|X\|_2 \|X^{-1}\|_2 \min_{p_k \in \Pi_k^1} \max_{\lambda_j} |p_k(\lambda_j)| \|\mathbf{r}_0\|_2$$

$$\|\mathbf{r}_k^{QMR}\|_2 \leq \kappa(V_{k+1}) \|\mathbf{r}_k^{GMRES}\|_2$$

Transpose-free Methods

Transpose-Free QMR (TFQMR)

Freund (1991), Chan et al. (1991), Freund and Szeto (1991)

- A^T can be eliminated by choosing a suitable starting vector

Transpose-free Methods

Transpose-Free QMR (TFQMR)

Freund (1991), Chan et al. (1991), Freund and Szeto (1991)

- A^T can be eliminated by choosing a suitable starting vector

Conjugate Gradients Squared (CGS)

Sonneveld (1989)

Construct iterates

$$\mathbf{x}_{2k} = \mathbf{x}_0 + \kappa(A, \mathbf{r}_0, 2k)$$

where $\mathbf{r}_{2k}^{CGS} = (p_k^{BiCG}(A))^2 \mathbf{r}_0$

- magnifies erratic convergence of BiCG
- may diverge when BiCG converges

BiCGSTAB

van der Vorst (1990)

Construct iterates

$$\mathbf{x}_{2k} = \mathbf{x}_0 + \kappa(A, \mathbf{r}_0, 2k)$$

where $\mathbf{r}_{2k}^{BiCGSTAB} = p_k^{BiCG}(A)\chi_k(A)\mathbf{r}_0$

- polynomial $\chi_k \in \Pi_k^1$ updated with a linear factor at each step

$$\chi_k(\lambda) = (1 - \mu_k\lambda)\chi_{k-1}(\lambda)$$

- free parameter μ_k determined via a local steepest descents problem
- convergence typically much smoother than CGS

BiCGSTAB

van der Vorst (1990)

