ITERATIVE PROJECTION METHODS FOR SPARSE LINEAR SYSTEMS AND EIGENPROBLEMS
CHAPTER 6 : MINIMIZING METHODS

Heinrich Voss

voss@tu-harburg.de

Hamburg University of Technology
Institute of Numerical Simulation

TUHH
Technische Universität Hamburg-Harburg

CGN Methods

There are several generalizations of CG (and CR) for nonsymmetric linear problems.

The most obvious approach is to apply CG to an equivalent linear systems, which is symmetric and positive definite, e.g.

$$A^T A x = A^T b$$

or

$$AA^T y = b, \quad x = A^T y.$$

These methods are called CGN method where the letter 'N' suggests that some sort of normal equations is considered.

CGN methods ct.

For $A^TAx = A^Tb$ the k-th CG iterate x^k minimizes the functional

$$\|x - x^*\|^2_{A^TA} = (x - x^*)^T A^T A(x - x^*) = \|A(x - x^*)\|_2^2 = \|r\|_2^2$$

in the Krylov space

$$\begin{aligned} & x^0 + \mathcal{K}_k(A^Tr^0, A^TA) \\ = \quad & x^0 + \text{span}\{A^Tr^0, (A^TA)A^Tr^0, \ldots, (A^TA)^{k-1}A^Tr^0\}. \end{aligned}$$

This method is called CGNR method where the letter 'R' indicates that the residuum is minimized.

The CGNR method was suggested by Hestenes und Stiefel (1952). The matrix A^TA in general is more populated than A, however, the algorithm does not take advantage of its explicit form.

CGNR

The matrix $A^T A$ which appears in the development of the CGNR method is usually not as sparse as the original matrix A. However, it has not to be formed explicitly in the algorithm.

1: $r^0 = b - Ax^0$
2: $d^1 = A^T r^0$
3: $\alpha_0 = \|d^1\|_2^2$
4: **for** $k = 1, 2, \ldots$ until convergence **do**
5: $\quad s^k = Ad^k$
6: $\quad \tau_k = \alpha_{k-1} / \|s^k\|_2^2$
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: $\quad t^k = A^T r^k$
11: $\quad \alpha_k = \|t^k\|_2^2$
12: $\quad \beta_k = \alpha_k \beta_k$
13: $\quad d^{k+1} = t^k + \beta_k d^k$
14: **end for**

CGNR

Cost of CGNR
2 matrix-vector products
2 scalar products
3 _axpy
Storage requirements: 4 vectors

To investigate the convergence of the CGNR method we note that

$$x^k = x^0 + q_{k-1}(A^TA)A^Tr^0$$

for some polynomial q_{k-1} of degree $k-1$. Subtracting the exact solution x^* on both sides and observing $r^0 = Ae^0$ one obtains

$$e^k = p_k(A^TA)e^0, \quad p_k(z) = 1 - zq_{k-1}(z),$$

and since $Ap_k(A^TA) = p_k(AA^T)A$ and $r^k = Ae^k$, multiplication with A gives

$$r^k = p_k(AA^T)r^0, \quad p_k(0) = 1.$$

Convergence of CGNR

Replacing p_k by a shifted and scaled Chebyshev polynomial one gets similar to the convergence result for the CG method

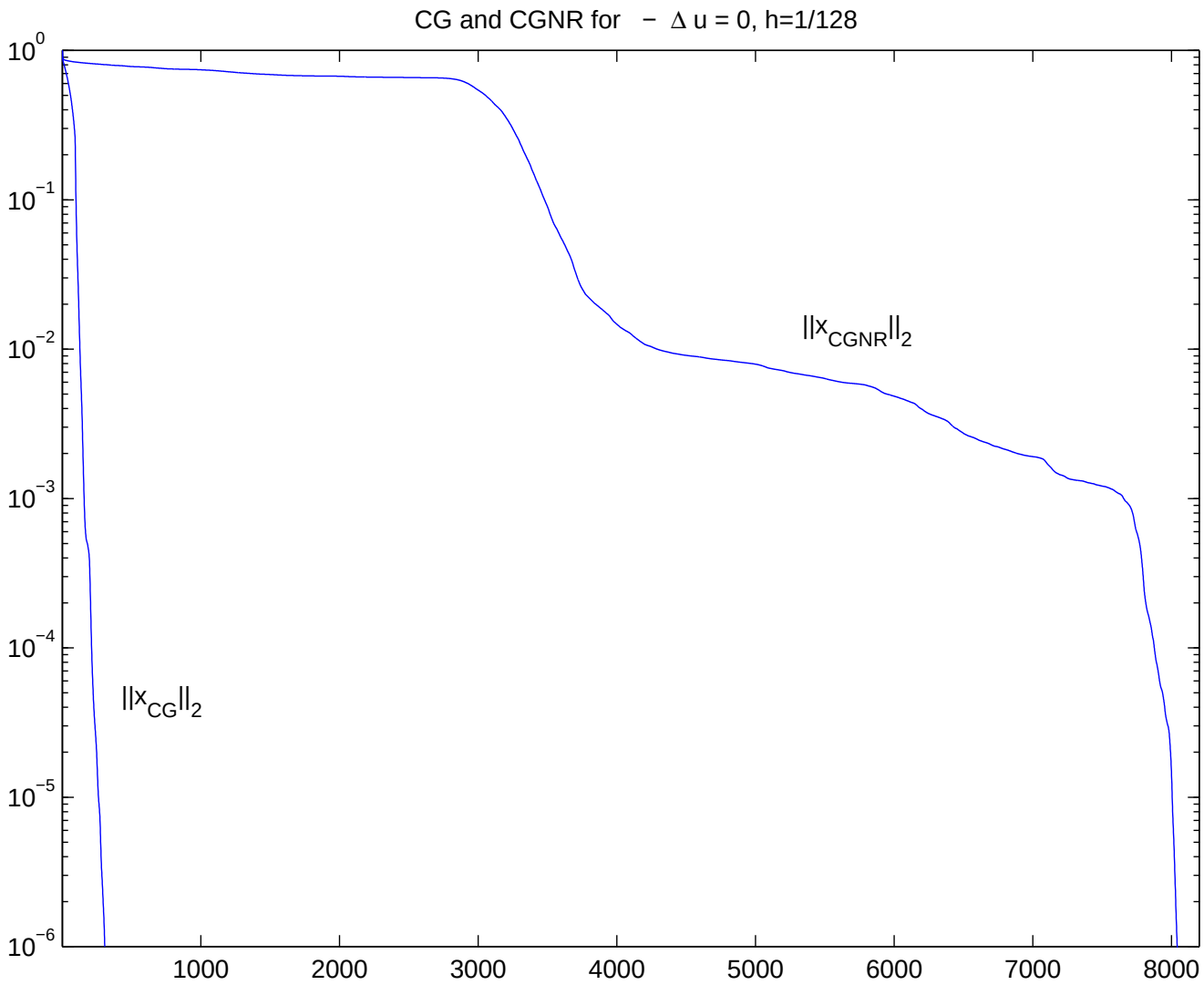
Theorem 6.1
Let A be nonsingular, σ_1 its largest and σ_n its smallest singular value, and let

$$R := \frac{\sigma_1 - \sigma_n}{\sigma_1 + \sigma_n} = \frac{\kappa_2(A) - 1}{\kappa_2(A) + 1}.$$

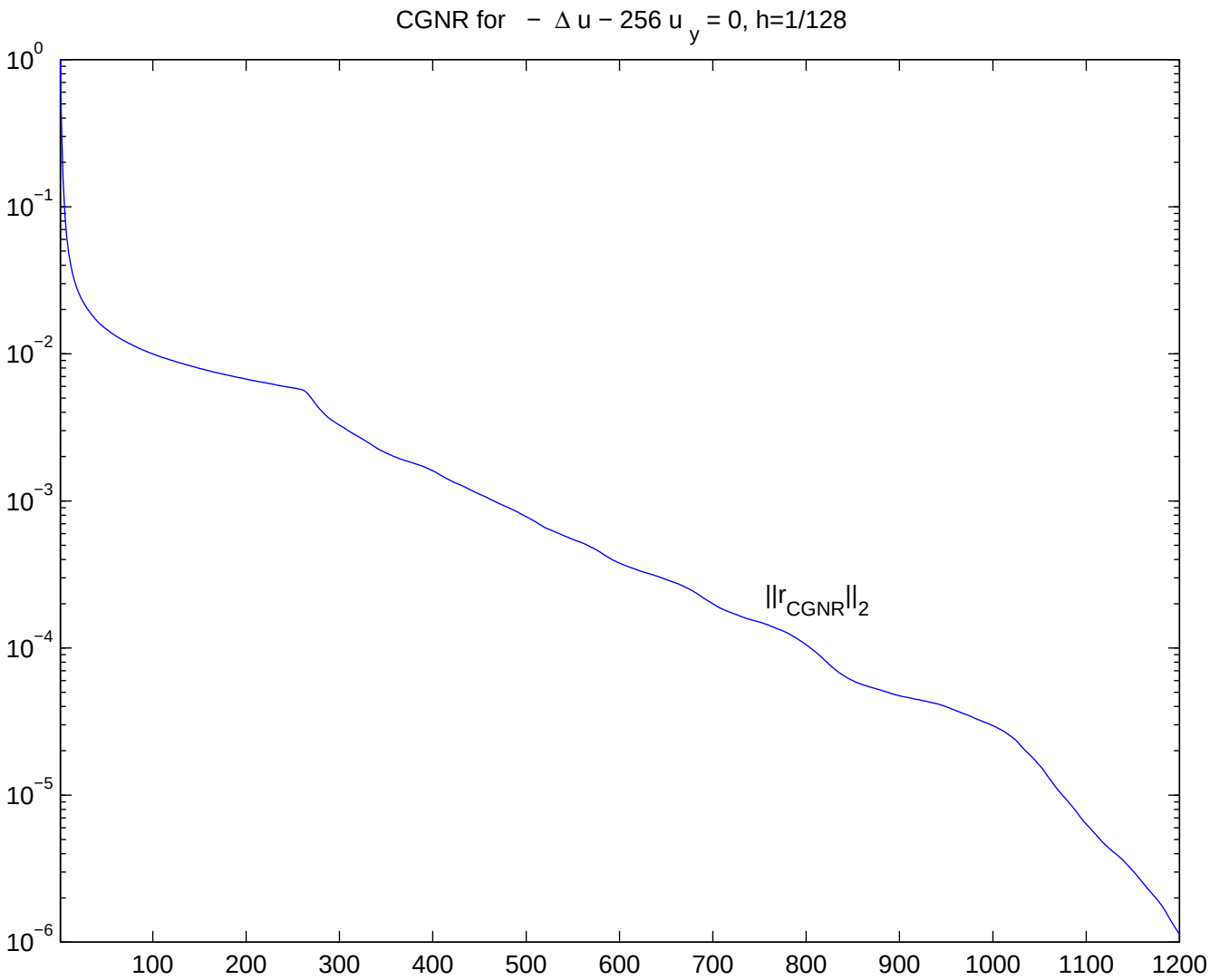
Then it holds

$$\|r^k\|_2 \leq \frac{2R^k}{1 + R^{2k}} \|r^0\|_2 \leq 2 \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1} \right)^k \|r^0\|_2.$$

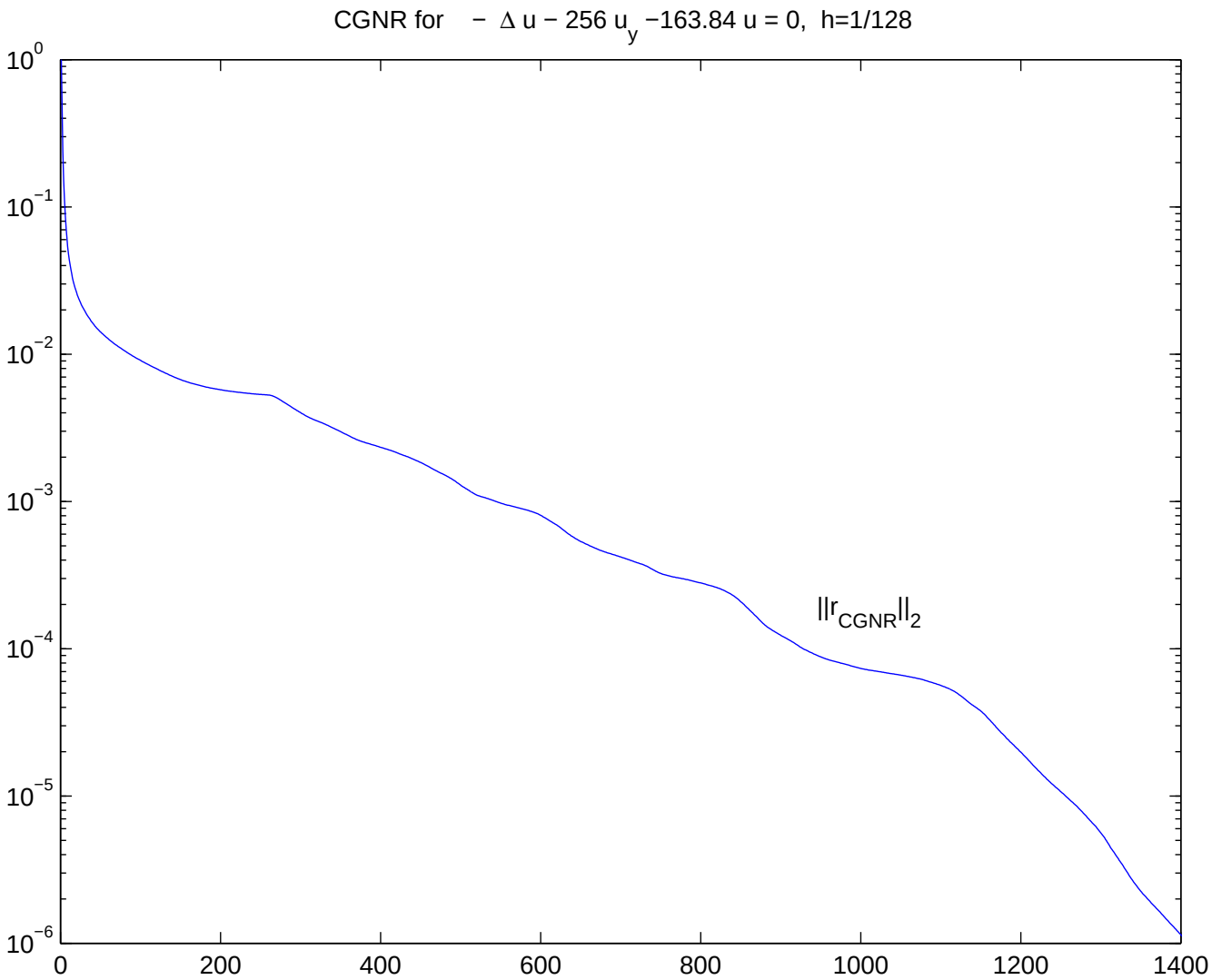
Example


CG and CGNR for − Δ u = 0, h=1/128
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
1000
2000
3000
4000
5000
6000
7000
8000
$||x_{CGNR}||_2$
$||x_{CG}||_2$

