ITERATIVE PROJECTION METHODS FOR SPARSE LINEAR SYSTEMS AND EIGENPROBLEMS
CHAPTER 5 : INDEFINITE PROBLEMS

Heinrich Voss

voss@tu-harburg.de

Hamburg University of Technology
Institute of Numerical Simulation

TUHH
Technische Universität Hamburg-Harburg

Indefinite problems

Consider

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

where A is symmetric, but not necessarily positive definite.

Although problem (1) is not equivalent to the minimum problem

$$\phi(x) := \frac{1}{2} x^T A x - b^T x = \min!$$

the CG algorithm makes sense for equation (2)

Conjugate gradient method

$r^0 := b - Ax^0$;
$\alpha_0 := (r^0)^T r^0$.
$d^1 = r^0$;
for $k = 1, 2, \ldots$ until $d^k == 0$ **do**
  $s^k = Ad^k$;
  $\tau_k = \alpha_{k-1} / (d^k)^T s^k$;
  $x^k = x^{k-1} + \tau_k d^k$;
  $r^k = r^{k-1} - \tau_k s^k$;
  $\beta_k = 1/\alpha_{k-1}$;
  $\alpha_k = (r^k)^T r^k$;
  $\beta_k = \beta_k \cdot \alpha_k$;
  $d^{k+1} = r^k + \beta_k d^k$;
end for

CG method ct.

If A is symmetric, but not positive definite, the CG method can break down with $(d^k)^T s^k = (d^k)^T A d^k = 0$ for some $k < n$.

As in the positive definite case the search directions satisfy $(d^j)^T A d^i = 0$ for $i \neq j$, and hence the d^j are linearly independent.

Thus, as long as it does not break down the CG method determines a basis of the Krylov space $\mathcal{K}_k(r^0, A)$, and the iterates $x^k = x^0 + \mathcal{K}_k(r^0, A)$ satisfy the Galerkin condition

$$(d^j)^T (Ax^k - b) = 0, \quad j = 1, \ldots, k.$$

This method can become unstable if the matrix

$$A_k := ((d^i)^T A d^j)_{i,j=1,\ldots,k}$$

is nearly singular.

CG method ct.

For $k \leq n$ let

$$\theta_1^{(k)} \leq \theta_2^{(k)} \leq \cdots \leq \theta_k^{(k)}$$

be the eigenvalues of the matrix A_k, the so called Ritz values of A with respect to $\mathcal{K}_k(r^0, A)$.

Since

$$\mathcal{K}_j(r^0, A) \subset \mathcal{K}_{j+1}(r^0, A),$$

$j = , \ldots, n-1$, the minmax-characterization of the eigenvalues of a symmetric matrix yields, that for fixed j the sequence of the Ritz values decreases monotonely and is bounded below by the j smallest eigenvalue of the system matrix A.

$$\theta_j^{(j)} \geq \theta_j^{(j+1)} \geq \theta_j^{(j+2)} \geq \cdots \geq \lambda_j.$$

CG method ct.

If the matrix A has exactly one negative eigenvalue (and if $(r^0)^T A r^0 > 0$) then there exists at most (in general exactly) one critical phase in the CG method, namely when the smallest Ritz value $\theta_1^{(k)}$ changes its sign.
Every other sequence $\{\theta_j^{(k)}\}$ is bounded below by $\lambda_j > 0$.

Correspondingly there exist at most (in general exactly) m critical phases in the CG method if the matrix A has m negative und $p \leq n - m \geq m$ positive eigenvalues.

CG method ct.

Consider

$$-\Delta u - 163.84u = 0 \text{ in } \Omega = (0,1) \times (0,1), \quad u = 0 \text{ on } \partial\Omega$$

Discretizing Δ with central differences with stepsize $h = 1/128$ yields a linear system

$$3.99U_{ij} - U_{i-1,j} - U_{i,j-1} - U_{i,j+1} - U_{i+1,j} = 0, \ i,j = 1,\ldots,127,$$

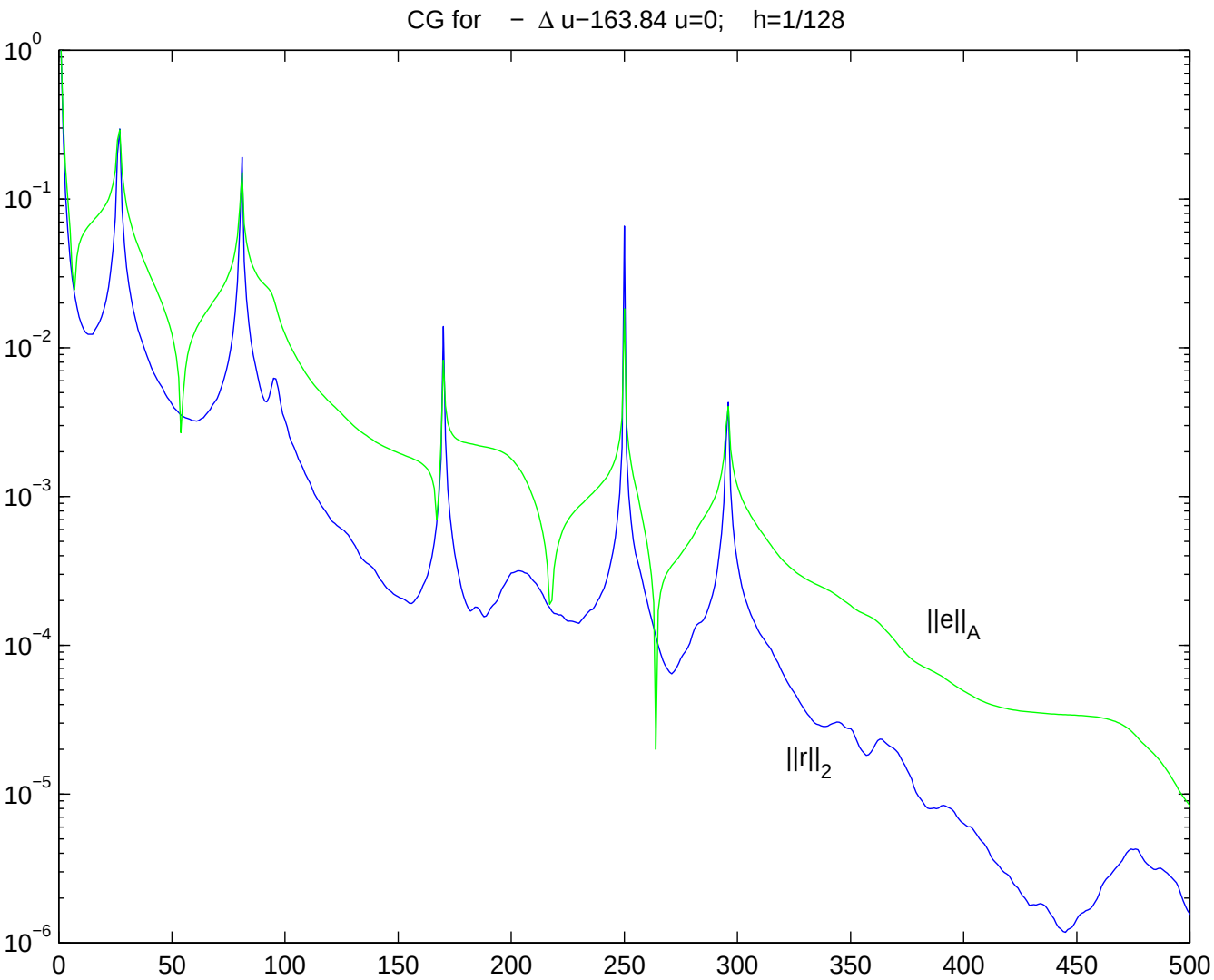
of dimension $n = 127^2 = 16129$.

The coefficient matrix has 8 negative eigenvalues

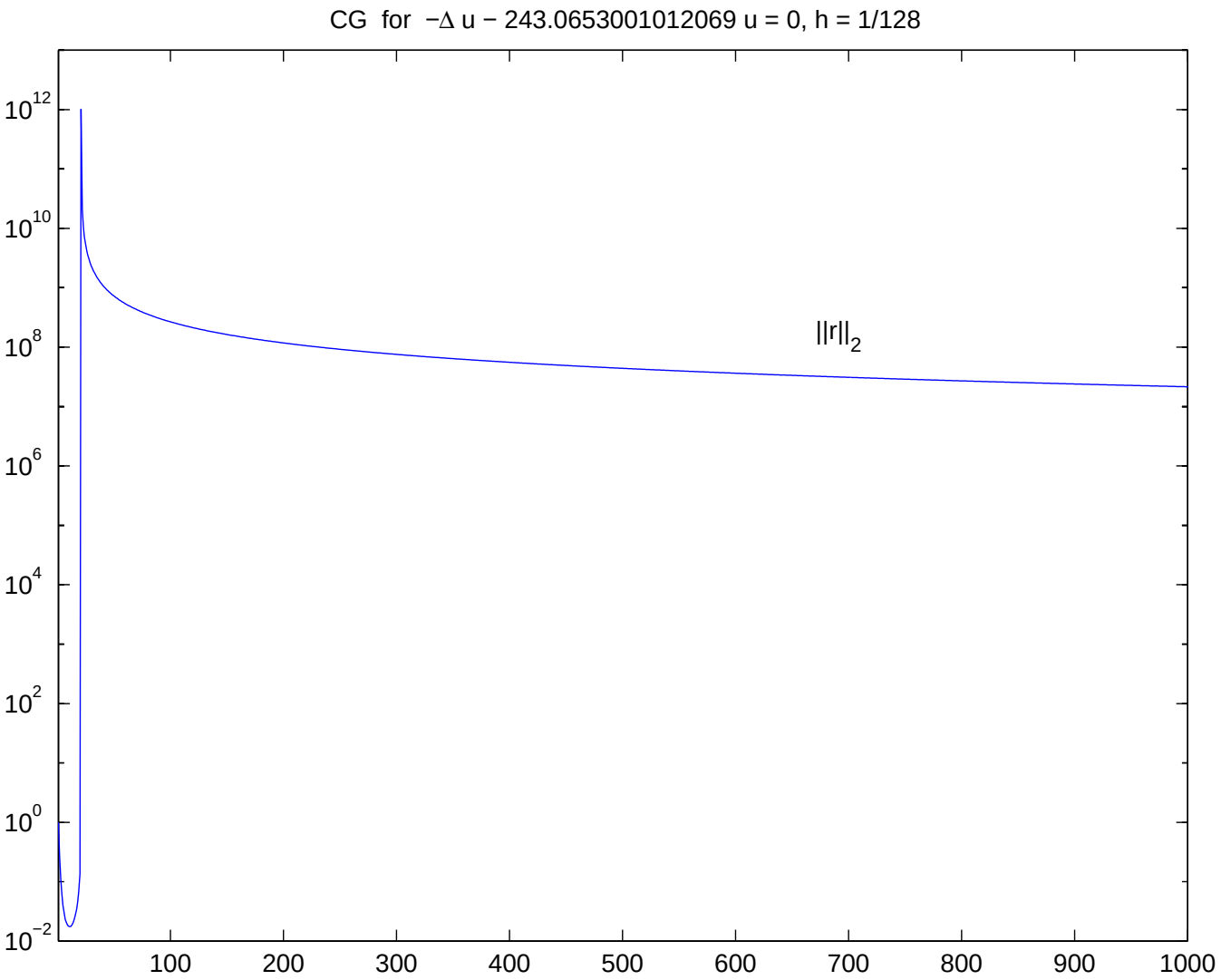
$$\begin{array}{ll} -8.795274784817014e-003 & -6.988549802753376e-003 \\ -6.988549802753343e-003 & -5.181824820689691e-003 \\ -3.978550749789028e-003 & -3.978550749789008e-003 \\ -2.171825767725394e-003 & -2.171825767725365e-003 \end{array}$$

5 of which are distinct.

Example


CG for − Δ u−163.84 u=0; h=1/128
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0
50
100
150
200
250
300
350
400
450
500
$\|e\|_A$
$\|r\|_2$

Example


CG for −Δ u − 243.0653001012069 u = 0, h = 1/128
10^{12}
10^{10}
10^{8}
10^{6}
10^{4}
10^{2}
10^{0}
10^{-2}
100
200
300
400
500
600
700
800
900
1000
$||r||_2$

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 (cf. Chapter 4).

Lanczos method ct.

$q^0 = 0$; $k = 1$
$\beta_0 = \|r^0\|$
while $r^{k-1} \neq 0$ **do**
  $q^k = r^{k-1}/\beta_{k-1}$
  $r^k = Aq^k$
  $r^k = r^k - \beta_{k-1} q^{k-1}$
  $\alpha_k = (q^k)^T r^k$
  $r^k = r^k - \alpha_k q^k$
  $\beta_k = \|r^k\|$
end while

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

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

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 ϕ 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.

Lanczos method ct.