Construct iterates

$$\mathbf{x}_{2k} = \mathbf{x}_0 + \kappa(A, \mathbf{r}_0, 2k)$$

where $\mathbf{r}_{2k}^{BiCGSTAB} = p_k^{BiCG}(A)\chi_k(A)\mathbf{r}_0$

- polynomial $\chi_k \in \Pi_k^1$ updated with a linear factor at each step

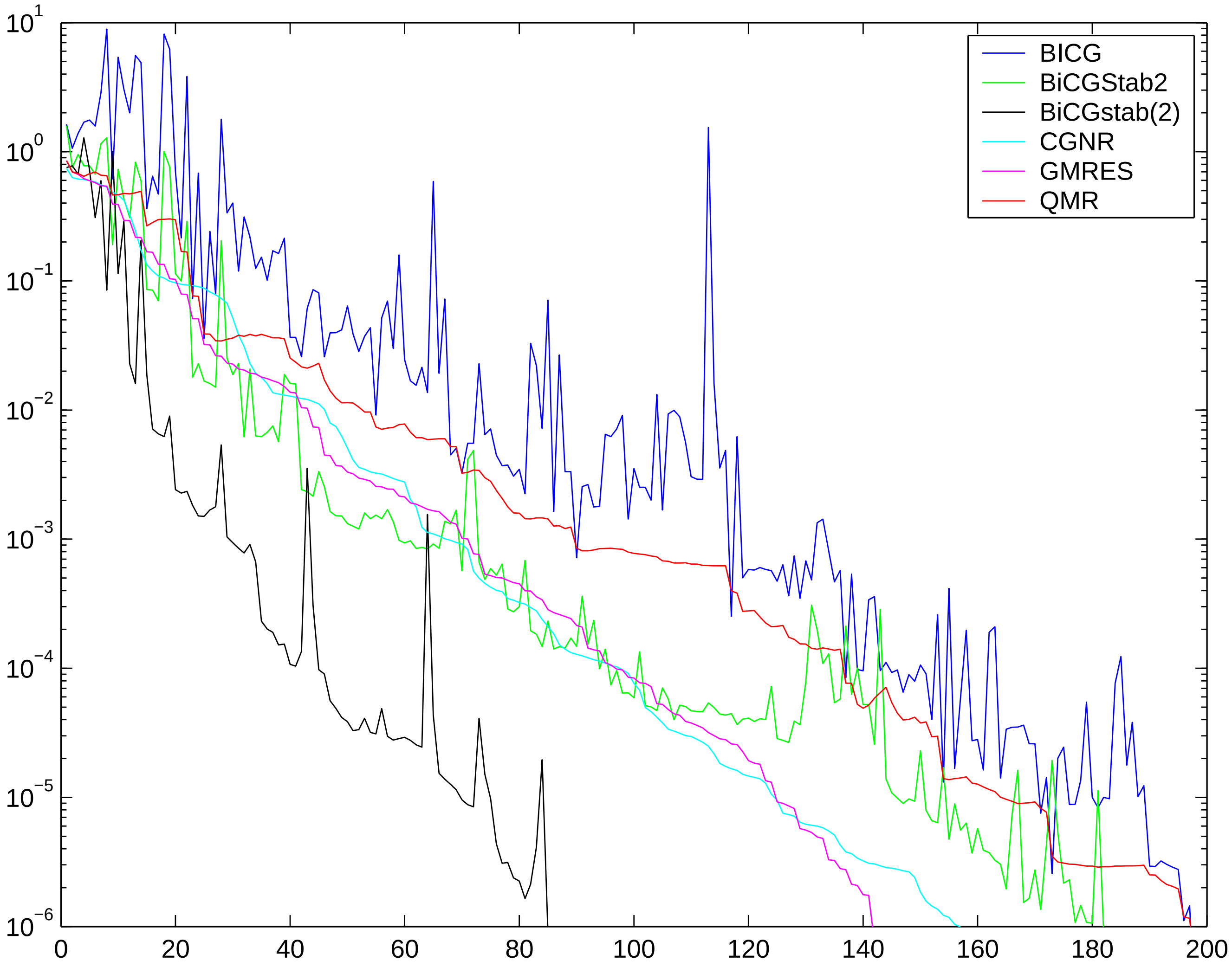
$$\chi_k(\lambda) = (1 - \mu_k \lambda)\chi_{k-1}(\lambda)$$

- free parameter μ_k determined via a local steepest descents problem
- convergence typically much smoother than CGS

BiCGStab2, Gutnecht (1993)
BiCGstab(l), Sleijpen and Fokkema (1993)

Example: Calculation of Invariant Tori

$$a(X,Y)\frac{\partial s}{\partial X} + b(X,Y)\frac{\partial s}{\partial Y} + c(X,Y)s = \psi$$



Preconditioning

Idea: instead of solving $A\mathbf{x} = \mathbf{b}$, solve

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$$

for some preconditioner M

Choose M so that

(i) eigenvalues of $M^{-1}A$ are well clustered
(ii) $M\mathbf{u} = \mathbf{r}$ is easily solved

Preconditioning

Idea: instead of solving $A\mathbf{x} = \mathbf{b}$, solve

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$$

for some preconditioner M

Choose M so that

(i) eigenvalues of $M^{-1}A$ are well clustered
(ii) $M\mathbf{u} = \mathbf{r}$ is easily solved

Extreme cases:

- $M = A$: good for (i), bad for (ii)
- $M = I$: good for (ii), bad for (i)

Practical Implementation

- preconditioner $M = M_1 M_2$
- solve $M\mathbf{u} = \mathbf{r}$, $M^T\mathbf{u} = \mathbf{r}$
- new system

$$[M_1^{-1} A M_2^{-1}][M_2 \mathbf{x}] = M_1^{-1}\mathbf{b} \Rightarrow \tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$$

- $\mathbf{x}_k = M_2^{-1}\tilde{\mathbf{x}}_k$, $\mathbf{r}_k = M_1 \tilde{\mathbf{r}}_k$

$$\begin{aligned} \text{central}: &\quad \text{as above} \\ \text{left}: &\quad M_2 = I \\ \text{right}: &\quad M_1 = I \end{aligned}$$

symmetric positive definite: if $M_2 = M_1^T$, resulting system is also symmetric positive definite

symmetric indefinite: M must be symmetric positive definite for MINRES; M can be indefinite with QMR

nonsymmetric:

- central: analysis may be easier
- left: if $M^{-1}A \simeq I$, $\tilde{\mathbf{r}}_k = M^{-1}A(\mathbf{x}_k - \hat{\mathbf{x}}) \simeq \mathbf{x}_k - \hat{\mathbf{x}}$, i.e.

$$\|\tilde{\mathbf{r}}_k\|_2 \simeq \|\mathbf{x}_k - \hat{\mathbf{x}}\|_2$$

- right: minimise in correct norm, i.e.

$$\|\tilde{\mathbf{r}}_k\|_2 = \|\mathbf{r}_k\|_2$$

Preconditioned Conjugate Gradient Method

Concus, Golub & O’Leary (1976)

```
choose $\mathbf{x}_0$
compute $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
solve $M\hat{\mathbf{r}}_0 = \mathbf{r}_0$
set $\mathbf{p}_0 = \mathbf{r}_0$
for $k = 0$ until convergence do
    $\alpha_k = \mathbf{r}_k^T\hat{\mathbf{r}}_k/\mathbf{p}_k^T A\mathbf{p}_k$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k$
    $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k$
    solve $M\hat{\mathbf{r}}_{k+1} = \mathbf{r}_{k+1}$
    $\beta_k = \mathbf{r}_{k+1}^T\hat{\mathbf{r}}_{k+1}/\mathbf{r}_k^T\hat{\mathbf{r}}_k$
    $\mathbf{p}_{k+1} = \hat{\mathbf{r}}_{k+1} + \beta_k\mathbf{p}_k$
end do
```

Connection with Stationary Methods

matrix splitting $A = M - N$

Iterates

$$\mathbf{x}_{k+1} = M^{-1} N \mathbf{x}_k + M^{-1}\mathbf{b} = \mathbf{x}_k + M^{-1}\mathbf{r}_k$$

where the error satisfies

$$\mathbf{x}_k - \hat{\mathbf{x}} = (I - M^{-1}A)^k(\mathbf{x}_0 - \hat{\mathbf{x}}).$$

$(I - M^{-1}A)$ is small ⇒ rapid convergence

good preconditioner $\equiv$ good splitting operator

Introduce iteration parameter α:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha M^{-1}\mathbf{r}_k$$

with error

$$\mathbf{x}_k - \hat{\mathbf{x}} = (I - \alpha M^{-1} A)^k (\mathbf{x}_0 - \hat{\mathbf{x}}).$$

Vary α from step to step: error becomes

$$\mathbf{x}_k - \hat{\mathbf{x}} = (I - \alpha_{k-1} M^{-1} A) \ldots (I - \alpha_0 M^{-1} A)(\mathbf{x}_0 - \hat{\mathbf{x}})$$

polynomial of degree k with constant term 1

$M = I$ gives CG method

Common Matrix Splittings

$$A = M - N, \qquad A = D + L + U$$

$$\begin{aligned}
\text{Richardson: } A &= I - (I - A) \\
\text{Jacobi: } A &= D - [-(L + L^T)] \\
\text{Gauss-Seidel: } A &= (D + L) - (-L^T) \\
\text{SOR: } A &= M_\omega - N_\omega \\
&= \frac{1}{\omega}(D + \omega L) - \frac{1}{\omega}[(1 - \omega)D - \omega L^T] \\
\text{SSOR: } A &= \frac{\omega}{2 - \omega}(M_\omega D^{-1} M_\omega^T - N_\omega D^{-1} N_\omega^T)
\end{aligned}$$

Incomplete LU Factorisation

Step 1: select set $J = \{(i,j) : 1 \leq i,j \leq N\}$ of index pairs (including all (i,i))

Step 2: perform LU factorisation and restrict all non-zeros to entries in J

$$A = LU - R = M - R$$

$$r_{ij} = 0,\ (i,j) \in J, \qquad r_{ii} = \alpha \sum_{i \neq j} r_{ij}$$

- ILU factorisations do not always exist
- very sequential in nature
- block matrix analogues