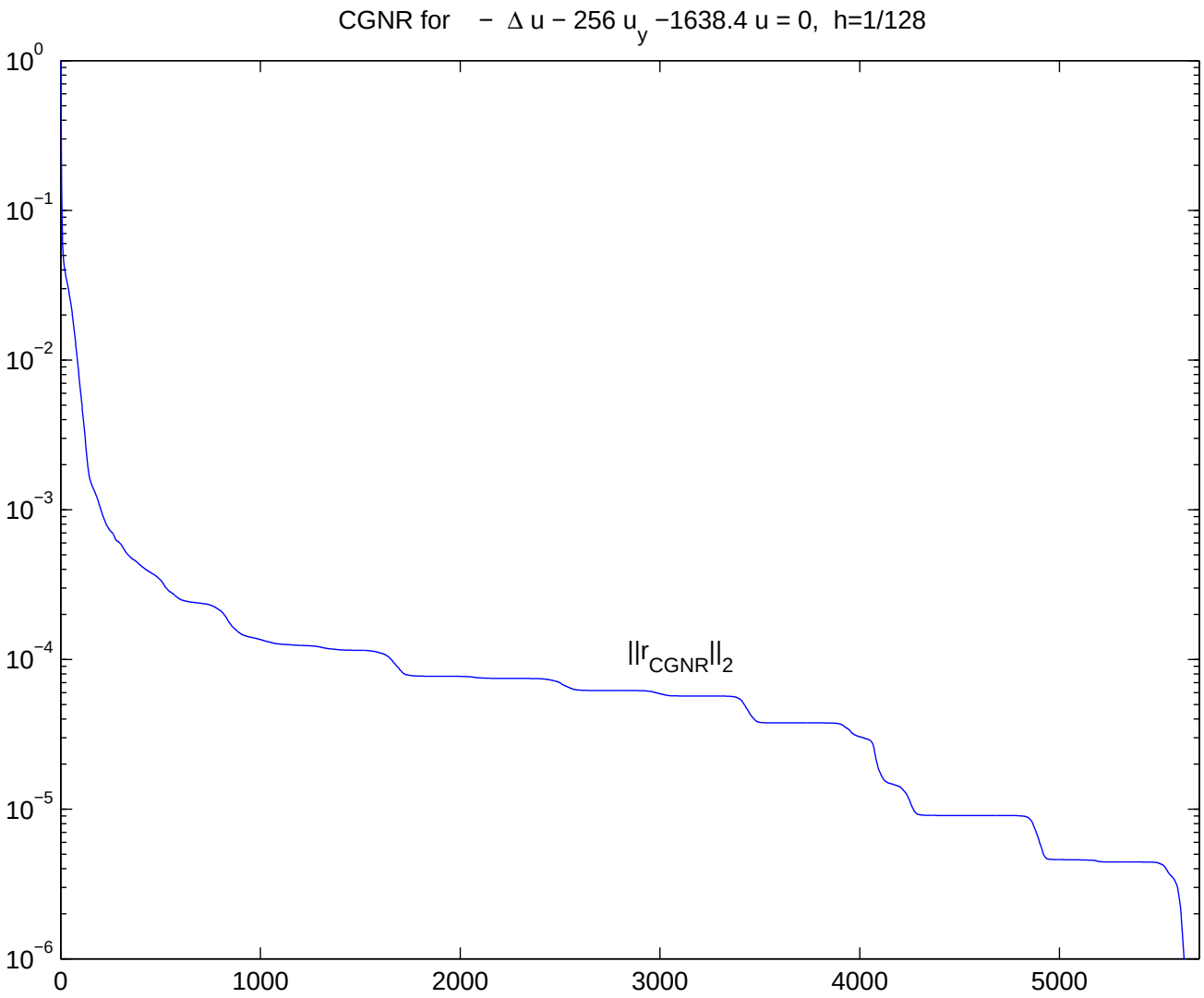
Example


CGNR for − Δ u − 256 u y = 0, h=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
1100
1200
||r_CGNR||_2

Example


CGNR for − Δ u − 256 u_y −163.84 u = 0, h=1/128
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
0
200
400
600
800
1000
1200
1400
||r_CGNR||_2

Example


CGNR for − Δ u − 256 u_y −1638.4 u = 0, h=1/128
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
0
1000
2000
3000
4000
5000
||r_CGNR||_2

CGNE

For $AA^T y = b$, $x = A^T y$, the iterates y^k minimize the functional

$$\|y - y^*\|^2_{AA^T} = (y - y^*)AA^T(y - y^*) = \|(A^T(y - y^*)\|_2^2 = \|x - x^*\|_2^2$$

in the Krylov

$$\begin{aligned} & A^T x^0 + A^T \mathcal{K}_k(r^0, A^T A) \\ = \quad & A^T x^0 + \text{span}\{A^T r^0, A^T(A^T A)r^0, \ldots, A^T(A^T A)^{k-1} r^0\}. \end{aligned}$$

This method is called CGNE method where the letter 'E' indicates that the error is minimized.

CGNE was introduced by Craig (1955).

CGNE

Again AA^T has not to be formed explicitly in the algorithm. Moreover, the method can be implemented in terms of the vectors x^k without reference to $y^k = A^T x^k$. The cost and storage requirements are the same as for CGNR.

$r^0 = b - Ax^0$
$d^1 = A^T r^0$
$\alpha_0 = \|r^0\|_2^2$
for $k = 1, 2, \ldots$ until convergence **do**
  $\tau_k = \alpha_{k-1} / \|d^k\|_2^2$
  $x^k = x^{k-1} + \tau_k d^k$
  $r^k = r^{k-1} - \tau_k A d^k$
  $\beta_k = 1/\alpha_{k-1}$
  $\alpha_k = \|r^k\|_2^2$
  $\beta_k = \alpha_k \beta_k$
  $d^{k+1} = A^T r^k + \beta_k d^k$
end for

Convergence of CGNE

The k-th iterate $x^k = A^T y^k$ satisfies

$$\begin{aligned} A^{-T}x^k = y^k &= y^0 + \tilde{q}_{k-1}(AA^T)(AA^T y^0 - b) \\ &= A^{-T}x^0 + \tilde{q}_{k-1}(AA^T)r^0, \end{aligned}$$

i.e.

$$x^k = x^0 + A^T \tilde{q}_{k-1}(AA^T)r^0.$$

Subtracting the solution x^* yields

$$e^k = e^0 + A^T \tilde{q}_{k-1}(AA^T)Ae^0,$$

and from

$$\begin{aligned} A^T \tilde{q}_{k-1}(AA^T)A &=: A^T\Big(\sum_{j=0}^{k-1} \alpha_j (AA^T)^j\Big)A \\ &= \sum_{j=0}^{k-1} \alpha_j A^T (AA^T)^j A = \sum_{j=0}^{k-1} \alpha_j (A^T A)^{j+1} \end{aligned}$$

we finally get

$$e^k = \tilde{p}_k(A^T A)e^0 \quad \tilde{p}_k(0) = 1.$$

Convergence of CGNE ct.

Replacing $\tilde{p}_k$ by a shifted and scaled Chebyshev polynomial one gets

Theorem 6.1a
Let A be nonsingular, σ_1 its largest and σ_n its smallest singular value, and let

$$R := \frac{\sigma_1 - \sigma_n}{\sigma_1 + \sigma_n} = \frac{\kappa_2(A) - 1}{\kappa_2(A) + 1}.$$

Then it holds

$$\|e^k\|_2 \leq \frac{2R^k}{1 + R^{2k}} \|e^0\|_2 \leq 2 \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1} \right)^k \|e^0\|_2.$$

CGNR, CGNE

The bounds given in Theorems 6.1 and 6.1a illustrate the main drawback of the CGN methods. In particular, if A is symmetric and positive definite the speed of convergence is slowed down to the one of the steepest descent method.

However, there are special situations where *CGNR* and *CGNE* are the optimal extensions of CG (cf. Nachtigal, Reddy & Treffethen (1992) and Freund, Golub & Nachtigal (1992)), but usually they are not satisfactory generalizations of the CG method.

As for the CG method the convergence properties of CGNE and CGNR can be improved considerably by preconditioning. Since the matrices $A^T A$ and AA^T are much more populated than the sparse matrix A the incomplete factorization techniques can not be adapted in a straightforward way.

Preconditioning by incomplete LQ factorization

Let $A = LQ$ be the LQ factorization of A, where L is a lower triangular matrix and Q is an orthogonal matrix. Then

$$AA^T = (LQ)(LQ)^T = LQQ^TL^T = LL^T,$$

and LL^T is the exact Cholesky factorization of AA^T which can be achieved without forming the matrix AA^T.

To obtain a sparse approximation Saad (1988) proposed the incomplete LQ factorization of A.Analogously to the incomplete Cholesky factorization most of the elements of L and Q are dropped during the factorization process according to some strategy to avoid excessive fill-in.

The LQ factorization can be calculated by applying the Gram-Schmidt process to the rows of A. This approach is known to be unstable if it is used to orthogonalize a large number of vectors. Notice, however that the rows of Q remain very sparse in the incomplete LQ factorization, and therefore any given row of A will be orthogonal to most of the previous rows of Q. Hence, within incomplete LQ factorization the Gram-Schmidt process is much less susceptible to numerical difficulties.

Incomplete LQ factorization ct.

Details of the implementation of the incomplete LQ factorization and its use as a preconditioner in CGNE method can be found in Saad (1988) and the book of Saad on Iterative Methods for Linear Systems.

Similarly the CGNR method can be preconditioned using the incomplete LQ factorization of A^T.

Preconditioning by SSOR

Consider the j-elementary step of the Gauß-Seidel method for the linear system

$$AA^T y = b$$

where the j-th component of y is modified such that the j-equation of the system is satisfied.

Let $y_{\text{new}} = y_{\text{old}} + \delta_j e^j$, then $\delta_j \in \mathbb{R}$ is chosen such that

$$0 = (e^j)^T(b - AA^T y_{\text{new}}) = (e^j)^T(b - AA^T y_{\text{old}} - \delta_j AA^T e^j)$$

$$\Rightarrow \quad \delta_j = \frac{b_j - (A^T e^j)^T A^T y_{\text{old}}}{\|A^T e^j\|_2^2} = \frac{b_j - a_j' x_{\text{old}}}{\|(a_j')^T\|_2^2},$$

where a_j' denotes the j-th row Of A, and $x_{\text{old}} = A^T y_{\text{old}}$. The update of x is given by

$$x_{\text{new}} = A^T y_{\text{new}} = A^T y_{\text{old}} + \delta_j A^T e^j = x_{\text{old}} + \delta_j (a_j')^T,$$

and the whole SSOR sweep with relaxation parameter ω reads:

SSOR for $AA^T y = b$, $x = A^T y$

1: Let y be an initial approximation
2: $x = A^T y$
3: **for** $j = 1, 2, \ldots, n, n, n-1, \ldots, 1$ **do**
4: $\quad \delta = \omega(b_j - a_j' x)/\|a_j'\|_2^2$
5: $\quad y_j = y_j + \delta$
6: $\quad x = x + \delta(a_j')^T$
7: **end for**

Comments

[1.] Since A is nonsingular the matrix AA^T is positive definite, and therefore the SSOR method converges for every $\omega \in (0, 2)$ if the **for**-statement is repeated. A serious difficulty is to determine the optimum relaxation parameter.

[2.] The matrix AA^T can be dense or in general much less sparse than A. However, the cost of the implementation above depend only on the nonzero structure of A.

CGNE-SSOR

In the following implementation of CGNE with preconditioning by SSOR the call "SSOR-CGNE(r,ω,z)" indicates that one sweep of the SSOR method with parameter ω and initial approximation $z = 0$ is executed for the linear system $AA^T z = r$.

1: $r^0 = b - Ax^0$
2: SSOR-CGNE(r^0, ω, z^0)
3: $d^1 = A^T z^0$
4: $\alpha_0 = (z^0)^T r^0$
5: **for** $k = 1, 2, \ldots$ until convergence **do**
6: $\quad \tau_k = \alpha_{k-1} / \|d^k\|_2^2$
7: $\quad x^k = x^{k-1} + \tau_k d^k$
8: $\quad r^k = r^{k-1} - \tau_k A d^k$
9: $\quad \beta_k = 1/\alpha_{k-1}$
10: $\quad$ SSOR-CGNE(r^k, ω, z^k)
11: $\quad \alpha_k = (z^k)^T r^k$
12: $\quad \beta_k = \alpha_k \beta_k$
13: $\quad d^{k+1} = A^T z^k + \beta_k d^k$
14: **end for**

SSOR for $A^TAx = A^Tb$

Similarly, if a^j denotes the j-th column of A the SSOR method for $A^TAx = A^Tb$ with relaxation parameter ω reads

1: Let x be an initial approximation
2: $z = Ax$
3: $c = A^Tb$
4: **for** $j = 1, 2, \ldots, n, n, n-1, \ldots, 1$ **do**
5: $\quad \delta = \omega(c_j - z^Ta^j)/\|a^j\|_2^2$
6: $\quad x_j = x_j + \delta$
7: $\quad z = z + \delta a^j$
8: **end for**

Comment
The SSOR method above can be applied to nonquadratic matrices $A \in \mathbb{R}^{m\times n}$ where $m > n$ and $\text{rank}(A) = n$. In this case it converges to the unique solution of the least squares problem