T_k is SPD. Hence, it allows an LDL^T factorization $T_k = L_k D_k L_k^T$,

$$D_k = \operatorname{diag}\{d_1, \ldots, d_k\},\ d_j > 0,$$

$$L_k = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ \mu_2 & 1 & 0 & \ldots & 0 & 0 \\ 0 & \mu_3 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \mu_k & 1 \end{pmatrix}$$

Comparing the elements in $T_k = L_k D_k L_k^T$ yields

$d_1 = \alpha_1$;
for $j = 2 : k$
$\mu_j = \beta_{j-1}/d_{j-1}$;
$d_j = \alpha_j - \beta_{j-1}\mu_j$;
end

Lanczos method ct.

With

$$z^k := L_k^T y^k, \quad C_k := Q_k L_k^{-T}$$

it holds

$$L_k D_k z^k = Q_k^T b, \quad x^k = C_k z^k,$$

and

$$C_k L_k^T = (c^1, \mu_2 c^1 + c^2, \ldots, \mu_k c^{k-1} + c^k) = (q^1, \ldots, q^k) = Q_k.$$

Since $L_k D_k$ is a lower triangular matrix and since $Q_k^T b = \|b\|_2 e^1$, one obtains the solution of

$$L_k D_k z^k = \|b\|_2 e^1$$

from the solution z^{k-1} of $L_{k-1} D_{k-1} z^{k-1} = \|b\|_2 e^1$ just adding the last component

$$\zeta_k = -\mu_k d_{k-1} \zeta_{k-1} / d_k.$$

CG with Lanczos process

1: $r^0 = b - Ax^0$; $q^0 = 0$; $c^0 = 0$; $\beta_0 = \|r^0\|_2$;
2: **for** $k = 1, 2, \ldots$ until convergence **do**
3: $\quad q^k = r^{k-1}/\beta_{k-1}$
4: $\quad r^k = Aq^k - \beta_{k-1} q^{k-1}$
5: $\quad \alpha_k = (q^k)^T r^k$
6: $\quad r^k = r^k - \alpha q^k$
7: $\quad \beta_k = \|r^k\|_2$
8: $\quad$ **if** $k == 1$ **then**
9: $\quad\quad d_1 = \alpha_1$; $\zeta_1 = \beta_0/d_1$; $c^1 = q^1$;
10: $\quad$ **else**
11: $\quad\quad \mu_k = \beta_{k-1}/d_{k-1}$; $d_k = \alpha_k - \beta_{k-1}\mu_k$; $\zeta_k = -\mu_k d_{k-1}\zeta_{k-1}/d_k$;
12: $\quad\quad c^k = q^k - \mu_k c^{k-1}$;
13: $\quad$ **end if**
14: $\quad x^k = x^{k-1} + \zeta_k c^k$;
15: **end for**

Conclusion

The CG approximations x^k can be obtained by the Lanczos method.

Cost of CG with Lanczos:
1 matrix-vector products
2 scalar products
5 _axpy

Storage requirements: 4 vectors

Indefinite problems

Lanczos' method does not need the definiteness of A, hence the projected system

$$T_k y^k = Q_k^T A Q_k y^k = \|b\|_2 e^1 \qquad (1)$$

can be determined in a stable way.

The LDL^T factorization of T_k does not necessarily exist and can not be computed in a stable way.

Paige & Saunders (1975): Use the LQ factorization

$$T_k = \tilde{L}_k U_k, \quad U_k \text{ orthogonal}, \quad \tilde{L}_k \text{ lower triangular.}$$

Indefinite problems ct.

Determine U_k as a product of Givens reflections

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

where

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

Indefinite problems ct.

$$
\begin{aligned}
T_k U_k^T &= \begin{pmatrix} T_{k-1} & \beta_{k-1} e^{k-1} \\ \beta_{k-1}(e^{k-1})^T & \alpha_k \end{pmatrix} \begin{pmatrix} U_{k-1}^T & 0 \\ 0^T & 1 \end{pmatrix} G_k \\
&= \begin{pmatrix} T_{k-1} U_{k-1}^T & \beta_{k-1} e^{k-1} \\ \beta_{k-1}(U_{k-1} e^{k-1})^T & \alpha_k \end{pmatrix} G_k \\
&= \begin{pmatrix} \tilde{L}_{k-1} & \beta_{k-1} e^{k-1} \\ \beta_{k-1}(e^{k-1})^T U_{k-1}^T & \alpha_k \end{pmatrix} G_k,
\end{aligned}
$$

where G_k is chosen such that β_{k-1} at the position $(k-1, k)$ is annihilated.

$$
\begin{aligned}
\beta_{k-1}(e^{k-1})^T U_{k-1}^T &= \beta_{k-1}(e^{k-1})^T \begin{pmatrix} U_{k-2}^T & 0 \\ 0^T & 1 \end{pmatrix} G_{k-1} \\
&= \beta_{k-1}(e^{k-1})^T G_{k-1} = (0, \ldots, 0, \beta_{k-1} s_{k-1}, -\beta_{k-1} c_{k-1}).
\end{aligned}
$$

Indefinite problems ct.

Multiplying a matrix with G_k from the right only the last two columns are combined.

Hence, in $\tilde{L}_k = T_k U_k^T$ only 3 diagonals are different from zero, i.e. $\tilde{L}_k$ has the form

$$\tilde{L}_k = \begin{pmatrix} \gamma_1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \delta_1 & \gamma_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ \varepsilon_1 & \delta_2 & \gamma_3 & 0 & \dots & 0 & 0 & 0 \\ 0 & \varepsilon_2 & \delta_3 & \gamma_4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \varepsilon_{k-2} & \delta_{k-1} & \tilde{\gamma}_k \end{pmatrix}$$

Indefinite problems ct.

In the transition from $\tilde{L}_k$ to

$$\begin{aligned}
\tilde{L}_{k+1} &= T_{k+1} \begin{pmatrix} U_k^T & 0 \\ 0^T & 1 \end{pmatrix} G_{k+1} \\
&= \begin{pmatrix} \tilde{L}_k & \beta_k e^k \\ \beta_k (e^k)^T U_k^T & \alpha_k \end{pmatrix} G_{k+1} \\
&= \begin{pmatrix} L_k & 0 \\ 0, \ldots, 0, \varepsilon_{k-1}, \delta_k & \tilde{\gamma}_{k+1} \end{pmatrix}
\end{aligned}$$

in the leading principal (k, k) matrix $\tilde{L}_k$ only the diagonal element $\tilde{\gamma}_k$ is changed.

Indefinite problems ct.

With

$$\tilde{W}_k := (w^1, \ldots, w^{k-1}, \tilde{w}^k) := Q_k U_k^T$$

and

$$\tilde{z}_k := (\zeta_1, \ldots, \zeta_{k-1}, \tilde{\zeta}_k)^T := U_k y^k$$

the linear system

$$T_k y^k = \|b\|_2 e^1, \quad x^k = Q_k y^k$$

can be rewritten as

$$\tilde{L}_k \tilde{z}_k = \|b\|_2 e^1, \quad x^k = \tilde{W}_k \tilde{z}^k.$$

(As in the computation of the CG approximation with the Lanczos method for the SPD case) this decomposition can be used to solve the projected problem.

Indefinite problems ct.

For $k = 2$ the solution $\tilde{z}^2$ is

$$\zeta_1 = \|b\|_2/\gamma_1, \quad \tilde{\zeta}_2 = -\delta_1\zeta_1/\tilde{\gamma}2.$$

If $\tilde{z}^{k-1}$ is already known, the leading $k-2$ components of $\tilde{z}^k$ coincide with those of $\tilde{z}^{k-1}$, and it holds that

$$\begin{aligned} \zeta_{k-1} &= \tilde{\zeta}_{k-1}\tilde{\gamma}_{k-1}/\gamma_{k-1} = c_k\tilde{\zeta}_{k-1} \\ \tilde{\zeta}_k &= -(\varepsilon_{k-2}\zeta_{k-2} + \delta_{k-1}\zeta_{k-1})/\tilde{\gamma}_k. \end{aligned}$$

Indefinite problems ct.

From

$$\begin{aligned}
\tilde{W}_k &= (w^1,\dots,w^{k-1},\tilde{w}^k) = (q^1,\dots,q^k)U_k^T \\
&= (q^1,\dots,q^k)\begin{pmatrix} U_{k-1}^T & 0 \\ 0^T & 1 \end{pmatrix} G_k = ((q^1,\dots,q^{k-1})U_{k-1}^T, q^k)G_k \\
&= (w^1,\dots,w^{k-2},\tilde{w}^{k-1},q^k)\begin{pmatrix} I_{k-2} & 0 & 0 \\ 0^T & c_k & s_k \\ 0^T & s_k & -c_k \end{pmatrix}
\end{aligned}$$

we obtain w^{k-1} and $\tilde{w}^k$ in the k-th step from

$$(w^{k-1},\tilde{w}^k) = (\tilde{w}^{k-1},q^k)\begin{pmatrix} c_k & s_k \\ s_k & -c_k \end{pmatrix}, \quad \tilde{w}^1 = q^1.$$

Indefinite problems ct.

Although the LQ factorization is determined in a stable way (even for indefinite matrices) the algorithm should not be implemented directly since intermediate matrices $\tilde{L}_k$ may become singular or $|\tilde{\gamma}_k|$ can become very small as a Ritz value passes the origin.

Since $\gamma_k = \sqrt{\beta_k^2 + \tilde{\gamma}_k^2} \neq 0$ as long as $\beta_k \neq 0$ (i.e. es long as the Lanczos process did not detect an invariant subspace of A) the matrix L_k is nonsingular. Hence, the linear system

$$L_k z^k = \|b\|_2 e^1$$

for every k has a unique solution z^k, which can be determined in a more stable way than $\tilde{z}^k$.

Indefinite problems ct.

Paige and Saunders propose to update the vectors

$$\hat{x}^k := W_k z^k := \hat{x}^{k-1} + \zeta_k w^k$$

instead of x^k from which we can get the CG approximation

$$x^{k+1} = \hat{x}^k + \tilde{\zeta}_{k+1} \tilde{w}^{k+1}$$

after the method has converged.

This method is called SYMMLQ.

SYMMLQ

$c_1 = -1$; $s_1 = 0$; $c_0 = 1$; $s_0 = 0$; $\zeta_0 = -1$; $\zeta_{-1} = 0$;
$\hat{x}^0 = x^0$; $\tilde{w}^0 = 0$; $q^0 = 0$; $\tilde{q} = b - Ax^0$; $\beta_0 = \|\tilde{q}\|_2$; $q^1 = \tilde{q}/\beta_0$;
for $k = 1, 2, \ldots$ until convergence **do**
  $\tilde{q} = Aq^k - \beta_{k-1} q^{k-1}$
  $\alpha_k = \tilde{q}^T q^k$
  $\tilde{q} = \tilde{q} - \alpha_k q^k$
  $\beta_k = \|\tilde{q}\|_2$
  $\tilde{\gamma}_k = -c_k \alpha_k - c_{k-1} s_k \beta_{k-1}$; $\gamma_k = \sqrt{\tilde{\gamma}_k^2 + \beta_k^2}$
  $\delta_{k-1} = s_k \alpha_k - c_k c_{k-1} \beta_{k-1}$
  $\varepsilon_{k-2} = s_{k-1} \beta_{k-1}$
  $c_{k+1} = \tilde{\gamma}_k / \gamma_k$; $s_{k+1} = \beta_k / \gamma_k$
  $\tilde{w}^k = s_k \tilde{w}^{k-1} - c_k q^k$
  $\zeta_k = -(\varepsilon_{k-2}\zeta_{k-2} + \delta_{k-1}\zeta_{k-1})/\gamma_k$
  $q^{k+1} = \tilde{q}/\beta_k$
  $\hat{x}^k = \hat{x}^{k-1} + \zeta_k (c_{k+1} \tilde{w}^k + s_{k+1} q^{k+1})$
end for
$x^{k+1} = \hat{x}^k + \zeta_k \tilde{w}^k / c_{k+1}$

Cost of SYMMLQ

1 matrix-vector product
2 scalar products
7 _axpy

Storage requirements: 5 vectors

Termination

The iteration is terminated, if the norm of the residual is sufficiently small.

Although x^k is not determined in every step, the corresponding residual $r^k = b - Ax^k$ can be monitored during the iteration.

$$\begin{aligned} r^k &= b - Ax^k = \|b\|_2 q^1 - AQ_k y^k \\ &= \|b\|_2 q^1 - Q_k T_k y^k - \beta_k q^{k+1} (e^k)^T y^k = -\beta_k \eta_k q^{k+1}, \end{aligned}$$

where η_k denotes the k-th element of y^k.

BUT The vector y^k is not formed explicitly in the algorithm.

