# ITERATIVE PROJECTION METHODS FOR SPARSE LINEAR SYSTEMS AND EIGENPROBLEMS
# CHAPTER 4 : CONJUGATE GRADIENT METHOD

Heinrich Voss

voss@tu-harburg.de

Hamburg University of Technology
Institute of Numerical Simulation

TUHH
Technische Universität Hamburg-Harburg

# Introduction

The conjugate gradient method, CG method for short, was developed independently by Hestenes (1951) and Stiefel (1952). Theoretically, it is a direct method for symmetric and positive definite linear systems, i.e. it yields the exact solution in a finite and predetermined number of operations. Differently from elimination methods, however, at different stages of the CG method one obtains approximate solutions of the system.

For a system of dimension $n$, at most $n$ steps of the CG method (using exact arithmetic) are needed, where at each step the approximate solution is refined progressively. Generally it is found that after a very small number of iterations a sufficiently good approximation is obtained and the process can be terminated. Hence, in practice the CG method is used as an iterative method.

# Introduction ct.

The iterative nature of the CG method was already mentioned in Hestenes & Stiefel (1952) but its importance for the numerical solution of large and sparse system was recognized only after the seminal paper of Reid (1971).

The history of the CG method and the Lanczos algorithm (which is closely related to the CG method) up to 1976 is presented in a review paper of Golub & O'Leary (1989) which contains an annotated bibliography of 291 items.

# A minimization problem

Suppose that the system matrix $A$ of the linear system of equations

$$Ax = b \qquad (1)$$

is symmetric and positive definite.

Then the quadratic functional

$$\phi(x) := \frac{1}{2}x^T Ax - x^T b, \quad x \in \mathbb{R}^n,$$

is strictly convex.

Hence, there exists a unique solution $x^*$ of the minimum problem

$$\phi(x) = \min!, \quad x \in \mathbb{R}^n, \qquad (2)$$

and from $\nabla\phi(x) = Ax - b$ we obtain that the minimum problem (2) and the linear system of equations (1) are equivalent.

# 1D minimizer

We now take advantage of the characterization of $x^*$ as the minimizer of $\phi$ to construct iterative methods for the numerical solution of (1).

Let $x^{k-1}$, $k \geq 1$, be an approximate solution of (2), and assume that the residual

$$r^{k-1} := b - Ax^{k-1} \neq 0$$

(for otherwise, $x^{k-1}$ already would be the solution of problem (2)).

We improve $x^{k-1}$ by a one dimensional line search. To this end we choose a search direction $d^k \neq 0$ and determine the minimum of $\phi$ on the line $x^{k-1} + \tau d^k$, $\tau \in \mathbb{R}$, i.e. we minimize the real equation

$$\begin{aligned} \psi(\tau) &:= \phi(x^{k-1} + \tau d^k) \\ &= \frac{1}{2}(x^{k-1} + \tau d^k)^T A(x^{k-1} + \tau d^k) - (x^{k-1} + \tau d^k)^T b \\ &= \frac{1}{2}\tau^2 (d^k)^T A d^k - \tau (d^k)^T r^{k-1} + \phi(x^{k-1}). \qquad (3) \end{aligned}$$

# 1D minimizer ct.

From $(d^k)^T A d^k > 0$ we obtain that $\psi$ has a unique minimum, which is characterized by $\psi'(\tau) = \tau (d^k)^T A d^k - (d^k)^T r^{k-1} = 0$, i.e.

$$\tau_k = \frac{(d^k)^T r^{k-1}}{(d^k)^T A d^k}. \qquad (4)$$

As a new approximation to the minimizer $x^*$ of (2) we therefore choose

$$x^k := x^{k-1} + \tau_k d^k.$$

If $r^k := b - Ax^k \neq 0$, then we repeat the one dimensional line search with a new search direction $d^{k+1}$.

# Reduction of $\phi$

From (3) and (4) we obtain the following reduction of the functional $\phi$ in the step above:

$$\begin{aligned}\phi(x^{k-1}) - \phi(x^k) &= \phi(x^{k-1}) - \phi(x^{k-1} + \tau_k d^k) = \phi(x^{k-1}) - \psi(\tau_k) \\ &= -\frac{1}{2}\left(\frac{(d^k)^T r^{k-1}}{(d^k)^T A d^k}\right)^2 \cdot (d^k)^T A d^k + \frac{(d^k)^T r^{k-1}}{(d^k)^T A d^k} \cdot (d^k)^T r^{k-1} \\ &= \frac{1}{2} \cdot \frac{((d^k)^T r^{k-1})^2}{(d^k)^T A d^k}.\end{aligned}$$

Hence, a reduction of the size of $\phi$ occurs in the line search step, if and only if the residual $r^{k-1}$ and the search direction $d^k$ are not orthogonal.

# Energy norm

Obviously, for the solution $x^*$ of $Ax = b$

$$\begin{aligned}(x-x^*)^T A(x-x^*) &= x^TAx - 2x^TAx^* + (x^*)^TAx^* \\ &= x^TAx - 2x^Tb - (x^*)^TAx^* + 2(x^*)^Tb \\ &= 2\Big(\phi(x) - \phi(x^*)\Big) \qquad \text{for every } x \in \mathbb{R}^n.\end{aligned}$$

Hence for any subset $D \subset \mathbb{R}^n$ the vector $\tilde{x} \in D$ is a minimizer of $\phi$ in $D$, if and only if it minimizes the energy norm

$$\|x^* - \tilde{x}\|_A := \sqrt{(x^* - \tilde{x})^T A(x^* - \tilde{x})}$$

of the error of $\tilde{x}$ in $D$. □

# Example

The Gauß-Seidel method can be interpreted as a line search method if we choose the unit vectors cyclically as search directions, i.e.

$$d^{jn+k} := u^k = (\delta_{ik})_{i=1,\dots,n}, \quad k = 1,\dots,n,\ j \in \mathbb{N} \cup \{0\}.$$

Suppose that we arrived at the approximate solution $x^{jn+k-1}$. Then the search direction is $u^k$, and hence,

$$\tau_{jn+k} = \frac{(r^{jn+k-1})^T u^k}{(u^k)^T A u^k} = \frac{r_k^{(nj+k-1)}}{a_{kk}} = \frac{1}{a_{kk}}\left(b_k - \sum_{i=1}^{n} a_{ki} x_i^{(nj+k-1)}\right),$$

and the new iterate is given by

$$x_i^{(nj+k)} = \begin{cases} x_i^{(nj+k-1)} & \text{if} \quad i \neq k \\ x_k^{(nj+k-1)} + \frac{1}{a_{kk}}\left(b_k - \sum_{\nu=1}^{n} a_{k\nu} x_\nu^{(nj+k-1)}\right) & \text{if} \quad i = k \end{cases}$$

# Example ct.

If additionally we relax the increment:

$$x_k^{(nj+k)} := x_k^{(nj+k-1)} + \omega \frac{1}{a_{kk}} \left( b_k - \sum_{\nu=1}^{n} a_{k\nu} x_\nu^{(nj+k-1)} \right)$$

for some $\omega \in \mathbb{R}$, then we obtain the SOR method.

Similarly as in (4) we obtain for a relaxed line search the decrease of the functional

$$\phi(x^{k-1}) - \phi(x^{k-1} + \omega \tau_k d^k) = \omega(1 - 0.5\omega) \frac{((d^k)^T r^{k-1})^2}{(d^k)^T A d^k}.$$

Hence, the SOR method decreases the functional $\phi$ in every step if and only if $\omega \in (0, 2)$, and this is exactly the set of parameters for which convergence of the SOR method occurs. □

# Steepest descent method

Given the approximation $x^{k-1}$ to the minimizer $x^*$, the functional $\phi$ decreases most rapidly in the direction of the negative gradient

$$-\nabla\phi(x^{k-1}) = b - Ax^{k-1} = r^{k-1}.$$

Hence, locally the optimum choice of the search direction is the residual $d^k := r^{k-1}$. The appertaining algorithm is the steepest descent method.

1: $r^0 = b - Ax^0$
2: **for** $k = 1, 2, \ldots$ until convergence **do**
3: $\quad s^k = Ar^{k-1}$
4: $\quad \tau_k = \|r^{k-1}\|_2^2 \,/\, (r^{k-1})^T s^k$
5: $\quad x^k = x^{k-1} + \tau_k r^{k-1}$
6: $\quad r^k = r^{k-1} - \tau_k s^k$
7: **end for**

Cost of steepest descent method: 1 matrix-vector product
2 scalar products
2 _axpys

# Zig-zagging

For ill-conditioned problems the convergence rate of the steepest descent method can be very slow.

Example

The steepest descent method yields for the functional

$$\phi(x) := x^T \begin{pmatrix} 10^{-3} & 0 \\ 0 & 10^3 \end{pmatrix} x$$

and the initial vector $x^0 := (100, 10^{-3})^T$ after 100 and 1000 iterations, respectively, the approximations

$$x^{100} = (99.4912, 0.000994912)^T, \quad x^{1000} = (95.0276, 0.000950276)^T$$

to the solution $x^* = 0$, which is not a substantial improvement upon the initial guess.

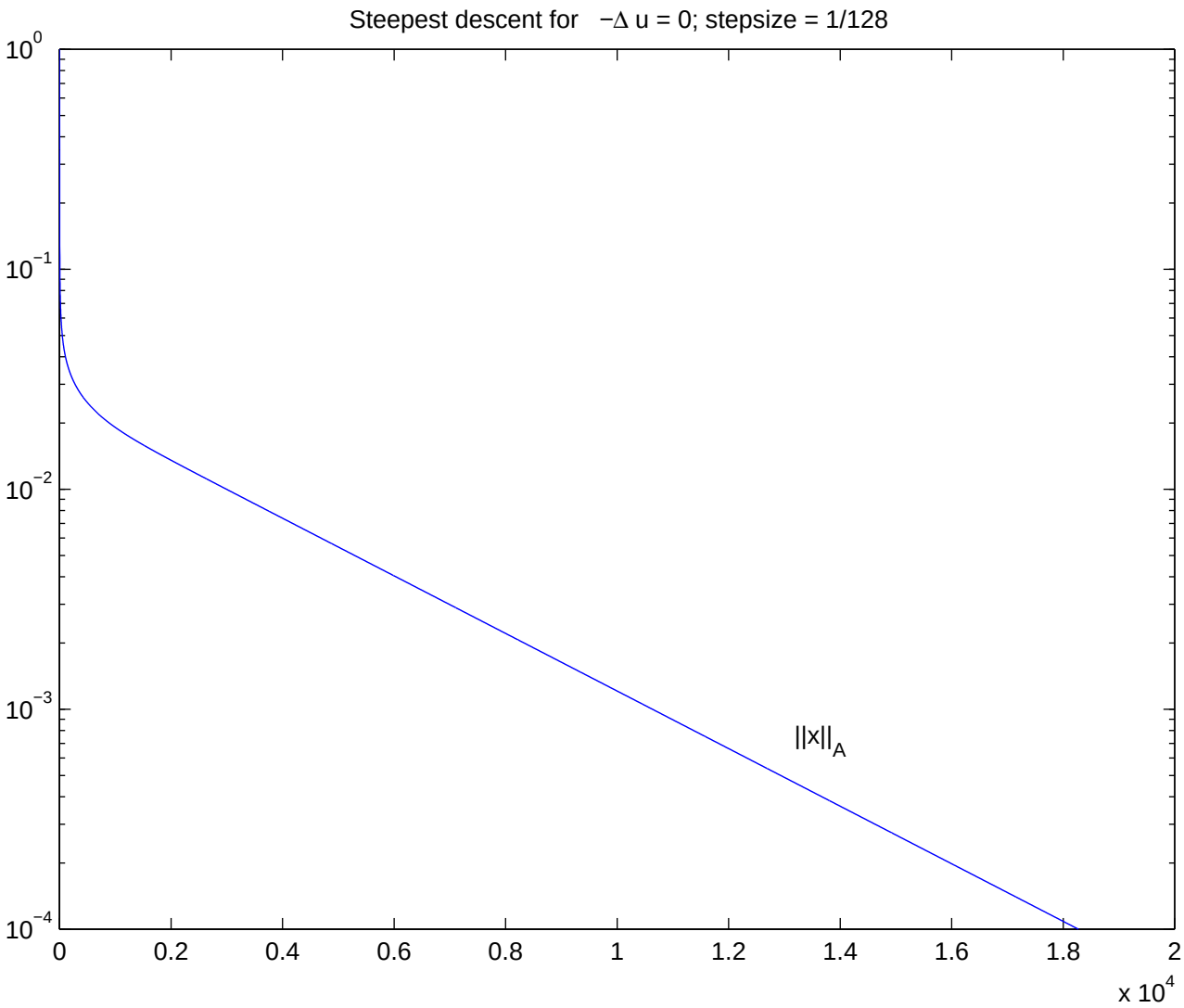
# Zig-zagging ct.

The reason for the bad performance is that in this example the level curves of $\phi$ are very elongated ellipses.

The gradient directions that arise in the iteration mainly point into the $x_2$-direction whereas the minimum is situated mainly in the $x_1$-direction. Hence, the method is zig-zagging through the flat and steep-sided valley $\{x \in \mathbb{R}^2 : \phi(x) \leq \phi(x^0)\}$.

