Quasi-Minimal Residuum Methods

Twosided Lanczos method

Choose v^1, w^1 with $(v^1)^T w^1 \neq 0$

$v^0 = 0$; $w^0 = 0$; $\beta_0 = 0$; $\gamma_0 = 0$;

for $k = 1, 2, \ldots$

$\tilde{v} = Av^k$; $\quad \tilde{w} = A^T w^k$;

$\alpha_k = \tilde{v}^T w^k$;

$\tilde{v} = \tilde{v} - \alpha_k v^k - \beta_{k-1} v^{k-1}$;

$\tilde{w} = \tilde{w} - \alpha_k w^k - \gamma_{k-1} w^{k-1}$;

$\gamma_k = \|\tilde{v}\|_2$;

if $\gamma_k == 0$ STOP;

$v^{k+1} = \tilde{v}/\gamma_k$;

$\beta_k = \tilde{w}^T v^{k+1}$;

if $\beta_k == 0$ STOP;

$w^{k+1} = \tilde{w}/\beta_k$; end

The twosided Lanczos process determines basis vectors v^j and w^j of the Krylov spaces $\mathcal{K}_k(v^0, A)$ and $\mathcal{K}_k(w^0, A^T)$, i.e. for "both sides of the matrix".

The coefficients α_j, β_j and γ_j are chosen such that the biorthogonality of w^j and v^k holds in the following way:

$$(w^j)^T v^k = \delta_{jk}.$$

Other "normalizations" are possible and are in use.

With $V_k := (v^1, \dots, v^k)$, $W_k := (w^1, \dots, w^k)$ and

$$H_k = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \dots & 0 & 0 \\ \gamma_1 & \alpha_2 & \beta_2 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ & & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \alpha_{k-1} & \beta_{k-1} \\ 0 & 0 & 0 & \dots & \gamma_{k-1} & \alpha_k \end{pmatrix}$$

the twosided Lanczos process can be written as

$$AV_k = V_k H_k + \gamma_k v^{k+1} (e^k)^T, \;\; A^T W_k = W_k H_k^T + \beta_k w^{k+1} (e^k)^T.$$

The biorthogonality obtains the form

$$W_k^T V_k = I_k.$$

From

$$\mathcal{K}_k(v^1, A) = \text{span}\{v^1, \dots, v^k\}$$

and

$$\mathcal{K}_k(w^1, A^T) = \text{span}\{w^1, \dots, w^k\}$$

it follows that the k-th approximation of the BiCG method can be determined from the tridiagonal system

$$W_k^T (A(x^0 + V_k y^k) - b) = H_k y^k - W_k^T r^0 = 0,$$

$$x^k = x^0 + V_k y^k.$$

The twosided Lanczos process can break down for two reasons. If $\gamma_k = 0$, i.e. $v^{k+1} = 0$, then $\mathcal{K}_k(r^0, A)$ is an invariant subspace of A, and for $v^1 = \beta_0 r^0$

$$AV_k y^k = V_k H_k y^k = r^0 = \beta V_k e^1 \quad \Longleftrightarrow \quad V_k(H_k y^k - \beta_0 e^1) = 0.$$

Hence, the solution of $Ax = b$ has the representation $x^0 + V_k y^k$, and the system is solved exactly. This breakdown is called regular termination or lucky termination.

Secondly the method can break down with $v^{k+1} \neq 0$ and $\beta_k = 0$, i.e. $(v^{k+1})^T \tilde{w} = 0$. This case is called serious breakdown. One does not obtain biorthogonal bases of the Krylov spaces of dimension $k+1$ or higher from the twosided Lanczos process (with initial vectors v^1 and w^1).

With $r^0 = b - Ax^0 = \beta_0 v^1$ and

$$\tilde{H}_m = \begin{pmatrix} H_m \\ (e^m)^T \end{pmatrix}$$

the residuum of $x := x^0 + V_m y$ is given by

$$\begin{aligned} b - Ax &= b - A(x^0 + V_m y) = r^0 - AV_m y \\ &= \beta_0 v^1 - V_{m+1}\tilde{H}_m y = V_{m+1}(\beta_0 e^1 - \tilde{H}_m y). \end{aligned}$$

The GMRES solution is obtained minimizing

$$\|b - Ax\|_2 = \|V_{m+1}(\beta_0 e^1 - \tilde{H}_m y)\|_2.$$

Observe however, that V_{m+1} does not have orthonormal columns, and therefore this least squares problem is fully populated.

QMR method (quasi minimal residuum):

We neglect the matrix V_{m+1}, solve the least square problem

$$\|\beta_0 e^1 - \tilde{H}_m y\|_2 = \min!$$

and set $x^m = x^0 + V_m y^m$.

This can be interpreted as minimizing the residuum with respect to the transformed norm

$$\|W_{m+1}^T r\|_2 = \|W_{m+1}^T V_{m+1}(\beta_0 e^1 - \tilde{H}_m y)\|_2 = \|\beta_0 e^1 - \tilde{H}_m y\|_2 = \min!$$

Hence, to enable a short recurrence the norm is modified in every single step.

Similarly as in the GMRES method the least squares problem in the QMR method can be solved using Givens reflections.

Similarly as in MINRES $x^m = x^0 + V_m y^m$ can be updated. We do not have to store all previous search directions, but only two of them.

The residual norms are not monotonely decreasing, however, magnifications usually appear only in very few iterations and are of small magnitude.

QMR method with 3 term recurrence

$r = b - Ax$; $\beta = \|r\|$; $v^1 = r/\beta_0$; $w^1 = v^1$; $z_1 = 1$;
$\beta_0 = 0$; $\gamma_0 = 0$; $v^0 = 0$; $w^0 = 0$; $p^{-1} = 0$; $p^{-2} = 0$;
for $k = 1, 2, \dots$ until convergence
$\tilde{v} = Av^k$;
$\tilde{w} = A^T w^k$;
$\alpha_k = \tilde{v}^T w^k$;
$\tilde{v} = \tilde{v} - \alpha_k v^k - \beta_{k-1} v^{k-1}$;
$\tilde{w} = \tilde{w} - \alpha_k w^k - \gamma_{k-1} w^{k-1}$;
$\gamma_k = \|\tilde{v}\|_2$;
$v^{k+1} = \tilde{v}/\gamma_k$;
$\beta_k = \tilde{w}^T v^{k+1}$;
$w^{k+1} = \tilde{w}/\beta_k$;
if $k > 2$ $\begin{pmatrix} \rho \\ \beta_{k-1} \end{pmatrix} = \begin{pmatrix} c_{k-2} & s_{k-2} \\ s_{k-2} & -c_{k-2} \end{pmatrix} \begin{pmatrix} 0 \\ \beta_{k-1} \end{pmatrix}$; end

if $k > 1$ $\begin{pmatrix} \beta_{k-1} \\ \alpha_k \end{pmatrix} = \begin{pmatrix} c_{k-1} & s_{k-1} \\ s_{k-1} & -c_{k-1} \end{pmatrix} \begin{pmatrix} \beta_{k-1} \\ \alpha_k \end{pmatrix}$; end
$\eta = \sqrt{\alpha_k^2 + \gamma_k^2}$;
$s_k = \gamma_k/\eta$;
$c_k = \alpha_k/\eta$;
$\alpha_k = \eta$;
$z_{k+1} = s_k z_k$;
$z_k = c_k z_k$;
$p^{k-1} = (v^k - \beta_{k-1} p^{k-2} - \rho p^{k-3})/\alpha_k$;
$x = x + \beta z_k p^{k-1}$;
end