Termination

From

$$T_k = T_k^T = U_k^T \tilde{L}_k^T$$

and (*) ($T_k y^k = Q_k^T A Q_k y^k = \|b\|_2 e^1$) one obtains

$$\tilde{L}_k^T y^k = \|b\|_2 U_k e^1,$$

and comparing the last components of this equation we get

$$\tilde{\gamma}_k \eta_k = \|b\|_2 \cdot s_2 s_3 \ldots s_k.$$

Hence,

$$r^k = -\frac{1}{c_{k+1}} s_2 \cdot \ldots \cdot s_{k+1} \|b\|_2 q^{k+1},$$

and $\|r^k\|_2$ is directly available without forming x^k. □

Theorem 5.1

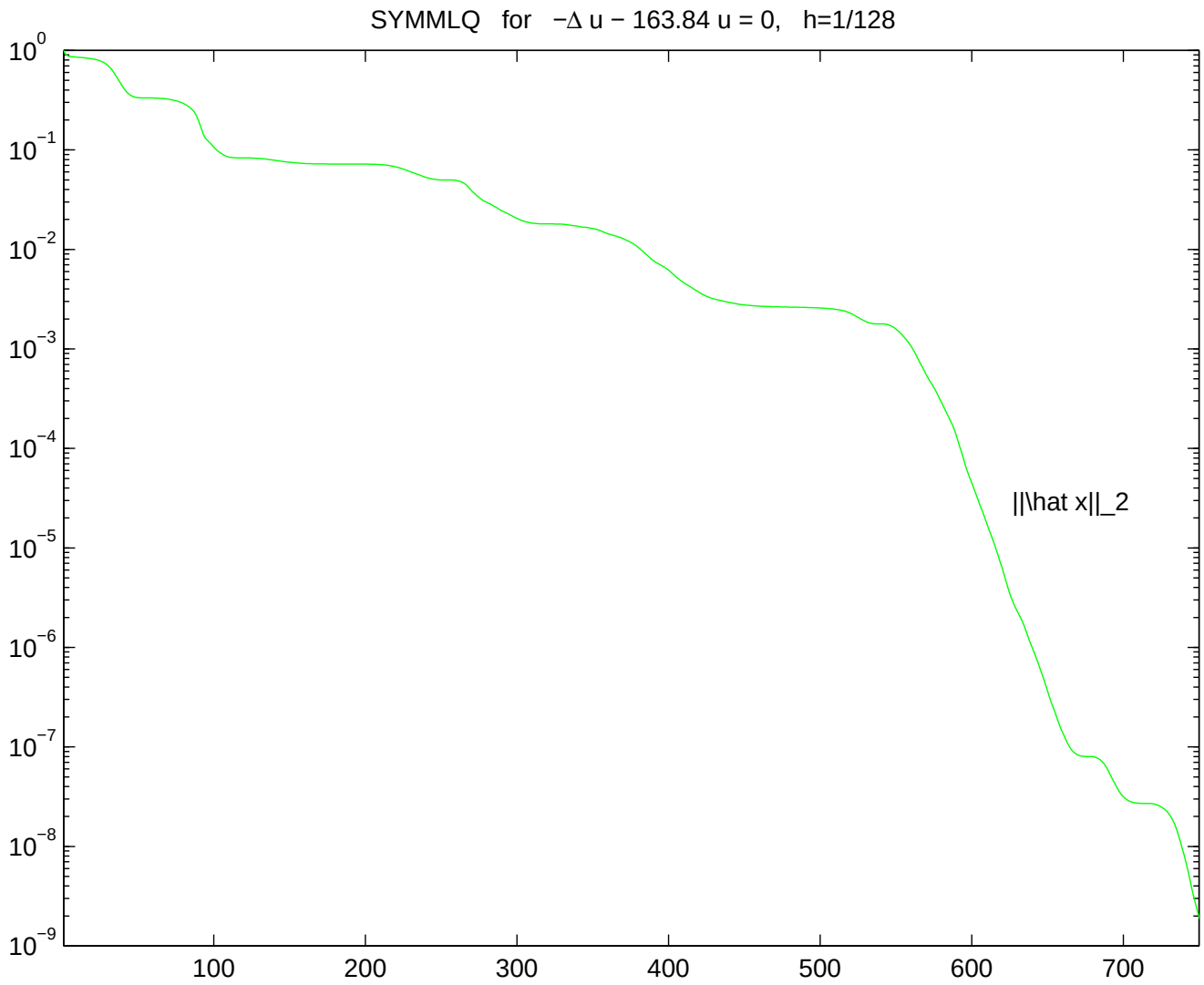
The approximation $\hat{x}^k$ is the unique solution of the minimization problem

$$\|e\|_2 = \min!,\ x \in A\mathcal{K}_k(b, A) = \text{span}\{Ab, A^2b, \ldots, A^kb\},$$

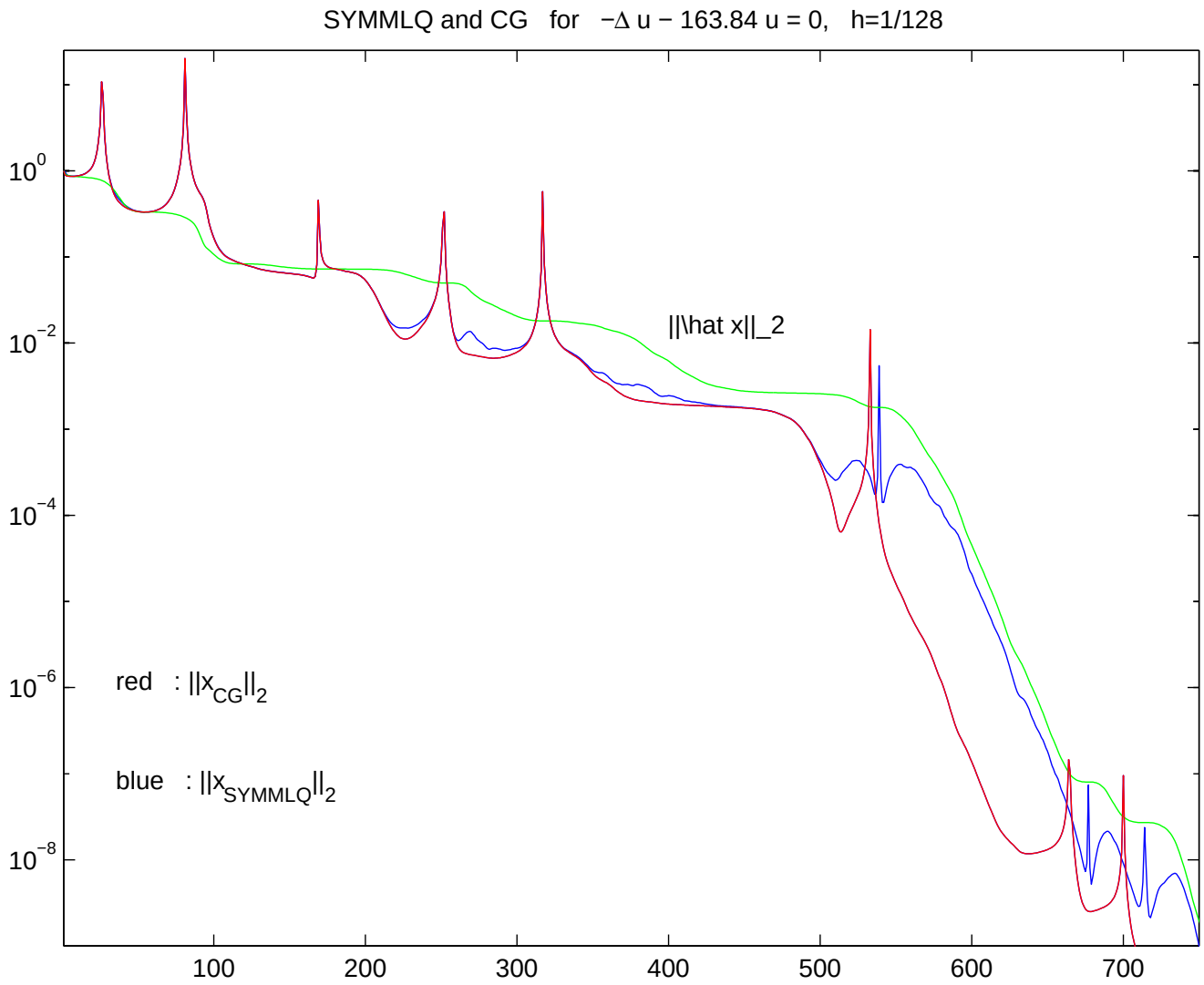
where $e := A^{-1}b - x$ denotes the error of x.

The CG iterates x^k do not minimize an error norm in the indefinite case. However, they are usually better approximations to the solution than the vectors $\hat{x}^k$.

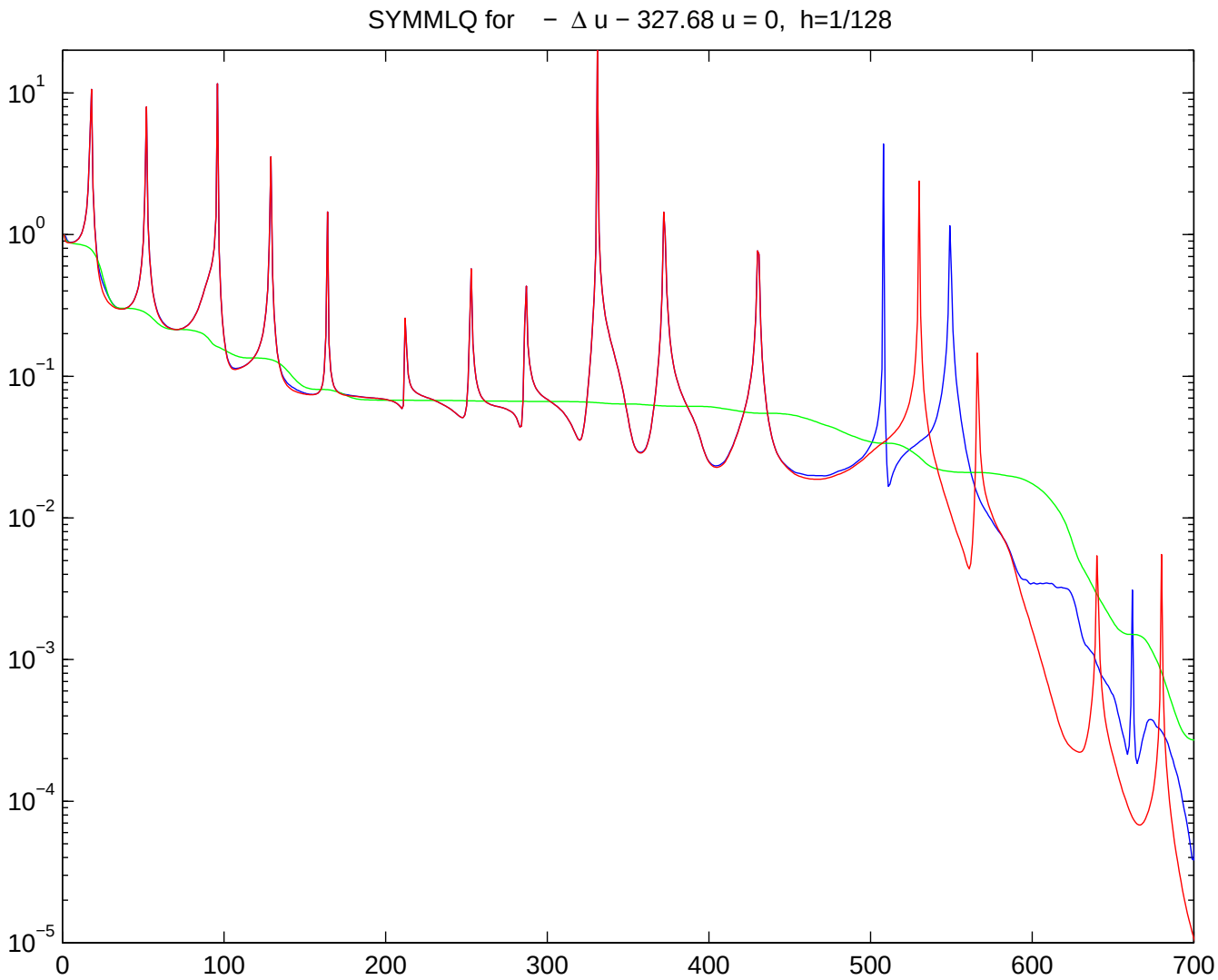
Example


SYMMLQ for −Δ u − 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
100
200
300
400
500
600
700
||\hat x||_2