For ill-conditioned problems this is the typical behavior of the steepest descent method. □

# Example


Steepest descent for −Δ u = 0; stepsize = 1/128
10^0
10^−1
10^−2
10^−3
10^−4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x 10^4
||x||_A

# Theorem 4.1

Let $A$ be a symmetric and positive definite matrix and denote by $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ the eigenvalues of $A$.

Then for every initial vector $x^0$ the sequence $\{x^k\}$, which is constructed by the steepest descent method, converges to the solution $x^*$ of $Ax = b$, and the following a priori error estimate with respect to the energy norm holds:

$$\|x^k - x^*\|_A \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^k \|x^0 - x^*\|_A.$$

# Proof

For $x \in \mathbb{R}^n$ and $r := b - Ax$ one step of the steepest descent method yields the iterate

$$y := x + \frac{\|r\|_2^2}{r^T A r}(b - Ax) =: (I - \theta(x)A)x + \theta(x)b,$$

from which we obtain for the error vectors

$$y - x^* = (I - \theta(x)A)(x - x^*).$$

Let

$$z := (I - \sigma A)x + \sigma b$$

Then it follows

$$\|y - x^*\|_A \leq \|z - x^*\|_A \quad \text{for every } \sigma \in \mathbb{R}.$$

# Proof ct.

We prove that for $\sigma = \tilde{\sigma} := 2/(\lambda_n + \lambda_1)$ the inequality

$$\|z - x^*\|_A \leq \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \|x - x^*\|_A$$

holds. Then the error bound is obtained by induction.

For $e := x - x^*$ and $\tilde{e} := z - x^*$ we get

$$\tilde{e} = (I - \tilde{\sigma} A)e =: Me.$$

The matrix $M$ is symmetric with eigenvalues $1 - \tilde{\sigma}\lambda_j$.

# Proof ct.

Hence, the spectral norm of $M$ is given by

$$\|M\|_2 = \max\left\{\left|1 - \frac{2\lambda_1}{\lambda_n + \lambda_1}\right|, \left|1 - \frac{2\lambda_n}{\lambda_n + \lambda_1}\right|\right\} = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} =: \eta,$$

from which we obtain

$$\begin{aligned} \|y - x^*\|_A &\leq \|\tilde{e}\|_A = \|A^{1/2}\tilde{e}\|_2 = \|A^{1/2}(I - \tilde{\sigma}A)e\|_2 \\ &= \|(I - \tilde{\sigma}A)A^{1/2}e\|_2 \leq \eta\|A^{1/2}e\|_2 = \eta\|e\|_A. \square \end{aligned}$$

# Remark

If $\lambda_1$ and $\lambda_n$ are the extreme eigenvalues of $A$, and if

$$\kappa_2(A) := \frac{\lambda_n}{\lambda_1}$$

denotes the condition number of $A$ with respect to the Euclidean norm then the error reduction factor of the steepest descent method with respect to the energy norm can be written as

$$\eta := \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} = \frac{\kappa_2(A) - 1}{\kappa_2(A) + 1}.$$

Hence, the smaller the condition number of $A$ is, the faster the steepest descent method converges.

In particular, for $\kappa_2(A) = 1$ one obtains $\eta = 0$.

Actually in this case $A = I$, and the gradient method yields the minimizer in the first step. □

# Example

We consider the discrete version of the model problem with stepsize $h = 1/m$.

The eigenvalues are

$$\lambda_{i,j} = 2\left(2 - \cos\frac{i\pi}{m+1} - \cos\frac{j\pi}{m+1}\right), \quad i,j = 1,\ldots,m.$$

Hence, the minimum eigenvalue is $\lambda_{1,1} = 4(1 - \cos\frac{\pi}{m+1})$ and the maximum eigenvalue is $\lambda_{m,m} = 4(1 + \cos\frac{\pi}{m+1})$, and we obtain the error reduction factor

$$\eta = \frac{\lambda_{m,m} - \lambda_{1,1}}{\lambda_{m,m} + \lambda_{1,1}} = \cos\frac{\pi}{m+1}.$$

For $h = 1/128$ one gets $\eta \approx 0.9997$. Hence, the convergence of the steepest descent is as fast (or as better as slow) as Jacobi's method.

# Ideal line search

To avoid the zig-zagging of the steepest descent method we ask for more suitable search directions. An ideal line search method would have the following property:

> *If the approximations $x^1, x^2, \ldots$ are determined by exact line searches along the search directions $d^1, d^2, \ldots$, then the minimizer $x^k$ of $\phi$ on the line $x^{k-1} + \tau d^k$, $\tau \in \mathbb{R}$, even minimizes the functional $\phi$ on the affine space $x^0 + \text{span}\{d^1, \ldots, d^k\}$, which is spanned by all preceding directions $d^j$, $j = 1, \ldots, k$.*

If we are able to construct linear independent directions with this property, then the corresponding line search method is not only convergent but it is finite, because after at most $n$ steps it finds the minimizer of $\phi$ on $x^0 + \text{span}\{d^1, \ldots, d^n\} = \mathbb{R}^n$. The line search method is said to have the finite termination property.

# Ideal line search ct.

Let $x \in x^0 + \text{span}\{d^1, \ldots, d^k\}$. Then $x$ has the representation

$$x = \tilde{x} + \tau d^k, \quad \tau \in \mathbb{R}, \ \tilde{x} = x^0 + \sum_{j=1}^{k-1} \tau_j d^j.$$

Similarly as for the 1D-line search

$$\phi(x) = \phi(\tilde{x}) + \tau \sum_{j=1}^{k-1} \tau_j (d^j)^T A d^k + \tau (d^k)^T r^0 + \frac{1}{2}\tau^2 (d^k)^T A d^k.$$

If $(d^j)^T A d^k = 0$ for $j = 1, \ldots, k-1$, then

$$\phi(x) = \phi(\tilde{x}) + \tau (d^k)^T r^0 + \frac{1}{2}\tau^2 (d^k)^T A d^k,$$

and the problem of minimizing $\phi$ over $x^0 + \text{span}\{d^1, \ldots, d^k\}$ decouples into a minimization over $x^0 + \text{span}\{d^1, \ldots, d^{k-1}\}$ and the minimization of the real equation

$$\psi(\tau) = \frac{1}{2}\tau^2 (d^k)^T A d^k - \tau (d^k)^T r^0,$$

# Ideal line search ct.

The minimizer $\tau_k$ of $\psi$ is $\tau_k = (d^k)^T r^0/(d^k)^T A d^k$, and from

$$\begin{aligned}(d^k)^T r^{k-1} &= \left(d^k\right)^T \left(b - A(x^0 + \sum_{j=1}^{k-1} \tau_j d^j)\right) \\ &= (d^k)^T r^0 - \sum_{j=1}^{k-1} \tau_j (d^k)^T A d^j = (d^k)^T r^0\end{aligned}$$

one obtains again

$$\tau_k = \frac{(d^k)^T r^{k-1}}{(d^k)^T A d^k}.$$

Hence, if the search directions $d^j$ satisfy $(d^i)^T A d^j = 0$ for $i \neq j$, then the minimization of $\phi$ over the affine space $x^0 + \text{span}\{d^1, \ldots, d^k\}$ can be replaced by the consecutive line searches along $d^1, \ldots, d^k$.

# Conjugate directions

Definition: Let $A \in \mathbb{R}^{n\times n}$ be symmetric and positive definite. The set of vectors $d^1, \ldots, d^k$ is called $A$-conjugate, if $d^j \neq 0$ for $j = 1, \ldots, k$ and

$$(d^i)^T A d^j = 0 \quad \text{for } i, j = 1, \ldots, k,\ i \neq j.$$

With this notion we can formulate our result as follows:

Theorem 4.2

Let $x^0 \in \mathbb{R}^n$ be given and assume that $x^1, x^2, \ldots$ have been determined by a line search method, where the search directions $d^1, d^2, \ldots$ are mutually $A$-conjugate. Then $x^k$ minimizes the functional $\phi$ over the affine space $x^0 + \text{span}\{d^1, \ldots, d^k\}$.

# Remark

From the minimality of $\phi(x^k)$ over $x^0 + \text{span}\{d^1, \ldots, d^k\}$ it follows that

$$(d^j)^T r^k = 0, \quad j = 1, \ldots, k,$$

for otherwise, if $(d^i)^T r^k \neq 0$ for some $i \in \{1, \ldots, k\}$, then there exists $\alpha \in \mathbb{R}$ such that $\phi(x^k + \alpha d^i) < \phi(x^k)$. □

# Method of conjugate directions

1: $r^0 = b - Ax^0$;
2: **for** $k = 1, 2, \ldots$ until $r^k = 0$ **do**
3: choose $d^k$ such that
$$(d^k)^T A d^j = 0,\ j = 1, \ldots, k-1, \quad \text{and} \quad (d^k)^T r^{k-1} \neq 0$$
4: $\tau_k = (d^k)^T r^{k-1} \big/ (d^k)^T A d^k$
5: $x^k = x^{k-1} + \tau_k d^k$
6: $r^k = b - Ax^k$
7: **end for**

# Remark

Two vectors $x, y \in \mathbb{R}^n \setminus \{0\}$ are $A$-conjugate, if and only if they are orthogonal with respect to the scalar product

$$\langle x, y \rangle_A := y^T A x.$$

Hence, if $d^1, \ldots, d^k$ are $A$-conjugate, then they are linearly independent.

Moreover, given any set of linearly independent vectors $v^1, \ldots, v^n \in \mathbb{R}^n$, one can construct a set of $n$ $A$-conjugate vectors by Gram-Schmidt orthogonalization of $v^1, \ldots, v^n$ with respect to the inner product $\langle \cdot, \cdot \rangle_A$. We will see that there is a less costly method to obtain an $A$-conjugate basis of $\mathbb{R}^n$. $\square$

# Conjugate Gradient Method

In order to execute the conjugate direction method we have to choose in the $k$-th step a vector $d^k$ such that $(d^k)^T A d^j = 0$, $j = 1, \ldots, k-1$, and $(d^k)^T r^{k-1} \neq 0$.

The first condition is satisfied by the $A$-orthogonal complement of any vector $v \notin \text{span}\{d^1, \ldots, d^{k-1}\}$ with respect to $\text{span}\{d^1, \ldots, d^{k-1}\}$.

To fulfill also the second requirement, we choose $v := r^{k-1}$, i.e.

$$d^k = r^{k-1} - \sum_{j=1}^{k-1} \frac{(d^j)^T A r^{k-1}}{(d^j)^T A d^j} d^j.$$

# CG method

For $d^k$ we actually obtain

$$\begin{aligned}(d^k)^T r^{k-1} &= (r^{k-1})^T r^{k-1} - \sum_{j=1}^{k-1} \frac{(d^j)^T A r^{k-1}}{(d^j)^T A d^j} \underbrace{(d^j)^T r^{k-1}}_{=0} \\ &= (r^{k-1})^T r^{k-1} > 0,\end{aligned}$$

and hence, $d^k$ is an admissible search direction.

The choice

$$d^k = r^{k-1} - \sum_{j=1}^{k-1} \frac{(d^j)^T A r^{k-1}}{(d^j)^T A d^j} d^j.$$

seems to have the disadvantages that it is very costly to perform and that all previous search directions $d^j$ have to be stored.

Actually the sum reduces to the last term. This follows from the following Lemma.

# Lemma 4.3

Let $x^0 \in \mathbb{R}^n$ be given and assume that the approximate solutions $x^j$, $j = 1, \ldots, k$, are determined by the conjugate direction method, where the search directions are given by .

$$d^k = r^{k-1} - \sum_{j=1}^{k-1} \frac{(d^j)^T A r^{k-1}}{(d^j)^T A d^j} d^j. \qquad (1)$$

Then the vectors $r^j$ and $d^j$ satisfy

$$(r^k)^T r^j = 0, \quad j = 0, \ldots, k-1, \qquad (2)$$

and

$$(d^j)^T A r^k = 0, \quad j = 1, \ldots, k-1. \qquad (3)$$

# Proof

From (1) it follows that

$$r^j = d^{j+1} + \sum_{i=1}^{j} \frac{(d^i)^T A r^j}{(d^i)^T A d^i} d^i,$$

and therefore, $(d^j)^T r^k = 0, \quad j = 1, \ldots, k$ yields

$$(r^k)^T r^j = \underbrace{(r^k)^T d^{j+1}}_{=0} + \sum_{i=1}^{j} \frac{(d^i)^T A r^j}{(d^i)^T A d^i} \underbrace{(r^k)^T d^i}_{=0} = 0$$

for $j = 1, \ldots, k-1$, i.e. (2).