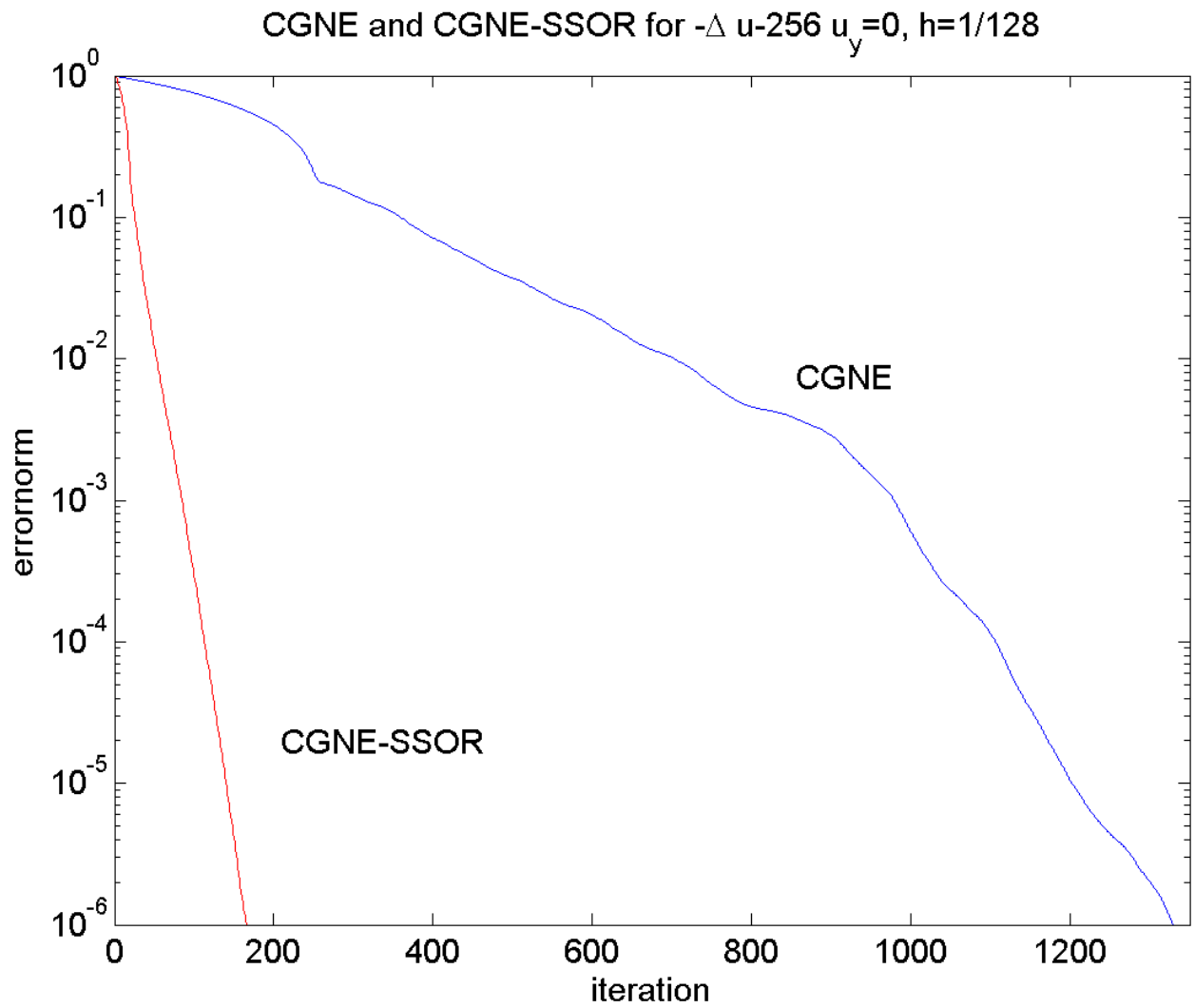
$$\|Ax - b\|_2 = \min!$$

CGNR-SSOR

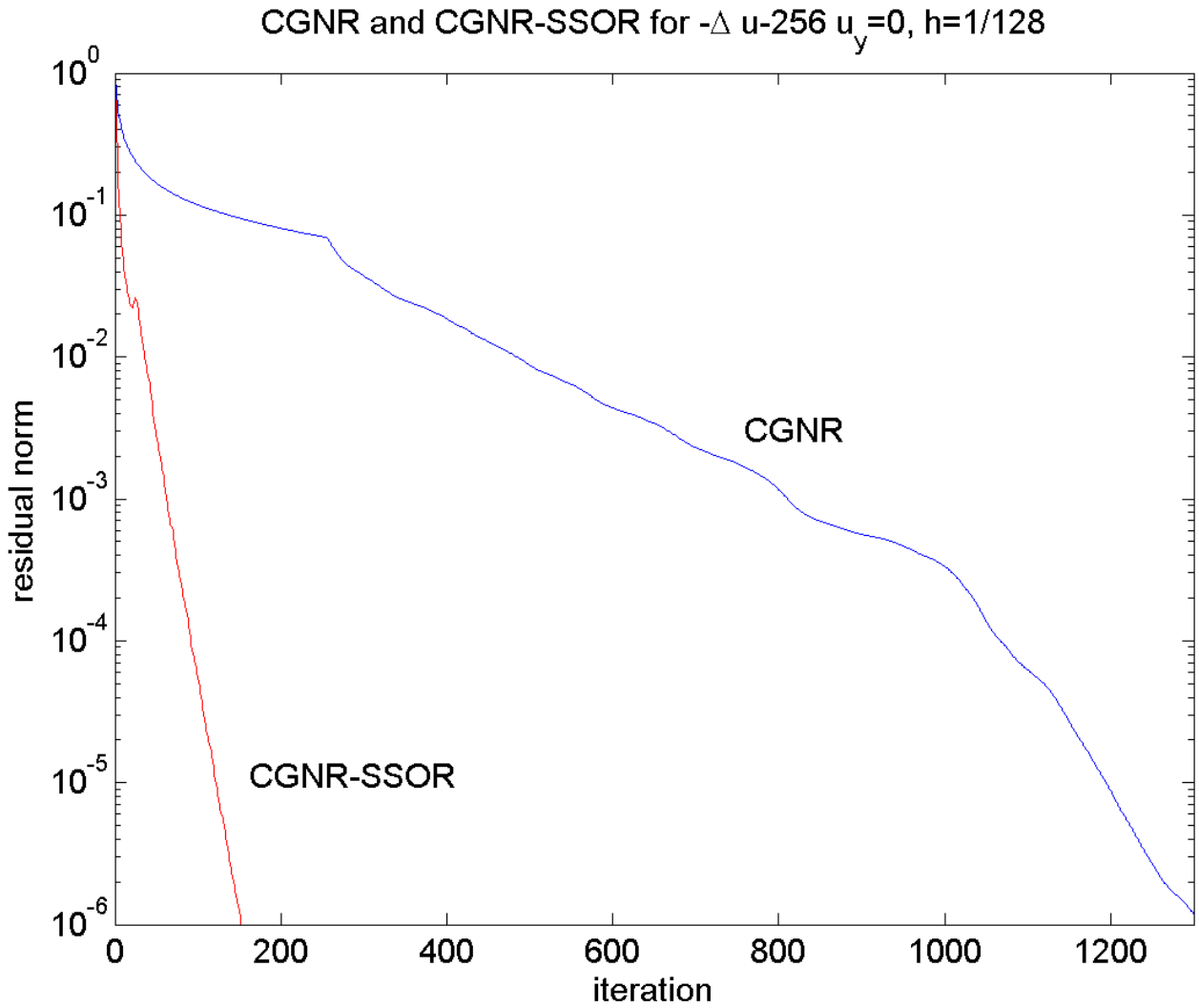
In the following implementation of preconditioned CGNR the call "SSOR-CGNR(r,ω,z)" indicates that one sweep of SSOR with parameter ω and initial approximation $z = 0$ is executed for the linear system $AA^T z = r$.

$r^0 = A^T(b - Ax^0)$
SSOR-CGNR(r^0, ω, z^0)
$d^1 = z^0$
$\alpha_0 = (z^0)^T r^0$
for $k = 1, 2, \ldots$ until convergence **do**
  $s^k = Ad^k$
  $\tau_k = \alpha_{k-1} / \|s^k\|_2^2$
  $x^k = x^{k-1} + \tau_k d^k$
  $r^k = r^{k-1} - \tau_k A^T s^k$
  $\beta_k = 1/\alpha_{k-1}$
  SSOR-CGNR(r^k, ω, z^k)
  $\alpha_k = (z^k)^T r^k$
  $\beta_k = \alpha_k \beta_k$
  $d^{k+1} = z^k + \beta_k d^k$
end for

Example


CGNE and CGNE-SSOR for -Δ u-256 u_y=0, h=1/128
errornorm
iteration
CGNE
CGNE-SSOR
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
0
200
400
600
800
1000
1200

Example


CGNR and CGNR-SSOR for -Δ u-256 u_y=0, h=1/128
residual norm
iteration
CGNR
CGNR-SSOR
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
0
200
400
600
800
1000
1200

Preconditioning

CGNR (and CGNE) can be precondioned not taking advantage of the special structure of the system matrix applying the methods to the system

$$M^{-1}Ax = M^{-1}b \qquad \text{left preconditioning}$$

or to

$$AM^{-1}y = b,\ y = Mx \qquad \text{right preconditioning}$$

where in both cases M is an approximation to A such that linear systems $Mu = c$ can be solved easily.

In the following examples we used $M = LU$ where

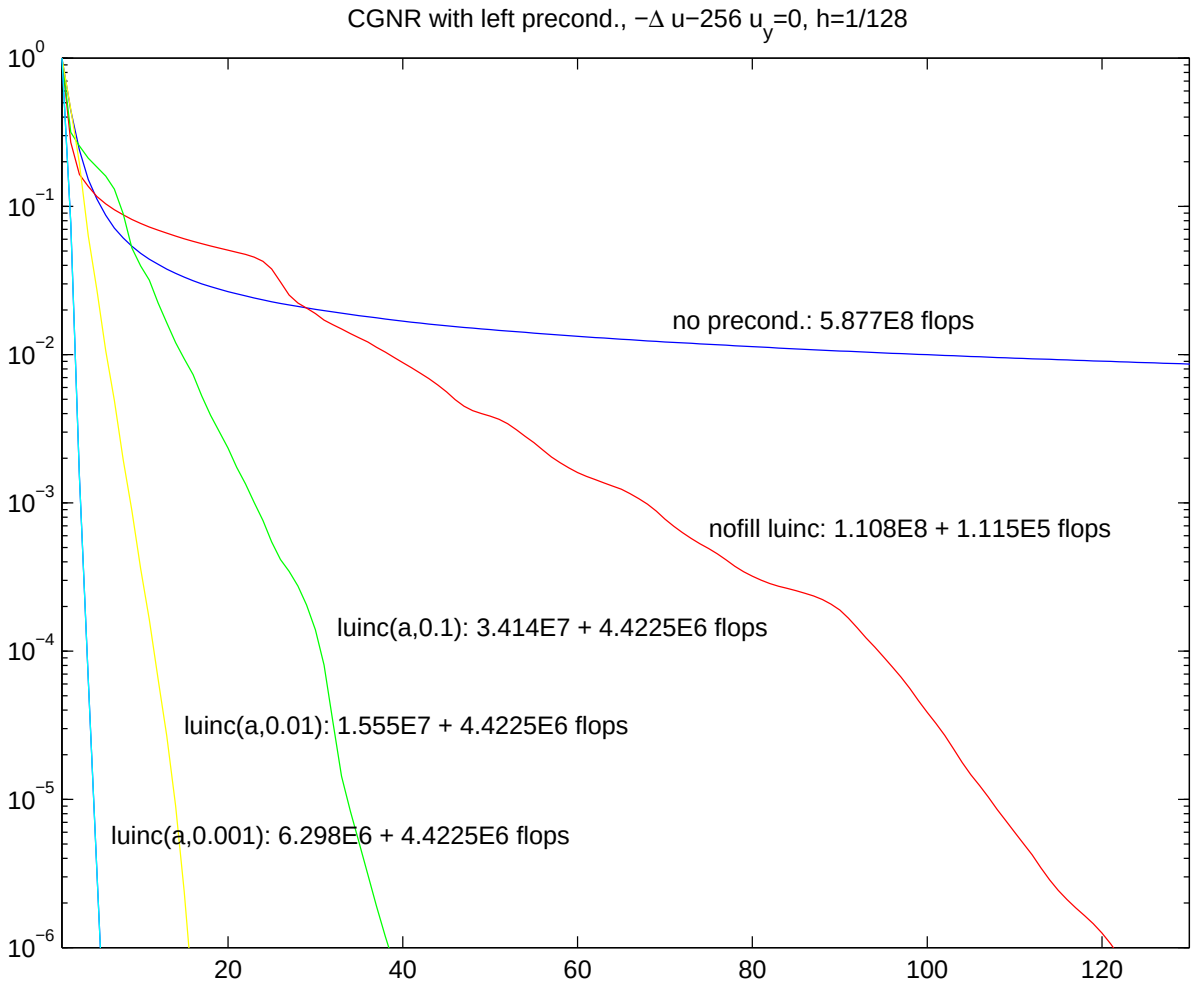
$$[L, U] = \text{luinc}(A,'0') \quad \text{no fill incomplete LU decomposition}$$

or

$$[L, U] = \text{luinc}(A, \tau)$$

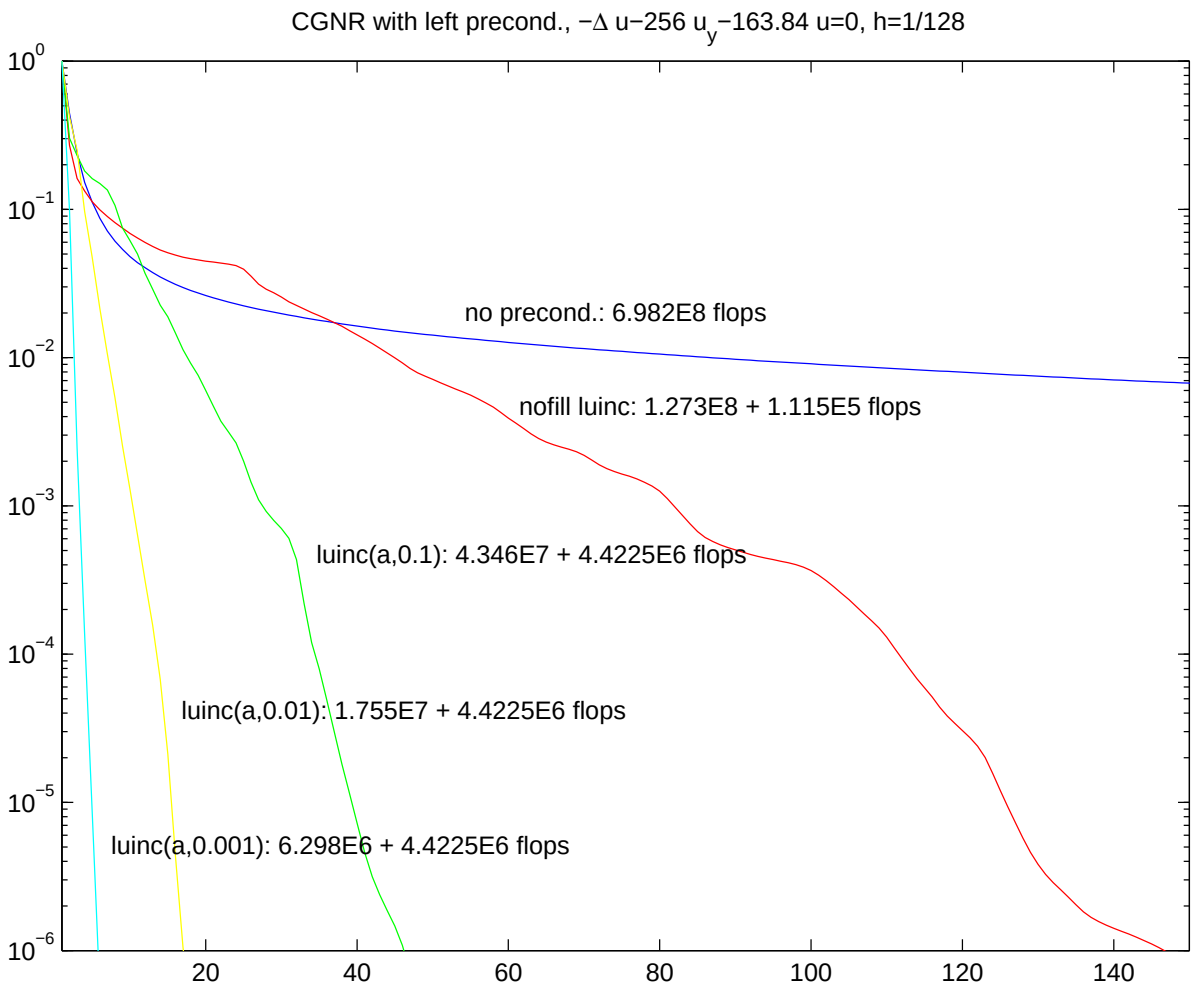
is an incomplete LU factorization of A with threshold τ.