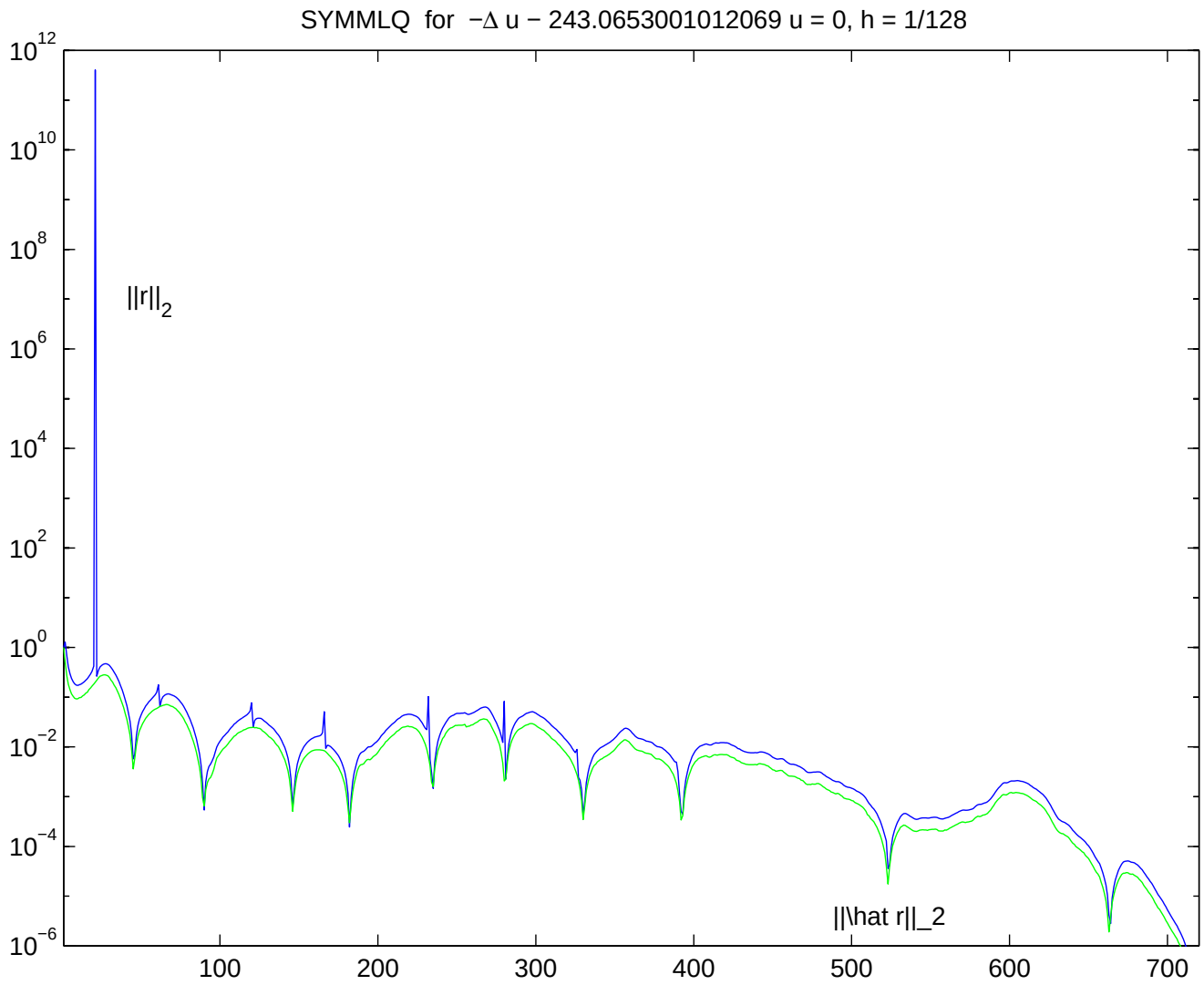
Example


SYMMLQ and CG for −Δ u − 163.84 u = 0, h=1/128
10^0
10^−2
10^−4
10^−6
10^−8
100
200
300
400
500
600
700
||\hat x||_2
red : ||x_CG||_2
blue : ||x_SYMMLQ||_2

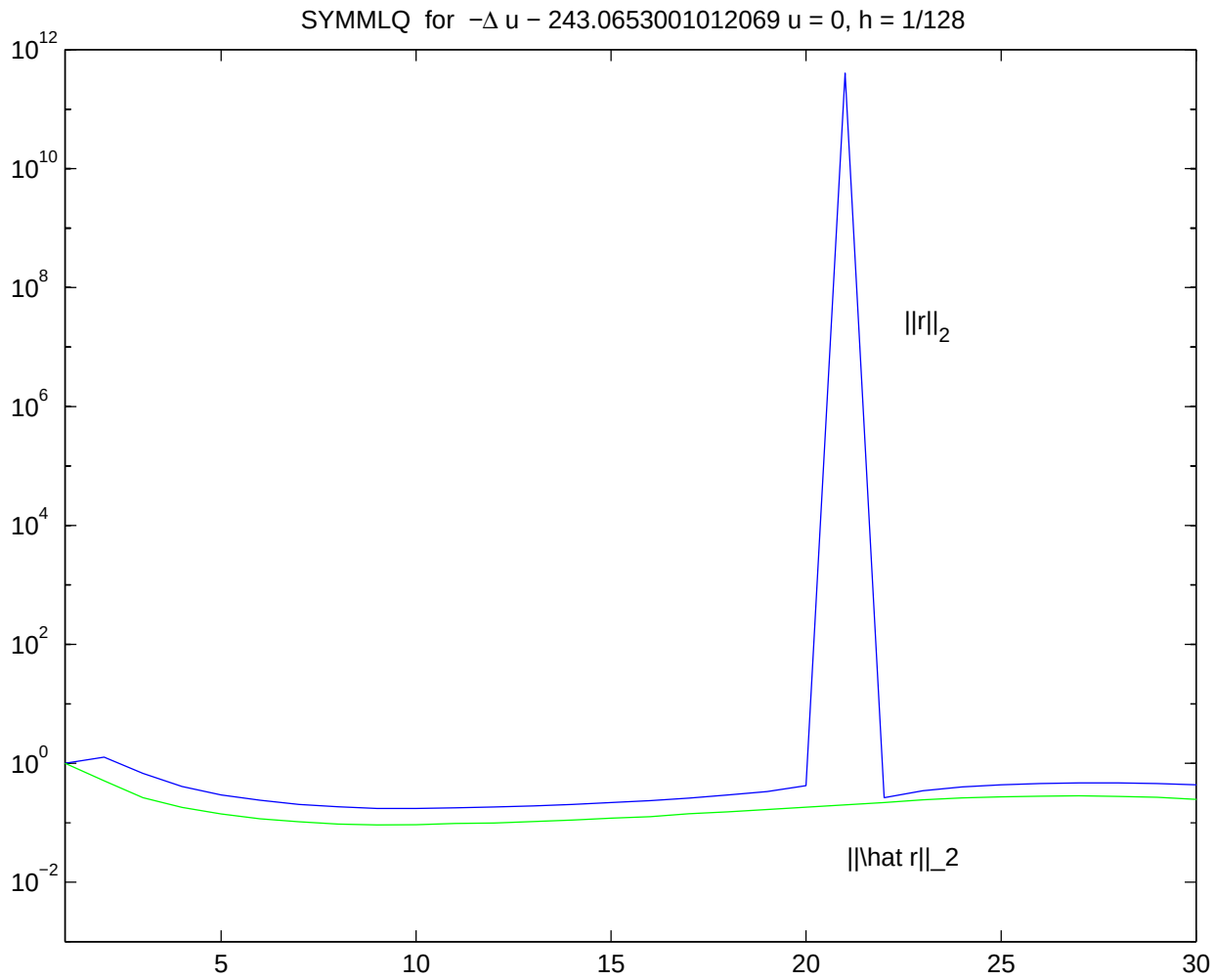
Example


SYMMLQ for − Δ u − 327.68 u = 0, h=1/128
10^1
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
0
100
200
300
400
500
600
700

Example


SYMMLQ for −Δ u − 243.0653001012069 u = 0, h = 1/128
10^12
10^10
10^8
10^6
10^4
10^2
10^0
10^−2
10^−4
10^−6
||r||_2
||\hat r||_2
100
200
300
400
500
600
700

Example


SYMMLQ for −Δ u − 243.0653001012069 u = 0, h = 1/128
10^12
10^10
10^8
10^6
10^4
10^2
10^0
10^−2
5
10
15
20
25
30
||r||_2
||\hat r||_2

Orthogonal direction method

Fridman (1963) suggested the orthogonal direction method, OD method for short, to compute the minimizer $\hat{x}^k$ of the error $\|x - x^*\|_2$ upon $A\mathcal{K}_k(b, A)$.

The method (which was rediscovered by Fletcher (1976)) constructs a sequence of orthogonal directions in $A\mathcal{K}_k(b, A)$ by the Lanczos procedure, and minimizes $\|x - x^*\|_2$ along these directions.

OD method

Choose initial approximation $\hat{x}^0$
$r^0 = b - A\hat{x}^0$
$d^0 = r^0$; $d^1 = Ar^0$;
$\delta_1 = 0$; $k = 0$,
while $\|r^k\|_2 > \text{tol}$ **do**
  $k = k + 1$;
  $\alpha_k = (d^k)^T d^k$
  $s^k = Ad^k$
  $\tau_k = (r^{k-1})^T d^{k-1}/\alpha_k$
  $\hat{x}^k = \hat{x}^{k-1} + \tau_k d^k$
  $r^k = r^{k-1} - \tau_k s^k$
  $\gamma_k = (d^k)^T s^k/\alpha_k$
  if k>1 **then**
    $\delta_k = \alpha_k/\alpha_{j-1}$
  end if
  $d^{k+1} = s^k - \gamma_k d^k - \delta_k d^{k-1}$
end while

OD method

The OD method requires 1 matrix-vector product and 7 level-1-operations, and five vectors have to be stored. Hence, it is less expensive than the SYMMLQ method.

However, it turned out to be unstable. The error decreases at the beginning of the iteration, but then it diverges rapidly.

Due to the loss of orthogonality of the search directions d^j the identity

$$(r^{k-1})^T A^{-1} d^k = (r^{k-1})^T d^{k-1}$$

(which is the basis of the step size formula in the OD method and which can be proved easily by induction) loses its validity. Thus, for k sufficently large x^k no longer minimizes $\|x - x^*\|_2$ upon $A\mathcal{K}_k(b, A)$.

Stabilized OD method

Stoer & Freund (1982) published a stable version of the OD method.

The orthogonal directions $d^1, \ldots, d^k$ with $d^1 = Ar^0$ span the space $A\mathcal{K}_k(b, A)$ if and only if they have the form $d^j = Av^j$, $v^1 = r^0$, where the vectors v^j satisfy

$$\text{span}\{v^1, \ldots, v^k\} = A\mathcal{K}_k(b, A) \quad \text{and} \quad (v^i)^T A^2 v^j = 0 \text{ for } i \neq j.$$

A set of vectors v^j with these properties can be obtained by the Lanczos method with respect to the scalar product $\langle x, y \rangle_{A^2} := y^T A^2 x$.

In terms of these vectors the step size of the OD method is given by

$$\tau_k = \frac{(r^{k-1})^T A^{-1} d^k}{\|d^k\|_2^2} = \frac{(r^{k-1})^T v^k}{(x^k)^T A^2 v^k},$$

and one obtains the stabilized OD method.

Stabilized OD method

Choose initial approximation $\hat{x}^0$
$r^0 = b - A\hat{x}^0$
$v^1 = r^0$; $v^0 = 0$; $w^1 = Av^1$; $w^0 = 0$;
$\delta_1 = 0$; $k = 0$,
while $\|r^k\|_2 >$ tol **do**
  $k = k + 1$;
  $\alpha_k = (w^k)^T w^k$
  $s^k = Aw^k$
  $\tau_k = (r^{k-1})^T v^k / \alpha_k$
  $\hat{x}^k = \hat{x}^{k-1} + \tau_k w^k$
  $r^k = r^{k-1} - \tau_k s^k$
  $\gamma_k = (w^k)^T s^k / \alpha_k$
  if k>1 **then**
    $\delta_k = \alpha_k / \alpha_{j-1}$
  end if
  $v^{k+1} = w^k - \gamma_k v^k - \delta_k v^{k-1}$
  $w^{k+1} = s^k - \gamma_k w^k - \delta_k w^{k-1}$
end while