# Proof ct.

To prove (3) (i.e. $(d^j)^T A r^k = 0$) we rewrite $d^{j+1}$, $j = 1, \ldots, k-1$, in the following way:

$$\begin{aligned} d^{j+1} &= r^j - \sum_{i=1}^{j} \frac{(d^i)^T A r^j}{(d^i)^T A d^i} d^i \\ &= b - Ax^j - \sum_{i=1}^{j} \frac{(d^i)^T A r^j}{(d^i)^T A d^i} d^i \\ &= b - A(x^{j-1} + \tau_j d^j) - \sum_{i=1}^{j} \frac{(d^i)^T A r^j}{(d^i)^T A d^i} d^i \\ &= b - Ax^{j-1} - \tau_j A d^j - \sum_{i=1}^{j} \frac{(d^i)^T A r^j}{(d^i)^T A d^i} d^i. \end{aligned}$$

# Proof ct.

From $\tau_j \neq 0$ (otherwise, $x^{j-1}$ would have been the minimizer of $\phi$ and the algorithm would have been terminated) we obtain

$$Ad^j = \frac{1}{\tau_j}\left(r^{j-1} - d^{j+1} - \sum_{i=1}^{j} \frac{(d^i)^T A r^j}{(d^i)^T A d^i}\, d^i\right),$$

and therefore, for $j = 1, \ldots, k-1$

$$(r^k)^T A d^j = \frac{1}{\tau_j}\left((r^k)^T r^{j-1} - (r^k)^T d^{j+1} - \sum_{i=1}^{j} \frac{(d^i)^T A r^j}{(d^i)^T A d^i}\,(r^k)^T d^i\right).$$

Finally from $(r^k)^T r^j = 0$ for $j = 0, \ldots, k-1$ and from $(d^j)^T r^k = 0$, for $j = 1, \ldots, k$, we get that all terms on the right hand side vanish. This completes the proof. □

From (1) and the last Lemma it follows that the current search direction $d^k$ is given by

$$d^k = r^{k-1} - \frac{(d^{k-1})^T A r^{k-1}}{(d^{k-1})^T A d^{k-1}} d^{k-1} =: r^{k-1} + \beta_k d^{k-1}.$$

Hence, we do not have to store the hole history of search directions to perform the $A$-orthogonalization of $r^{k-1}$, but only the most recent direction is needed.

It will turn out that the execution of the conjugate direction method requires the storage of four vectors.

# Improvement of CG method

We can even improve the efficiency of the algorithm by some simple conversions.

Firstly, the current residual can be updated according to

$$r^k = b - Ax^k = b - A(x^{k-1} + \tau_k d^k) = r^{k-1} - \tau_k A d^k.$$

This manipulation saves one matrix-vector product $Ax^k$ in every iteration because the matrix-vector product $Ad^k$ is needed in the calculation of $\tau_k$, anyhow.

From $r^k = r^{k-1} - \tau_k A d^k$ one gets for $k \geq 2$

$$Ad^{k-1} = \frac{1}{\tau_{k-1}}(r^{k-2} - r^{k-1}),$$

and the orthogonality of the residuals yields

$$(d^{k-1})^T A r^{k-1} = \frac{1}{\tau_{k-1}}(r^{k-2} - r^{k-1})^T r^{k-1} = -\frac{1}{\tau_{k-1}}(r^{k-1})^T r^{k-1},$$

and from $d^k = r^{k-1} + \beta_k d^{k-1}$ and $r^k = r^{k-1} - \tau_k A d^k$ we obtain

$$\begin{aligned}
(d^{k-1})^T A d^{k-1} &= (d^{k-1})^T A(d^{k-1} - \beta_{k-2} d^{k-2}) \\
&= (d^{k-1})^T A r^{k-2} = \frac{1}{\tau_{k-1}}(r^{k-2} - r^{k-1})^T r^{k-2} \\
&= \frac{1}{\tau_{k-1}}(r^{k-2})^T r^{k-2}.
\end{aligned}$$

Thus, the current search direction reads

$$d^k = r^{k-1} + \frac{(r^{k-1})^T r^{k-1}}{(r^{k-2})^T r^{k-2}} d^{k-1}.$$

Finally, we can replace $(d^k)^T r^{k-1}$ in $\tau_k$ by

$$(d^k)^T r^{k-1} = (r^{k-1} + \beta_k d^{k-1})^T r^{k-1} = (r^{k-1})^T r^{k-1},$$

which is needed in the determination of $\beta_k$, anyway.

Putting these simplifications together we finally arrive at the following method of conjugate gradients, which is essentially the method of Hestenes and Stiefel.

# Conjugate Gradient Algorithm

1: $r^0 := b - Ax^0$;
2: $\alpha_0 := (r^0)^T r^0$.
3: $d^1 = r^0$;
4: **for** $k = 1, 2, \ldots$ until $d^k == 0$ **do**
5: $\quad s^k = Ad^k$;
6: $\quad \tau_k = \alpha_{k-1} / (d^k)^T s^k$;
7: $\quad x^k = x^{k-1} + \tau_k d^k$;
8: $\quad r^k = r^{k-1} - \tau_k s^k$;
9: $\quad \beta_k = 1/\alpha_{k-1}$;
10: $\quad \alpha_k = (r^k)^T r^k$;
11: $\quad \beta_k = \beta_k \cdot \alpha_k$;
12: $\quad d^{k+1} = r^k + \beta_k d^k$;
13: **end for**

# Cost of CG method

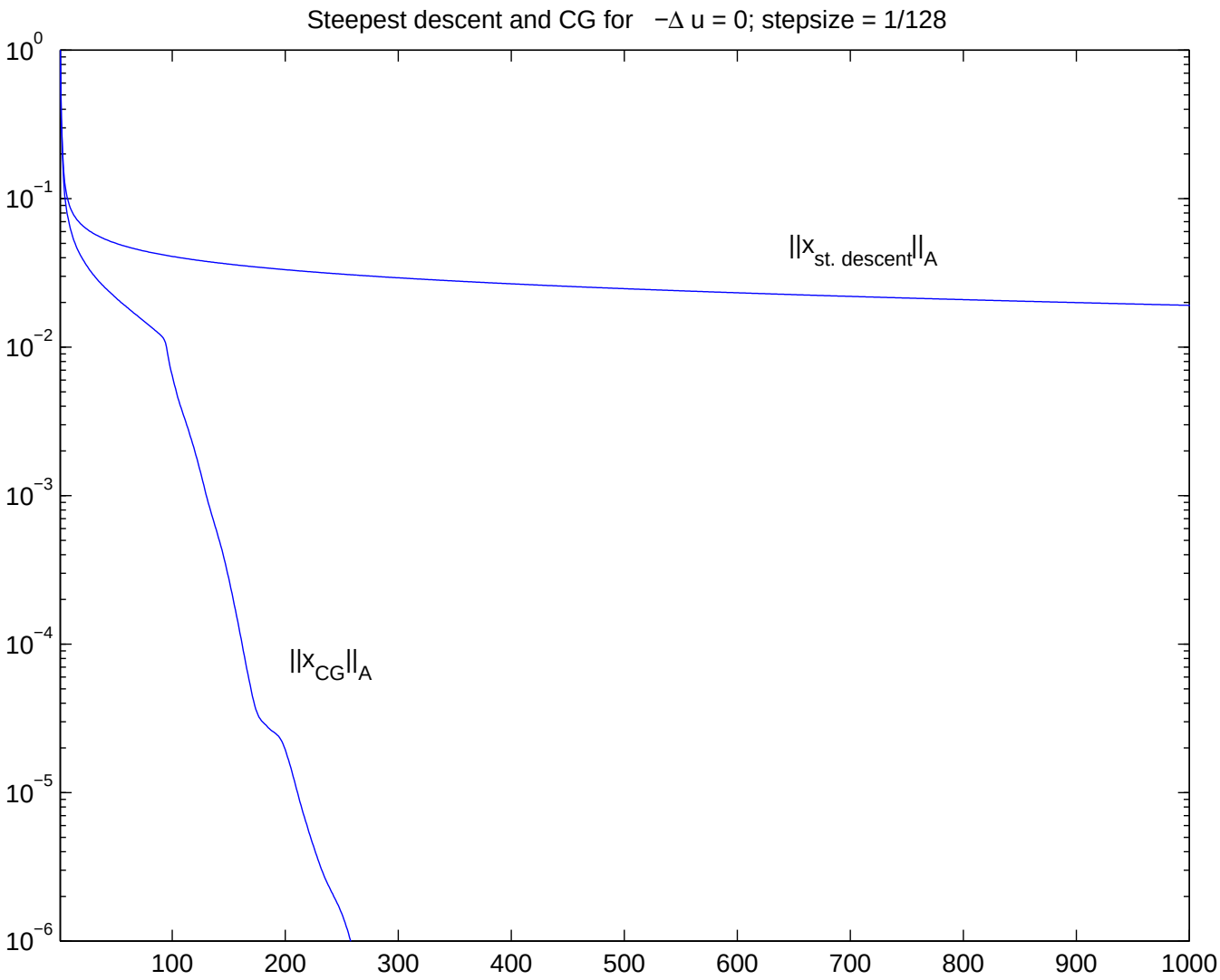
1 matrix-vector product
2 scalar products
3 _axpys

Storage requirements: 4 vectors

The termination criterion in the CG algorithm is unrealistic because rounding errors destroy the conjugacy among the search directions and finite termination usually will not appear.

The termination criterion should be based on a combination of a sufficient decrease of the residual and a maximum number of iterations.

# Example


Steepest descent and CG for −Δ u = 0; stepsize = 1/128
$10^{0}$
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
$10^{-5}$
$10^{-6}$
100
200
300
400
500
600
700
800
900
1000
$||x_{\text{st. descent}}||_A$
$||x_{CG}||_A$

# Error bound

Because the conjugate gradient method is a direct method, and hence, in exact arithmetic the solution is obtained in a finite number of steps, terms like 'convergence' or 'asymptotic reduction rate' do not make sense. In this section we shall derive an error estimate.

We already realized that for the solution $x^*$ of $Ax = b$ it holds

$$\|x - x^*\|_A^2 = 2\Big(\phi(x) - \phi(x^*)\Big) \qquad \text{for every } x \in \mathbb{R}^n.$$

Hence, $x^k$ minimizes the error $e := x^* - x$ with respect to the energy norm $\|\cdot\|_A$ over the affine space $x^0 + \text{span}\{d^1, \ldots, d^k\}$.

The representation of $\text{span}\{d^1, \ldots, d^k\}$ which is contained in the following lemma is the key to the error bound for the conjugate gradient method.

# Lemma 4.4

Assume that the search directions $d^1, \ldots, d^k$ are constructed by the conjugate gradient method. Then

$$\text{span}\{d^1, \ldots, d^k\} = \text{span}\{r^0, Ar^0, \ldots, A^{k-1}r^0\}.$$

**Definition**

Let $B \in \mathbb{R}^{n\times n}$ and $v \in \mathbb{R}^n$ be given. Then the linear space

$$\mathcal{K}_k(v, B) := \text{span}\{v, Bv, \ldots, B^{k-1}v\}$$

is called the $k$-th Krylov space of $B$ corresponding to $v$.

# Proof

For $k = 1$ the statement is trivial since $d^1 = r^0$.

Assume that it already has been shown for some $k < n$. Then (with the notation $d^0 = 0$) we obtain

$$\begin{aligned} d^{k+1} &= r^k + \beta_k d^k = r^{k-1} - \tau_k A d^k + \beta_k d^k \\ &= d^k - \beta_{k-1} d^{k-1} - \tau_k A d^k + \beta_k d^k, \qquad (*) \end{aligned}$$

and from $d^k, d^{k-1} \in \mathcal{K}_k(r^0, A)$ it follows that
$\text{span}\{d^1, \ldots, d^{k+1}\} \subset \mathcal{K}_{k+1}(r^0, A)$.

The converse inclusion also follows from (*) and $\tau_k \neq 0$. □

# Error bound ct.

As before we denote by $e^0 := x^* - x^0$ the error of $x^0$.Then from

$$r^0 = b - Ax^0 = A(x^* - x^0) = Ae^0$$

one obtains $A^j r^0 = A^{j+1} e^0$, and thus the elements of the Krylov space $z \in \mathcal{K}(r^0, A)$ can be written as

$$z = \sum_{j=1}^{k} \alpha_j A^j e^0, \quad \alpha_j \in \mathbb{R}.$$

Hence, it follows from (*) that the error $e^k := x^* - x^k$ of the $k$-th iterate $x^k$ satisfies

$$\begin{aligned} \|e^k\|_A^2 &= \min_{x \in x^0 + \mathcal{K}_k(r^0, A)} \|x^* - x\|_A^2 \\ &= \min_{z \in \mathcal{K}_k(r^0, A)} \|x^* - x^0 + z\|_A^2 \\ &= \min_{\alpha_1, \ldots, \alpha_k \in \mathbb{R}} \|e^0 + \alpha_1 Ae^0 + \cdots + \alpha_k A^k e^0\|_A^2. \qquad (**) \end{aligned}$$

# Error bound ct.

Let

$$\tilde{\Pi}_k := \{p \,:\, p \text{ is a polynomial of degree } \le k,\ p(0) = 1\}$$

be the set of polynomials $p$ of maximum degree $k$ such that $p(0) = 1$. Then (**) gets the form

$$\|e^k\|_A^2 = \min_{p \in \tilde{\Pi}_k} \|p(A)e^0\|_A^2.$$

Hence, the error of the $k$-th iterate can be represented as the product of a polynomial $p(A)$ in $A$ and the initial error $e^0$. Therefore, the CG method can be considered as a polynomial iteration, where in contrast to the Chebyshev iteration the underlying polynomial is constructed by the CG method itself.