Example



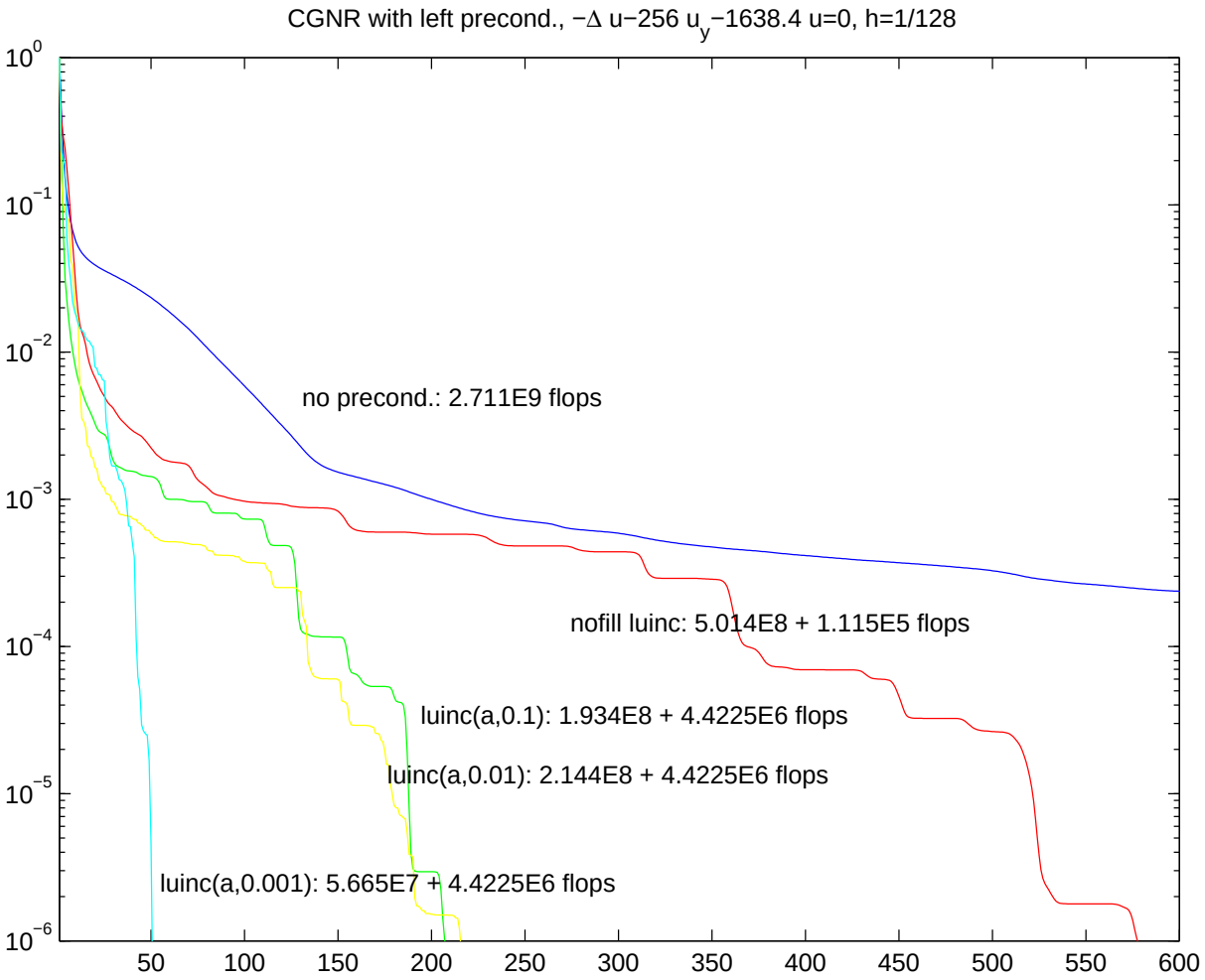
Direct solution using \: 1.825E7 flops

Example



Direct solution using \: 1.825E7 flops

Example



Direct solution using \: 1.825E7 flops

Sparse approximate inverses

For symmetric problems the preconditioner had to be symmetric as well. For nonsymmetric problems this is no longer necessary. We just have to find a matrix $C = M^{-1}$ such that $\|AC - I\| \approx 0$ (right preconditioning) or $\|CA - I\| \approx 0$ (left preconditioning). Then applying the preconditioner simply amounts in matrix multiplication.

The Frobenius norm is particularly interesting to choose since

$$\|AC - I\|_F^2 = \sum_{j=1}^{n} \|(AC - I)e^j\|_2^2.$$

SPAI ct.

Hence, if c^j denotes the j-th column of C, then we have to solve the n independent least squares problems

$$\|Ac^j - e^j\|_2 = \min!, \quad j = 1, \ldots, n.$$

Solving these problems is equivalent to determining the inverse of A, and will be much too expensive. Therefore, for every $j = 1, \ldots, n$ we choose a sparsity pattern $G(j)$, denote by $\hat{c}^j$ a vector which contains only elements in $G(j)$, and by $\hat{A}_j$ a matrix whose columns are the columns of A corresponding to $G(j)$ and whose rows i are such that there exists $a_{ik} \neq 0$ for some $k \in G(j)$. Since A is sparse, $\hat{A}_j$ will be a small matrix, and the resulting least squares problems

$$\|\hat{A}_j \hat{c}^j - \hat{e}^j\|_2 = \min! \qquad (*)$$

can be solved by QR factorization.

Choice of sparsity pattern

Grote & Huckle (1997) proposed an incremental method. Starting from a set of indices $G_0(j)$ (usually corresponding to a diagonal C or to the structure of A) they solve the least squares problem (*) and then, iteratively, enlarge the set of indices.

Let

$$\|\hat{A}_j^{(p)}\hat{c}^{j,p} - e^j\|_2 = \min! \qquad (**)$$

be the least squares problem in the p-th step of the iteration, let $c^j \in \mathbb{R}^n$ be the solution of (**) (i.e. $\hat{c}^{j,p}$ augmented by zero components), and denote by $r = Ac^j - e^j$ the residual.

Choice of sparsity pattern ct.

Let $\mathcal{L} = \{k : r_k \neq 0\}$, and for $\ell \in \mathcal{L}$ let $\mathcal{N}_\ell = \{k : a_{\ell k} \neq 0, k \notin G_p(j)\}$. Then the candidates for indices of new non-zero components in the solution vector are chosen in

$$\cup_{\ell \in \mathcal{L}} \mathcal{N}_\ell.$$

For k in this set we consider the problem

$$\|r + \mu_k A e^k\|_2 = \|A(c^j + \mu_k e^k) - e^j\|_2 = \min! \quad \Longrightarrow \quad \mu_k = -\frac{r^T A e^k}{\|A e^k\|_2^2}.$$

It can be shown that the norm of the new residual is

$$\|r\|_2^2 - \frac{(r^T A e^k)^2}{\|A e^k\|_2^2},$$

and that there exist indices such that $r^T A e^k \neq 0$. From these indices one chooses those which yield the smallest residual.

Choice of sparsity pattern ct.

This iteration step is repeated until the norm of the residual is smaller than a prescribed bound or until we have reached the maximum storage that is allowed for column j.

Note that putting new indices into $G_p(j)$ will add columns and rows to the matrix $\hat{A}_j^{(p)}$. Methods exist to update the QR factorization.

Approximate inverse preconditioning is an active field of research. It is particularly interesting for parallel machines. Gould & Scott (1995) described some improvements of the method. A symmetric approximate inverse preconditioner was proposed by Benzi, Meyer & Tuma (1996).

Nonsymmetric problems

Desirable are generalizations of CG (or CR) methods which have similar properties as CG without squaring the condition number of the system matrix.

Outstanding properties of the CG method:

— based on a short recurrence (low storage requirements)

— minimizes error on ascending sequence of Krylov spaces

Theorem of Faber & Manteuffel (1984)
Methods with these two properties exist only for a very limited class of matrices.
For 3-term-recurrences only for matrices

$$A = e^{i\theta}(H + \sigma I),\ H = H^H,\ \theta \in \mathbb{R},\ \sigma \in \mathbb{C}.$$

Hence, one has to decide which one of the two properties to keep.

Generalized Conjugate Residuals

Eisenstat, Elman & Schultz (1983) studied the Generalized Conjugate Residual Method, GCR method for short, which is a direct generalization of the CR method (and of MINRES) to nonsymmetric systems of linear equations.

By the Gram-Schmidt process a basis $\{d^1, \ldots, d^k\}$ of the Krylov space $\mathcal{K}_k(r^0, A)$ is constructed which is orthogonal with respect to the scalar product $\langle x, y \rangle_{A^TA} = x^T A^T A y$.

Since $d^1, \ldots, d^k$ are A^TA–conjugate directions the minimization problem

$$\|x - x^*\|_{A^TA} = \text{min!}, \quad x \in x^0 + \mathcal{K}_k(r^0, A)$$

is decoupled into a sequence of line searches along $d^1, \ldots, d^k$.

Hence, the following algorithm generates iterates x^k which minimize the Euclidean norm of the residual $\|b - Ax\|_2 = \|x - x^*\|_{A^TA}$ upon $x^0 + \mathcal{K}_k(r^0, A)$ (if it does not break down).

GCR method

Choose an initial guess x^0
$r^0 = b - Ax^0$
$d^1 = r^0$
$s^1 = Ad^1$
for $k = 1, 2, \ldots$ until convergence **do**
  $\alpha_k = \|s^k\|_2^2$
  $\tau_k = (r^{k-1})^T s^k / \alpha_k$
  $x^k = x^{k-1} + \tau_k d^k$
  $r^k = r^{k-1} - \tau_k s^k$
  $d^{k+1} = r^k$
  $s^{k+1} = Ad^{k+1}$
  for $j = 1 : k$ **do**
    $\gamma_{k,j} = (s^{k+1})^T s^j / \alpha_j$
    $d^{k+1} = d^{k+1} - \gamma_{k,j} d^j$
    $s^{k+1} = s^{k+1} - \gamma_{k,j} s^j$
  end for
end for

Generalized Conjugate Residuals

Without further assumptions on the matrix A the GCR method may break down because d^{k+1} becomes the null vector.

Consider for instance

$$A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad x^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Then at step $k = 1$ one obtains $Ad^1 = 0$, and therefore τ_1 is undefined.

It can be shown (cf. Elman (1982)) that the CGR method does not break down, if A is a positive real matrix, i.e. if

$$M = \frac{1}{2}(A + A^T)$$

is positive definite. Hence, for this type of matrices the GCR methods (in exact arithmetic) yields the solution in a finite number of steps.

Comments on GCR

Axelsson (1980) proposed the following method for nonsymmetric linear systems:

For x^0, r^0, and d^1 as in the GCR method let

$$\begin{aligned} x^k &= x^{k-1} + \sum_{j=1}^{k-1} \eta_{k,j} d^j \\ r^k &= b - Ax^k \\ \beta_k &= (Ar^k)^T Ad^k / \|Ad^k\|_2^2 \\ d^{k+1} &= r^k - \beta_k d^k \end{aligned}$$

where the scalars $\eta_{k,j}$ are chosen such that $\|r^k\|_2$ in minimal, i.e. from the linear system of equations

$$(d^j)^T A^T (Ay - r^{k-1}) = 0,\ j = 1, \ldots, k,\ y = \sum_{j=1}^{k-1} \eta_{k,j} d^j.$$

Obviously, this method is equivalent to the GCR method.

Comments on GCR

The work and storage requirements of the GCR method are prohibitively high.

Vinsome (1976) proposed a method called ORTHOMIN which is a truncated GCR method and which is considerably less expensive per iteration step. For an integer $m \ll n$ the search direction d^{k+1}, $k > m$, is forced to be $A^T A$–orthogonal only to the last m directions $d^k, \ldots, d^{k-m+1}$, but not to all preceding directions d^j. This method usually is referred to ORTHOMIN(m).

Eisenstat, Elman & Schultz (1983) alternatively suggested to restart the GCR method every $m+1$ steps, i.e. the current iterate $x^{j(m+1)}$, $j = 0, 1, 2, \ldots$ is chosen as a new initial guess for the GCR method. The storage requirements of this method, called GCR(m), are the same as for ORTHOMIN(m) since m preceding directions have to be stored, but the work per iteration is smaller, since on an average only $m/2$ vectors d^j are used to compute the current search direction.

ORTHODIR

Young & Jea (1980) studied methods which are closely related to ORTHOMIN. In ORTHOMIN they replaced the minimization problem in the k-th step by

$$\|x^k - x^*\|_{ZA} \leq \|x - x^*\|_{ZA} \quad \text{for every } x \in x^0 + \mathcal{K}_k(r^0, A)$$

where $Z \in \mathbb{R}^{n\times n}$ is an arbitrary matrix such tht ZA is positive definite.

Moreover, they considered an algorithm called ORTHODIR or ORTHODIR(m) for the truncated form, which differs from the GCR method only by the choice of the search directions. These are given by

$$\tilde{d}^{k+1} = A\tilde{d}^k - \sum_{j=1}^{k} \tilde{\gamma}_{k,j}\tilde{d}^j$$

where the scalars $\tilde{\gamma}_{k,j}$ are chosen such that the search directions $\tilde{d}^j$ are ZA-conjugate.

Notice that ORTHODIR can only break down with $\tilde{d}^{k+1} = 0$ if the solution x^* lies in $x^0 + \mathcal{K}_k(r^0, A)$ assuming exact arithmetic.

ORTHORES

Young & Jea (1980) introduced a method called ORTHORES and ORTHORES(m) for the truncated variant, where the iterates are constructed by the recurrence relation

$$x^k = \gamma_k r^{k-1} r^{k-1} + \sum_{j=0}^{k-1} \delta_{k,j} x^j$$

and γ_k and $\delta_{k,j}$ are determined such that the residual vectors r^j are semi-orthogonal with respect to Z, i.e.

$$(Zr^k)^T r^j = 0 \quad \text{for } j < k.$$

For the truncated version ORTHORES(m) this requirement must be fulfilled for $k - m \le j < k$

For the choice of Z Young and Jea considered among others

$$Z = A^T, \quad Z = A^T(A + A^T), \quad \text{and} \quad Z = A^T M$$

where $M \in \mathbb{R}^{n \times n}$ is a preconditioner of A.

Generalized Minimal Residual Method (GMRES)

A different approach to the iterative solution of general linear systems which generalizes the MINRES method and which is mathematically equivalent to the GCR method was introduced by Saad & Schultz (1986). Differently from the GCR method this method, which is called Generalized Minimal RESidual Method (GMRES for short) does not break down for general matrices, which are not positive real.

GMRES constructs a sequence

$$x^k \in x^0 + \mathcal{K}_k(r^0, A)$$

of vectors such that the Euclidean norm of the residuum becomes minimal.

To this end an orthonormal basis $q^1, \ldots, q^k$ of $\mathcal{K}_k(r^0, A)$ is determined using Arnoldi's method (i.e. the modified Gram-Schmidt process), such that the problem

$$\|r^k\|_2 = \|b - A(x^0 + Q_k y)\|_2 = \min!, \quad y \in \mathbb{R}^k,$$

can be solved easily.

Arnoldi method

1: Let q^1 be an initial vector such that $\|q^1\|_2 = 1$.
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad q^{k+1} = Aq^k$
4: $\quad$ **for** $j = 1 : k$ **do**
5: $\quad\quad h_{jk} = (q^{k+1})^T q^j$
6: $\quad\quad q^{k+1} = q^{k+1} - h_{jk} q^j$
7: $\quad$ **end for**
8: $\quad h_{k+1,k} = \|q^{k+1}\|_2$
9: $\quad q^{k+1} = q^{k+1} / h_{k+1,k}$
10: **end for**

GMRES

With $Q_k = (q^1, \ldots, q^k)$ and

$$\tilde{H}_k = (h_{j\ell}) \in \mathbb{R}^{(k+1)\times k} \quad \text{such that } h_{j\ell} = 0 \text{ for } j - \ell > 2$$

it holds

$$AQ_k = Q_{k+1}\tilde{H}_k,$$

and for $x = x^0 + Q_k y$, $r = b - Ax$ and $q^1 = r^0/\|r^0\|_2$

$$\begin{aligned} \|r\|_2 &= \|b - A(x^0 + Q_k y)\|_2 = \|r^0 - AQ_k y\|_2 = \|r^0 - Q_{k+1}\tilde{H}_k y\|_2 \\ &= \left\| Q_{k+1}(\|r^0\|_2 e^1 - \tilde{H}_k y) \right\|_2 = \left\| \|r^0\|_2 e^1 - \tilde{H}_k y \right\|_2 . \end{aligned}$$

Hence, the problem to minimize the Euclidean norm of the residuum on a Krylov space yields a least squares problem where the system matrix $\tilde{H}_k$ has Hessenberg form .

GMRES ct.