Cost of QMR: 2 matrix-vector products
3 scalar products
9 _axpy

Storage requirements: 10 vectors

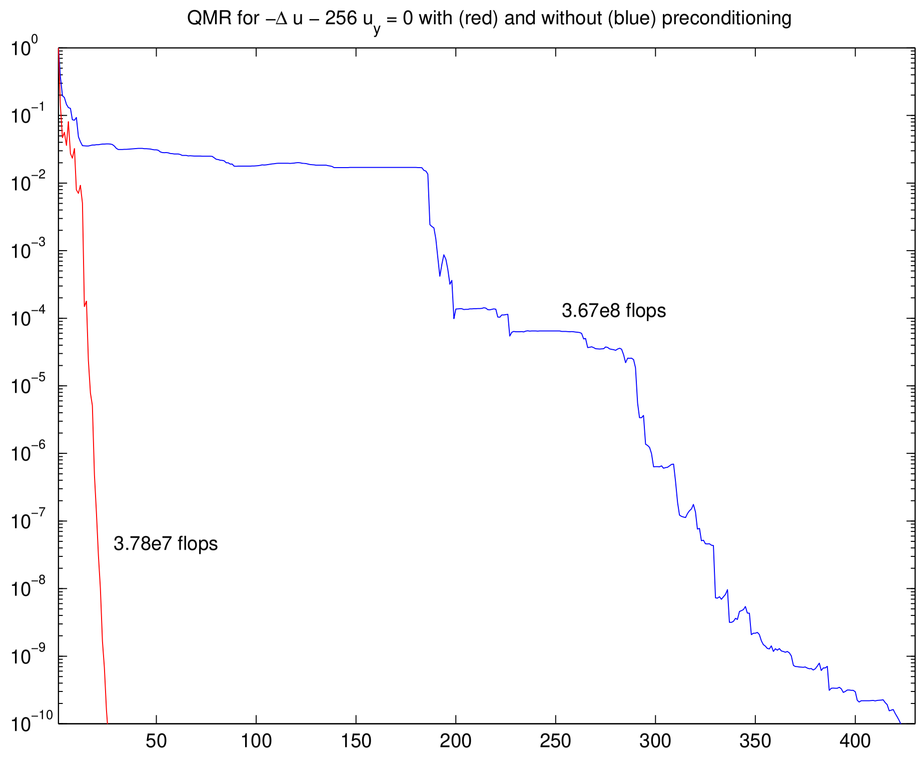
$\|r^m\|_2$ is not determined in the QMR method. However, it holds

$$b - Ax = V_{m+1}(\beta e^1 - \tilde{H}_m y).$$

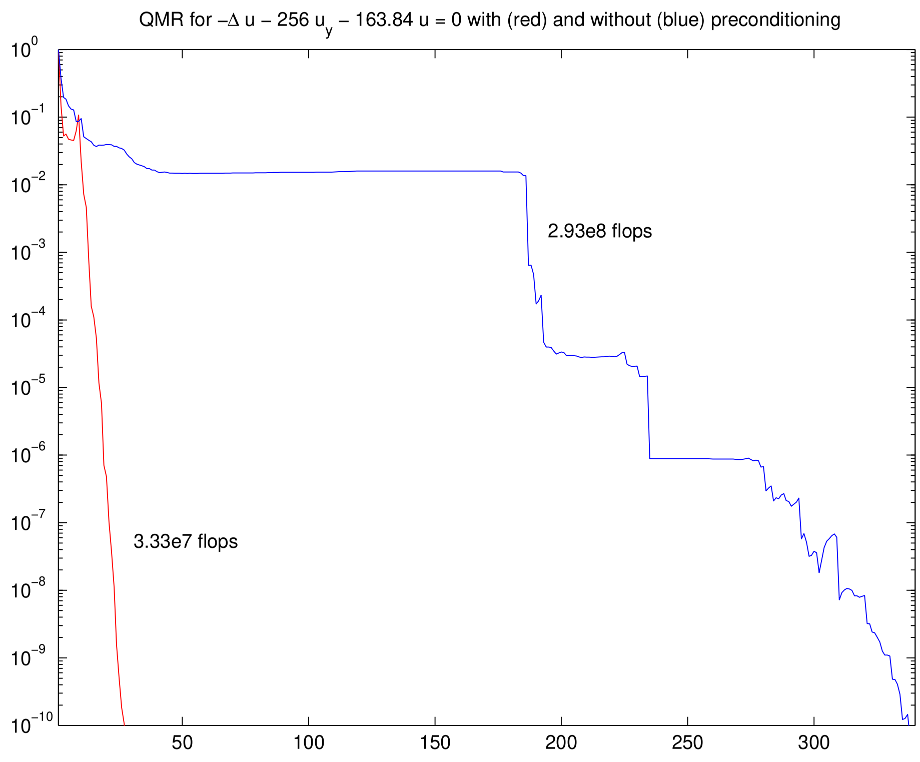
If the twosided Lanczos process is normalized such that $\|v^k\|_2 = 1$ for every k, then it follows

$$\|r^m\|_2 \leq \|V_{m+1}\|_S \|\beta_0 e^1 - \tilde{H}_m y\|_2 \leq \sqrt{m+1}\|\beta_0 e^1 - \tilde{H}_m y\|_2,$$