More important, the polynomial does not depend on parameters (estimates of the bounds of the spectrum of $A$) which have to be prepared by the user in the Chebyshev method.

# Error bound ct.

To obtain a more convenient form of the error representation we consider the spectral decomposition

$$A = \sum_{j=1}^{n} \lambda_j z^j (z^j)^T$$

of $A$, where $z^j$, $j = 1, \ldots, n$, is an orthonormal set of eigenvectors of $A$ and $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the corresponding eigenvalues.

Then

$$A^2 = \sum_{i,j=1}^{n} \lambda_i \lambda_j z^i \underbrace{(z^i)^T z^j}_{=\delta_{ij}} (z^j)^T = \sum_{j=1}^{n} \lambda_j^2 z^j (z^j)^T,$$

and by induction

$$A^i = \sum_{j=1}^{n} \lambda_j^i z^j (z^j)^T.$$

# Error bound ct.

Therefore, from

$$e^0 =: \sum_{j=1}^{n} \gamma_j z^j$$

one gets

$$\|e^k\|_A^2 = \min_{p \in \tilde{\Pi}_k} \|\sum_{j=1}^{n} \gamma_j p(\lambda_j) z^j\|_A^2.$$

The definition of the energy norm yields

$$\begin{aligned} \Big\| \sum_{j=1}^{n} \gamma_j p(\lambda_j) z^j \Big\|_A^2 &= \left( \sum_{i=1}^{n} \gamma_i p(\lambda_i) z^i \right)^T A \left( \sum_{j=1}^{n} \gamma_j p(\lambda_j) z^j \right) \\ &= \sum_{j=1}^{n} \gamma_j^2 \lambda_j \Big( p(\lambda_j) \Big)^2. \end{aligned}$$

# Error bound ct.

Hence, we obtain the very important representation of the error of $x^k$ in the energy norm

$$\|e^k\|_A^2 = \min_{p \in \tilde{\Pi}_k} \sum_{j=1}^{n} \gamma_j^2 \lambda_j \Big(p(\lambda_j)\Big)^2 \qquad (1)$$

which will be discussed in detail at the end of this section.

# Coarse upper bound

First we deduce from (1) a coarse upper bound of the convergence properties of the CG method.

From

$$\|e^0\|_A^2 = \sum_{j=1}^{n} \lambda_j \gamma_j^2$$

we obtain by estimating the values $|p(\lambda_j)|$ by the maximum of $|p|$ on the spectrum of $A$

$$\|e^k\|_A \leq \min_{p \in \tilde{\Pi}_k} \max_{\lambda \in \mathrm{spec}(A)} |p(\lambda)| \cdot \|e^0\|_A,$$

and even coarser

$$\|e^k\|_A \leq \min_{p \in \tilde{\Pi}_k} \max_{\lambda_1 \leq \lambda \leq \lambda_n} |p(\lambda)| \cdot \|e^0\|_A,$$

and the special choice

$$p(\lambda) = C_k\left(\frac{2\lambda - \lambda_n - \lambda_1}{\lambda_n - \lambda_1}\right) \Big/ C_k\left(\frac{-\lambda_n - \lambda_1}{\lambda_n - \lambda_1}\right)$$

as shifted and scaled Chebyshev polynomial yields (cf. Chapter 3):

# Theorem 4.5

Let $A \in \mathbb{R}^{n\times n}$ be symmetric and positive definite, and denote by $\lambda_n$ and $\lambda_1$ the largest and smallest eigenvalue of $A$, respectively.

Let $x^0$ be any initial vector, and denote by $x^k$ the $k$-th approximate to the solution $x^*$ of $Ax = b$, which is obtained by the CG method.

Then the following a priori error estimate with respect to the energy norm holds:

$$\|x^k - x^*\|_A \le \frac{2R^k}{1+R^{2k}}\|x^0 - x^*\|_A, \quad R := \frac{\sqrt{\lambda_n}-\sqrt{\lambda_1}}{\sqrt{\lambda_n}+\sqrt{\lambda_1}}. \qquad (2)$$

Replacing the denominator in (2) by 1 we receive the error estimate

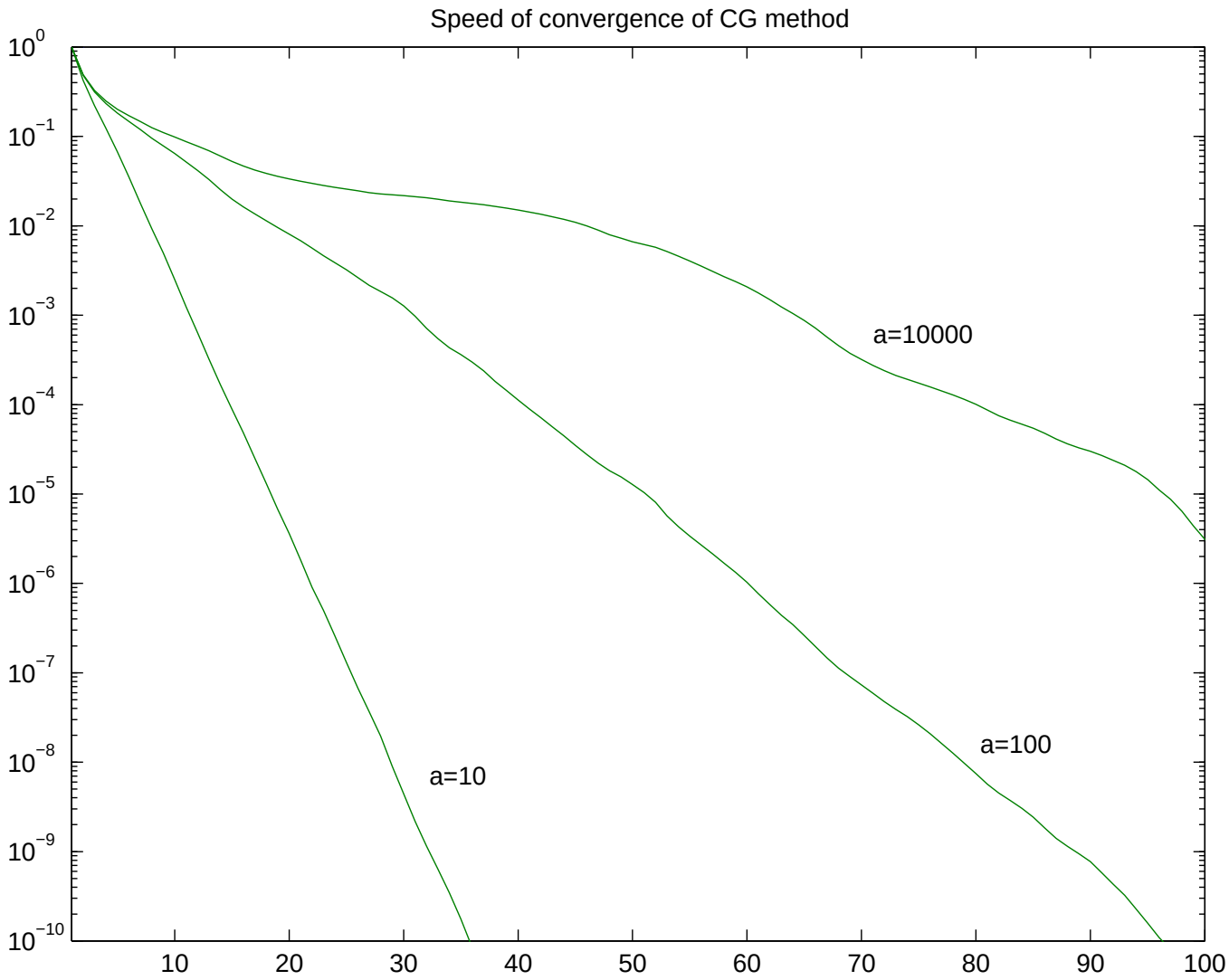
$$\|x^k - x^*\|_A \le 2\left(\frac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1}\right)^k \|x^0 - x^*\|_A. \qquad (3)$$

# Example

To demonstrate the dependence of the convergence of the CG method on the condition number, we consider the linear system of equations $Ax = 0$, where $A$ is a diagonal matrix of dimension 500, such that the entries are pseudo random numbers in the interval $[1, a]$.

The next figure contains the reduction of the Euclidean norms of the errors for the cases $a = 10$, $a = 100$ and $a = 10000$ in a semilogarithmic scale. □

# Example ct.


Speed of convergence of CG method
$10^{0}$
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
$10^{-5}$
$10^{-6}$
$10^{-7}$
$10^{-8}$
$10^{-9}$
$10^{-10}$
10
20
30
40
50
60
70
80
90
100
a=10000
a=100
a=10

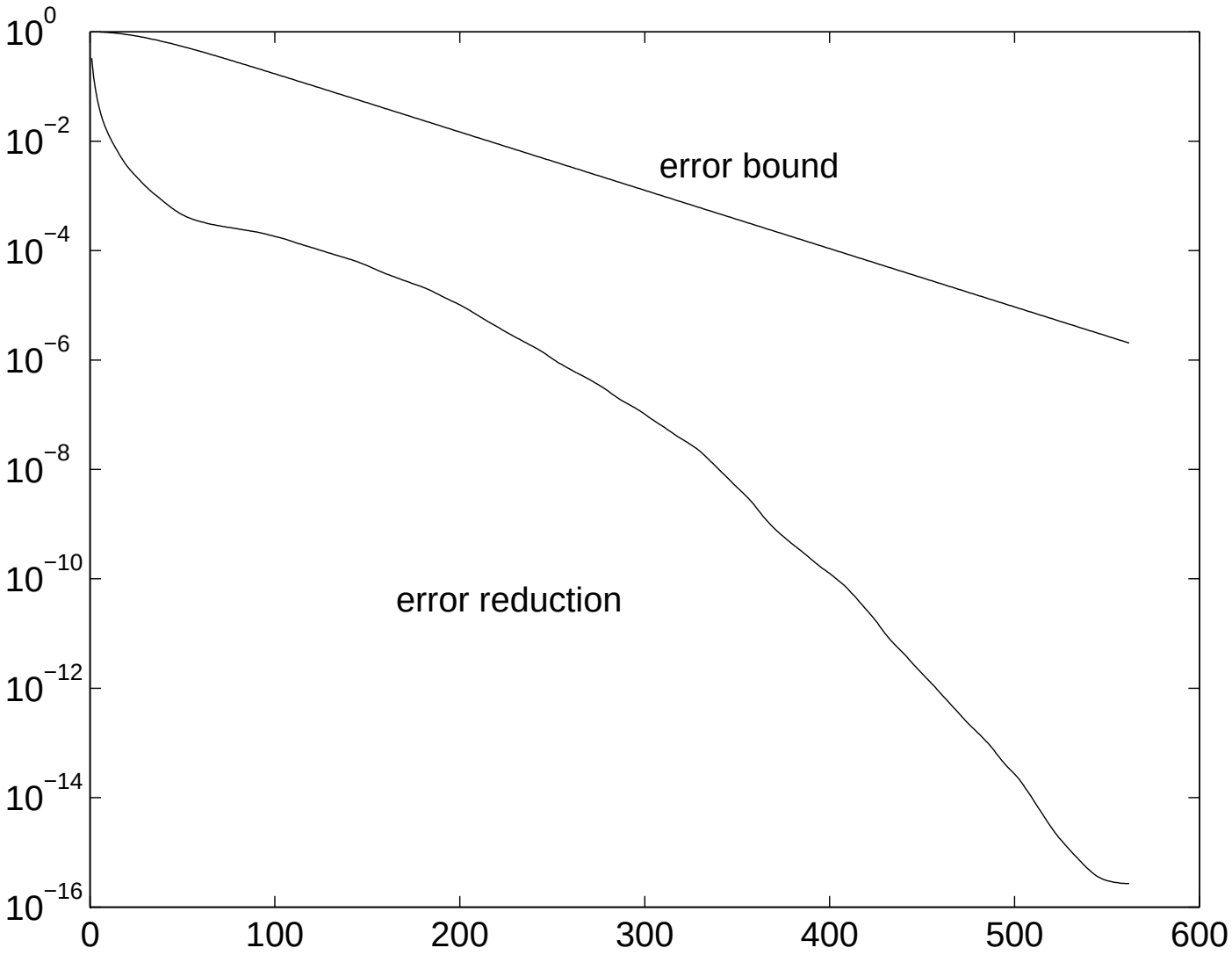
# Example

To demonstrate the quality of last Theorem the following figure contains the real reduction factors $\|e^k\|_A/\|e^0\|_A$ as well as those ones, which are predicted by (2) for the discrete version of the model problem with stepsize $h = 1/128$.

In this problem the condition number is

$$\kappa_2(A) = \frac{1+\cos\frac{\pi}{128}}{1-\cos\frac{\pi}{128}} \approx 6639.5.$$

# Example ct.


$10^{0}$
$10^{-2}$
$10^{-4}$
$10^{-6}$
$10^{-8}$
$10^{-10}$
$10^{-12}$
$10^{-14}$
$10^{-16}$
0
100
200
300
400
500
600
error bound
error reduction

# Improved error bound

The last Theorem provides only very coarse upper bounds of the speed of convergence of the CG method.

