# The construction of bases for Krylov subspaces

Gérard MEURANT

Joint work with Lothar Reichel

ICS, March 2016

# Introduction

GMRES is one of the best Krylov method for solving non symmetric linear systems

$$Ax = b$$

where $A$ is a real non symmetric matrix

The Krylov subspace is

$$\mathcal{K}_m(A, r) = \{r, Ar, A^2 r, \ldots, A^{m-1} r\}$$

where $r = b - Ax_0$ (usually $x_0 = 0$)

Generally GMRES is restarted every $m$ iterations $\rightarrow$ GMRES($m$)

GMRES uses an "orthonormal" basis constructed with the Arnoldi process
It is modified Gram-Schmidt adapted to the Krylov subspace

$v_1 = r/\|r\|$
For $k = 1, \ldots, m$
 $w = Av_k$
 For $j = 1, \ldots, k$
  $h_{j,k} = v_j^T w$
  $w = w - h_{j,k} v_j$
 end (for $j$)
 $h_{k+1,k} = \|w\|, \;\; v_{k+1} = w/h_{k+1,k}$
end (for $k$)

$$AV_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T$$

with $V_k = (v_1, \dots, v_k)$, $H_k = (h_{i,j})$

GMRES-MGS has been proved to be backward stable (Paige, Rozloznik and Strakoš, SIMAX, v 28 (2006))

But, there are data dependencies which are not good on a parallel computer

How to introduce more parallelism in GMRES?

Many people have been working on this problem; see, for instance:

Z. Bai, D. Hu, and L. Reichel, *A Newton basis GMRES implementation*, IMA J. Numer. Anal., 14 (1994)

J. Erhel, *A parallel GMRES version for general sparse matrices*, Electron. Trans. Numer. Anal., 3 (1995)

R.B. Sidje, *Alternatives to parallel Krylov subspace basis computation*, Numer. Linear Algebra Appl., 4 (1997)

B. Philippe and L. Reichel, *On the generation of Krylov subspace bases*, Applied Numerical Mathematics, 62 (2012)

# Newton basis

One possibility is to use a Newton basis constructed as

$$v_1 = r/\|r\|, \quad v_{i+1} = \frac{1}{\eta_i}(A - \xi_i I)v_i, \; i = 1, \dots, m-1$$

where the $\eta_i$'s are normalizing factors to have vectors of unit norm and the $\xi_i$'s $\in \mathbb{C}$ are known as *shifts*

If $V_m = [v_1, v_2, \dots, v_m]$ is of rank $m$, we have a basis of the Krylov subspace

If we use a preconditioner $M$, $A$ has to be replaced either by $M^{-1}A$ or $AM^{-1}$

Note that $v_{m+1} = p_m(A)v_1$ and the $\xi_i$'s are the roots of $p_m$

If the $\xi_i$'s appear in complex conjugate pairs, everything can be implemented in real arithmetic

Algorithm GMRESN($m$)

For $k = 1, \ldots, m$
 Add a vector to the known Newton basis $V_k$
end (for $k$)

Orthogonalize $V_{m+1}$ with (parallel) QR

Compute $H_{m+1,m}$

Compute the iterates $x_k$ by reducing $H_{m+1,m}$ to upper triangular structure

Note that $V_m$ is tall and skinny ($m \ll n$)

There exist good (recent) parallel QR algorithms for such matrices

Problem: How to compute "good" shifts?