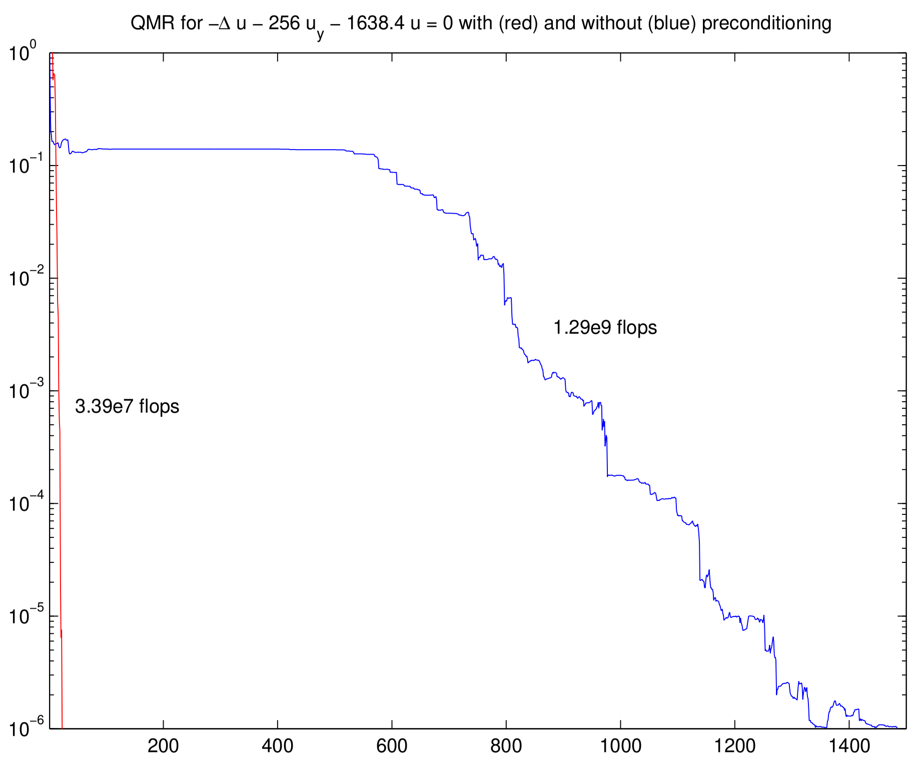
and the termination can be based on the moderate overestimation of $\|r^m\|_2$ on the right.



QMR for –Δ u – 256 u_y = 0 with (red) and without (blue) preconditioning
3.67e8 flops
3.78e7 flops

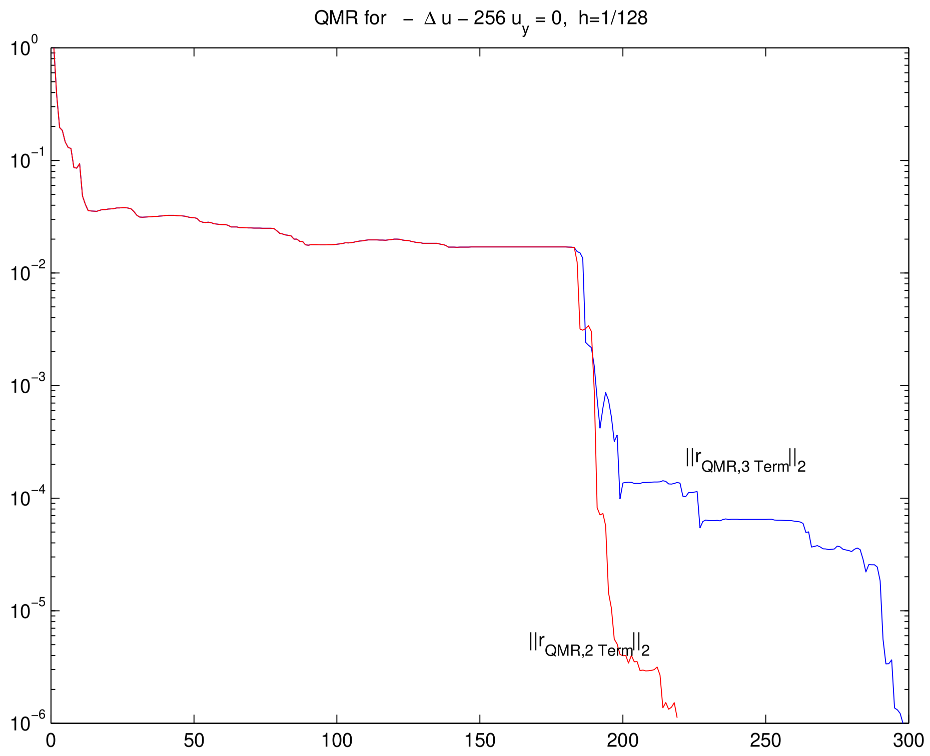


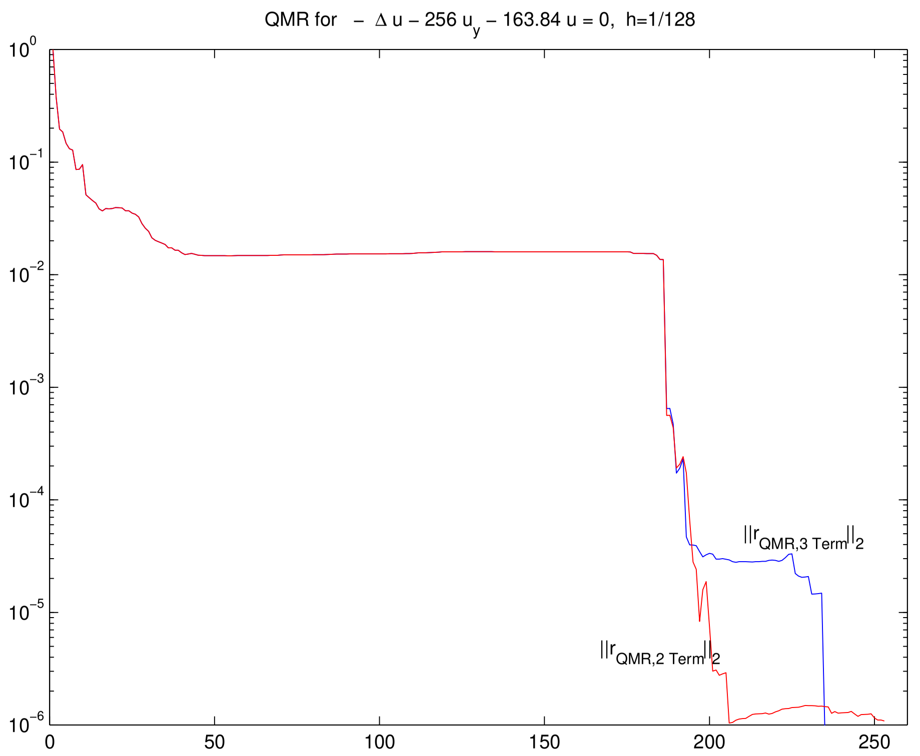
QMR for –Δ u – 256 u_y – 163.84 u = 0 with (red) and without (blue) preconditioning
2.93e8 flops
3.33e7 flops