In particular, the finite termination property is not reflected by (2). This can be obtained from the finer estimate (1).

Because the matrix $A \in \mathbb{R}^{n\times n}$ has at most $n$ distinct eigenvalues, we obtain from (1) for every polynomial $q \in \tilde{\Pi}_k$, such that $q_k(\lambda_j) = 0$, $j = 1, \ldots, n$, that

$$\|e^k\|_A^2 = \min_{p\in\tilde{\Pi}_k} \sum_{j=1}^{n} \gamma_j^2 \lambda_j \Big(p(\lambda_j)\Big)^2 \le \sum_{j=1}^{n} \gamma_j^2 \lambda_j \Big(q_k(\lambda_j)\Big)^2 = 0. \qquad (4)$$

Moreover, we get

# Theorem 4.6

Let $A \in \mathbb{R}^{n\times n}$ be symmetric and positive definite, and let $x^k$ be the approximates to the solution $x^*$ of $Ax = b$, which are obtained by the CG method.

If $A$ has $m \leq n$ distinct eigenvalues, then in exact arithmetic the CG algorithm stops with $x^k = x^*$ after at most $m$ steps.

Proof

If $A$ has the distinct eigenvalues $\lambda_1 < \cdots < \lambda_m$, then

$$q_m(\lambda) := \prod_{j=1}^{m}(\lambda - \lambda_j)$$

in (4) yields $\|e^m\|_A^2 = 0$. □

# Clustered eigenvalues

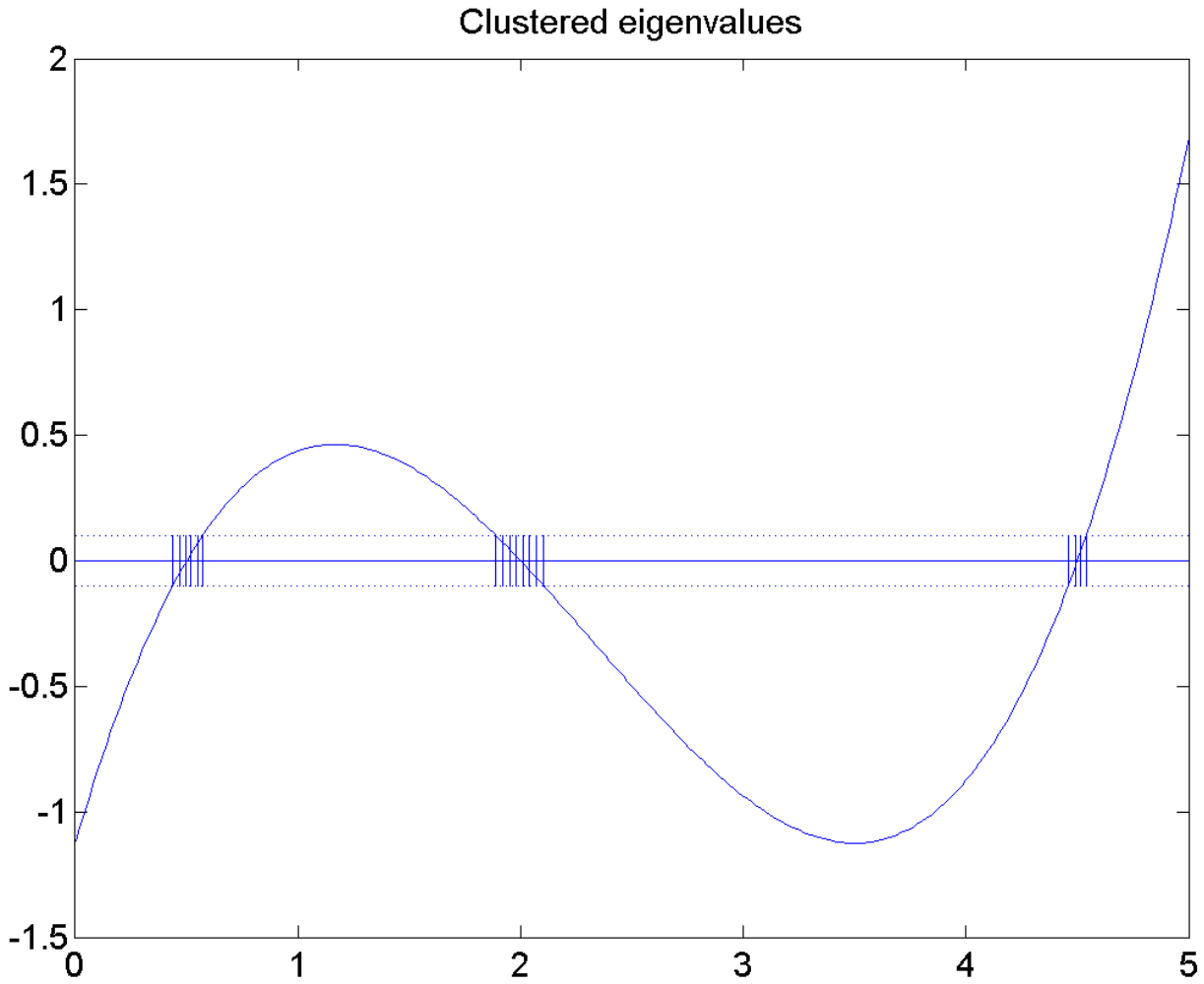
If the matrix $A$ does not have a very small number $m \ll n$ of distinct eigenvalues but they are contained in $m$ disjoint intervals $I_j$ of very small length, then the CG method will not stop with the exact solution after $m$ steps, but it will produce a very small residual after $m$ steps.

To see this we choose a polynomial $q$ of degree $m$ such that $q(0) = 1$, which possesses a root in each of the intervals $I_j$. Then by assumption all eigenvalues are close to one of the roots of $q$, and thus

$$\max\{|q(\lambda_j)| \ : \ \lambda_j \text{ is an eigenvalue of } A\},$$

which is an upper bound of the reduction of the error in $m$ steps, will be small.

# Clustered eigenvalues ct.

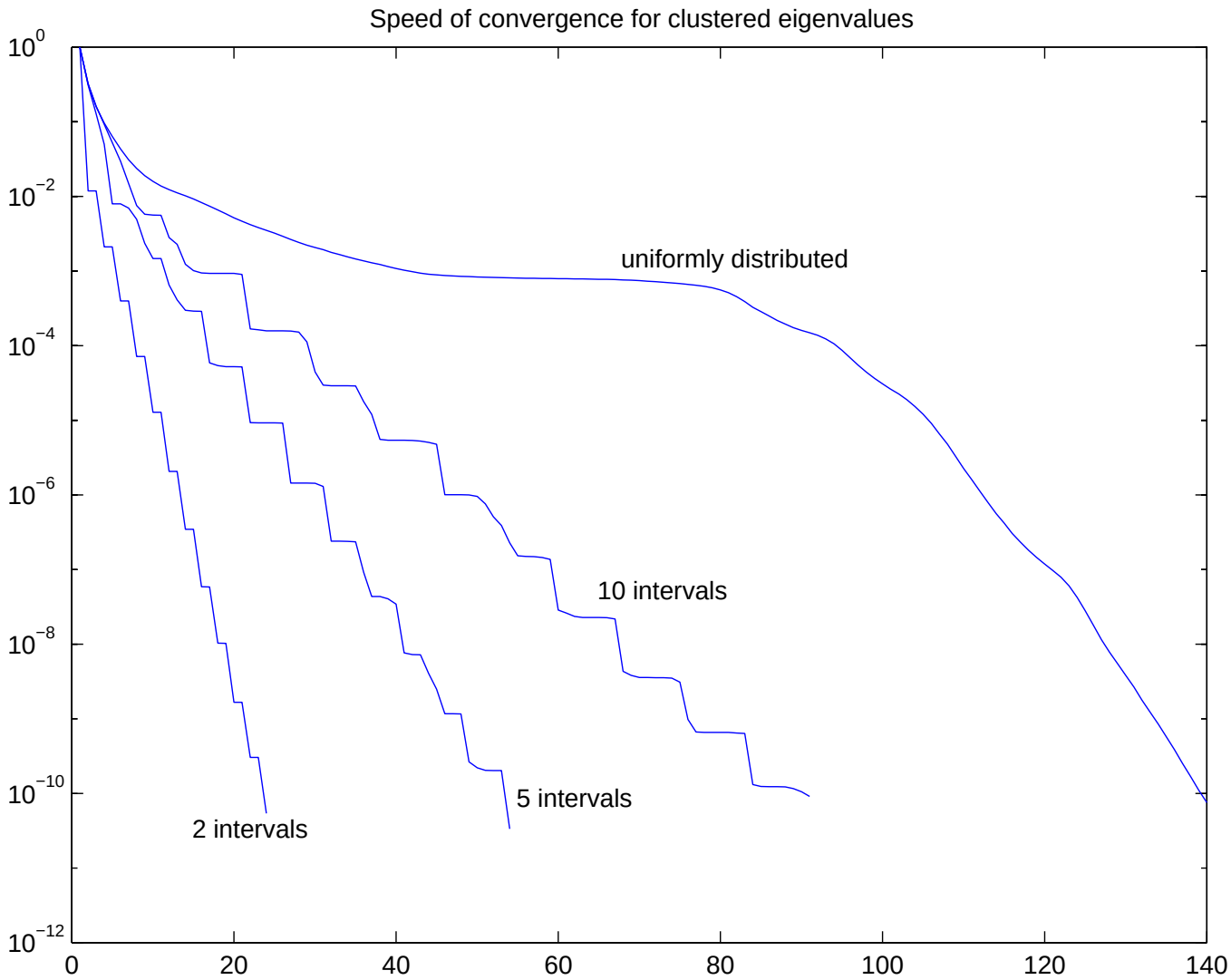


# Clustered eigenvalues ct.

To demonstrate the effect of clustered eigenvalues on the convergence properties of the CG method, we consider a homogeneous linear system $Ax = 0$, where $A$ is a diagonal matrix of dimension $n = 500$ with condition number $\kappa_2(A) = 10000$.

The following figure contains the reduction of the energy norms of the errors for the cases that all eigenvalues are uniformly distributed in 2, 5 and 10 intervals of length 1, respectively, and for the case that the eigenvalues are uniformly distributed in the interval $[1, 10000]$ in a semilogarithmic scale.

# Clustered eigenvalues ct.



# Outlying eigenvalues

Knowing only the largest and the smallest eigenvalues $\lambda_n$ and $\lambda_1$ of a positive definite matrix $A$, the bound in Theorem 4.5 is the best possible. If the interior eigenvalues of $A$ lie at the points where the shifted Chebyshev polynomial $C_k$ attains its maximum absolute value in $[\lambda_1, \lambda_n]$, then for a certain initial error $e^0$, the CG polynomial will be equal to the Chebyshev polynomial, and the bound in Theorem 4.5 will be actually be attained at step $k$.

If additional information is available about interior eigenvalues of $A$, one can often improve the simpler estimate of Th. 4.5 while maintaining a simple expression.

Suppose, for example that $A$ has one eigenvalue $\lambda_n$ which is much larger than the others. Then we replace the polynomial in the minmax characterization of the error by the product of a linear factor which is zero at $\lambda_n$, and the shifted and scaled Chebyshev polynomial on the interval $[\lambda_1, \lambda_{n-1}]$ of degree $k-1$:

# Outlying eigenvalues ct.

$$p_k(\lambda) = \left[ C_{k-1}\left(\frac{2\lambda - \lambda_{n-1} - \lambda_1}{\lambda_{n-1} - \lambda_1}\right) \Big/ C_{k-1}\left(\frac{-\lambda_{n-1} - \lambda_1}{\lambda_{n-1} - \lambda_1}\right)\right] \left(\frac{\lambda_n - \lambda}{\lambda_n}\right).$$

Since the second factor is zero at $\lambda_n$ and less than one in absolute value on each of the other eigenvalues, the maximum absolute value of this polynomial on $\{\lambda_1, \ldots, \lambda_n\}$ is less than the maximum absolute value of the first factor on $\{\lambda_1, \ldots, \lambda_{n-1}\}$.