Parameterised Incomplete Factorisation

```
for $i = 1, \ldots, n$ do
    for $j = 1, \ldots, n$ do
        $s_{ij} = a_{ij} - \sum_{k=1}^{\min(i,j)-1} l_{ik} u_{kj}$
        if $(i, j) \in J$ then
            if $(i \geq j)$ then $l_{ij} = s_{ij}$
            if $(i < j)$ then $u_{ij} = s_{ij}$
        else
            $l_{ii} = l_{ii} + \alpha s_{ij}$
        endif
    enddo
    $u_{ii} = 1$
    for $j = i + 1, \ldots, n$ do
        $u_{ij} = u_{ij} / l_{ii}$
    enddo
enddo
```

Some Variations on ILU

- $J \equiv$ nonzero entries in A

  $\alpha = 0$: ILU, Meijerink and van der Vorst (1977)
  $\alpha = 1$: MILU, Gustafsson (1978)

- ILU(N), MILU(N)
  J includes N extra diagonals

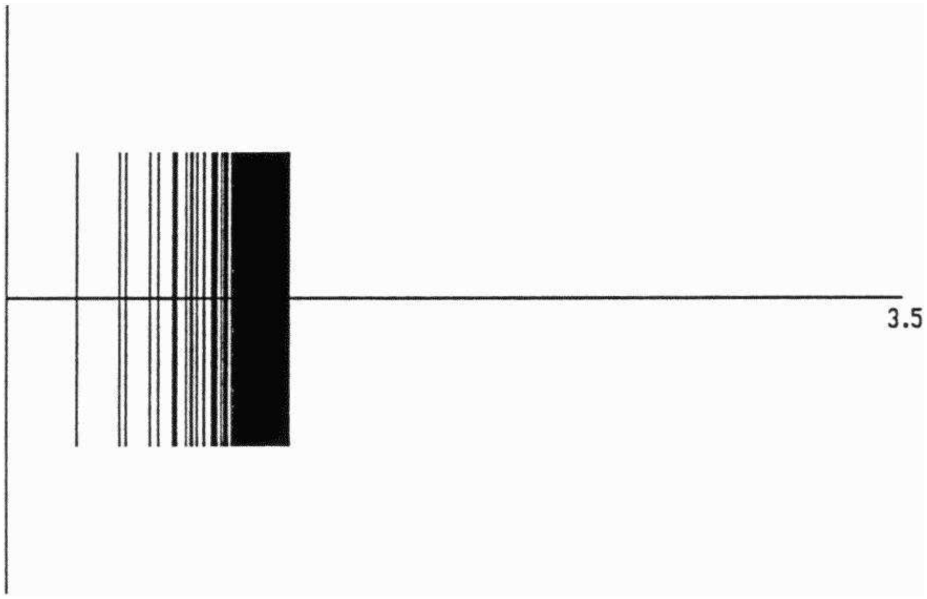
- ILU with Drop Tolerance, Munksgaard (1980)
  Drop all entries of fill-in with absolute value less than $\tau \in [10^{-4}, 10^{-2}]$.

- Shifted ILU, Manteuffel (1978,1980)
  Make A more diagonally dominant by factorising