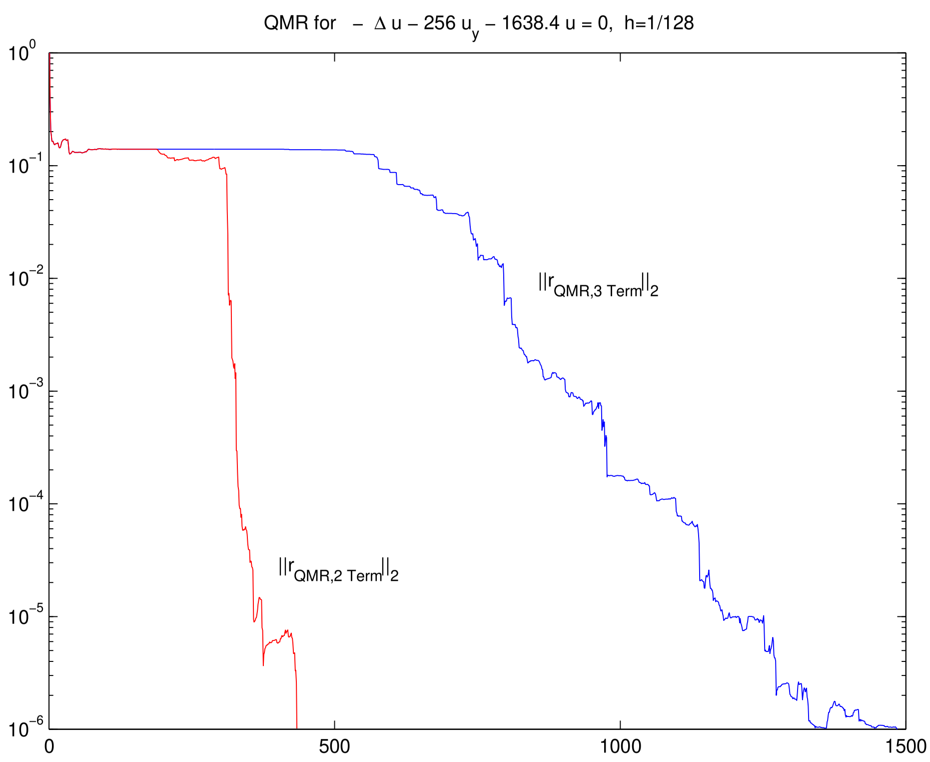


In addition to the twosided Lanczos process with 3 term recurrence a coupled 2 term Lanczos process is in use to determine biorthogonal bases $\{v^j\}$ and $\{w^j\}$. In this case additional bases $\{p^j\}$ of $\mathcal{K}_k(v^1, A)$ and $\{q^j\}$ of $\mathcal{K}_k(w^1, A^T)$ are determined.

The "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods" contain a (preconditioned) QMR method based on the coupled 2 term Lanczos process. The following plots compare the convergence history of the QMR method with 2 term and with 3 term recurrence.


QMR for $-\Delta u - 256 u_y = 0$, h=1/128
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0
50
100
150
200
250
300
$\|r_{QMR,3\ Term}\|_2$
$\|r_{QMR,2\ Term}\|_2$


QMR 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}
0
50
100
150
200
250
$\|r_{QMR,3\ Term}\|_2$
$\|r_{QMR,2\ Term}\|_2$



Serious breakdowns of the twosided Lanczos method can be avoided by so called look ahead strategies. Only in very few cases (which are called incurable breakdowns) this is not possible.

Look ahead Lanczos methods were introduced by Parlett, Taylor & Liu (1985) solving eigenvalue problems numerically; in connection with QMR methods for sparse linear systems they were studied by Freund, Gutknecht & Nachtigall (1993).

$$\mathcal{K}_k(v^1, A) = \{\phi(A)v^1 \,:\, \phi \in \Pi_{k-1}\},$$
$$\mathcal{K}_k(w^1, A^T) = \{\phi(A^T)w^1 \,:\, \phi \in \Pi_{k-1}\}.$$

The twosided Lanczos process determines a unique polynomial $\psi_{k-1} \in \Pi_{k-1}$, $\psi_{k-1} = t^{k-1} + \tilde{\psi}$, $\tilde{\psi} \in \Pi_{k-2}$ (monic polynomial) such that

$$v^1 = \psi_{k-1}(A)v^1,\ w^1 = \psi_{k-1}(A^T)w^1.$$

With the formal scalar product

$$\langle \phi, \psi \rangle := (\phi(A^T)w^1)^T(\psi(A)v^1) = (w^1)^T\phi(A)\psi(A)w^1$$

the biorthogonality can be rewritten as

$$\langle \phi, \psi_{k-1} \rangle = 0 \quad \text{for every } \phi \in \Pi_{k-2}, \quad \langle \psi_{k-1}, \psi_{k-1} \rangle \neq 0.$$

A polynomial with this property is called

formal orthogonal polynomial (FOP).

Let

$$\mu_j := (w^1)^T A^j v^1 = \langle 1, \iota^j \rangle,\ j = 1, 2, \ldots$$

be the moments corresponding to $\langle \cdot, \cdot \rangle$. Then

$$\phi_{k-1}(t) := \sum_{j=0}^{k-1} \gamma_j t^j,\ \gamma_{k-1} \neq 0,$$

is a FOP of degree $k-1$ if and only if

$$\begin{pmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_{k-2} \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{k-2} & \mu_{k-1} & \mu_k & \cdots & \mu_{2k-3} \end{pmatrix} \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{k-2} \end{pmatrix} = -\gamma_{k-1} \begin{pmatrix} \mu_{k-1} \\ \mu_k \\ \vdots \\ \mu_{2k-3} \end{pmatrix}$$

has a nontrivial solution $(\gamma_0, \gamma_1, \ldots, \gamma_{k-1})^T$.

A FOP ϕ_{k-1} is called regular if it is uniquely determined by the requirement

$$\langle \phi, \psi_{k-1} \rangle = 0 \quad \text{for every } \phi \in \Pi_{k-2}$$

up to a multiple scalar, else it is called singular.

Every polynomial $\phi_0 \equiv \gamma_0 \neq 0$ is a regular FOP of degree 0.

For $k \geq 1$ there exists a regular FOP if and only if the matrix $M := (\mu_{i+j})_{i,j=1,\dots,k-1}$ is nonsingular.