Using arguments like those in Chapter 3 it follows that

$$\|x^k - x^*\|_A \le 2\left(\frac{\sqrt{\kappa_{n-1}} - 1}{\sqrt{\kappa_{n-1}} + 1}\right)^{k-1} \|x^0 - x^*\|_A, \qquad \kappa_{n-1} = \frac{\lambda_{n-1}}{\lambda_1}$$

# Outlying eigenvalues ct.

Similarly, if $A$ has a few outlying eigenvalues:

$$\lambda_1 \le \lambda_2 \le \cdots \le \lambda_{n-\ell} \ll \lambda_{n-\ell+1} \le \cdots \le \lambda_n, \quad \lambda_{n-\ell}/\lambda_{n-\ell+1} \ll 1,$$

one can consider a polynomial $p_k$ which is the product of a polynomial of degree $\ell$ that is zero at each of the outlyers and less than one in magnitude at each of the other eigenvalues, and a shifted and scaled Chebyshev polynomial of degree $k-\ell$ on the interval $[\lambda_1, \lambda_{n-\ell}]$.

$$p_k(\lambda) = \left[ C_{k-\ell}\left(\frac{2\lambda - \lambda_{n-\ell} - \lambda_1}{\lambda_{n-\ell} - \lambda_1}\right) \Big/ C_{k-1}\left(\frac{-\lambda_{n-\ell} - \lambda_1}{\lambda_{n-\ell} - \lambda_1}\right) \right] \prod_{j=n-\ell+1}^{n} \left(\frac{\lambda_j - \lambda}{\lambda_j}\right).$$

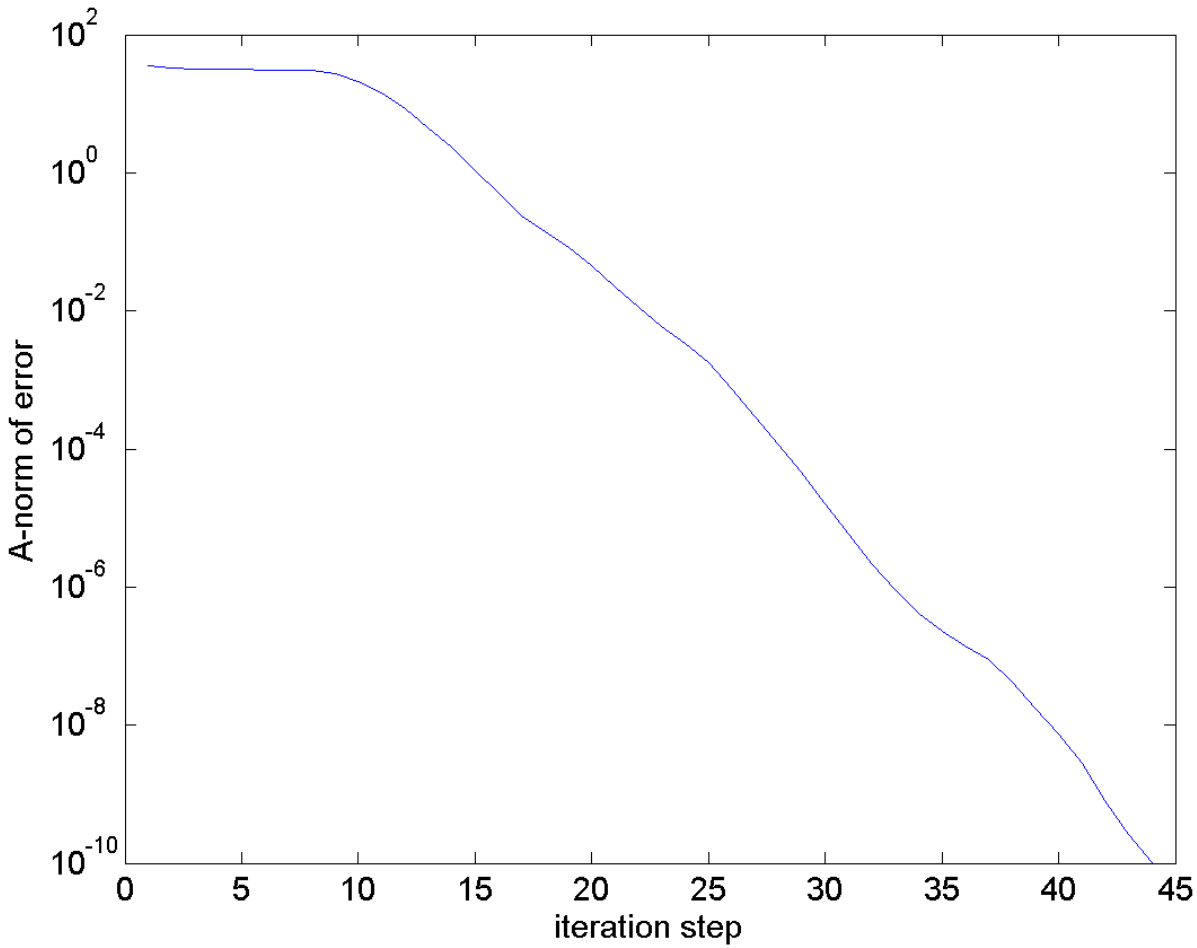
With this polynomial one obtains from the minmax characterization of the error the upper bound

$$\|x^k - x^*\|_A \le 2\left(\frac{\sqrt{\kappa_{n-\ell}} - 1}{\sqrt{\kappa_{n-\ell}} + 1}\right)^{k-\ell} \|x^0 - x^*\|_A, \qquad \kappa_{n-\ell} = \frac{\lambda_{n-\ell}}{\lambda_1}$$

# Local effects in convergence behavior

The following figure shows the convergence behavior of the CG method for

$A = \text{diag}([0.001; 0.1 + rand(98, 1); 1.2]);\ b = ones(100, 0);\ x0 = zeros(100, 1).$



# Local effects in convergence behavior

In the beginning the convergence is quite slow in agreement with the condition number $\kappa(A) = 1200$.

After iteration 10 the convergence gets noticeably faster than in the first phase.

To explain this effect we need a different variant of the CG method, namely its connection to the Lanczos process (Lanczos 1950).

# Lanczos method

The Lanczos algorithm determines an orthonormal basis $\{q^1, \ldots, q^k\}$ of the Krylov space

$$\mathcal{K}_k(r^0, A) := \text{span}\{r^0, Ar^0, \ldots, A^{k-1}r^0\}, \; k = 1, \ldots, n,$$

such that

$$T_k := Q_k^T A Q_k, \quad Q_k := (q^1, \ldots, q^k)$$

is tridiagonal.

The vectors $q^k$ can be obtained by a three term recurrence.

# Lanczos method ct.

Assume that we already computed the orthonormal basis $q^1, \ldots, q^k$ of the Krylov space $\mathcal{K}_k(v, A)$.

Then $A^{k-1}v \in \operatorname{span}\{q^1, \ldots, q^k\}$, and therefore, there exist $\gamma_1, \ldots, \gamma_k \in \mathbb{R}$ such that

$$A^k v = A(A^{k-1}v) = A\Big(\sum_{j=1}^{k} \gamma_k q^j\Big) = \gamma_k A q^k + A\Big(\sum_{j=1}^{k-1} \gamma_j q^j\Big).$$

The second term on the right hand side is contained in $\mathcal{K}_k(v, A)$. Hence, to obtain an orthonormal basis of $\mathcal{K}_{k+1}(v, A)$ it suffices to compute the orthonormal complement of the $u^k := Aq^k$ with respect to the vectors $q^1, \ldots, q^k$.

# Lanczos method ct.

Since $Aq^j \in \mathcal{K}_{j+1}(v, A) \subset \mathcal{K}_{k-1}(v, A)$ for every $j < k-1$, we have

$$(q^j)^T u^k = (q^j)^T A q^k = (Aq^j)^T q^k = 0.$$

Hence, $(u^k)^T q^j = 0$ for $j = 1, \ldots, k-2$, and therefore

$$Aq^k = \gamma_k q^{k+1} + \alpha_k q^k + \beta_{k-1} q^{k-1}.$$

# Lanczos method ct.

The coefficients are obtained from

$$\begin{aligned}\beta_{k-1} &= (q^{k-1})^T A q^k = (Aq^{k-1})^T q^k \\ &= (\gamma_{k-1} q^k + \alpha_{k-1} q^{k-1} + \beta_{k-2} q^{k-2})^T q^k = \gamma_{k-1},\end{aligned}$$

i.e.

$$Aq^k = \beta_k q^{k+1} + \alpha_k q^k + \beta_{k-1} q^{k-1}. \qquad (1)$$

Thus,

$$\alpha_k = (q^k)^T A q^k,$$

and the condition $\|q^{k+1}\|_2 = 1$ yields

$$\beta_k = 1/\|Aq^k - \alpha_k q^k - \beta_{k-1} q^{k-1}\|_2. \qquad (2)$$

If the denominator in (2) vanishes, then $Aq^k \in \mathcal{K}_k(v, A)$, and therefore, $\mathcal{K}_k(v, A)$ is a $k$-dimensional invariant subspace of $A$.

# Lanczos method ct.

1: $q^0 = 0$; $k = 1$
2: $\beta_0 = \|r^0\|$
3: **while** $r^{k-1} \neq 0$ **do**
4: $\quad q^k = r^{k-1}/\beta_{k-1}$
5: $\quad r^k = Aq^k$
6: $\quad r^k = r^k - \beta_{k-1} q^{k-1}$
7: $\quad \alpha_k = (q^k)^T r^k$
8: $\quad r^k = r^k - \alpha_k q^k$
9: $\quad \beta_k = \|r^k\|$
10: **end while**

Then with $T_k = \text{tridiag}\{\beta_{j-1}, \alpha_j, \beta_j\}$

$$AQ_k = Q_k T_k + r^k (e^k)^T, \quad r^k = Aq^k - \alpha_k q^k - \beta_{k-1} q^{k-1}.$$

and $r^k = \|r^k\|_2 q^{k+1} = \beta_k q^{k+1}$

# Lanczos method ct.

Consider the CG method for the linear system

$$Ax = b, \quad A \text{ symmetric and positive definite.}$$

Let $x^0 = 0$ (else consider $Ay = b - Ax^0 =: \tilde{b}$)

The approximation $x^k$ after $k$ steps minimizes

$$\phi(x) := \frac{1}{2} x^T A x - b^T x \quad \text{in } \mathcal{K}_k(r^0, A) = \mathcal{K}_k(b, A).$$

# Lanczos method ct.

Restricting $\phi$ to $\mathcal{K}_k(b, A)$ yields

$$\phi_k(y) := \frac{1}{2} y^T Q_k^T A Q_k y - y^T Q_k^T b, \quad y \in \mathbb{R}^k,$$

and the CG iterate $x^k$ satisfies

$$T_k y^k = Q_k^T b, \quad x^k = Q_k y^k.$$

Disadvantage (at a first glance): All $q^j$, $j = 1, \ldots, k$, are needed to determine $x^k$. We will come back to this point in the next section.

# Local convergence behavior

The following analysis of the local convergence behavior of the CG method is contained in a paper of van der Sluis & van der Vorst (1986)

Multiplying

$$AQ_k = Q_k T_k + \beta_k q^{k+1} (e^k)^T$$

by $e^1$ from the right yields

$$Aq^1 = Q_k T_k e^1.$$

Premultiplying this with $A$ from the left gives

$$A^2 q^1 = AQ_k T_k e^1 = (Q_k T_k + \beta_k q^{k+1} (e^k)^T) T_k e^1 = Q_k T_k^2 e^1 \text{ for k>2.}$$

And similarly one obtains

$$A^j q^1 = Q_k T_k^j e^1, \text{ for } j = 3, \ldots, k-1, \; A^k q^1 = Q_k T_k^k e^1 + c q^{k+1}$$

for some constant $c$

# Local convergence behavior ct.

Since $x^k \in \mathcal{K}_k(b, A)$ it follows that there exists some polynomial $p_k(t) = \sum_{j=0}^{k} \gamma_j t^j$ of degree $k$ such that

$$r^k = p_k(A)b,$$

and from $r^k \perp \mathcal{K}_k(b, A)$ and $q^1 = b$ it follows

$$\begin{aligned} 0 &= Q_k^T p_k(A) q^1 = Q_k^T \sum_{j=0}^{k} \gamma_j A^j q^1 \\ &= \sum_{j=0}^{k} \gamma_j Q_k^T Q_k T_k^j e^1 + \gamma_k c Q_k^T q^{k+1} \\ &= p_k(T_k) e^1 \end{aligned}$$

# Local convergence behavior ct.

Since $T_k$ is symmetric there exists an orthonormal basis of $\mathbb{R}^k$ of eigenvectors $y^j$, $j = 1, \ldots, k$ of $T_k$. Denote by $\theta_j$ the eigenvalue corresponding to $y^j$, and assume that the eigenvalues are ordered by magnitude $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_k$.

Let $e^1 = \sum_{j=1}^{k} \tau_j y^j$. Then $\tau_j \neq 0$ for $j = 1, \ldots, k$:

Assume that $\tau_j = 0$, i.e. $(y^j)^T e^1 = 0$. Then it follows from $T_k y^j = \theta_j y^j$

$$0 = (T_k y^j)^T e^1 = (y^j)^T (T_k e^1) = (y^j)^T (\alpha_1 e^1 + \beta_1 e^2) \;\Rightarrow\; (y^j)^T e^2 = 0,$$

and by induction we obtain the contradiction $(y^j)^T e^i = 0$, $i = 1, \ldots, k$.

# Local convergence behavior ct.

Since $\tau_j \neq 0$ for every $j$ we obtain from

$$0 = p_k(T_k)e^1 = \sum_{j=1}^{k} \tau_j p(T_k)y^j = \sum_{j=1}^{k} \tau_j p(\theta_j)y^j$$

that the eigenvalues of $\theta_j$ are the roots of the polynomial $p_k$, and from $p_k(0) = 1$ we therefore obtain

$$p_k(t) = \prod_{j=1}^{k}(\theta_j - t) \Big/ \prod_{j=1}^{k} \theta_j \qquad (1)$$

Exploiting this characterization of the residual polynomial $p_k$ we can further analyze the convergence behavior of the CG method.