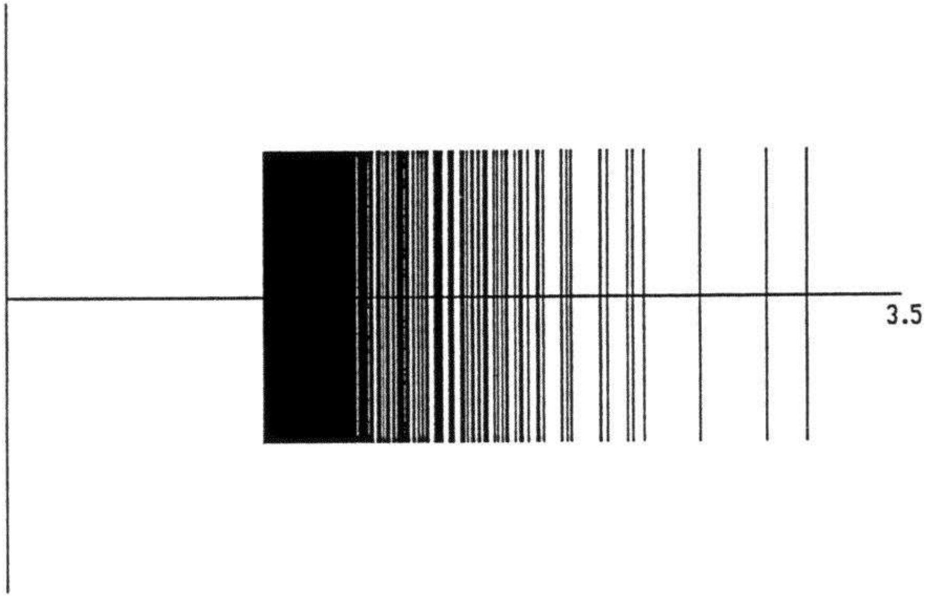
$$\bar{A} = D + \frac{1}{1+\gamma} C.$$

Sample Eigenvalue Plots

- seven point finite difference stencil



Incomplete Cholesky



Modified Incomplete Cholesky

Polynomial Preconditioning

Apply CG to

$$p(B^{-1}A)B^{-1}A\mathbf{x} = p(B^{-1}A)B^{-1}\mathbf{b}$$

i.e. use

$$M^{-1} = p(B^{-1}A)B^{-1}$$

so that $\mathbf{u} = M^{-1}\mathbf{r}$ is easily solved

Choose B to be a matrix splitting from stationary methods e.g.

- B from SSOR gives m-step CG method (Adams (1985))
- $B = I$ from Richardson (Ashby (1987))

Grid-based Preconditioners

- hierarchical basis preconditioners
  Yserentant (1986), Ong (1989)

  $$M = S^{-T} S^{-1}$$

  S is the linear transformation from standard to hierarchical finite element basis

- multigrid methods
  Hackbusch (1985)
  solve on a series of coarse to fine grids
  algrbraic multigrid

- domain decomposition
  Bramble, Pasciak and Schatz (1986)
  different preconditioners for different parts of the grid

Element-By-Element Method

Hughes, Levit & Winget (1983)

- assemble one element matrix A_e into G_e
- form $\bar{G}_e = I + D^{-\frac{1}{2}}(G_e - D_e)D^{-\frac{1}{2}}$
- factorise $\bar{G}_e = \mathcal{L}_e \mathcal{D}_e \mathcal{L}_e^T$
  ($\mathcal{L}_e$ is the assembly of lower $\triangle$ factor of related $\bar{A}_e$)
- preconditioner is

$$M = D^{\frac{1}{2}} \left[\prod_{e=1}^{E} \mathcal{L}_e \right] \left[\prod_{e=1}^{E} \mathcal{D}_e \right] \left[\prod_{e=E}^{1} \mathcal{L}_e^T \right] D^{\frac{1}{2}}$$

Element Factorisation Method

Kaasschieter (1989)

- factor element matrices $A_e = (D_e + L_e)D_e^+(D_e + L_e)^T$ (D_e^+ is the generalised inverse of D_e)
- number without maximal global node numbers (using e.g. Reverse Cuthill-McKee numbering)
- form $\mathcal{L} = L^T[L_e]L$, $\mathcal{D} = L^T[D_e]L$
- preconditioner is

$$M = (\mathcal{D} + \mathcal{L})\mathcal{D}^{-1}(\mathcal{D} + \mathcal{L}^T)$$

Examples of Stopping Criteria

- standard tests: $\|\mathbf{r}_k\|_2 \leq \epsilon, \quad \frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq \epsilon$

Examples of Stopping Criteria

- standard tests: $\|\mathbf{r}_k\|_2 \leq \epsilon, \quad \frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq \epsilon$
- with left preconditioning: $\|M_1\mathbf{r}_k\|_2 \leq \epsilon, \quad \frac{\|M_1\mathbf{r}_k\|_2}{\|M_1\mathbf{r}_0\|_2} \leq \epsilon$

Examples of Stopping Criteria

- standard tests: $\|\mathbf{r}_k\|_2 \leq \epsilon, \quad \frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq \epsilon$
- with left preconditioning: $\|M_1\mathbf{r}_k\|_2 \leq \epsilon, \quad \frac{\|M_1\mathbf{r}_k\|_2}{\|M_1\mathbf{r}_0\|_2} \leq \epsilon$
- condition number dependent (e.g. Ashby et al. (1990)):

$$\frac{\|\mathbf{x}_k - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}_0 - \hat{\mathbf{x}}\|_2} \leq \kappa(A)\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq \epsilon, \qquad \frac{\|\mathbf{x}_k - \hat{\mathbf{x}}\|_A}{\|\mathbf{x}_0 - \hat{\mathbf{x}}\|_A} \leq \left(\kappa_A(A)\left|\frac{\gamma_k}{\gamma_0}\right|\right)^{\frac{1}{2}}$$