Singular FOPs appear if the matrix M is singular and if the nonhomogeneous system is consistent for $\gamma_k \neq 0$. If it is inconsistent for $\gamma_k \neq 0$ then no FOP of degree k exists. This case is called deficient.

Lemma Let ψ_{k-1} be a regular FOP of degree $k-1$ with respect to the formal scalar product

$$\langle \phi, \psi \rangle := (w^1)^T \phi(A) \psi(A) w^1.$$

Then a regular FOP of degree k exists if and only if $\langle \psi_{k-1}, \psi_{k-1} \rangle \neq 0$.

Hence a serious breakdown of the twosided Lanczos process occurs for some ℓ if and only if no regular FOP of degree ℓ exists.

Let $\{k_1, k_2, \dots\}$, $1 = k_1 < k_2 < \dots$, the set of indices such that for each $j = 1, \dots, m$ there exists a unique monic regular FOP $\psi_{k_j-1} \in \Pi_{k_j-1}$, and let

$$v^{k_j} := \psi_{k_j-1}(A) v^1, \; w^{k_j} := \psi_{k_j-1}(A^T) w^1.$$

These vectors are called regular vectors.

For $\ell := \min\{j : j \neq k_j\} - 1$ the vectors $v^1, \dots, v^\ell$ and $w^1, \dots, w^\ell$ can be constructed by the classical twosided Lanczos method. More generally, it can be shown (Gutknecht (1992)) that three consecutive FOPs $\psi_{k_{j-1}-1}$, ψ_{k_j-1}, $\psi_{k_{j+1}-1}$ are connected via a three term recurrence.

The look-ahead Lanczos process prevents possible break downs of the classical Lanczos method by relaxing the biorthogonality condition. It fills the gap between regular vectors v^{k_j} and $v^{k_{j+1}}$ (and between w^{k_j} and $w^{k_{j+1}}$) such that $k_{j+1} > k_j + 1$ adding vectors

$$v^k \in \mathcal{K}_k(v^1, A) \setminus \mathcal{K}_{k-1}(v^1, A),\ \ w^k \in \mathcal{K}_k(w^1, A^T) \setminus \mathcal{K}_{k-1}(w^1, A^T),$$

$k = k_j, \dots, k_{j+1} - 1,$. These inner vectors can be chosen such that the vectors v^j and w^k from different blocks are biorthogonal sind, where the vectors v^i, $i = k_j, \dots, k_{j+1} - 1$, are referred to as the j-th block.

Look-ahead Lanczos method

Choose v^1, w^1 with $\|v^1\|_2 = 1$; $\|w^1\|_2 = 1$;
$\tilde V_1 = v^1$; $\tilde W_1 = w^1$; $\tilde D_1 := \tilde W_1^T \tilde V_1$;
$\tilde V_0 = \emptyset$; $\tilde W_0 = \emptyset$; $v^0 = 0$; $w^0 = 0$;
$k_1 = 1$; $p = 1$; $\rho_1 = 1$; $\xi_1 = 1$;
for $k = 1, 2, \dots$
  if det $\tilde D_p \neq 0$;
    $\tilde v^{k+1} = Av^k - \tilde V_p \tilde D_p^{-1} \tilde W_p^T A v^k - \tilde V_{p-1} \tilde D_{p-1}^{-1} \tilde W_{p-1}^T A v^k$;
    $\tilde w^{k+1} = A^T w^k - \tilde W_p \tilde D_p^{-1} \tilde V_p^T A^T w^k - \tilde W_{p-1} \tilde D_{p-1}^{-1} \tilde V_{p-1}^T A^T w^k$;
    $k_{p+1} = k + 1$; $p = p + 1$; $\tilde V_p = \emptyset$; $\tilde W_p = \emptyset$;
  else
    $\tilde v^{k+1} = Av^k - \zeta_k v^k - (\eta_k/\rho_k) v^{k-1} - \tilde V_{p-1} \tilde D_{p-1}^{-1} \tilde W_{p-1}^T A v^k$;
    $\tilde w^{k+1} =$
$A^T w^k - \zeta_k w^k - (\eta_k/\rho_k) w^{k-1} - \tilde W_{p-1} \tilde D_{p-1}^{-1} \tilde V_{p-1}^T A^T w^k$;
    end

$\rho_{k+1} = \|\tilde{v}^{k+1}\|_2;$
$\xi_{k+1} = \|\tilde{w}^{k+1}\|_2;$
if $\rho_{k+1}\xi_{k+1} == 0$ STOP
$v^{k+1} = \tilde{v}^{k+1}/\rho_{k+1};$
$w^{k+1} = \tilde{w}^{k+1}/\xi_{k+1}$
$\tilde{V}_p = (\tilde{V}_p, v^{k+1});$
$\tilde{W}_p = (\tilde{W}_p, w^{k+1});$
$\tilde{D}_p = \tilde{W}_p^T \tilde{V}_p;$
end

Remarks:

1. ζ_k and η_k are coefficients which (with the exception of $\eta_{k_j} = 0$) can be chosen arbitrarily. Since the blocks, which are built in the look-ahead Lanczos method, are usually small (FG&N report that the maximum size that they observed in their numerical examples was 4) the choice of ζ_k and η_k is not of great influence on the total iteration.

2. In the algorithm above inner vectors are only determined if a serious breakdown would appear in exact arithmetic. In practice a look-ahead Lanczos algorithm must also be able to handle near break-downs, i.e. $((w^k)^T v^k \approx 0$, but $v^k \not\approx 0$, $w^k \not\approx 0)$. To check the regularity of $\tilde{D}_p$ in QMRPACK the minimal singular value of $\tilde{D}_p$ is monitored.

3. If only regular steps are performed then the look-ahead Lanczos process reduces to the twosided Lanczos method.

There exists a transpose free QMR method (Freund (1993)).

This is not a transpose free variant of the QMR method, but a QMR variant of the transpose free CGS method.