If $U_k \in \mathbb{R}^{(k+1)\times(k+1)}$ is an orthogonal matrix such that

$$U_k \tilde{H}_k = \begin{pmatrix} R_k \\ 0^T \end{pmatrix}, \quad R_k \text{ upper triangular,}$$

and if u^k is the first column of U_k then it holds

$$\begin{aligned} \left\| \|r^0\|_2 e^1 - \tilde{H}_k y \right\|_2^2 &= \left\| U_k(\|r^0\|_2 e^1 - \tilde{H}_k y) \right\|_2^2 \\ &= \left\| \|r^0\|_2 u^k - \begin{pmatrix} R_k \\ 0^T \end{pmatrix} y \right\|_2^2 \\ &= \left\| \|r^0\|_2 (u_1^k, \ldots, u_k^k)^T - R_k y \right\|_2^2 \\ &\quad + \|r^0\|_2^2 \cdot |u_{k+1}^k|^2. \end{aligned}$$

GMRES ct.

Hence, the least squares problem is equivalent to the linear system

$$R_k y = \|r^0\|_2 \begin{pmatrix} u_1^k \\ \vdots \\ u_k^k \end{pmatrix}$$

where R_k is an upper triangular matrix, and the norm of the residuum satisfies

$$\|b - Ax^k\|_2 = \|r^0\| \cdot |u_{k+1}^k|,$$

i.e. the accuracy of the method, can be monitored without explicitly forming x^k or y^k.

If the residual norm is sufficiently small, GMRES is terminated, and only after this final step of the Arnoldi method, the triangular system is solved and $x^k = x^0 + Q_k y^k$ is determined.

GMRES ct.

Since $\tilde{H}_k$ is an upper Hessenberg matrix, the matrix U_k can be calculated efficiently using Givens reflections. Let

$$G_j = \begin{pmatrix} I_{j-2} & 0 & 0 \\ 0^T & c_j & s_j \\ 0^T & s_j & -c_j \end{pmatrix} \in \mathbb{R}^{j\times j},\ c_j^2 + s_j^2 = 1$$

denote a Givens reflection acting on the $(j-1)$-st and j-th component of a vector.

Let $U_1 = G_2$ be such that

$$U_1 \tilde{H}_1 = \begin{pmatrix} c_2 & s_2 \\ s_2 & -c_2 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{21} \end{pmatrix} = \begin{pmatrix} \sqrt{h_{11}^2 + h_{21}^2} \\ 0 \end{pmatrix} =: \begin{pmatrix} \hat{h}_{11} \\ 0 \end{pmatrix}.$$

GMRES ct.

Assume that the orthogonal matrix

$$U_{k-1} = G_k \begin{pmatrix} G_{k-1} & 0 \\ 0^T & 1 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} G_2 & O \\ O & I_{k-1} \end{pmatrix} =: G_k \begin{pmatrix} U_{k-2} & 0 \\ 0^T & 1 \end{pmatrix}$$

is chosen such that

$$U_{k-1}\tilde{H}_{k-1} =: \begin{pmatrix} \hat{h}_{11} & \hat{h}_{12} & \ldots & \hat{h}_{1,k-1} \\ 0 & \hat{h}_{22} & \ldots & \hat{h}_{2,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \hat{h}_{k-1,k-1} \\ 0 & 0 & \ldots & 0 \end{pmatrix} =: \begin{pmatrix} R_{k-1} \\ 0^T \end{pmatrix} \in \mathbb{R}^{k\times(k-1)}$$

is an upper triangular matrix.

GMRES ct.

Since the multiplication of a matrix by G_{k+1} from the left leaves invariant the first $k-1$ rows one obtains the upper triangular matrix in the k-th step of the method by

$$\begin{aligned} U_k \tilde{H}_k &= G_{k+1} \begin{pmatrix} U_{k-1} & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \tilde{H}_{k-1} & h^k \\ 0^T & h_{k+1,k} \end{pmatrix} \\ &= \begin{pmatrix} I_{k-1} & 0 & 0 \\ 0^T & c_{k+1} & s_{k+1} \\ 0^T & s_{k+1} & -c_{k+1} \end{pmatrix} \begin{pmatrix} R_{k-1} & \tilde{h}^k \\ 0^T & g_k \\ 0^T & h_{k+1,k} \end{pmatrix} = \begin{pmatrix} R_k \\ 0^T \end{pmatrix} \end{aligned}$$

where

$$U_{k-1} \begin{pmatrix} h_{1k} \\ \vdots \\ h_{k-1,k} \\ h_{kk} \end{pmatrix} =: U_{k-1} h^k =: \begin{pmatrix} \hat{h}_{1k} \\ \vdots \\ \hat{h}_{k-1,k} \\ g_k \end{pmatrix} =: \begin{pmatrix} \hat{h}^k \\ g_k \end{pmatrix}, \ldots$$

GMRES ct.

$$\ldots, s_{k+1} := \frac{h_{k+1,k}}{\sqrt{g_k^2 + h_{k+1,k}^2}}, \; c_{k+1} := \frac{g_k}{\sqrt{g_k^2 + h_{k+1,k}^2}}.$$

In a similar manner as in the SYMMLQ method the elements $\hat{h}_{j\ell}$ can be obtained along with the Arnoldi process, and the reflections can be applied to the right hand side $\|r^0\|_2 e^1$ currently.

Observe however, that differently from the SYMMLQ method the nontrivial entries c_j and s_j of G_j have to be stored since they are needed in the transformation of the k-th column h^k of $\tilde{H}_k$.

The diagonal elements of the triangular matrix R_k are given by $\hat{h}_{jj} = \sqrt{g_j^2 + h_{j+1,j}^2}$. Hence, R_k is nonsingular, if the Arnoldi process constructs k orthogonal vectors q^j. GMRES breaks down if and only if $h_{j+1,j} = 0$ for some j, and this happens if and only if $r^j = 0$, i.e. $x^j = x^*$. (lucky breakdown)

GMRES algorithm

$q^1 = b - Ax^0$
$z_1 = \|q^1\|_2$
$q^1 = q^1/z_1$
$k = 1$
while $|z_k| >$tol **do**
  $q^{k+1} = Aq^k$
  for $i = 1 : k$ **do**
    $h_{ik} = (q^i)^T q^{k+1}$
    $q^{k+1} = q^{k+1} - h_{ik} q^i$
  end for
  $h_{k+1,k} = \|q^{k+1}\|_2$
  $q^{k+1} = q^{k+1}/h_{k+1,k}$
  for $i = 1 : k-1$ **do**

$$\begin{pmatrix} h_{i,k} \\ h_{i+1,k} \end{pmatrix} = \begin{pmatrix} c_{i+1} & s_{i+1} \\ s_{i+1} & -c_{i+1} \end{pmatrix} \begin{pmatrix} h_{i,k} \\ h_{i+1,k} \end{pmatrix}$$

  end for

GMRES algorithm ct.

$\alpha = \sqrt{h_{kk}^2 + h_{k+1,k}^2}$
$s_{k+1} = h_{k+1,k}/\alpha;$
$c_{k+1} = h_{kk}/\alpha$
$h_{kk} = \alpha$
$z_{k+1} = s_{k+1} z_k$
$z_k = c_{k+1} z_k$
$k = k + 1$
end while
$y_k = z_k / h_{kk}$
for $i = k - 1 : -1 : 1$ **do**
$y_i = (z_i - \sum_{j=i+1}^{k} h_{ij} y_j)/h_{ii}$
end for
$x^k = x^0 + \sum_{j=1}^{k} y_j q^j$

Cost of GMRES

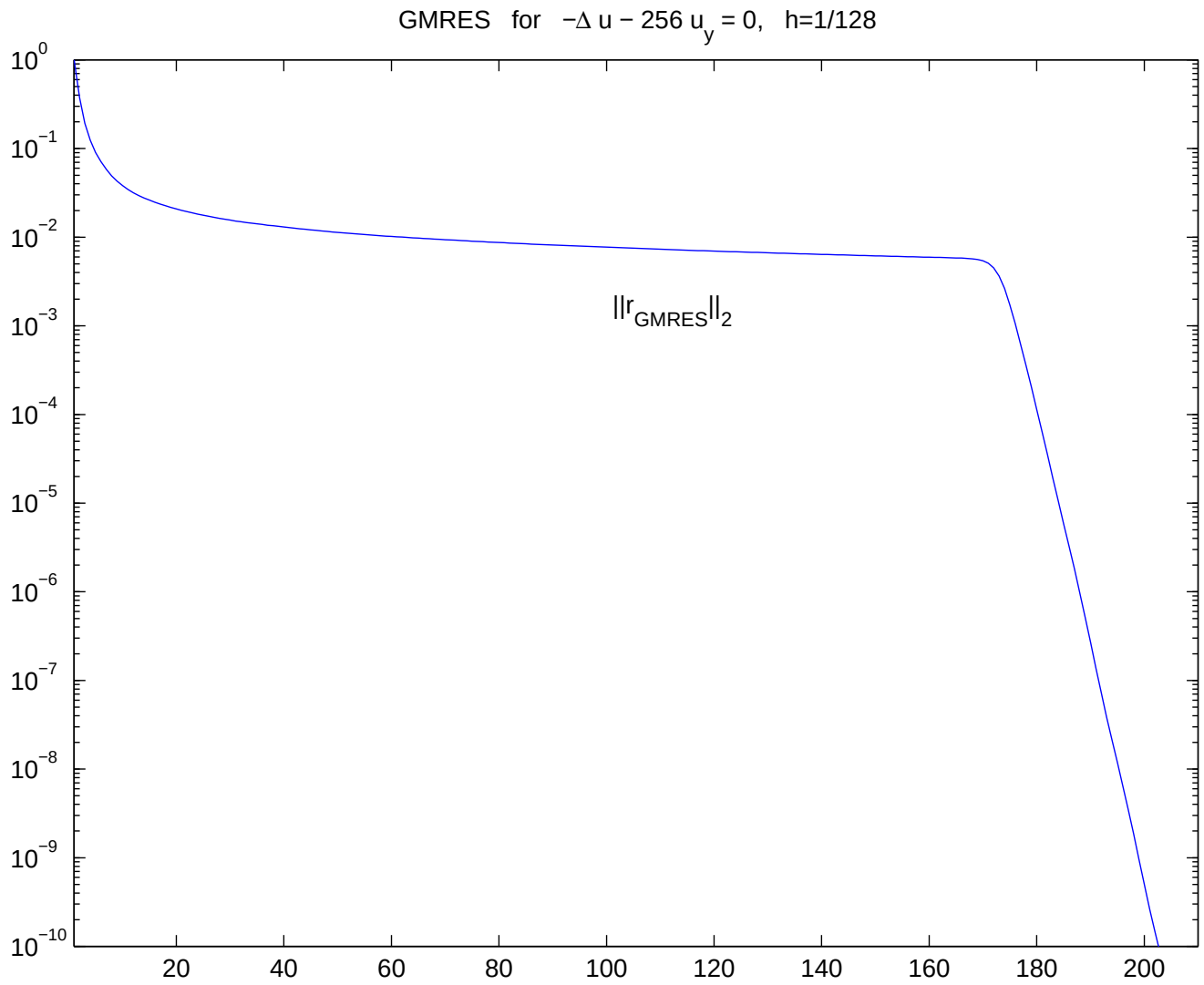
Cost of the k-th step (neglecting the transformation of $\tilde{H}_k$ to triangular form):

1	matrix-vector product
k+1	scalar products
k+1	_axpy

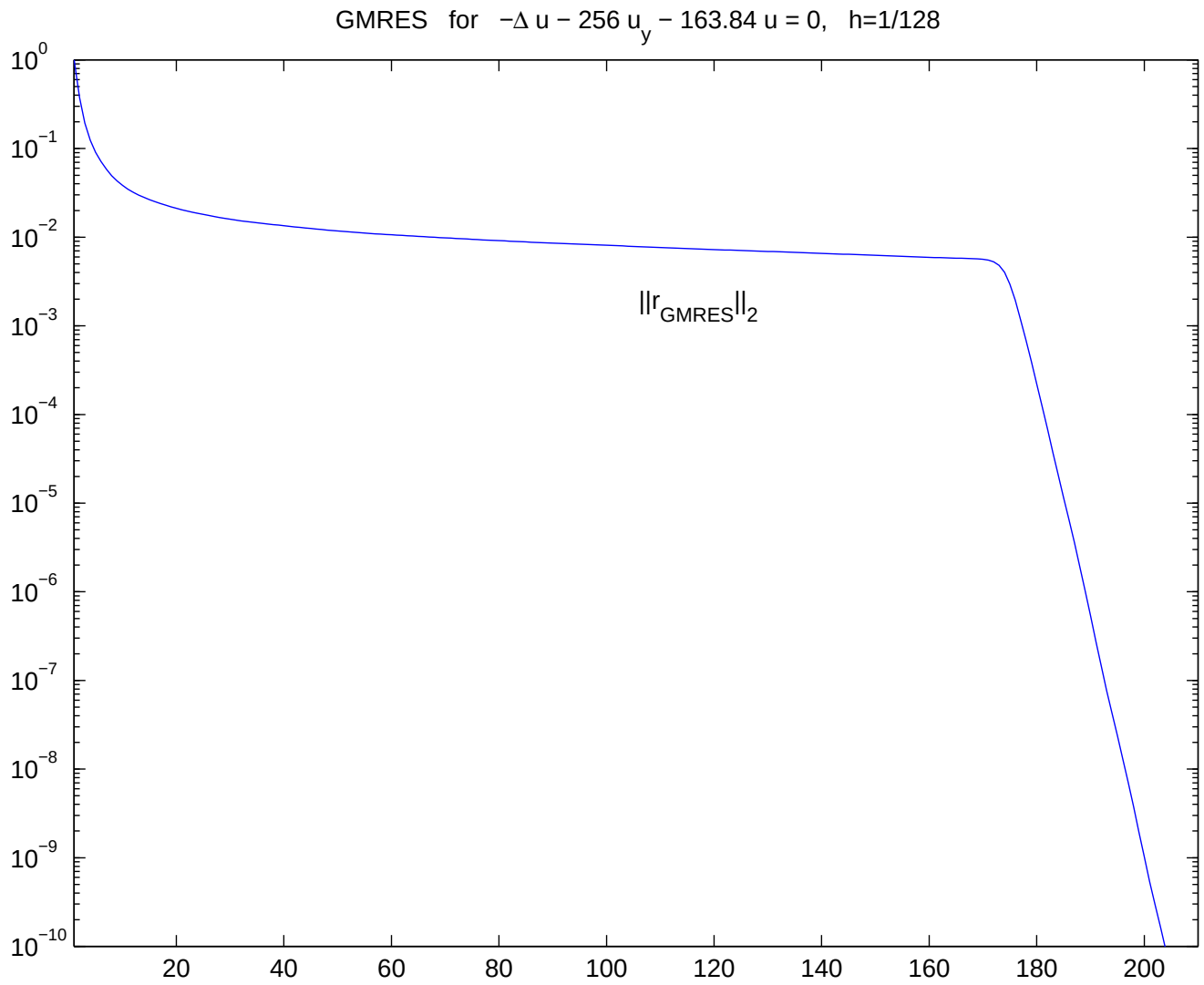
Determining $x^m \in x^0 + \mathcal{K}_m(r^0, A)$ costs