Examples of Stopping Criteria

- standard tests: $\|\mathbf{r}_k\|_2 \leq \epsilon, \quad \frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq \epsilon$
- with left preconditioning: $\|M_1\mathbf{r}_k\|_2 \leq \epsilon, \quad \frac{\|M_1\mathbf{r}_k\|_2}{\|M_1\mathbf{r}_0\|_2} \leq \epsilon$
- condition number dependent (e.g. Ashby et al. (1990)):

$$\frac{\|\mathbf{x}_k - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}_0 - \hat{\mathbf{x}}\|_2} \leq \kappa(A)\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq \epsilon, \qquad \frac{\|\mathbf{x}_k - \hat{\mathbf{x}}\|_A}{\|\mathbf{x}_0 - \hat{\mathbf{x}}\|_A} \leq \left(\kappa_A(A)\left|\frac{\gamma_k}{\gamma_0}\right|\right)^{\frac{1}{2}}$$

- backward error analysis (e.g. Arioli et al. (1991)):

$$\frac{\|\mathbf{r}_k\|_\infty}{\|A\|_\infty\|\mathbf{x}_k\|_1 + \|\mathbf{r}_0\|_\infty} \leq \epsilon$$

Software Packages

Available from netlib:

- by WWW:

  http://www.netlib.org/master/expanded_liblist.html

- by email: mail **netlib**@**netlib.org** with

  send index from linalg

- by anonymous ftp:

  ftp.netlib.org

Some Examples

Netlib:

itpack	Young and Kincaid	FORTRAN
slap	Seager and Greenbaum	FORTRAN
linalg/laspack	Skalicky	C
linalg/qmrpack	Freund and Nachtigal	FORTRAN
linalg/templates	Barrett et al.	C, FORTRAN, MATLAB
linalg/cg	Eijkhout	PVM

Also:

cgcode	Ashby et al., LLNL	FORTRAN
AZTEC	Tuminaro et al. Sandia National Lab.	nCUBE2, IBM SP2, Intel Paragon, MPI

Some Relevant Books

- Templates for the Solution of Linear Systems. . . , Barrett et al. , SIAM (1994)
- Iterative Solution Methods, Axelsson, CUP (1996)
- Iterative Methods for Sparse Linear Systems, Saad, PWS (1996)
- Iterative Methods for Solving Linear Systems, Greenbaum, SIAM (1997)
- Computer Solution of Large Linear Systems, Meurant, North-Holland (1999)
- Iterative Krylov Methods for Large Linear Systems, van der Vorst, CUP (2003)

Some Summary Papers

- Iterative Solution of Linear Systems, Freund, Golub and Nachtigal, Acta Numerica (1991)
- Developments and Trends in the Parallel Solution of Linear Systems, Duff and van der Vorst, Parallel Computing 25 (1999)
- Numerical Progress in Eigenvalue Computation in the 20th Century, Golub and van der Vorst, J. Comp. and Appl. Math. 123 (2000)
- Iterative Solution of Linear Systems in the 20th Century, Saad and van der Vorst, J. Comp. and Appl. Math. 123 (2000)
- Preconditioning Techniques for Large Linear Systems: A Survey, Benzi, J. Comput. Phys. (2003)

Summary

- symmetric positive definite CG
- symmetric indefinite CR, MINRES, SYMMLQ
- nonsymmetric
  - normal equations

    CGNR
  - minimisation

    GMRES, GMRES(m)
  - biorthogonalisation

    BiCG, CGS, BiCGSTAB, BiCGstab(l), QMR

Summary

- symmetric positive definite CG
- symmetric indefinite CR, MINRES, SYMMLQ
- nonsymmetric
  - normal equations

    CGNR
  - minimisation

    GMRES, GMRES(m)
  - biorthogonalisation

    BiCG, CGS, BiCGSTAB, BiCGstab(l), QMR

PRECONDITION WISELY!