# Local convergence behavior ct.

Let $z^j$, $j = 1, \ldots, n$ be an orthonormal set of eigenvectors of $A$, and let $\lambda_j$ be the eigenvalue corresponding to $z^j$, where the eigenvalues again are ordered by magnitude.

Write the initial error as

$$x - x^0 = \sum_{j=1}^{n} \mu_j z^j.$$

From

$$A(x - x^k) = b - Ax^k = r^k = p_k(A)r^0 = p_k(A)A(x - x^0) = Ap_k(A)(x - x^0)$$

it follows

$$x - x^k = p_k(A)(x - x^0) = \sum_{j=1}^{n} \mu_j p_k(A) z^j = \sum_{j=1}^{n} \mu_j p_k(\lambda_j) z^j.$$

# Local convergence behavior ct.

We now replace the first component of $x^k$ (with respect to the basis $z^j$) by the first component of $x$ and take the modified $x^k$ as starting vector $\tilde{x}^0$ of another CG process (with the same $A$ and $b$):

$$x - \tilde{x}^0 = \sum_{j=2}^{n} \mu_j p_k(\lambda_j) z^j.$$

The new CG process generates a sequence $\tilde{x}^m$ for which the error is characterized by a polynomial $\tilde{p}_m$.

# Local convergence behavior ct.

Let

$$q_k(t) = \frac{(\lambda_1 - t)(\theta_2 - t)\dots(\theta_k - t)}{\lambda_1\theta_2\dots\theta_k} = \frac{\theta_1(\lambda_1 - t)}{\lambda_1(\theta_1 - t)}p_k(t).$$

From $\tilde{p}_m(0)q_k(0) = 1$ it follows

$$\|x - x^{k+m}\|_A^2 \le \|\tilde{p}_m(A)q_k(A)(x - x^0)\|_A^2 = \sum_{j=2}^{n} \lambda_j \tilde{p}_m(\lambda_j)^2 q_k(\lambda_j)^2 \mu_j^2.$$

Defining

$$F_k := \frac{\theta_1}{\lambda_1} \max_{j=2,\dots,n} \left|\frac{\lambda_j - \lambda_1}{\lambda_j - \theta_1}\right|$$

it follows that

$$|q_k(\lambda_j)| \le F_k |p_k(\lambda_j)|$$

# Local convergence behavior ct.

Hence, the upper bound can be further simplified as

$$\begin{aligned}\|x - x^{k+m}\|_A^2 &\leq F_k^2 \sum_{j=2}^{n} \lambda_j \tilde{p}_m(\lambda_j)^2 p_k(\lambda_j)^2 \mu_j^2 \\ &= F_k^2 \|x - \tilde{x}^m\|_A^2 \leq F_k^2 \frac{\|x - \tilde{x}^m\|_A^2}{\|x - \tilde{x}^0\|_A^2} \|x - x_k\|_A^2.\end{aligned}$$

If the smallest eigenvalue $\theta_1$ is close to $\lambda_1$, then $F_k \approx 1$, and we may expect a reduction of the error in the next steps that is bounded by the reduction that we obtain for a CG process for $Ax = b$ in which $\lambda_1$ is missing.

With $\kappa_2(A) := \lambda_n/\lambda_2$ we have

$$\|x - x^{m+k}\|_A \leq 2 \left( \frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1} \right)^m \|x - x^k\|_A.$$

We stress the fact that this acceleration of convergence only holds in exact arithmetic. In finite precission arithmetic the behavior is more complicated (cf. exercises).

# Preconditioning

The estimate

$$\|x^k - x^*\|_A \le 2\left(\frac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1}\right)^k \|x^0 - x^*\|_A \qquad (1)$$

suggests that the convergence of the CG method depends heavily on the condition number of the system matrix $A$.

We therefore transform problem $Ax = b$ into a system

$$\tilde{A}\tilde{x} = \tilde{b}, \qquad (2)$$

where

$$\tilde{A} := C^{-1}AC^{-T}, \quad \tilde{b} := C^{-1}b, \quad x := C^{-T}\tilde{x}, \qquad (3)$$

and $C \in \mathbb{R}^{n\times n}$ is a nonsingular matrix, such that linear systems of equations

$$Cy = d \quad \text{and} \quad C^T y = d \quad (d \in \mathbb{R}^n \text{ given})$$

can be solved easily and that the condition number of $\tilde{A}$ is as small as possible.

Hence, $\tilde{A}$ should be close to $I$, and therefore, $CC^T \approx A$.

# CG for $\tilde{A}\tilde{x} = \tilde{b}$

1: $\tilde{r} = \tilde{b} - \tilde{A}\tilde{x}$
2: $\tilde{d} = \tilde{r}$
3: $\alpha = (\tilde{r})^T \tilde{r}$
4: **while** $\sqrt{\alpha} > \varepsilon$ **do**
5: $\quad \tilde{s} = \tilde{A}\tilde{d}$
6: $\quad \tau = \alpha \Big/ (\tilde{d})^T \tilde{s}$
7: $\quad \tilde{x} = \tilde{x} + \tau \tilde{d}$
8: $\quad \tilde{r} = \tilde{r} - \tau \tilde{s}$
9: $\quad \beta = 1/\alpha$
10: $\quad \alpha = (\tilde{r})^T \tilde{r}$
11: $\quad \beta = \beta \cdot \alpha$
12: $\quad \tilde{d} = \tilde{r} + \beta \tilde{d}$
13: **end while**

# CG for $\tilde{A}\tilde{x} = \tilde{b}$ original variables; $\tilde{d} := C^T d$; $\tilde{s} = C^{-1} s$

1: $C^{-1}r = C^{-1}b - C^{-1}AC^{-T}\tilde{x} = C^{-1}(b - Ax)$
2: $C^T d = C^{-1} r$
3: $\alpha = r^T C^{-T} C^{-1} r$
4: **while** $\sqrt{\alpha} > \varepsilon$ **do**
5: $\quad C^{-1}s = C^{-1}AC^{-T}C^T d = C^{-1}Ad$
6: $\quad \tau = \alpha \Big/ \Big((d^T C)(C^{-1}s)\Big) = \alpha / d^T s$
7: $\quad C^T x = C^T x + \tau C^T d$
8: $\quad C^{-1}r = C^{-1}r - \tau C^{-1}s$
9: $\quad \beta = 1/\alpha$
10: $\quad \alpha = r^T C^{-T} C^{-1} r$
11: $\quad \beta = \beta \cdot \alpha$
12: $\quad C^T d = C^{-1} r + \beta C^T d$
13: **end while**

# Preconditioned CG. $M = CC^T$

1: $r = b - Ax$
2: Solve $Md = r$ for $d$
3: $\alpha = r^T d$
4: **while** $\sqrt{\alpha} > \varepsilon$ **do**
5: $\quad s = Ad$
6: $\quad \tau = \alpha / d^T s$
7: $\quad x = x + \tau d$
8: $\quad r = r - \tau s$
9: $\quad \beta = 1/\alpha$
10: $\quad$ Solve $Mz = r$ for $z$
11: $\quad \alpha = r^T z$
12: $\quad \beta = \beta \cdot \alpha$
13: $\quad d = z + \beta d$
14: **end while**

# PCG

The preconditioned conjugate gradient algorithm, PCG algorithm for short, requires as the original CG method one matrix-vector product and five level-1-operations, and additionally the solution of one system of linear equations $Mz = r$.

Notice that the matrix $C$, which we used in the derivation of the PCG method, does not appear in the algorithm, but only the preconditioner $M$.

$M = CC^T$ obviously is symmetric and, by the regularity of $C$, positive definite, and every symmetric and positive definite matrix principally is a suitable preconditioner (we can choose $C := M^{1/2}$ in the derivation of the PCG method).

# Preconditioner

In practical problems, $M$ must have the following properties:

- Linear systems $Mz = r$ can be solved easily
- $M$ is a good approximation of $A$.

The choice of the preconditioner can effect the speed of the convergence dramatically. Preconditioning is a field of active research.

The most important preconditioner is obtained by incomplete Cholesky factorization.

# Preconditioning by Splitting Methods

To obtain a fast preconditioned CG method we have to find a symmetric and positive definite matrix $M$ such that firstly, $M$ approximates $A$ as good as possible, and secondly, the linear system of equations $Mz = r$ can be solved easily.

When we considered splitting methods, we studied nearly the same question. All splitting matrices constructed in Chapter 2, which inherit the symmetry and positive definiteness from the system matrix $A$ are suitable preconditioners.

# Preconditioning by Splitting ct.

If $A := D - E - E^T$ is the standard partition, where $D$ denotes the diagonal part of $A$ and $-E$ the strictly lower triangular part, then the following preconditioners are at hand:

| | |
|---|---|
| $M = D$ | Jacobi |
| $M := (D-E)D^{-1}(D-E^T)$ | symmetric GS |
| $M := \frac{1}{\omega(2-\omega)}(D-\omega E)D^{-1}(D-\omega E^T),\ \omega \in (0,2),$ | SSOR. |

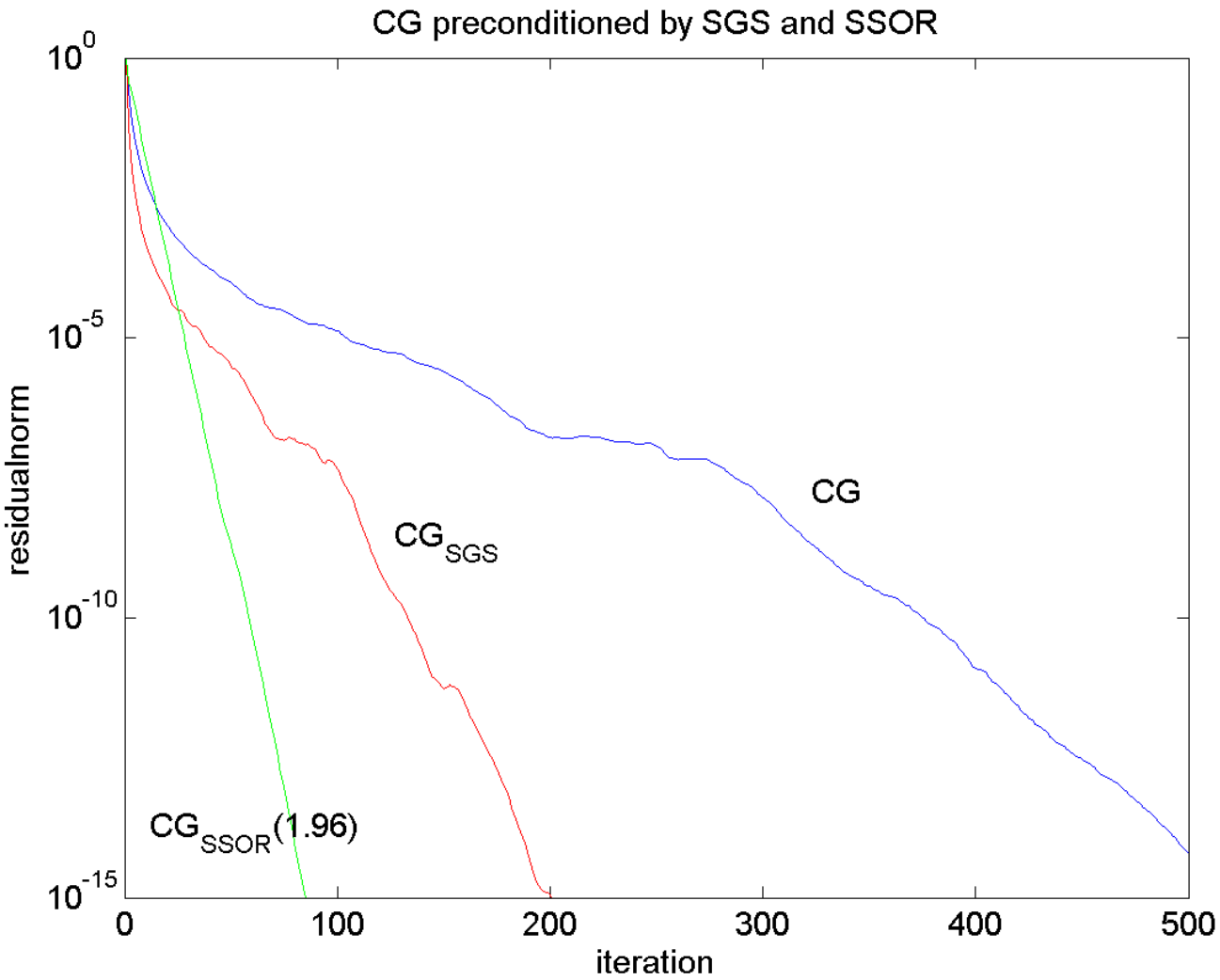
Of course, these matrices are symmetric and positive definite, if the system matrix $A$ has these properties.