Stabilized OD method

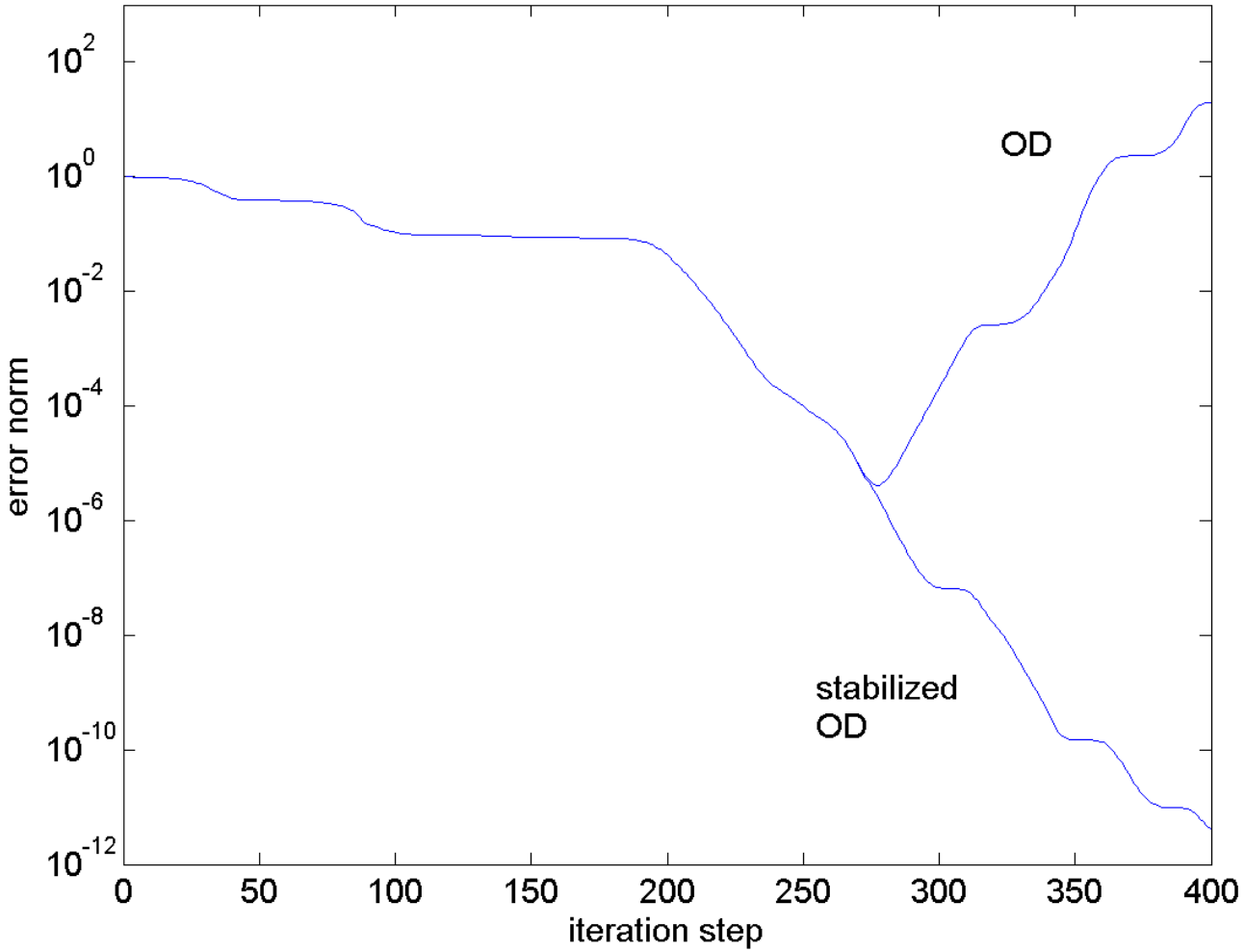
Every step of the stabilized OD method requires 1 matrix-vector product and 9 level-1-operations, and 7 vectors have to be stored.

Compared to SYMMLQ the same number of operations is required, but 2 additional vectors have to be stored. Moreover, there is no easy way to obtain the CG approximation x^k from $\hat{x}^k$ which is usually a better approximation to x^* than $\hat{x}^k$.

The following picture contains the convergence history of the OD and the stabilized OD method for the discrete Helmholtz equation

$$3.99U_{ij} - U_{i-1,j} - U_{i+1,j} - U_{i,j-1} - U_{i,j+1} = 0,\ i,j = 1,\ldots,128.$$

Example


error norm
iteration step
OD
stabilized
OD
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
0
50
100
150
200
250
300
350
400

MINRES

A further stable method for solving indefinite systems is a modification of the

Method of conjugate residuals (CR method)

Let A be symmetric and positive definite.

Determine $x^k \in x^0 + \mathcal{K}_k(r^0, A)$ such that the error with respect to the A^2 norm is minimal.

The conjugate residual method was introduced by Stiefel (1955). It was rediscovered by Luenberger (1970), who considered already indefinite problems and discussed a variant that handles exact breakdowns.

As for the CG method the approximations x_k can be obtained from a sequence of one dimensional line searches where the search directions satisfy a 3-term-recurrence.

CR method

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

Cost of CR

1 matrix-vector product
2 scalar products
4 _axpy
Storage requirements: 5 Vectors

For $x^* := A^{-1}b$ it holds

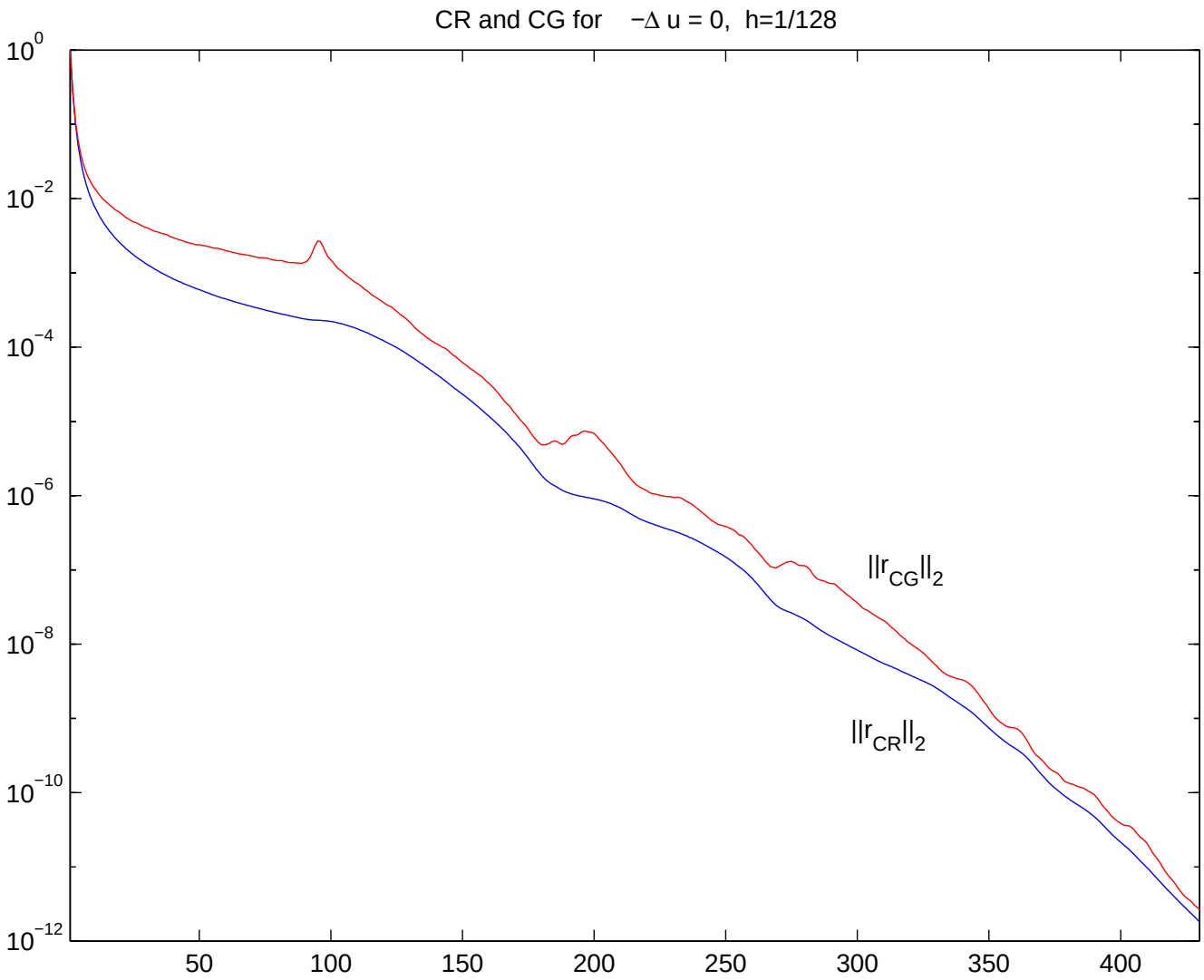
$$\|x - x^*\|_{A^2}^2 = \|A(x - x^*)\|_2^2 = \|Ax - b\|_2^2 = \|r\|_2^2$$

Hence, the CR method minimizes the Euclidean norm of the residuum in $x^0 + \mathcal{K}_k(r^0, A)$.

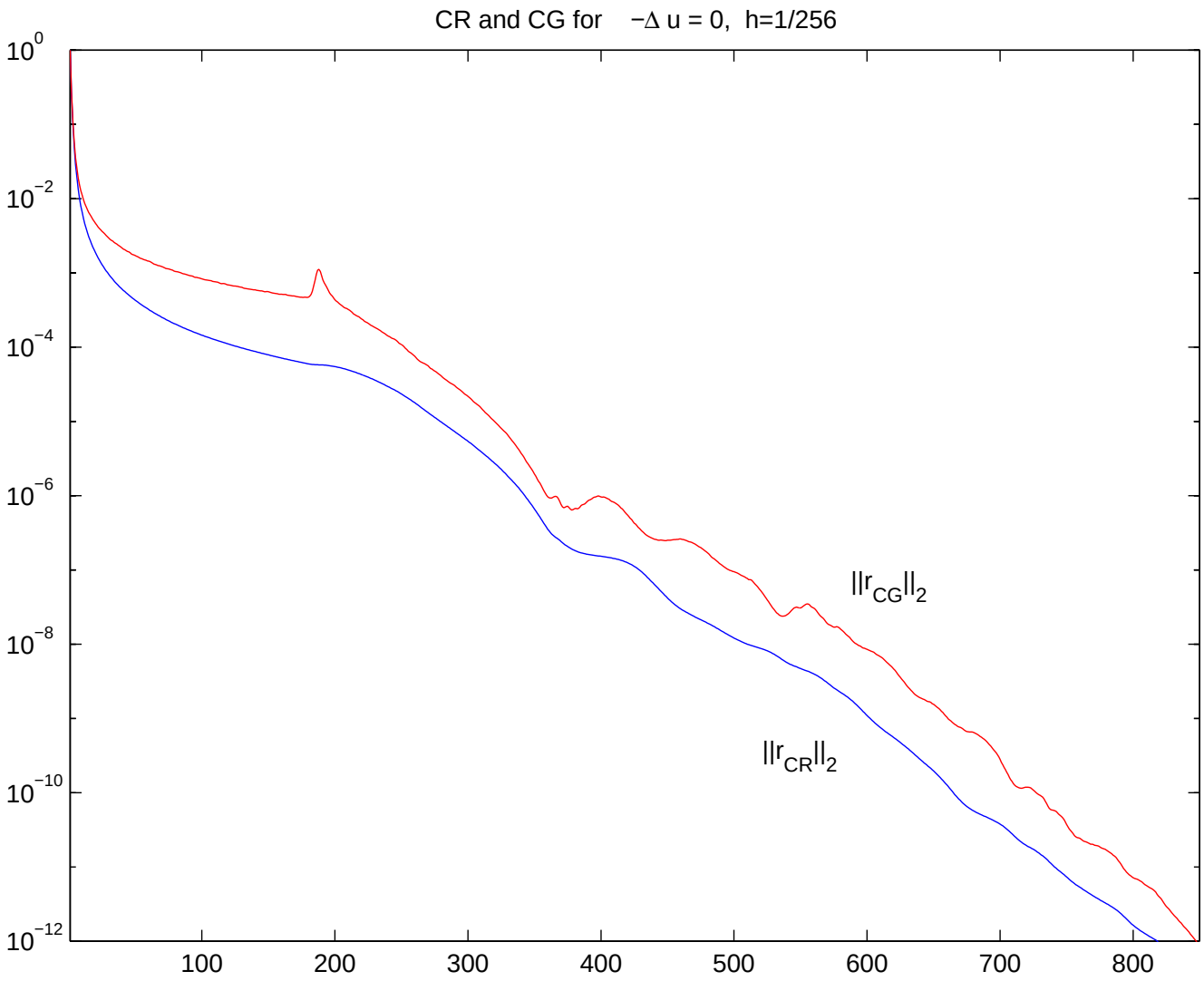
Moreover, it can be shown that the residuals $r^k := b - Ax^k$ determined by the CR method are A-conjugate (this explains the name of the method).

If A is symmetric but indefinite the approach of the CR method (to minimize the residuum on $\mathcal{K}_k(r^0, A)$) is still reasonable. However, the CR algorithm can break down with $\alpha_k = (r^k)^T A r^k = 0$.