m+1	matrix-vector products
(m+1)(m+2)+m	level-1-operations
m+1	vectors to store

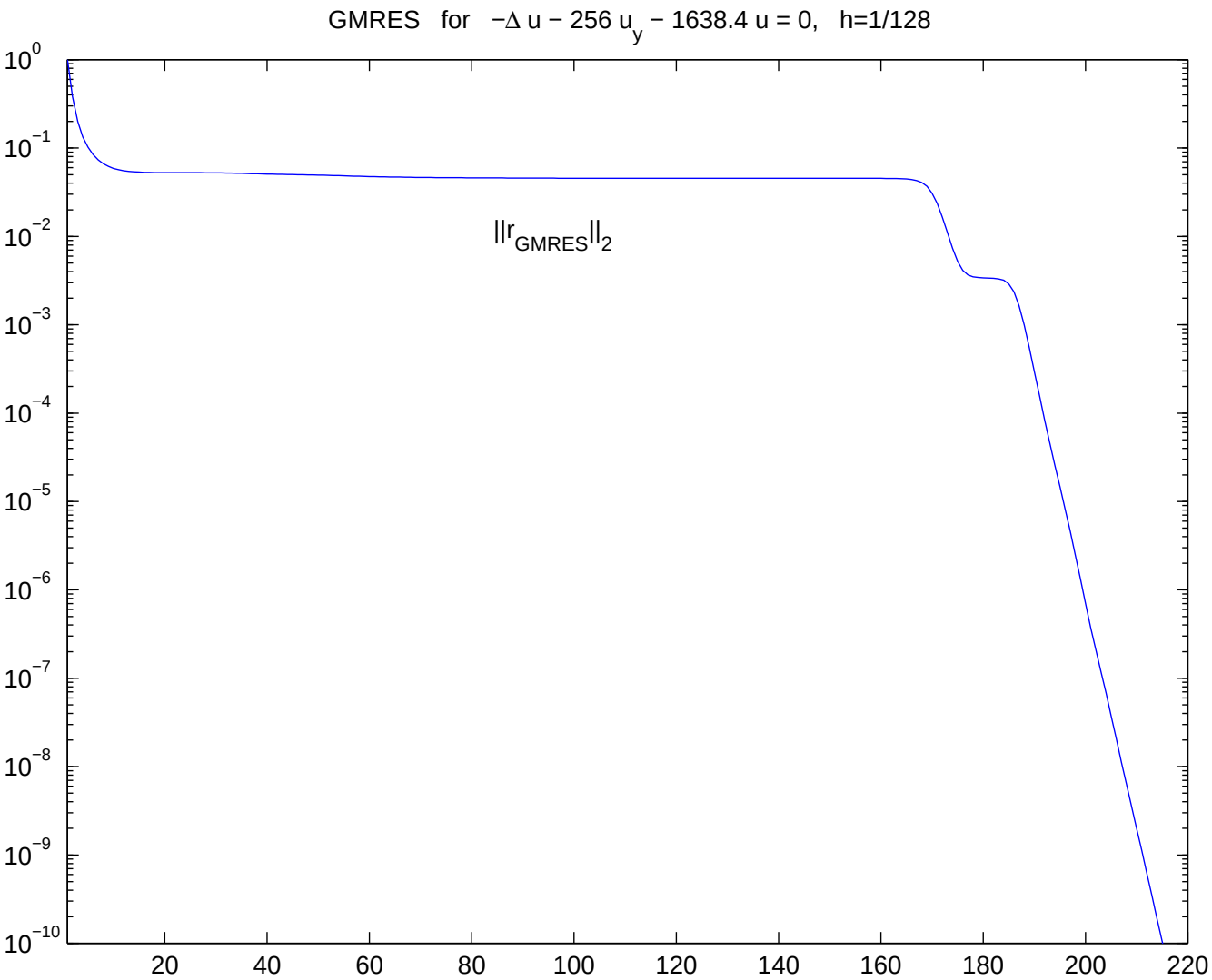
Example


GMRES for −Δ u − 256 u_y = 0, h=1/128
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
10^−7
10^−8
10^−9
10^−10
20
40
60
80
100
120
140
160
180
200
||r_GMRES||_2

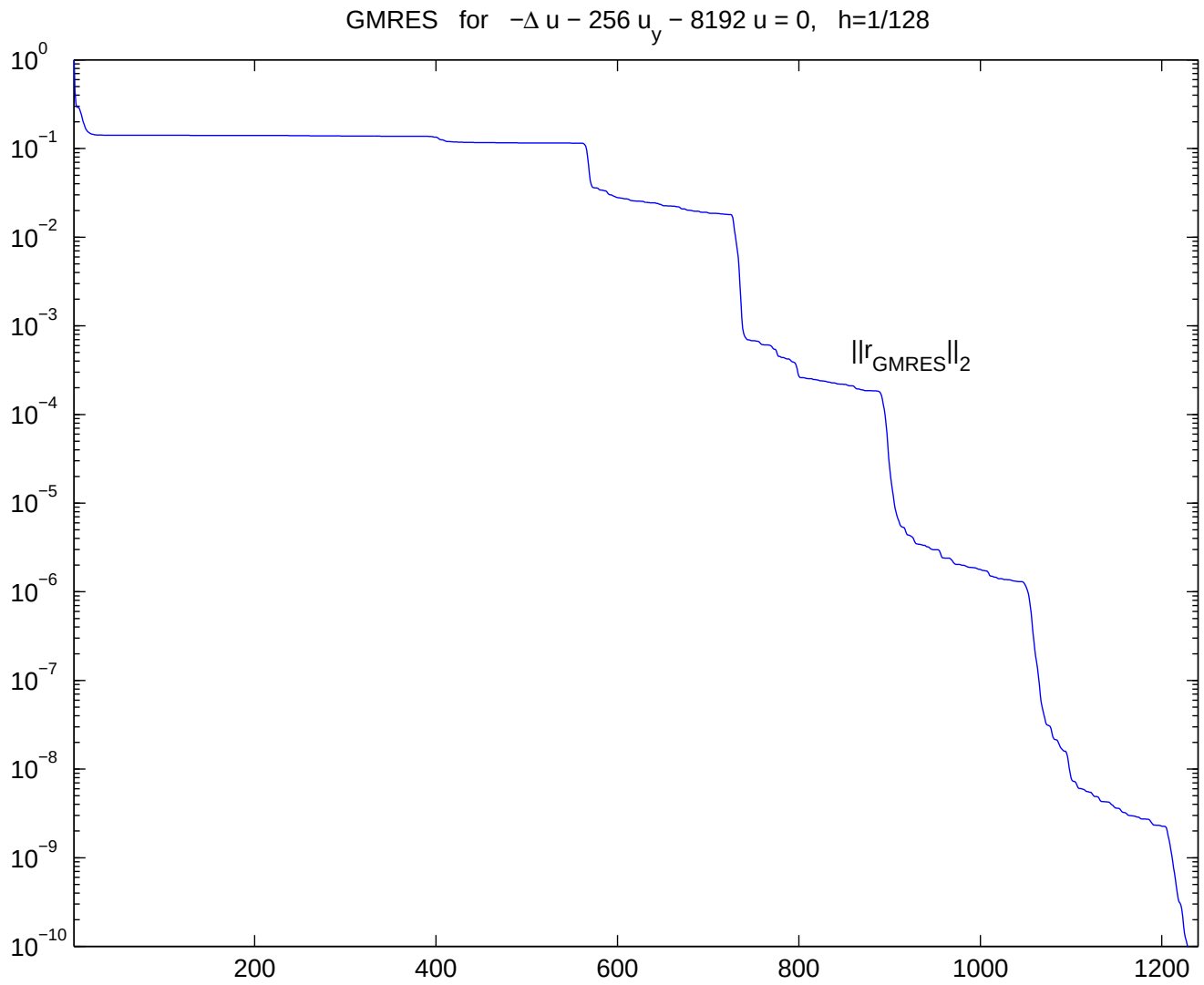
Example


GMRES for $-\Delta u - 256\, u_y - 163.84\, u = 0$, h=1/128
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
$\|r_{GMRES}\|_2$
20
40
60
80
100
120
140
160
180
200

Example


GMRES for −Δ u − 256 u_y − 1638.4 u = 0, h=1/128
||r_GMRES||_2
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
10^−7
10^−8
10^−9
10^−10
20
40
60
80
100
120
140
160
180
200
220

Example


GMRES for $-\Delta u - 256\, u_y - 8192\, u = 0$, h=1/128
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
200
400
600
800
1000
1200
$\|r_{GMRES}\|_2$

Theorem 6.2

Let A be positive real, i.e. the symmetric part $M := 0.5(A + A^T)$ of A is positive definite.

Then it holds

$$\|r^k\|_2 \leq \left(1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\max}(A^T A)}\right)^{k/2} \|r^0\|_2.$$

Here $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denotes the minimal and maximal eigenvalue, respectively.

Proof

By construction

$$\|r^k\|_2 = \min_{q\in\Pi_k,\ q(0)=1} \|q(A)r^0\|_2,$$

and for every $\alpha \in \mathbb{R}$ we get with the polynomial $\tilde{q}(\lambda) = (1+\alpha\lambda)^k$

$$\min_{q\in\Pi_k,\ q(0)=1} \|q(A)\|_2 \le \|\tilde{q}(A)^k\|_2 \le \|\tilde{q}(A)\|_2^k.$$

From $x^TMx \ge \lambda_{\min}(M)x^Tx$ and $x^TA^TAx \le \lambda_{\max}(A^TA)x^Tx$ for every $x \ne 0$ we obtain for every $\alpha \in \mathbb{R}$

$$\begin{aligned}\|\tilde{q}(A)\|_2^2 &= \max_{\|x\|_2=1} x^T(I+\alpha A)^T(I+\alpha A)x \\ &= \max_{\|x\|_2=1} (1+\alpha x^T(A+A^T)x + \alpha^2 x^TA^TAx)\end{aligned}$$

Proof ct.

and for $\alpha < 0$

$$\|\tilde{q}(A)\|_2^2 \leq 1 + 2\alpha\lambda_{\min}(M) + \alpha^2\lambda_{\max}(A^TA).$$

The quadratic expression on the right hand side attains its minimum for $\alpha = -\lambda_{\min}(M)/\lambda_{\max}(A^TA)$.

Hence,

$$\|\tilde{q}(A)\|_2 \leq \left(1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\max}(A^TA)}\right)^{1/2},$$

from which we get the statement. □

Theorem 6.3

Let $A = T\Lambda T^{-1}$ be diagonalizable with a regular matrix T and a diagonal matrix Λ.

Then it holds

$$\|r^k\|_2 \leq \kappa_2(T)\|r^0\|_2 \cdot \min_{q\in\Pi_k,\ q(0)=1} \max_{\lambda\in\sigma(A)} |q(\lambda)|.$$

If A is normal, then

$$\|r^k\|_2 \leq \|r^0\|_2 \cdot \min_{q\in\Pi_k,\ q(0)=1} \max_{\lambda\in\sigma(A)} |q(\lambda)|.$$

Here $\sigma(A) = \{\lambda_1, \ldots, \lambda_n\}$ denotes the spectrum of A.

Proof

With $\tilde{\Pi}_k := \{q \in \Pi_k \,:\, q(0) = 1\}$ it holds

$$\begin{aligned}
\|r^k\|_2 &= \min_{q\in\tilde{\Pi}_k} \|q(A)r^0\|_2 \\
&= \min_{q\in\tilde{\Pi}_k} \|q(T\Lambda T^{-1})r^0\|_2 \\
&= \min_{q\in\tilde{\Pi}_k} \|T\,q(\Lambda)T^{-1}r^0\|_2 \\
&\le \|T\|_2 \cdot \|T^{-1}\|_2 \min_{q\in\tilde{\Pi}_k} \|q(\Lambda)\|_2\|r^0\|_2 \\
&= \kappa_2(T)\|r^0\|_2 \min_{q\in\tilde{\Pi}_k} \max_{\lambda\in\sigma(A)} |q(\lambda)|.
\end{aligned}$$

If A is normal, then A can be transformed to diagonal form unitarily, and $\kappa_2(T) = 1$.

Remark

If A is normal, then (cf. Greenbaum & Gurvits 1994) the bound of the speed of convergence is sharp, i.e. for A there exists an initial residual r^0 such that the inequality holds with '='. In this case the speed of convergence depends on the spectrum of the matrix.

For nonnormal matrices it was shown by Greenbaum, Ptak and Strakos (1996), that for every monotonely decreasing sequence of positive real numbers there exists a matrix and an initial vector such that the residuals in the GMRES method attain the elements of the given sequence. The matrix can be determined such that it has predetermined eigenvalues.

Hence, the speed of convergence of the GMRES method is completely independent of the spectrum of the system matrix.

Example

The following example demonstrates that the GMRES may make progress only in the n-th step, although the spectrum may be clustered very well. Hence, the speed of convergence of GMRES for nonnormal matrices is independent of the spectrum of the system matrix.

Let

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{pmatrix}$$

be a Frobenius matrix. Assume that the initial vector is chosen such that $r^0 = e^1$.

Then Ar^0 is a multiple of e^n, $A^2 r^0$ is a linear combination of e^n and e^{n-1}, etc. All $A^k r^0$, $k = 1, \dots, n-1$ are orthogonal to r^0. Hence, the optimal approximation in $x^0 + \mathcal{K}_k(r^0, A)$ is $x^k = x^0$, $k = 1, \dots, n-1$, and only in the n-th step the solution is arrived. By choosing the a_js suitably every spectrum of A can be realized.

Remark

With this result in mind it is easy to construct an example for which CGNR clearly outperforms GMRES:

For $A \in \mathbb{R}^{10\times 10}$ as on the last slide where the roots of the polynomial

$$p(\lambda) = \lambda^{10} + \sum_{j=0}^{9} a_j \lambda^j$$

are

$$\lambda_j = 1 + 0.1 \exp(0.2\pi j i),\ j = 0, \ldots, 9,$$

$b = e^1 = (1, 0, \ldots, 0)^T$ and $x^0 = 0$ GMRES stagnates with $x^1 = \cdots = x^9 = 0$, $x^{10} = e^2$, whereas CGNR reduces the Euclidean norm of the residual to $4.1e-11$ within 3 steps.

Constructing a problem with given convergence curve

Following Greenbaum, Ptak & Strakos (1996) we construct a problem with a given convergence curve and any prescribed eigenvalues.

Let $x^0 = 0$ and $r^0 = b$, and let $\mathcal{W} = \{w^1, \ldots, w^n\}$ be an orthonormal basis of the Krylov space $\mathcal{K}_n(b, A)$ such that $\text{span}\{w^1, \ldots, w^k\} = \mathcal{K}_k(b, A)$ for $j = 1, \ldots, n$.

From the minimization property

$$\|r^k\|_2 = \min\{\|r^0 - u\| \ : \ u \in A\mathcal{K}_k(r^0, A)\}$$

it is clear that b can be expanded as

$$b = \sum_{j=1}^{n} \langle b, w^j \rangle w^j,$$

where $|\langle b, w^j \rangle| = \sqrt{\|r^{j-1}\|^2 - \|r^j\|^2}$, $r^0 = b$, $\|r^n\| = 0$.

Constructing a problem with . . . ct.

Given a nonincreasing positive sequence $f(0) \geq f(1) \geq \cdots \geq f(n-1) > 0$, define $f(n) = 0$, and

$$g(k) = \sqrt{f(k-1)^2 - f(k)^2}, \quad k = 1, 2, \ldots, n.$$

The conditions $\|b\| = f(0)$, $\|r^j\| = f(j)$, $j = 1, 2, \ldots, n-1$ will then be satisfied if the coordinates of b in the basis $\mathcal{W}$ are determined by the prescribed sequence of residual norms

$$W^H b = (g(1), \ldots, g(n))^T, \quad W = [w^1, w^2, \ldots, w^n].$$

Constructing a problem with . . . ct.

Let $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$, $\lambda_j \neq 0$, $j = 1, \ldots, n$ be a set of points in the complex plain. Consider the monic polynomial

$$\prod_{j=1}^{n}(\lambda - \lambda_j) =: \lambda^n - \sum_{j=0}^{n-1} \alpha_j \lambda^j.$$

Clearly, $\alpha_0 \neq 0$.