- $\|V_m\|$ small
- $\sigma_{\min}(V_m)$ large

We consider two possibilities:

- Spoke sets (Reichel and Philippe)
- "Good" interpolation points in the complex plane

We first do $p$ initial steps of GMRES and we compute the Ritz values (eigenvalues of $H_p$)

Typically $p = 10$

$H_p$ is real, the Ritz values are real or complex conjugate pairs

# Spoke sets

We compute the center of gravity $G$ (a point on the $x$-axis) of the Ritz values $R_i$

For each segment $GR_i$ we compute fast Leja points

see J. Baglama, D. Calvetti and L. Reichel, *Fast Leja points*, Electron. Trans. Numer. Anal., 7 (1998)

# “Good” interpolation points

A minimal volume ellipse containing the Ritz values is computed

Then a set $X$ of $N \geq m$ points is computed in the upper half-ellipse to obtain a small Lebesgue constant

The $N$ Lagrange polynomials $\ell_j$ related to these points are such that the polynomial $\ell_j$ is equal to 1 at point $\xi_j$ and zero at the other points

The Lebesgue function is

$$\Lambda_X(\xi) = \sum_{j=1}^{m} |\ell_j(\xi)|$$

The Lebesgue constant $\ell_X$ is the maximum of the Lebesgue function over $\xi \in \Omega$ where $\Omega$ is our given half-ellipse

We solve the problem

$$\min_{X,\,\xi_j \in \bar{\Omega}} I_{\Omega,\partial\Omega}, \quad I_{\Omega,\partial\Omega} = \left\{ I_\Omega^2 + \mu \int_{\partial\Omega} [\ell_1(\xi)^2 + \cdots + \ell_N(\xi)^2] \, ds \right\}^{\frac{1}{2}}$$

where $\mu$ is a positive real number and

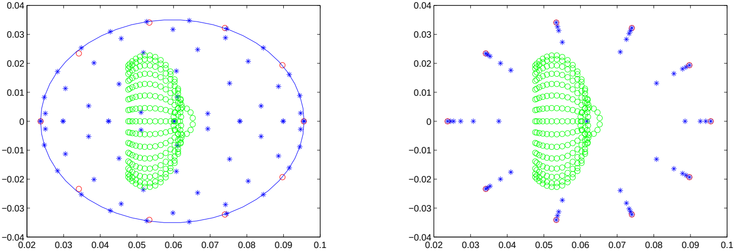
$$I_\Omega = \left\{ \int_\Omega [\ell_1(x,y)^2 + \cdots + \ell_N(x,y)^2] \, dxdy \right\}^{\frac{1}{2}}$$

This does not always give the smallest possible Lebesgue constants therefore we added heuristic refinement algorithms which, in many cases, improve the results

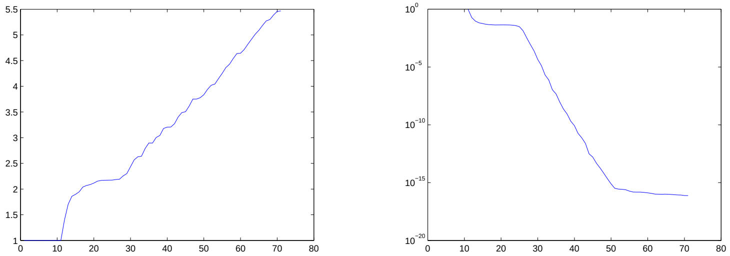
We pre-computed points for the half-ellipse $\Omega_0$ of center $(0, 0)$ with semi-axes 2 (in $x$) and 1 (in $y$) for different total degrees of the bivariate polynomial

When we have the minimal volume ellipse $\Omega$ we compute the needed degree for having at least $m$ points and we map the (symmetrized) points in $\Omega_0$ to $\Omega$