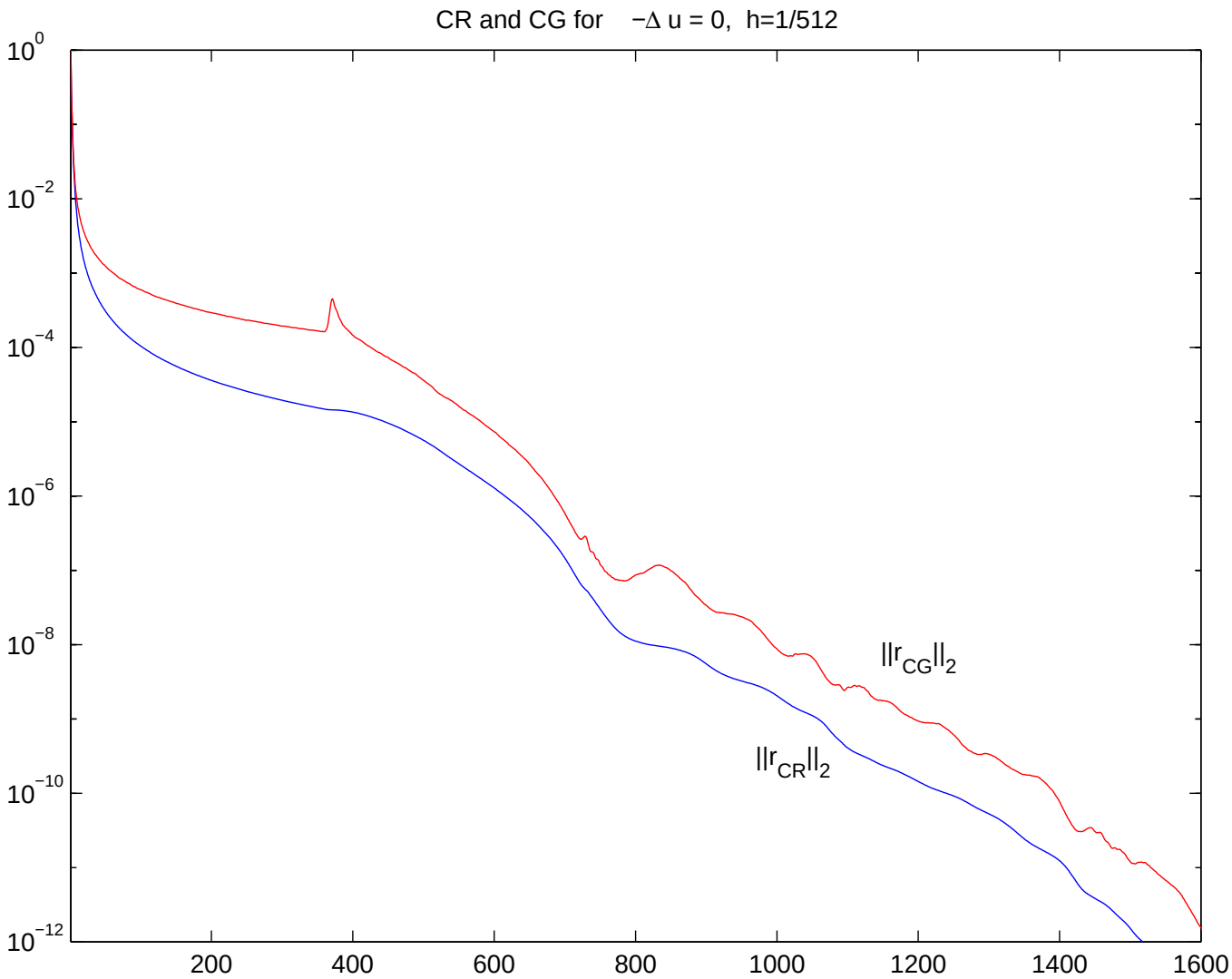
Example


CR and CG for −Δ u = 0, h=1/128
10^0
10^−2
10^−4
10^−6
10^−8
10^−10
10^−12
50
100
150
200
250
300
350
400
||r_CG||_2
||r_CR||_2

Example


CR and CG for −Δ u = 0, h=1/256
10^0
10^−2
10^−4
10^−6
10^−8
10^−10
10^−12
100
200
300
400
500
600
700
800
||r_CG||_2
||r_CR||_2

Example


CR and CG for −Δ u = 0, h=1/512
10^0
10^−2
10^−4
10^−6
10^−8
10^−10
10^−12
200
400
600
800
1000
1200
1400
1600
||r_CG||_2
||r_CR||_2

MINRES Paige & Saunders (1975)

Use the Lanczos process to tridiagonalize A and the LQ factorization of T_k to solve the normal equations

$$Q_k^T A^2 Q_k y^k = Q_k^T A b, \quad x^k = Q_k y^k$$

of the least squares problem

$$\|AQ_k y - b\|_2^2 = \min!$$

Again we assume without restricting generality $x^0 = 0$.

MINRES ct.

Since r^k is orthogonal to $q^1, \ldots, q^k$ it holds

$$\begin{aligned} Q_k^T A^2 Q_k &= (Q_k T_k + r^k (e^k)^T)^T (Q_k T_k + r^k (e^k)^T) \\ &= T_k^2 + \beta_k^2 e^k (e^k)^T \\ Q_k^T A b &= \beta_0 Q_k^T A q^1 = \beta_0 Q_k^T A Q_k e^1 \\ &= \beta_0 Q_k^T (Q_k T_k + (r^k)^T e^k) e^1 = \beta_0 T_k e^1 \end{aligned}$$

From the LQ factorization of $T_k = \tilde{L}_k U_k$ which can be obtained in the same manner as in the SYMMLQ method currently along with the Lanczos process one gets the Cholesky factorization

$$T_k^2 + \beta_k^2 e^k (e^k)^T = \tilde{L}_k \tilde{L}_k^T + \beta_k^2 e^k (e^k)^T = L_k L_k^T$$

without computing T_k^2 explicitly.

MINRES ct.

$c_1 = -1$; $s_1 = 0$; $c_0 = 1$; $s_0 = 0$; $q^0 = 0$; $v^0 = 0$; $v^{-1} = 0$

$\tilde{q} = b - Ax^0$; $\beta_0 = \|\tilde{q}\|_2$; $q^1 = \tilde{q}/\beta_0$; $\eta_1 = \beta_0$

for $k = 1, 2, \ldots$ until convergence **do**

  $\tilde{q} = Aq^k - \beta_{k-1} q^{k-1}$

  $\alpha_k = \tilde{q}^T q^k$

  $\tilde{q} = \tilde{q} - \alpha_k q^k$

  $\beta_k = \|\tilde{q}\|_2$

  $\tilde{\gamma}_k = -c_k \alpha_k - c_{k-1} s_k \beta_{k-1}$; $\gamma_k = \sqrt{\tilde{\gamma}_k^2 + \beta_k^2}$

  $\delta_{k-1} = s_k \alpha_k - c_k c_{k-1} \beta_{k-1}$; $\varepsilon_{k-2} = s_{k-1} \beta_{k-1}$

  $c_{k+1} = \tilde{\gamma}_k / \gamma_k$; $s_{k+1} = \beta_k / \gamma_k$

  $v^k = (q^k - \varepsilon_{k-2} v^{k-2} - \delta_{k-1} v^{k-1}) / \gamma_k$;

  $x^k = x^{k-1} + c_{k+1} \eta_k v^k$;

  $q^{k+1} = \tilde{q} / \beta_k$

  $\eta_{k+1} = s_{k+1} \eta_k$

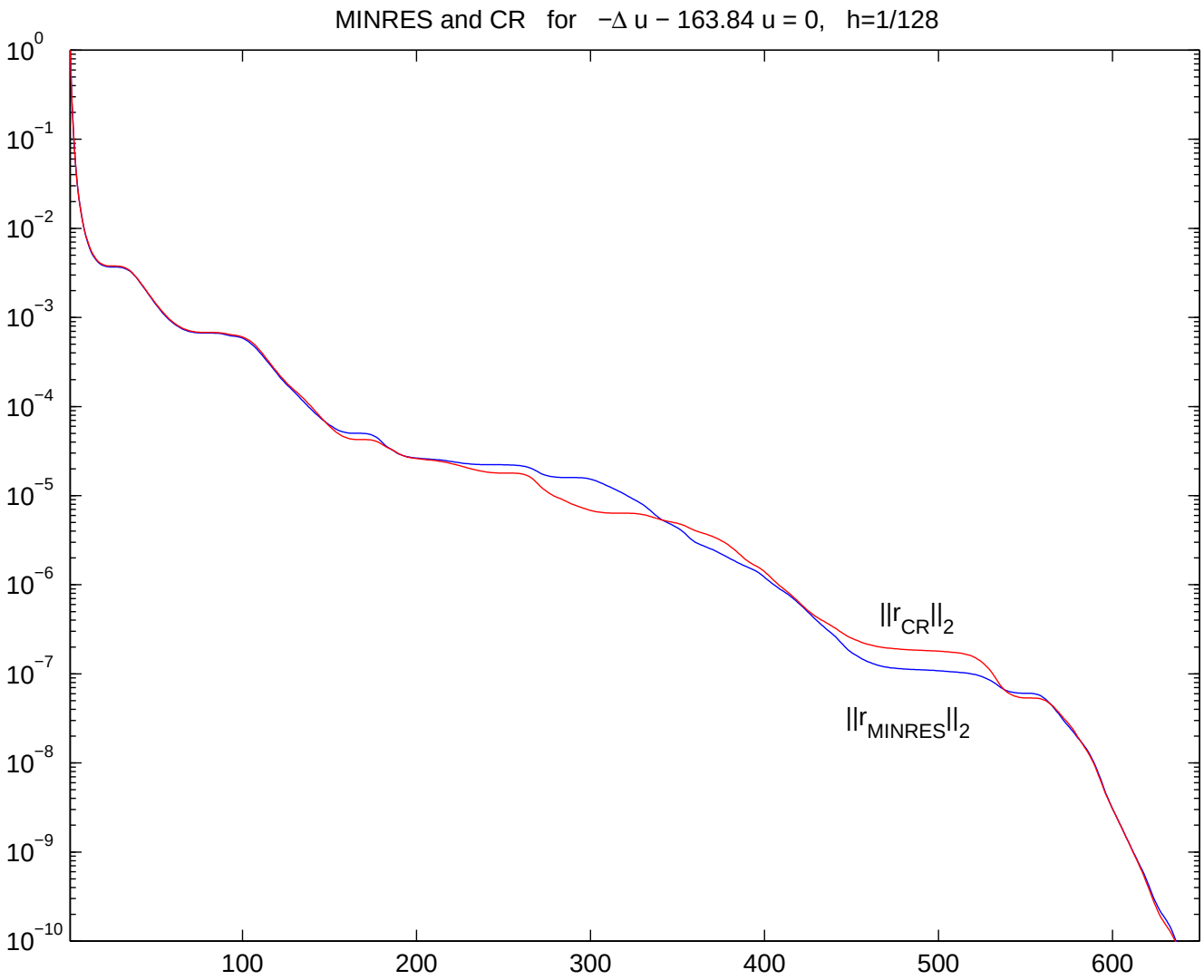
end for

Cost of MINRES

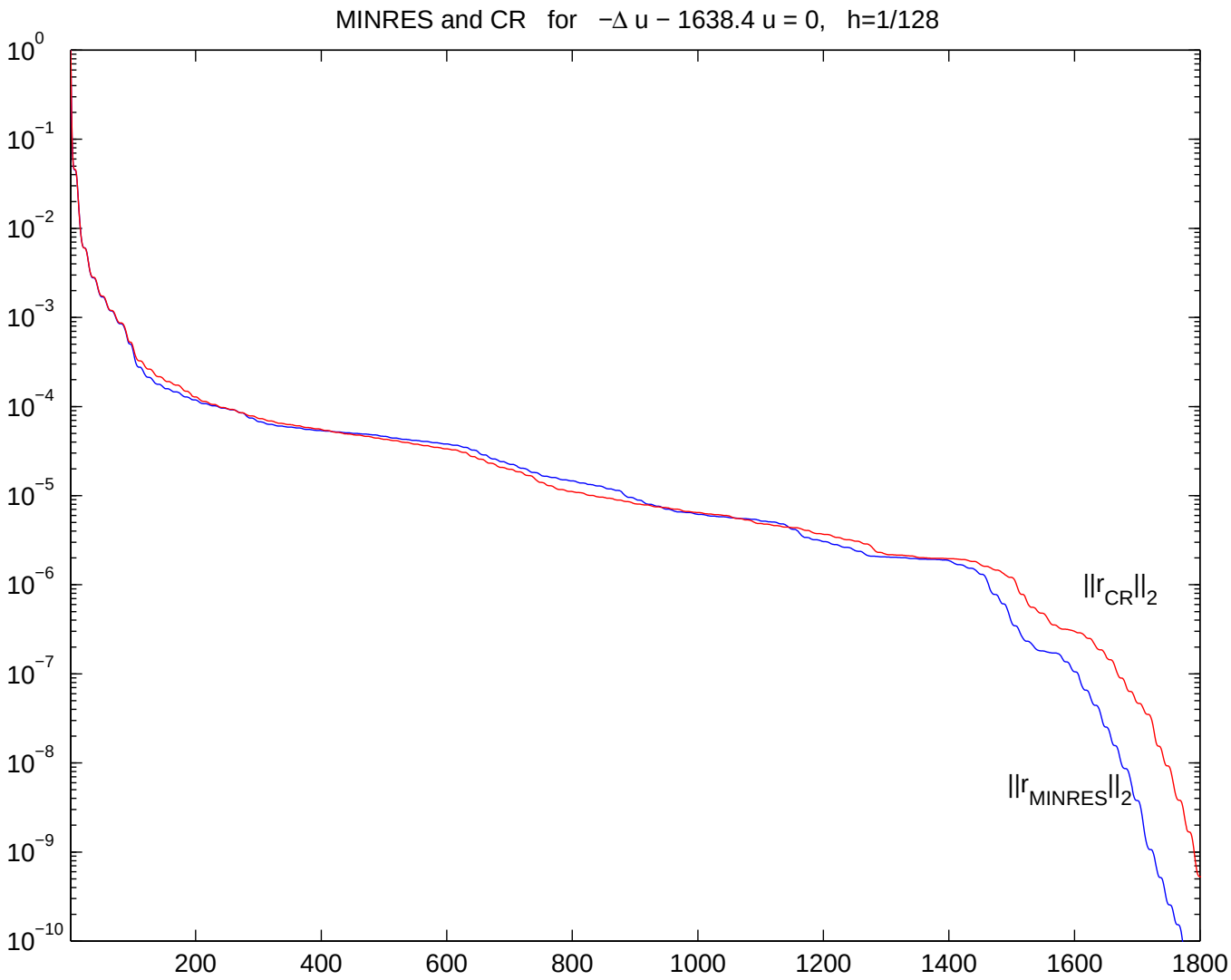
1 matrix-vector product
2 scalar products
7 _axpy