We construct the matrix A as a representation of a linear operator mapping $\mathcal{A} : \mathbb{C}^n \to \mathbb{C}^n$ with respect to the standard basis $e^1, \ldots, e^n$.

$\mathcal{A}$ is uniquely determined by its values on any basis vectors.

Constructing a problem with . . . ct.

Let $\mathcal{V} = \{v^1, \ldots, v^n\}$ be any orthonormal basis of $\mathbb{C}^n$, let $V = [v^1, \ldots, v^n]$, and let b satisfy

$$V^H b = (g(1), \ldots, g(n))^T.$$

Note that given b with $\|b\| = f(0)$, V can be chosen or, alternatively, given V, b can be chosen.

Since $g(n) \neq 0$, the set of vectors $\mathcal{B} = \{b, v^1, \ldots, v^{n-1}\}$ is linearly independent, and also forms a basis of $\mathbb{C}^n$.

Let $B = [b, v^1, \ldots, v^{n-1}]$. Then the operator $\mathcal{A}$ is determined by the equations

$$\begin{aligned} \mathcal{A}b &:= v^1 \\ \mathcal{A}v^j &:= v^{j+1},\ j = 1, \ldots, n-2 \\ \mathcal{A}v^{n-1} &:= \alpha_0 b + \alpha_1 v^1 + \cdots + \alpha_{n-1} v^{n-1}. \end{aligned}$$

Constructing a problem with . . . ct.

The matrix representation of $\mathcal{A}$ with respect to the basis $\mathcal{B}$ is

$$\mathcal{A}^{\mathcal{B}} = \begin{pmatrix} 0 & \dots & 0 & \alpha_0 \\ 1 & & 0 & \alpha_1 \\ & \ddots & \vdots & \vdots \\ & & 1 & \alpha_{n-1} \end{pmatrix}$$

which is the companion matrix corresponding to the set of eigenvalues Λ.

Changing the basis from $\mathcal{B}$ to the standard basis $\{e^1, \dots, e^n\}$ yields the desired matrix

$$A = B\mathcal{A}^{\mathcal{B}}B^{-1}.$$

Restarted GMRES

The main disadvantage of GMRES is that the cost and the storage requirements grow linearly with the number of iterations.

The usual way to cope with this problem is restarting. After a given number m of steps the current least squares problem is solved, and the solution x^m is chosen to be the initial vector for another m GMRES steps. This method called GMRES(m) is repeated until convergence.

There are no rules for choosing m. If m is chosen too large the method can become too expensive, and if m is chosen too small the convergence can become very slow, or the method can even stagnate without convergence.

Example

For

$$A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \; b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; x^0 = 0,$$

$x^1 = x^0$ solves the least squares problem

$$\|Ax - b\|_2 = \min!, \quad x = x^0 + \alpha b.$$

Hence, GMRES(1) stagnates.

Convergence of GMRES(m)

For positive real A we get from the first convergence theorem for GMRES(m) the residual bound

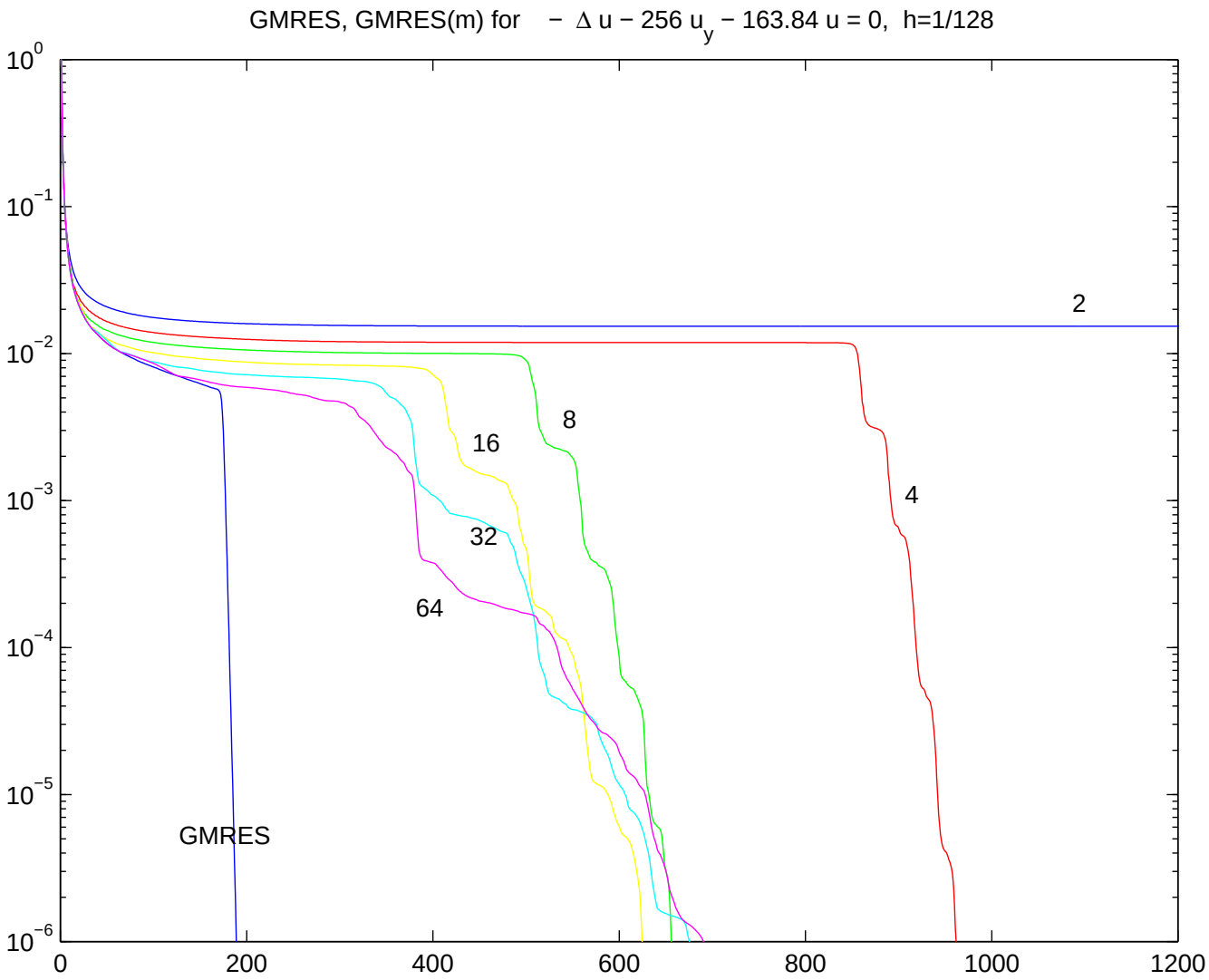
$$\|r^{jm+i}\|_2 \leq \left(1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\max}(A^TA)}\right)^{i/2} \|r^{jm}\|_2,$$

$$\|r^{jm}\|_2 \leq \left(1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\max}(A^TA)}\right)^{m/2} \|r^{(j-1)m}\|_2,$$

and therefore

$$\|r^{k}\|_2 \leq \left(1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\max}(A^TA)}\right)^{k/2} \|r^{0}\|_2.$$

Example


GMRES, GMRES(m) for − Δ u − 256 u_y − 163.84 u = 0, h=1/128
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
0
200
400
600
800
1000
1200
2
4
8
16
32
64
GMRES

Example

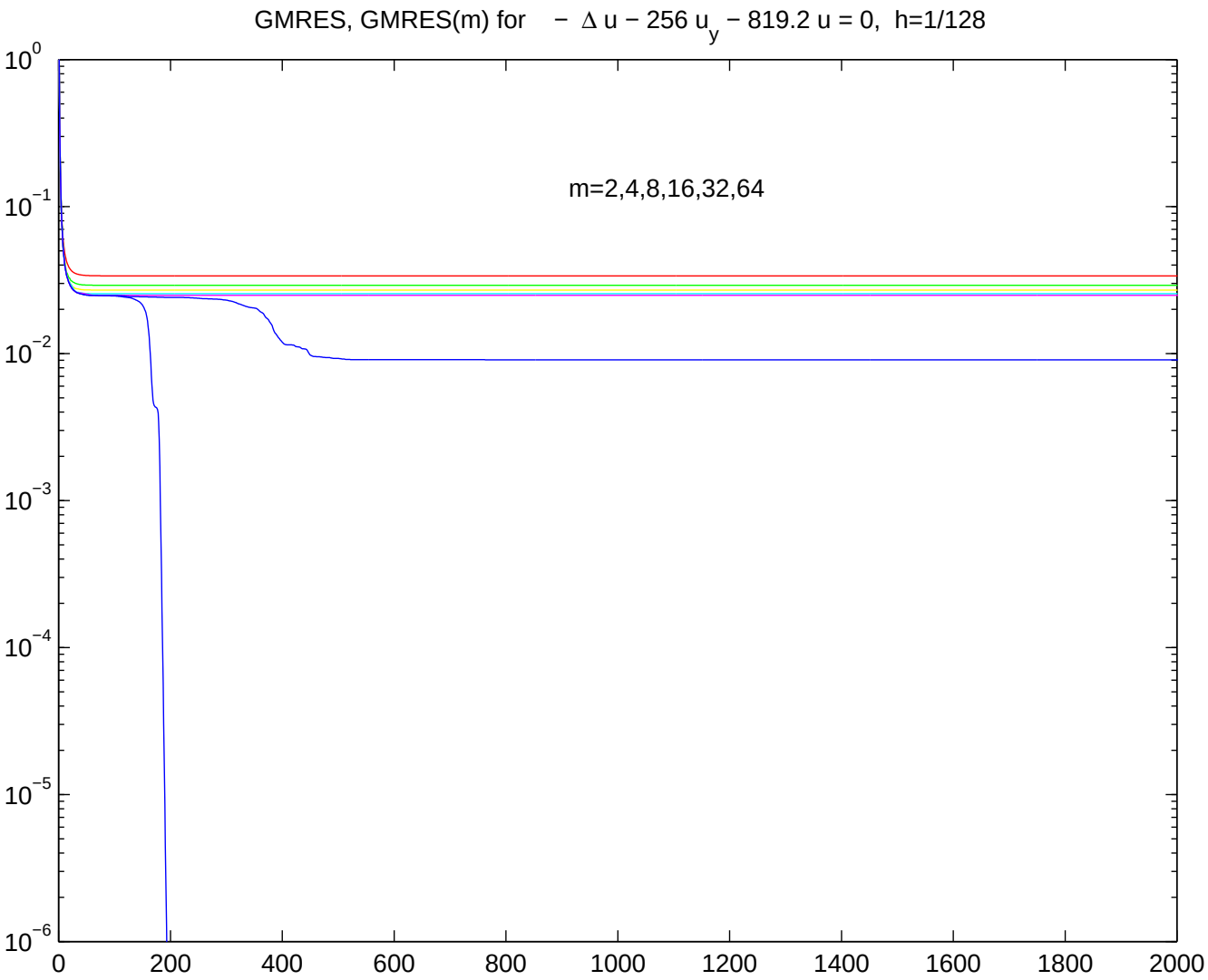
For

$$-\Delta u - 128u_y - 163.84u = 0, \quad h = 1/128,$$

we get

Type	iterations	flops
GMRES	190	1.20e9
GMRES(2)	stationary	
GMRES(4)	964	4.39e8
GMRES(8)	656	3.66e8
GMRES(16)	624	5.01e8
GMRES(32)	704	9.24e8
GMRES(64)	704	1.65e9

Example


GMRES, GMRES(m) for − Δ u − 256 u_y − 819.2 u = 0, h=1/128
m=2,4,8,16,32,64
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
0
200
400
600
800
1000
1200
1400
1600
1800
2000

Preconditioning

Again GMRES and GMRES(m) can be preconditioned applying the methods to the system

$$M^{-1}Ax = M^{-1}b \qquad \text{left preconditioning}$$

or to

$$AM^{-1}y = b,\ y = Mx \qquad \text{right preconditioning}$$

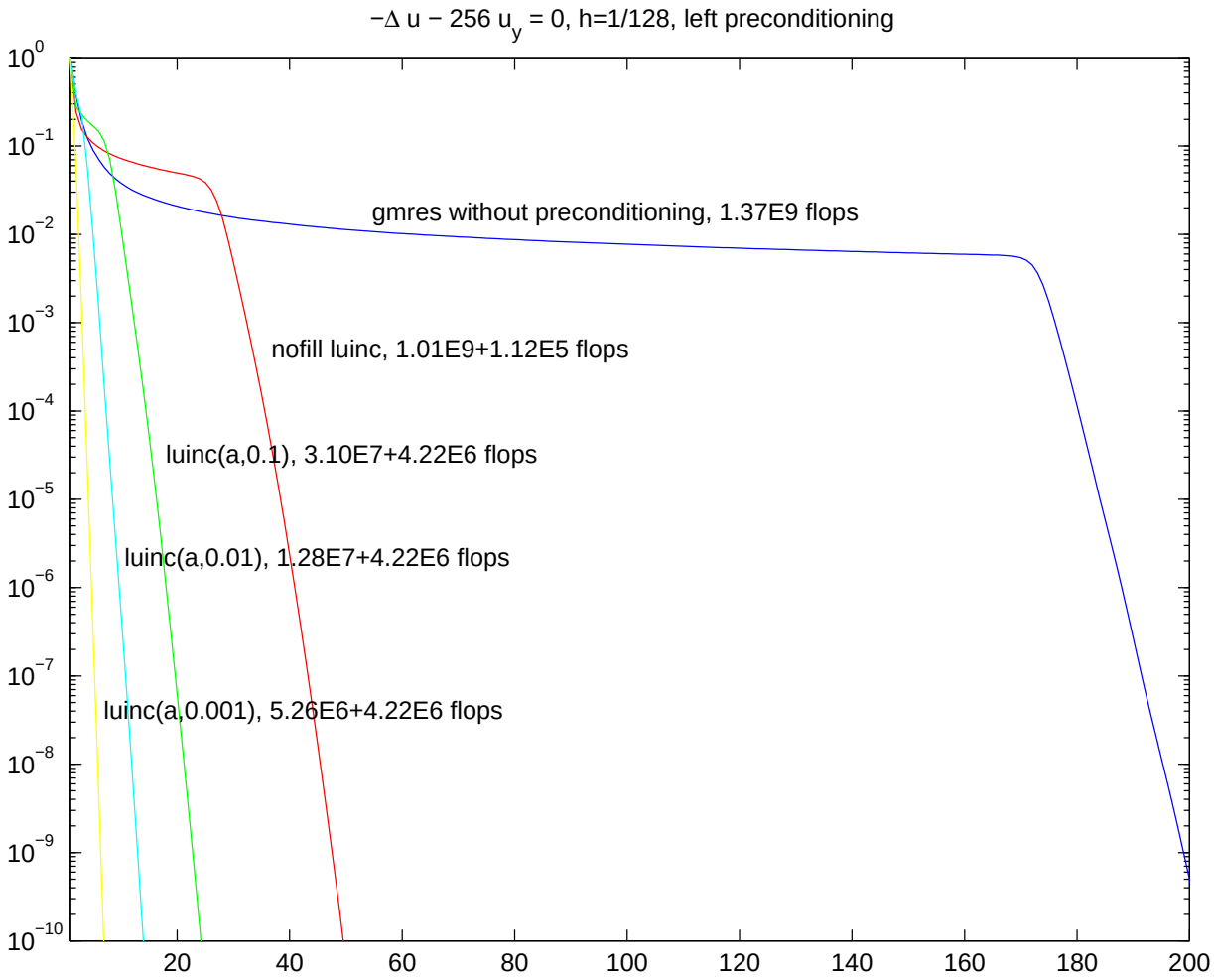
where in both cases M is an approximation to A such that linear systems $Mu = c$ can be solved easily.