CGS method

$r^0 = b - Ax^0$; $d^0 = r^0$; $u^0 = r^0$;

Choose $\tilde{r}^0$ with $(\tilde{r}^0)^T r^0 \neq 0$

for $k = 0, 1, 2, \dots$ until convergence

$\tau_k = (\tilde{r}^0)^T r^k / (\tilde{r}^0) A d^k$;

$q^k = u^k - \tau_k A d^k$;

$x^{k+1} = x^k + \tau_k (u^k + q^k)$;

$r^{k+1} = r^k - \tau_k A (u^k + q^k)$;

$\beta_k = (\tilde{r}^0)^T r^{k+1} / (\tilde{r}^0)^T r^k$;

$u^{k+1} = r^{k+1} + \beta_k q^k$;

$d^{k+1} = u^{k+1} + \beta_k (q^k + \beta_k d^k)$;

end

The CGS algorithm can be interpreted in the following way: x^k is updated in two steps, $x^{k+1/2} = x^k + \tau_k u^k$ and $x^{k+1} = x^{k+1/2} + \tau_k q^k$ corresponding to the increase of the degree of the residual polynomial in each step by 2. To avoid indices which are multiples of $1/2$ we double the indices in the CGS method. Then the statements in the for-loop obtain the following form:

$$
\begin{aligned}
&\tau_{2k} = (\tilde{r}^0)^T r^{2k} / (\tilde{r}^0)^T A d^{2k};\\
&q^{2k} = u^{2k} - \tau_{2k} A d^{2k};\\
&x^{2k+2} = x^{2k} + \tau_{2k}(u^{2k} + q^{2k});\\
&r^{2k+2} = r^{2k} - \tau_{2k} A(u^{2k} + q^{2k});\\
&\beta_{2k} = (\tilde{r}^0)^T r^{2k+2} / (\tilde{r}^0)^T r^{2k};\\
&u^{2k+2} = r^{2k+2} + \beta_{2k} q^{2k};\\
&d^{2k+2} = u^{2k+2} + \beta_{2k}(q^{2k} + \beta_{2k} d^{2k});
\end{aligned}
$$

x^{2k+2} is updated in two steps:

$$
\begin{aligned}
x^{2k+1} &= x^{2k} + \tau_{2k} u^{2k};\\
x^{2k+2} &= x^{2k+1} + \tau_{2k} q^{2k};
\end{aligned}
$$

Putting $u^{2k+1} := q^{2k}$ and $\tau_{2k+1} := \tau_{2k}$ both statements read

$$x^m = x^{m-1} + \tau_{m-1} u^{m-1}.$$

For even m this is exactly the approximation from the CGS method, for odd m these approximations of the solution do not appear in the CGS method.

With the matrix

$$U_m = (u^0, \dots, u^{m-1})$$

and the vector

$$z^m = (\tau_0, \tau_1, \dots, \tau_{m-1})^T$$

the approximation x^m can be written as

$$x^m = x^0 + U_m z^m = x^{m-1} + \tau_{m-1} u^{m-1},$$

and the residuum is

$$r^m = r^0 - AU_m z^m = r^{m-1} - \tau_{m-1} A u^{m-1}.$$

The last equation yields

$$Au^m = \frac{1}{\tau_m}(r^m - r^{m+1}).$$

Hence,

$$AU_m = R_{m+1}\tilde{B}_m$$

with

$$R_{m+1} = (r^0, r^1, \dots, r^m) \quad \text{and}$$

$$\tilde{B}_m = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ -1 & 1 & \dots & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & & & -1 & 1 \\ 0 & 0 & \dots & \dots & -1 \end{pmatrix} \operatorname{diag}\left\{\frac{1}{\tau_0}, \frac{1}{\tau_1}, \dots, \frac{1}{\tau_{m-1}}\right\},$$

and the residuum is

$$r^m = r^0 - AU_m z_m = R_{m+1}(e^1 - \tilde{B}_m z^m).$$

If the columns of R_{m+1} are scaled by the diagonalmatrix $\Delta_{m+1} = \text{diag}\{\delta_0, \dots, \delta_m\}$, then on eobtains

$$r^m = R_{m+1}\Delta_{m+1}^{-1}\Delta_{m+1}(e^1 - \tilde{B}_m z^m) = R_{m+1}\Delta_{m-1}^{-1}(\delta_0 e^1 - \tilde{H}_m)$$

with $\tilde{H}_m := \Delta_{m+1}\tilde{B}_m$.

It is easily seen that the approximations x^{2m} of the CGS method are given by

$$x^{2m} = x^0 + U_{2m}H_{2m}^{-1}(\delta_0 e^1),$$

where H_{2m} is obtained from $\tilde{H}_{2m}$ by deleting the last row.

Solving (similarly as in the transition from the BiCG to the GMRES method) the least squares problem

$$\|\delta e^1 - \tilde{H}_m z^m\| = \min!,$$

then one obtains the approximation

$$x^m = x^0 + U_m z^m$$

of the TFQMR method.

To implement this method we still have to deduce how to update x^m for otherwise all previous vectors u^m would have to be stored. This can be found in
Saad: Iterative Methods for Sparse Linear Systems.