Storage requirements: 6 Vectors

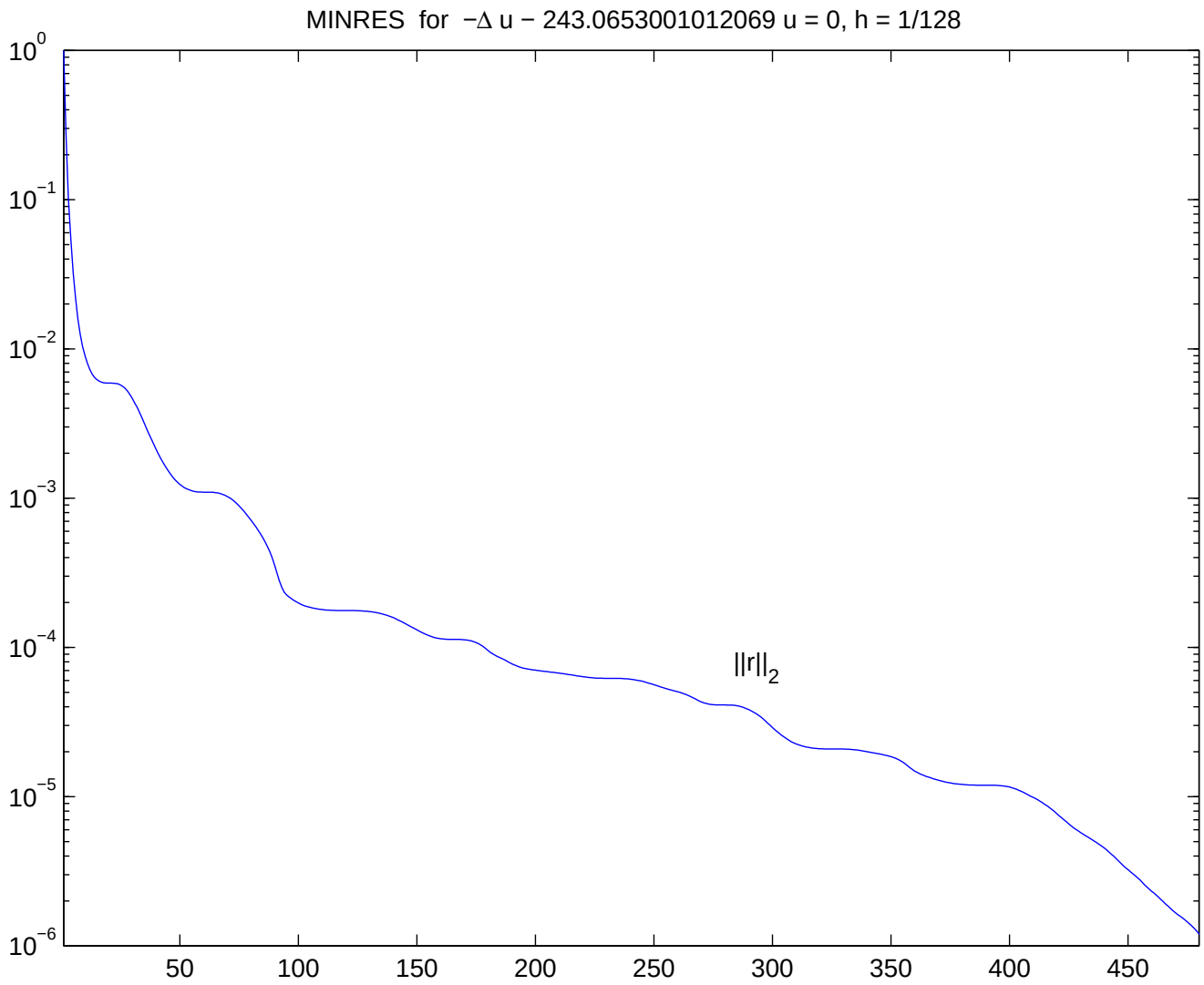
Example


MINRES and CR for −Δ u − 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
100
200
300
400
500
600
||r_CR||_2
||r_MINRES||_2

Example


MINRES and CR for −Δ u − 1638.4 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
1400
1600
1800
||r_CR||_2
||r_MINRES||_2

Example


MINRES for −Δ u − 243.0653001012069 u = 0, h = 1/128
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
50
100
150
200
250
300
350
400
450
||r||_2

Error bounds for MINRES

$x^k \in x^0 + \text{span}\{x^0, Ax^0, \ldots, A^{k-1}x^0\}$ is determined such that $\|r^k\|_2$ is minimal in

$$r^0 + \text{span}\{Ar^0, \ldots, A^k r^0\},$$

i.e. $r^k = p_k^M(A)r^0$ with $p_k^M \in \Pi_k$, $p_k^M(0) = 1$,

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

For $A = U\Lambda U^T$, $\Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_n\}$, it follows

$$\|r^k\|_2 \le \min_{p_k \in \Pi_k,\ p_k(0)=1} \|p_k(\Lambda)\|_2 \|r^0\|_2,$$

i.e.

$$\frac{\|r^k\|_2}{\|r^0\|_2} \le \min_{p_k \in \Pi_k,\ p_k(0)=1} \max_{j=1,\ldots,n} |p_k(\lambda_j)|.$$

Error bounds for MINRES ct.

If A is positive definite (CR method!) we obtain in the same way as for the CG method

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

For indefinite problems typically only very few ($\ell \ll n$) eigenvalues $\lambda_1 \leq \cdots \leq \lambda_\ell < 0$ are negative, and most of the eigenvalues $0 < \lambda_{\ell+1} \leq \cdots \leq \lambda_n$ are positive.

Error bounds for MINRES ct.

If

$$p_k(\lambda) = q_\ell(\lambda)\left[T_{k-\ell}\left(\frac{2\lambda-\lambda_{\ell+1}-\lambda_n}{\lambda_n-\lambda_{\ell+1}}\right)\Big/ T_{k-\ell}\left(\frac{-\lambda_{\ell+1}-\lambda_n}{\lambda_n-\lambda_{\ell+1}}\right)\right]$$

$$q_\ell(\lambda) = (-1)^\ell \prod_{j=1}^{\ell}(\lambda-\lambda_j)/\prod_{j=1}^{\ell}\lambda_j$$

and Chebyshev polynomials $T_{k-\ell}$,

one gets

$$\frac{\|r^k\|_2}{\|r^0\|_2} \le q_\ell(\lambda_n) 2\left(\frac{\sqrt{\kappa_{n-\ell-1}}-1}{\sqrt{\kappa_{n-\ell-1}}+1}\right)^{k-\ell}, \quad \kappa_{n-\ell-1} := \frac{\lambda_n}{\lambda_{\ell+1}}.$$

Preconditioning

The last error bound demonstrates that it is reasonable to use positive definite preconditioners even if the problem is indefinite.

To preserve symmetry of the problem we consider

$$C^{-1}AC^{-T}y = C^{-1}b,\ x := C^T y.$$

From

$$\begin{aligned} C^{-1}AC^{-T}z = \mu z \quad &\Longleftrightarrow \quad C^{-1}Ay = \mu C^T y,\ y = C^{-T}z \\ &\Longleftrightarrow \quad C^{-T}C^{-1}Ay = (CC^T)^{-1}Ay = \mu y \end{aligned}$$

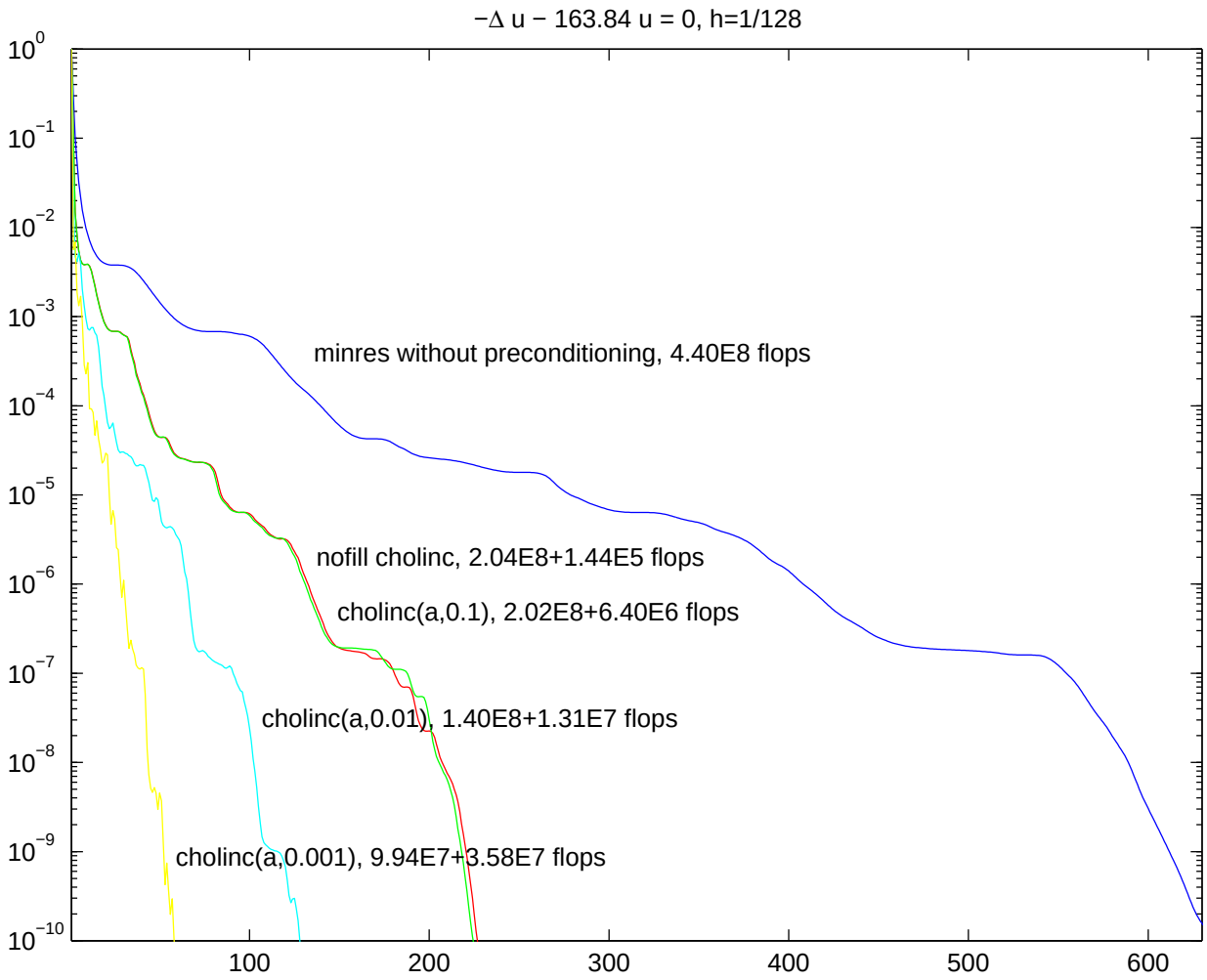
we obtain that the Cholesky factor C of the preconditioner $M := CC^T$ should be chosen such that:

$$\kappa_{n-\ell-1}(M^{-1}A) \ll \kappa_{n-\ell-1}(A)$$

MINRES with Preconditioning

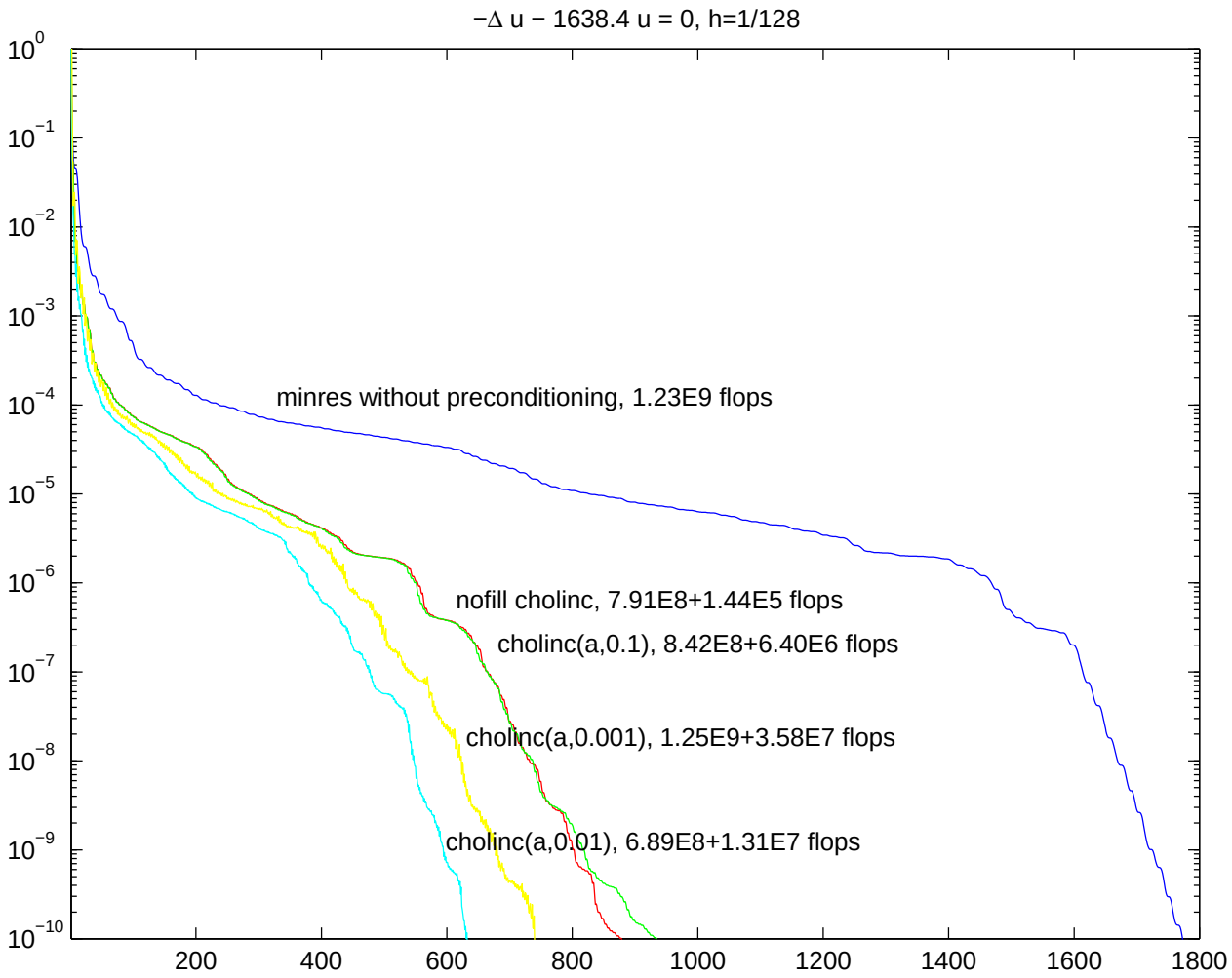
$c_1 = -1$; $s_1 = 0$; $c_0 = 1$; $s_0 = 0$; $q^0 = 0$; $v^0 = 0$; $v^{-1} = 0$
$q = b - Ax$; $w = M^{-1}q$; $\beta_0 = \sqrt{q^T w}$
$q^1 = q/\beta_0$; $w^1 = w/\beta_0$; $\eta_1 = \beta_0$
for $k = 1, 2, \ldots$ until convergence **do**
  $q = Aw^k - \beta_{k-1} q^{k-1}$
  $\alpha_k = q^T w^k$
  $q = q - \alpha_k q^k$
  $w^{k+1} = M^{-1}q$
  $\beta_k = \sqrt{q^T w^{k+1}}$
  $\tilde{\gamma}_k = -c_k \alpha_k - c_{k-1} s_k \beta_{k-1}$; $\gamma_k = \sqrt{\tilde{\gamma}_k^2 + \beta_k^2}$
  $\delta_{k-1} = s_k \alpha_k - c_k c_{k-1} \beta_{k-1}$; $\varepsilon_{k-2} = s_{k-1}\beta_{k-1}$
  $c_{k+1} = \tilde{\gamma}_k/\gamma_k$; $s_{k+1} = \beta_k/\gamma_k$
  $v^k = (w^k - \varepsilon_{k-2} v^{k-2} - \delta_{k-1} v^{k-1})/\gamma_k$
  $x^k = x^{k-1} + c_{k+1}\eta_k v^k$
  $q^{k+1} = q/\beta_k$
  $w^{k+1} = w^{k+1}/\beta_k$
  $\eta_{k+1} = s_{k+1}\eta_k$
end for

Example



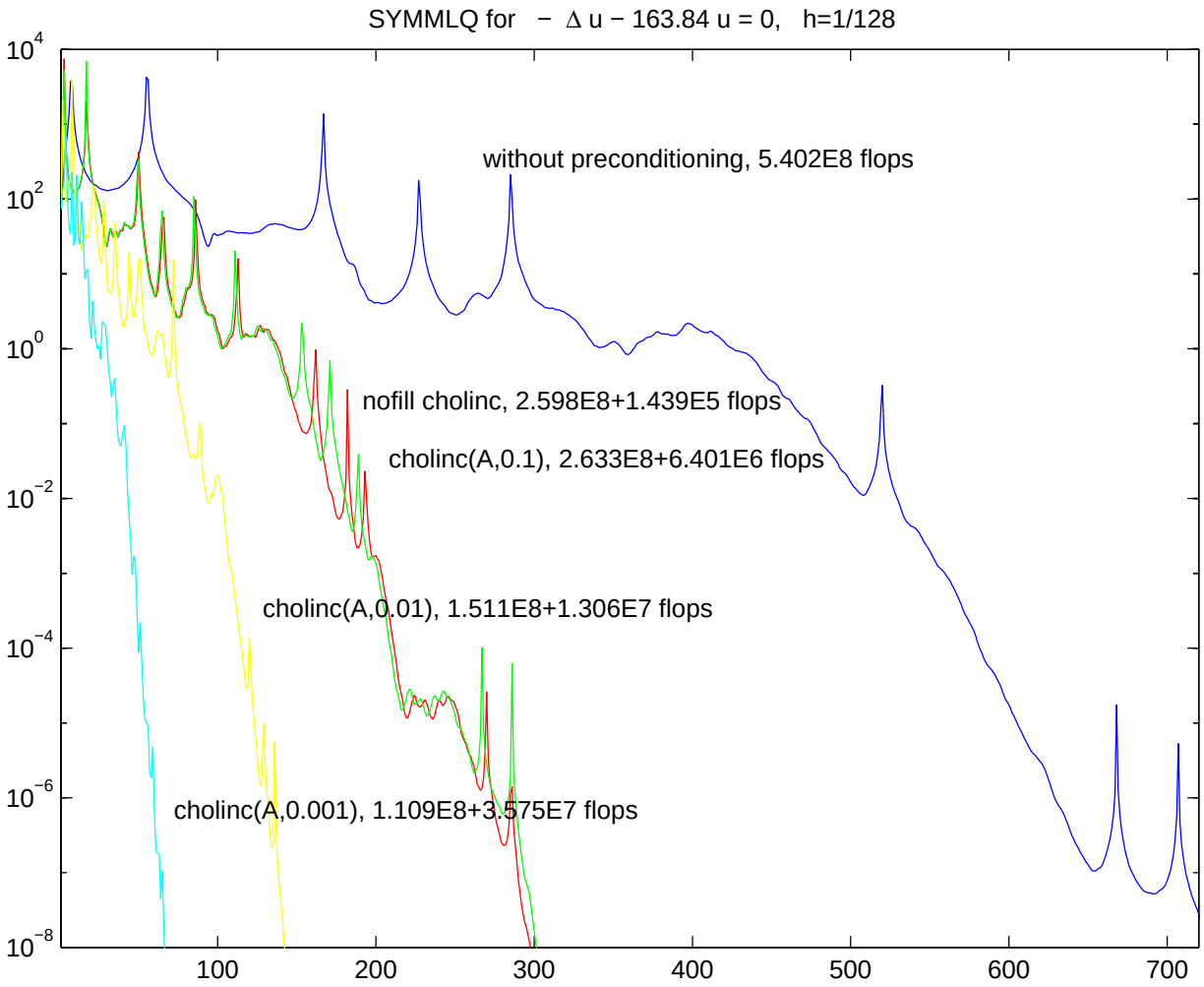
Direct solution using \: 1.625E8 flops

Example



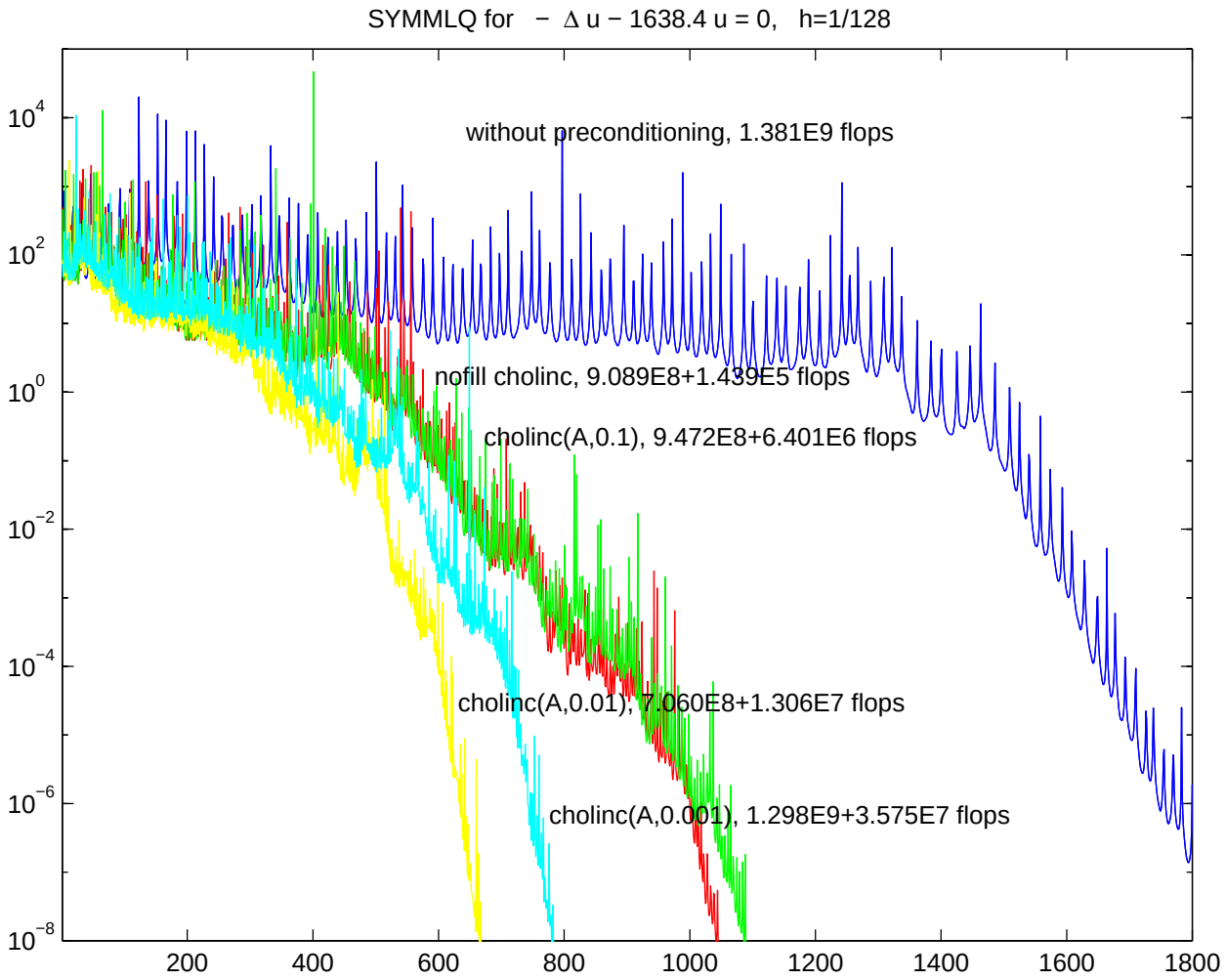
Direct solution using \: 1.625E8 flops

Example



Direct solution using \: 1.625E8 flops

Example



Direct solution using \: 1.625E8 flops