In the following examples we used $M = LU$ where

$$[L, U] = \text{luinc}(A, \tau)$$

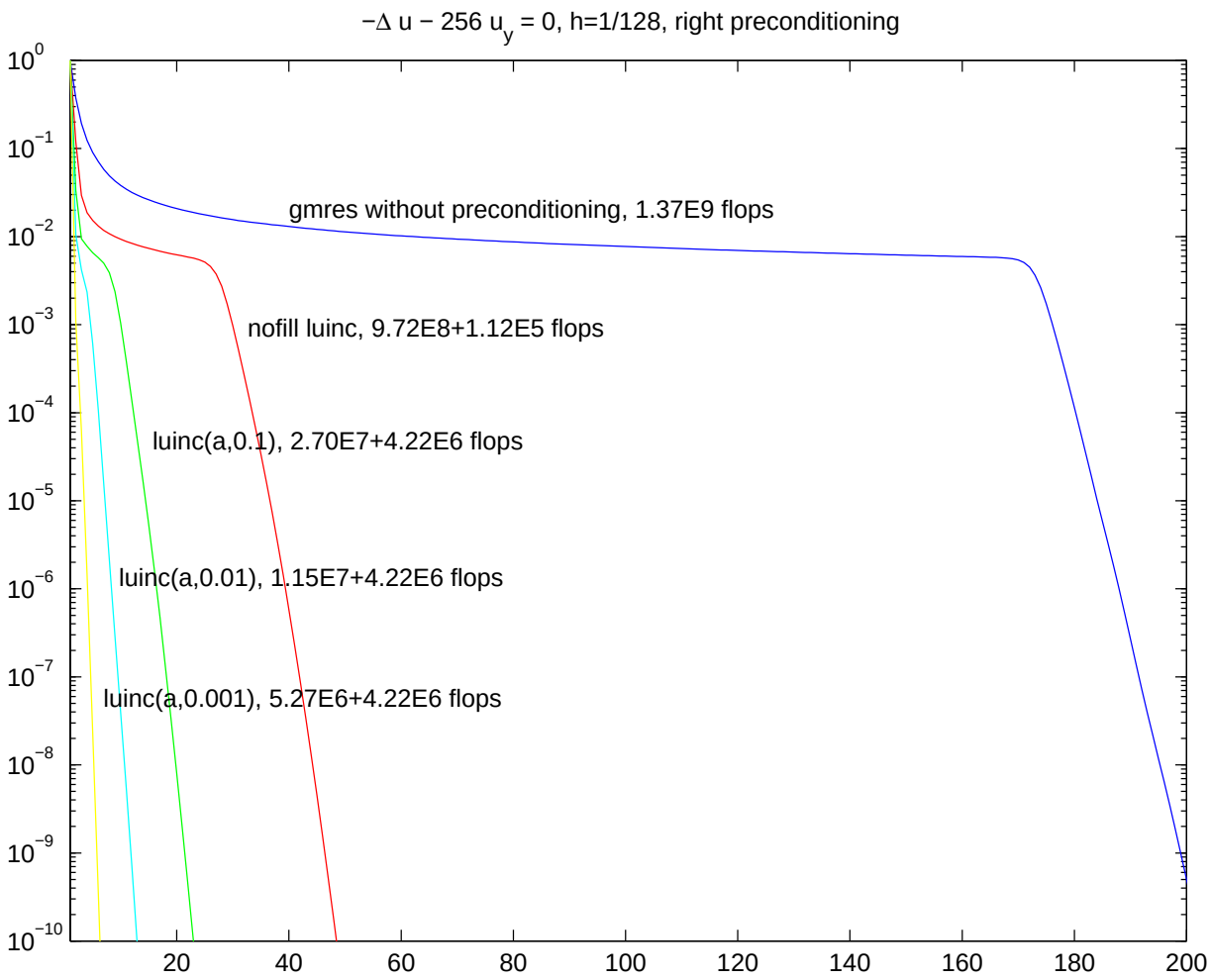
is an incomplete LU factorization of A with threshold τ.

Example



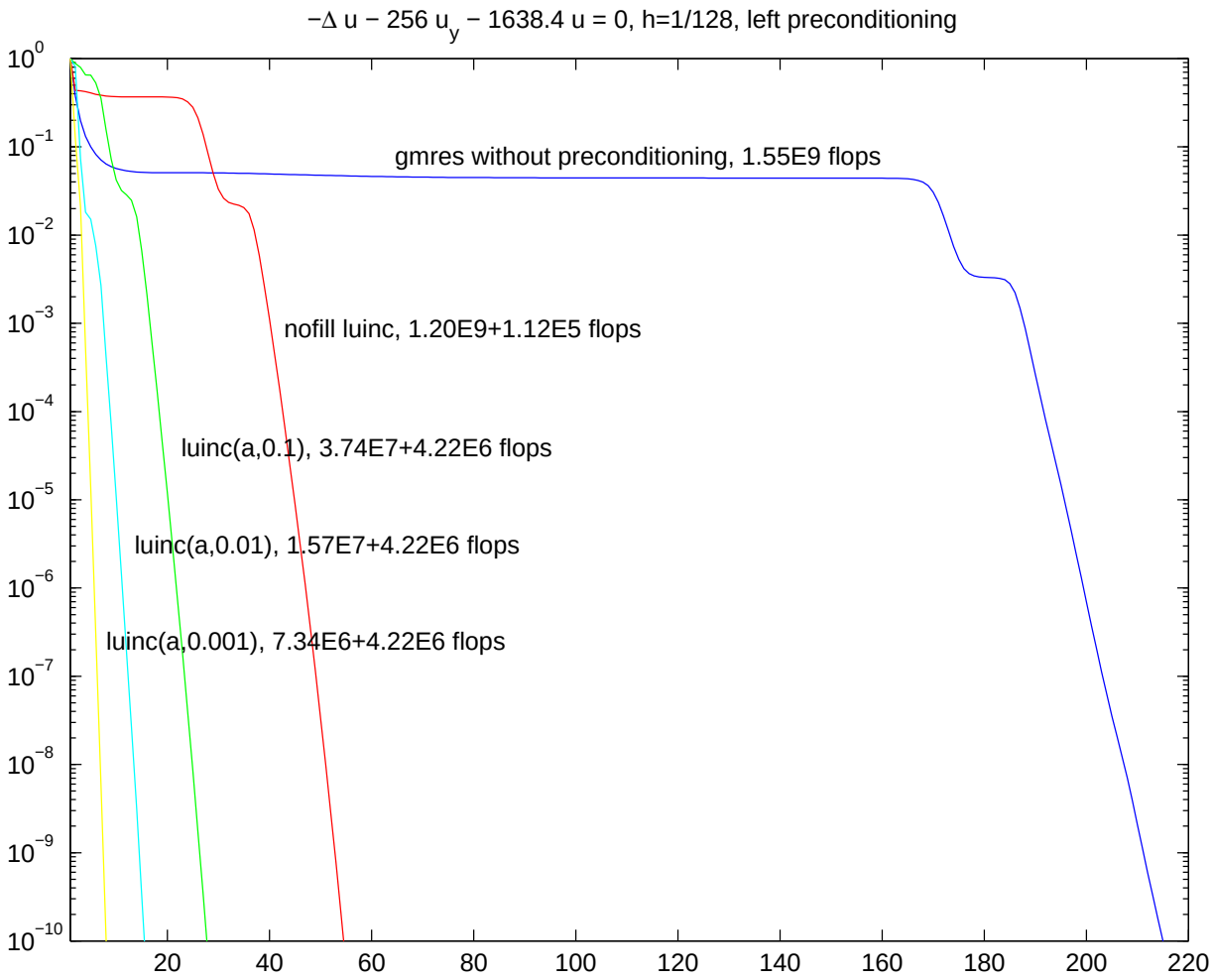
Direct solution using \: 1.825E7 flops

Example



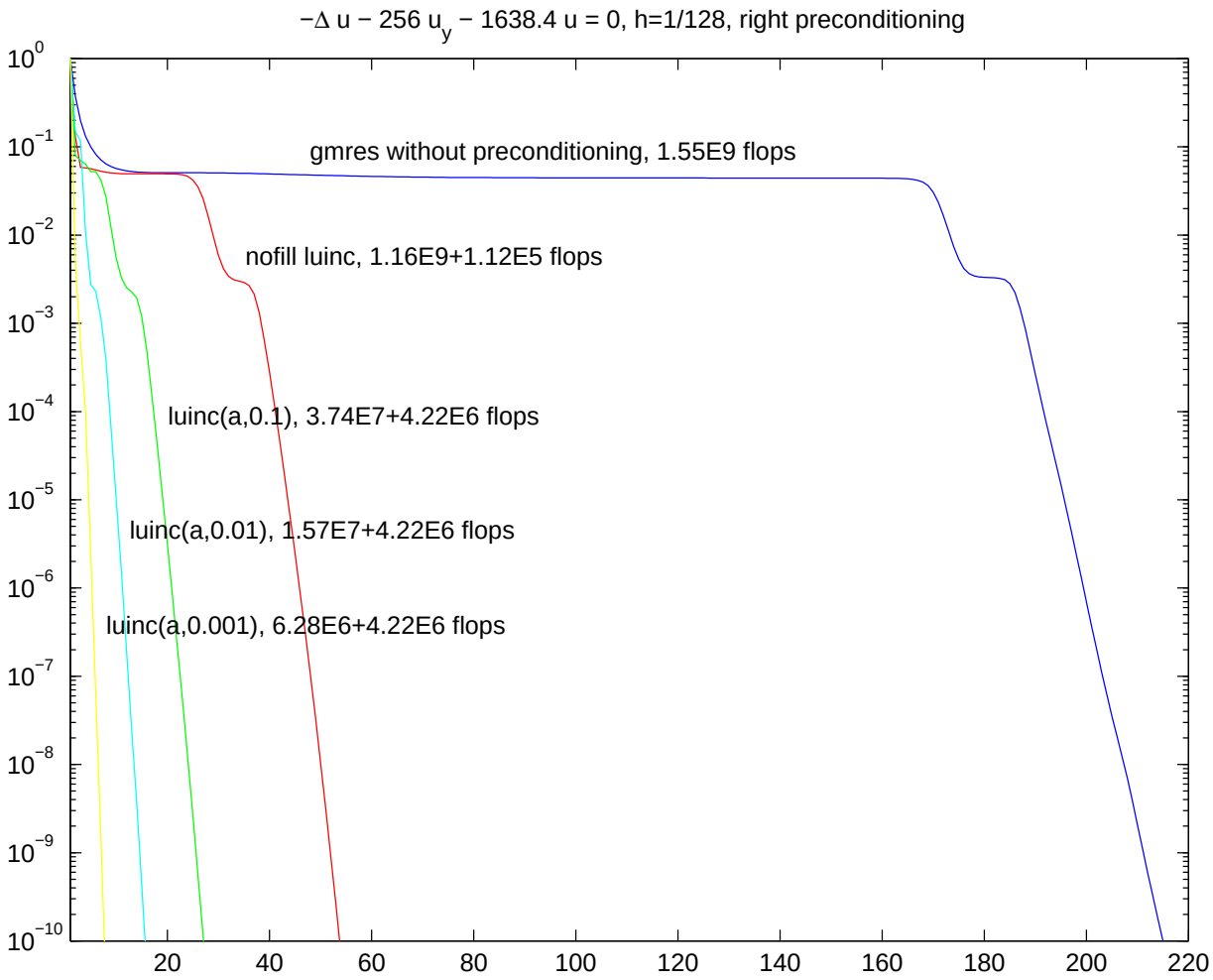
Direct solution using \: 1.825E7 flops

Example



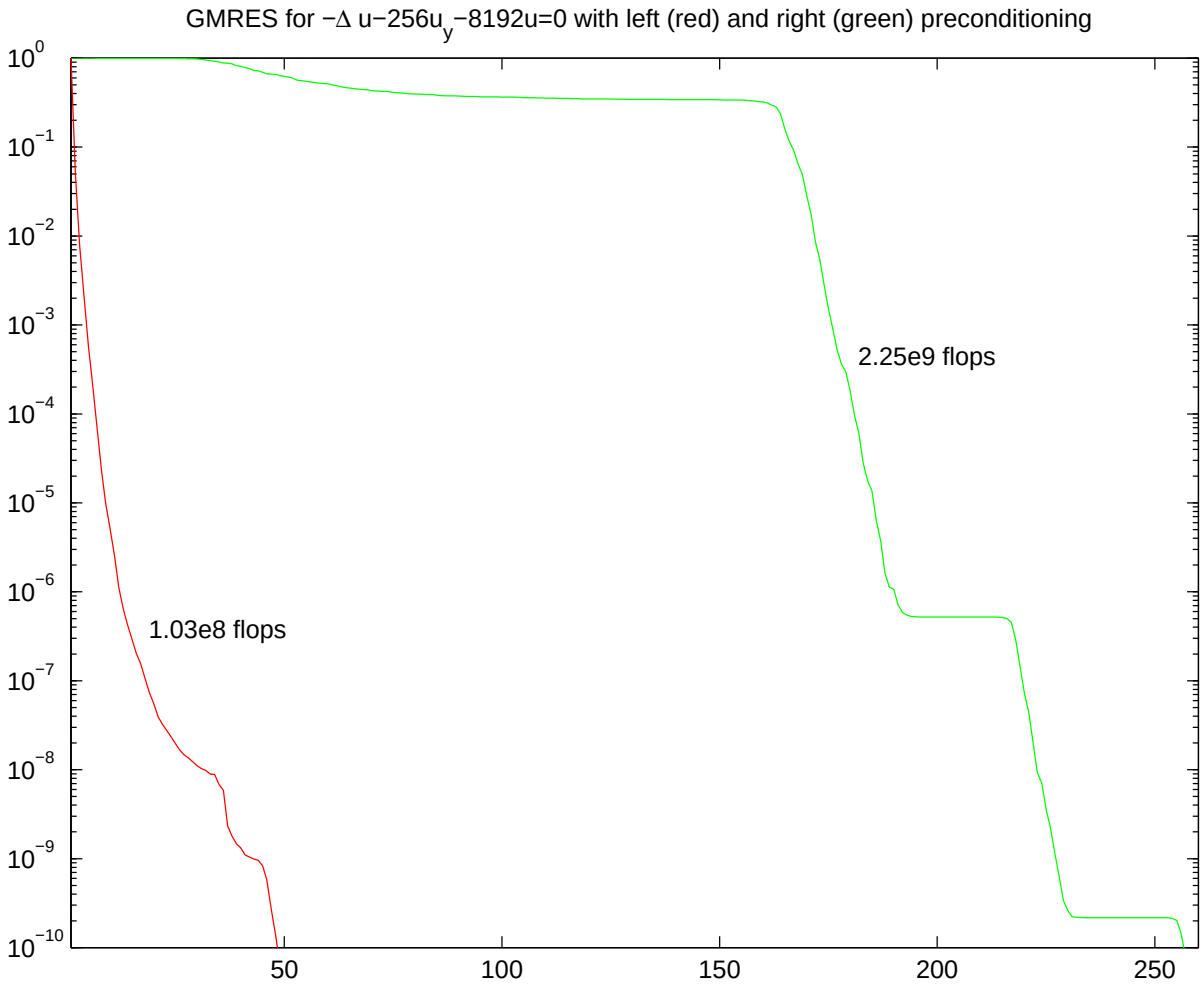
Direct solution using \: 1.825E7 flops

Example



Direct solution using \: 1.825E7 flops

Example



Direct solution using \: 1.825E7 flops