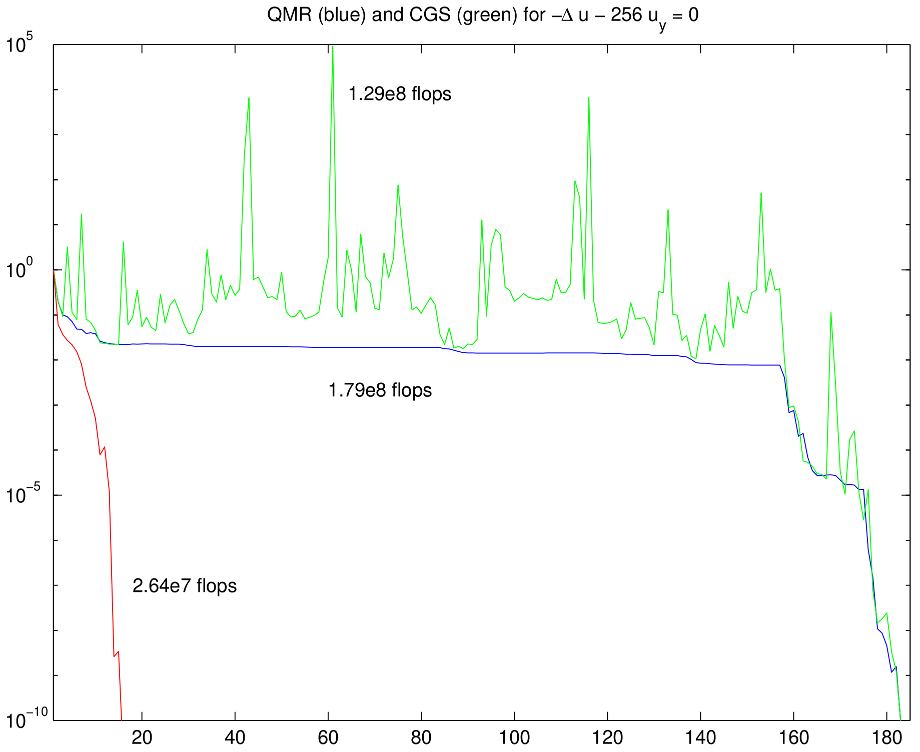
TFQMR method

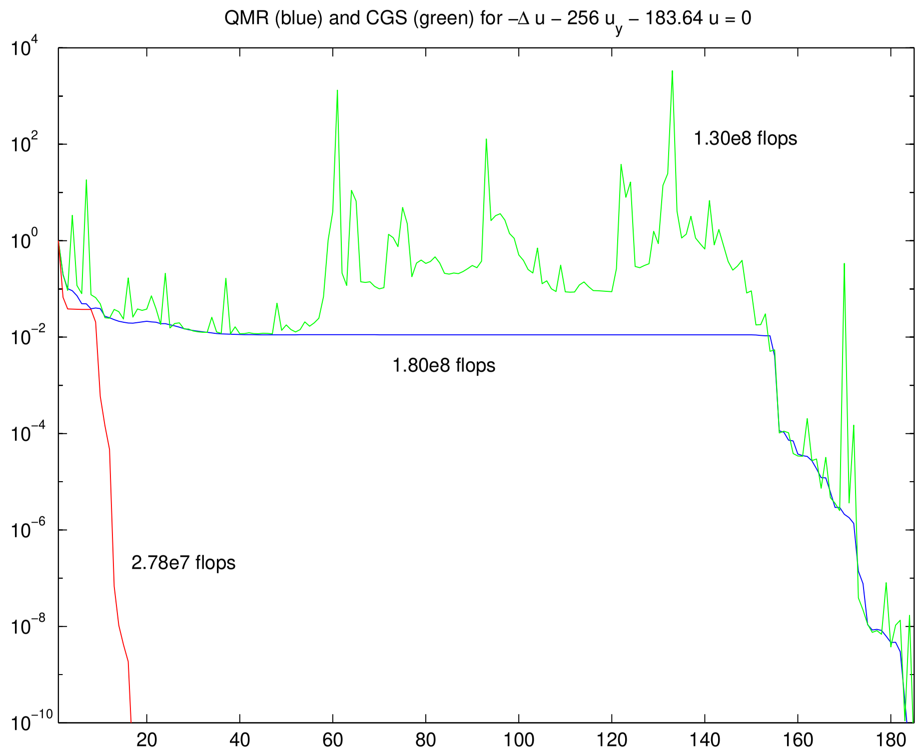
$r = b - Ax;\ \tilde{r} = r;\ w = r;\ u = r;\ v = Au;\ d = 0;$
$\tau = \|r\|_2;\ \theta = 0;\ \eta = 0;\ \rho = \tilde{r}^T r;\ z^0 = Au;$
for $k = 1, 2, \ldots$
  $\alpha_1 = \rho/(v^T \tilde{r});\ \alpha_0 = \alpha_1;$
  $u^0 = u;$
  $u = u - \alpha_0 v;$
  $w = w - \alpha_0 z^0;$
  $d = u^0 + \theta^2 \eta d/\alpha_0;$
  $\theta = \|w\|_2/\tau;$
  $c = 1/\sqrt{1+\theta^2};\ \tau = \tau\theta c;\ \eta = c^2\alpha_0;$
  $x = x + \eta d;$
  $z^1 = Au;$
  $w = w - \alpha_1 z^1;$
  $d = u + \theta^2 \eta d/\alpha_1;$

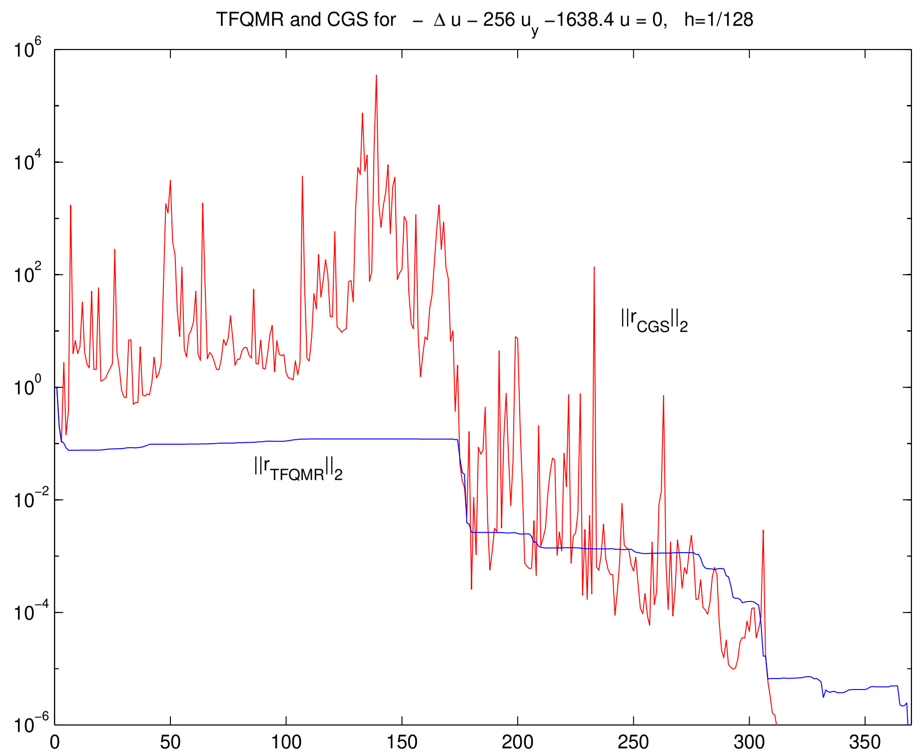
  $\theta = \|w\|_2/\tau;$
  $c = 1/\sqrt{1+\theta^2};\ \tau = \tau\theta c;\ \eta = c^2\alpha_1;$
  $x = x + \eta d;$
  $\beta = 1/\rho;\ \rho = \tilde{r}^T w;\ \beta = \beta\rho;$
  $u = w + \beta u;$
  $z^0 = Au;$
  $v = z^0 + \beta(z^1 + \beta v);$
end

Cost of TFQMR: 2 matrix-vector products
4 scalar products
10 _axpy

Storage requirements: 10 Vectors




QMR (blue) and CGS (green) for −Δ u − 256 u_y − 183.64 u = 0
1.30e8 flops
1.80e8 flops
2.78e7 flops


TFQMR and CGS for − Δ u − 256 u_y −1638.4 u = 0, h=1/128
||r_CGS||_2
||r_TFQMR||_2

In the same way as the QMR variant TFQMR was deduced from the CGS method Chan, Gallopoulos, Simoncini, Szeto und Tong (1994) derived the QMR variant QMRBiCGStab from the BiCGStab method.