The solution of $Mz = r$ for $M := B$ can be obtained by one step of the corresponding splitting iteration

$$y^{k+1} = B^{-1}(B-A)y^k + B^{-1}r, \qquad (1)$$

if one chooses as initial guess $y^0 = 0$.

# Example


CG preconditioned by SGS and SSOR
residualnorm
iteration
$10^{0}$
$10^{-5}$
$10^{-10}$
$10^{-15}$
0
100
200
300
400
500
CG
$CG_{SGS}$
$CG_{SSOR}(1.96)$

# Preconditioning by Splitting ct.

It is obvious that more than one step of the splitting method can be performed to get an improved preconditioning.

If in an inner iteration $k$ steps of (1) are in use, then the preconditioner is given by

$$M = \sum_{j=0}^{k-1} \left[ B^{-1}(B-A) \right]^j B^{-1}. \qquad (2)$$

Since $A$ and $B$ are symmetric, each term in (2) is symmetric. Thus $M$ is symmetric, too. Concerning the positive definiteness of $M$ the following theorem was proved by Adams (1985).

# Theorem 4.7

Let $A$ and $B$ be a symmetric matrices, and suppose that $A$ is positive definite and $B$ is nonsingular. Then the matrix $M$ given in (2) is symmetric.

(i) For $k$ odd, $M$ is positive definite if and only if $B$ is positive definite.

(ii) For $k$ even, $M$ is positive definite if and only if $2B - A$ is positive definite.

We next discuss the condition of Theorem 4.7 which seems to be quite strange.

# Theorem 4.8

Suppose that $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$ are symmetric and positive definite.

Then the matrix $2B - A$ is positive definite if and only if $\rho(B^{-1}(B - A)) < 1$.

Proof: Obviously, $\lambda$ is an eigenvalue of $B^{-1}(B - A)$ if and only if it is an eigenvalue of the generalized eigenproblem

$$(B - A)x = \lambda Bx.$$

# Proof ct.

Since $B$ is positive definite all eigenvalues of $B^{-1}(B-A)$ are real. From Rayleigh's principle we obtain that $\rho(B^{-1}(B-A)) < 1$ if and only if

$$-1 < \frac{x^T(B-A)x}{x^TBx} < 1 \quad \text{for every } x \neq 0,$$

which is equivalent to

$$-x^TBx < x^T(B-A)x < x^TBx \quad \text{for every } x \neq 0,$$

i.e.

$$x^T(2B-A)x > 0 \quad \text{and} \quad x^TAx > 0 \quad \text{for every } x \neq 0.$$

Since $A$ is supposed to be positive definite $\rho(B^{-1}(B-A)) < 1$ if and only if $2B-A$ is positive definite. □

# Remark

Adams (1985) studied $k$-step preconditioners.

For SSOR preconditioners she proved that the condition number is a nonincreasing function of $k$.

She reports however, that the actual decrease in the number of iterations is not enough to balance the extra work involved in the preconditioner, at least if the dimension of the problem is not too large.

# Incomplete Cholesky Preconditioner

## Cholesky Factorization

1: **for** $i = 1 : n$ **do**
2: **for** $j = 1 : i - 1$ **do**
3: $c_{ij} = \left( a_{ij} - \sum_{k=1}^{j-1} c_{ik} c_{jk} \right) / c_{jj}$
4: **end for**
5: $c_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} c_{ik}^2}$
6: **end for**

If the Cholesky factor $C$ of $A$ is known, then the linear system of equations $Ax = b$ can be solved directly:

$$\text{Solve } Cy = b, \quad \text{solve } C^T x = y$$

where both systems are triangular.

# Incomplete Cholesky ct.

Let

$$m(i) := \min\{j \,:\, 1 \le j \le i,\ a_{ij} \neq 0\}, \quad i = 1, \ldots, n,$$

i.e. $a_{i,m(i)}$ is the first nonzero entry in the $i$-th row of $A$.

Then, the set of index pairs

$$S(A) := \{(i,j) \,:\, m(i) \le j \le i,\ 1 \le i \le n\}$$

is called the envelope of the matrix $A$.

From the algorithm above it follows immediately, that all elements $c_{ij}$ such that $(i,j) \notin S(A)$ vanish. Hence, if $A$ is a band matrix, then $C$ has the same band structure.

# Incomplete Cholesky ct.

Matrices that arise from discretizations of boundary value problems of partial differential equations usually possess an envelope that is very large but only sparsely populated by nonzero elements. This sparsity pattern unfortunately is not inherited by the Cholesky factor, but most of the envelope is filled in during the elimination process.

To reduce the storage requirements one chooses a subset $J \subset S(A)$ of the envelope of $A$, which always contains the diagonal index pairs $(1,1),\ldots,(n,n)$, and modifies the algorithm above in the following way: All entries $c_{ij}$ such that $(i,j) \notin J$ are set to zero.

The algorithm then obtains the following form:

# Incomplete Cholesky Factorization

1: **for** $i = 1 : n$ **do**
2: **for** $j = 1 : i - 1$ **do**
3: **if** $(i, j) \notin J$ **then**
4: $c_{ij} = 0$
5: **else**
6: $c_{ij} = \left( a_{ij} - \sum_{k=1}^{j-1} c_{ik} c_{jk} \right) / c_{jj}$;
7: **end if**
8: **end for**
9: $c_{ii} = \sqrt{a_{ii} - \sum_{j=1}^{i-1} c_{ij}^2}$
10: **end for**

# Incomplete Cholesky Factorization ct.

If $J = S(A)$, then $C$ is the Cholesky factor and $A = CC^T$.

If $J \subset S(A)$, $J \neq S(A)$, then the factorization is called incomplete Cholesky decomposition, IC decomposition for short.

A reduction of $J$ usually decreases the storage requirements, the factorization time and the time needed to solve the linear systems $C^T y = r$ and $Cz = y$ in the CG step. It also increases the defect $A - CC^T$ which reduces the rate of convergence of the preconditioned CG method.

These conflicting influences make it difficult to determine the optimum choice of $J$.

# No fill principle

The most common choice, called no-fill principle, is

$$J = J(A) := \{(i,j) \,:\, 1 \le j \le i \le n,\ a_{ij} \neq 0\}.$$

For diagonally sparse matrices, where the nonvanishing entries appear on parallels of the main diagonal (e.g. finite difference approximations of boundary value problems on rectangles are of this type) Meijerink & van der Vorst (1977) proposed to allow the incomplete factor $C$ to have a few more nonzero diagonals than the matrix $A$ itself (counting only the number of diagonals of $A$ on one side of the main diagonal, of course).

If $m$ additional diagonals are considered one obtains the $ICCG(m)$ method.

# IC decomposition

One disadvantage of the IC factorization is, that there exist symmetric and positive definite matrices $A$ such that the IC decomposition does not exist. The algorithm may fail because the square root of a negative number is required to compute a diagonal element $c_{ii}$.

The following theorem, that we state without proof, contains a class of matrices for which the IC decomposition is guaranteed to exist.

Definition
A regular matrix $A \in \mathbb{R}^{n\times n}$ is called an $M$-matrix, if $A$ has nonpositive off-diagonal elements and if its inverse has nonnegative entries.

For $A \in \mathbb{R}^{n\times n}$ the matrix

$$\langle A\rangle = (\langle a_{ij}\rangle) \quad \text{where } \langle a_{ij}\rangle = \begin{cases} -|a_{ij}| & \text{for} \quad i \neq j \\ |a_{ii}| & \text{for} \quad I = j \end{cases}$$

is called Ostrowski's comparison matrix (Ostrowski 1937).
$A \in \mathbb{R}^{n\times n}$ is called $H$-matrix, if $\langle A\rangle$ is an $M$-matrix.

# IC factorization

**Theorem**
Let $A$ be a symmetric $H$-matrix. Then for any choice of the index set $J$, the IC factorization can be carried out.

In case that the conditions of the last Theorem are not satisfied the incomplete Cholesky decomposition may not exist. There have been several suggestions made to remedy the nonexistence.

Suppose that the computation of $c_{ii}$ requires the square root of a nonpositive number. Then, one simple idea is to replace $c_{ii}$ by an arbitrary positive number.

One strategy in use is the choice $c_{ii}^2 = \sum_{j=1}^{i-1} |c_{ij}|$ which was suggested by Kershaw (1978).

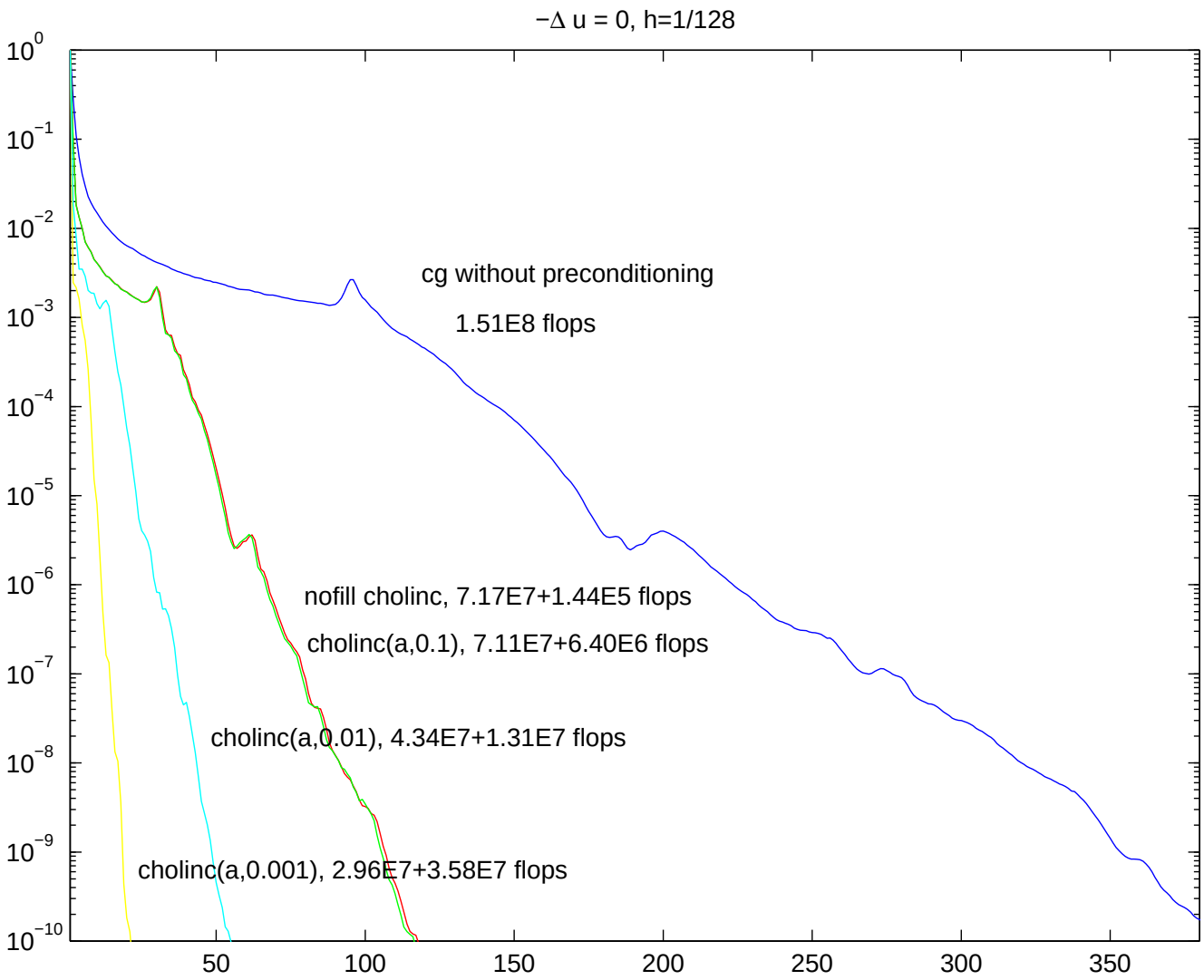
# IC factorization

A different approach which was advocated by Manteuffel (1980): It is well known that strictly diagonally dominant matrices with positive diagonal and nonpositive off-diagonal entries are *M*-matrices. Hence, for $\alpha \geq 0$ sufficiently large, the matrix $\hat{A} := A + \alpha I$ is an *H*-matrix, and by the last Theorem its IC factorization exists.

Gustafsson (1978) proposed the so called modified incomplete Cholesky decomposition, MIC decomposition for short, where the diagonal elements are increased whenever an off-diagonal element of *C* is neglected.

Munksgaard (1980) did not prescribe the set *J* of index pairs to be filled in the IC decomposition in advance, but he proposed to drop elements in the factorization if they are numerically small compared with the diagonal elements of their row and column. He combined this strategy with that of the MIC decomposition of Gustafsson.

# Example


−Δ u = 0, h=1/128
cg without preconditioning
1.51E8 flops
nofill cholinc, 7.17E7+1.44E5 flops
cholinc(a,0.1), 7.11E7+6.40E6 flops
cholinc(a,0.01), 4.34E7+1.31E7 flops
cholinc(a,0.001), 2.96E7+3.58E7 flops
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
10^−7
10^−8
10^−9
10^−10
50
100
150
200
250
300
350