QMRBiCGStab method

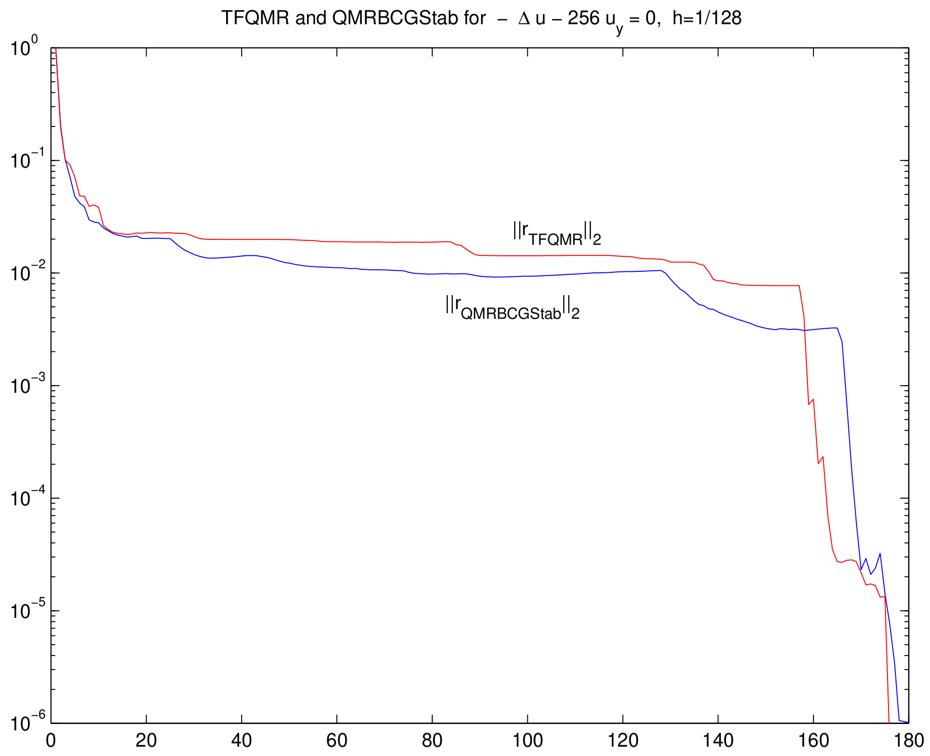
$r = b - Ax$; $\tilde{r} = r$; $u = r$; $v = Au$; $d = 0$;
$\tau = \|r\|_2$; $\rho = \tilde{r}^T r$; $\theta = 0$; $\eta = 0$;
for $k = 1, 2, \ldots$ until convergence

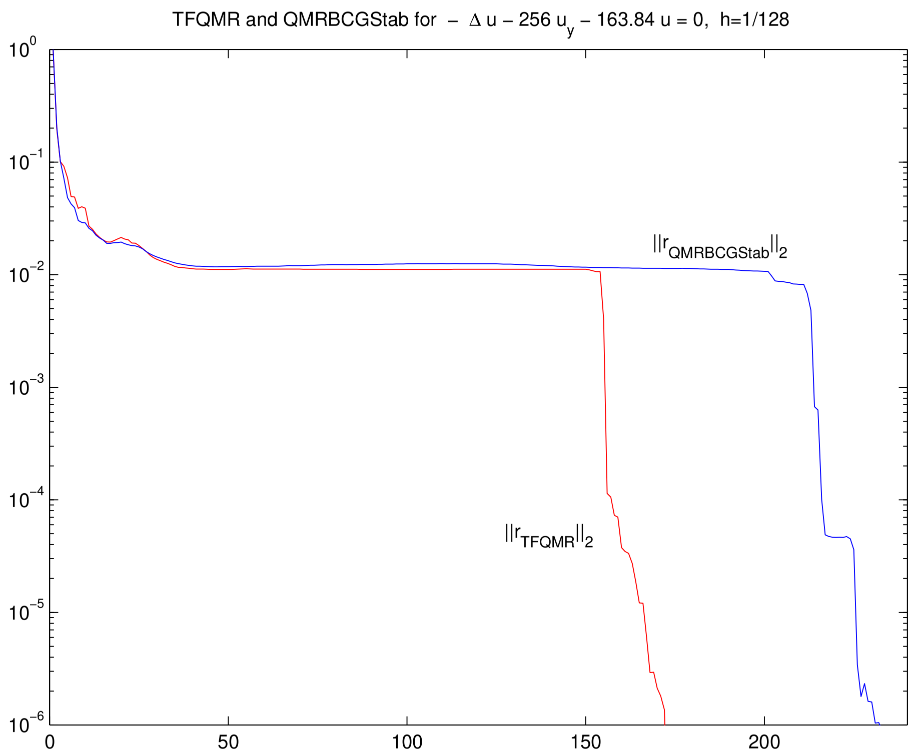
$$
\begin{aligned}
&\alpha = \rho/(\tilde{r}^T v);\\
&s = r - \alpha v;\\
&\tilde{\theta} = \|s\|_2/\tau;\\
&c = 1/\sqrt{1+\tilde{\theta}^2};\\
&d = u + \theta^2 \eta d/\alpha;\\
&\eta = c^2\alpha;\\
&x = x + \eta d;\\
&\tau = \tau\tilde{\theta}c;\\
&w = As;\\
&\omega = w^T s/(w^T w);\\
&r = s - \omega w;
\end{aligned}
$$

$\theta = \|r\|_2/\tau;$
$c = 1/\sqrt{1+\theta^2};$
$d = s + \tilde{\theta}^2 \eta d/\omega;$
$\eta = c^2\omega;$
$x = x + \eta d;$
$\tau = \tau\theta c;$
$\beta = 1/\rho;$
$\rho = \tilde{r}^T r;$
$\beta = \beta\rho\alpha/\omega;$
$u = r + \beta(u - \omega v);$
$v = Au;$
end

Cost of QMRBiCGStab: 2 matrix-vector products
6 scalar products
8 _axpy

Storage requirements: 8 vectors


TFQMR and QMRBCGStab for − Δ u − 256 u_y = 0, h=1/128
||r_TFQMR||_2
||r_QMRBCGStab||_2
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
0
20
40
60
80
100
120
140
160
180


TFQMR and QMRBCGStab for − Δ u − 256 u_y − 163.84 u = 0, h=1/128
||r_QMRBCGStab||_2
||r_TFQMR||_2
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
0
50
100
150
200

TFQMR and QMRBCGStab for $-\Delta u - 256\, u_y - 1638.4\, u = 0$, h=1/128