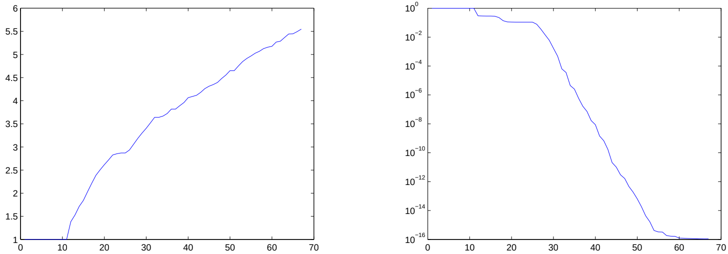
# Examples



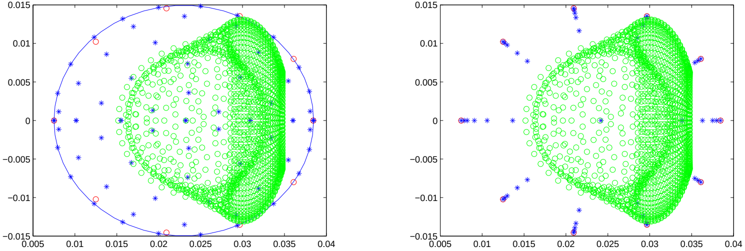

supg001 225, eigenvalues, half-ellipse and spoke sets



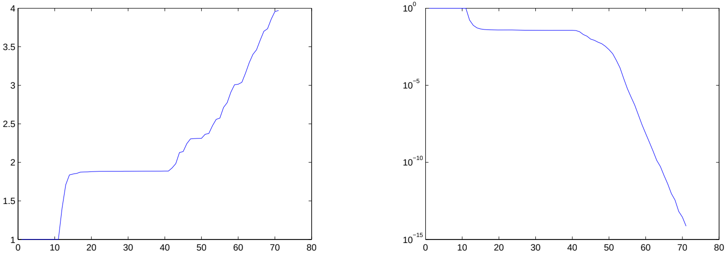

supg001 225, half-ellipse, norm of $V_k$ and $\sigma_{\min}(V_k)$



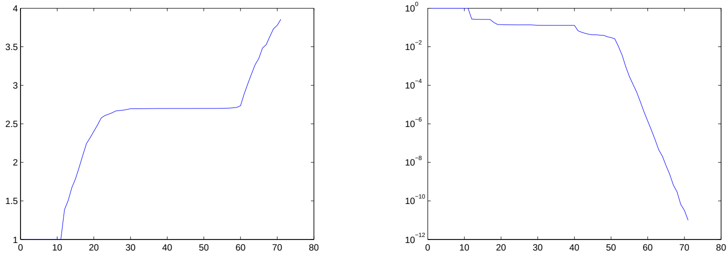

supg001 225, spoke sets, norm of $V_k$ and $\sigma_{\min}(V_k)$




supg001 1600, eigenvalues, half-ellipse and spoke sets




supg001 1600, half-ellipse, norm of $V_k$ and $\sigma_{\min}(V_k)$



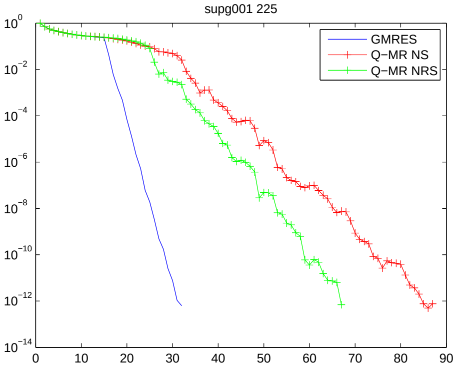

supg001 1600, spoke sets, norm of $V_k$ and $\sigma_{\min}(V_k)$

As long as the basis is full rank (for $m$ small) we obtain the same results as with GMRES(m)

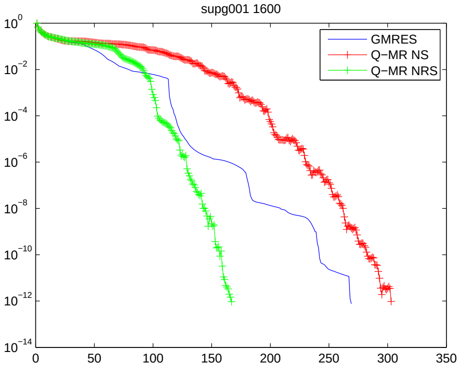
Could we use the Newton bases with Q-MR?

That is, without the QR step

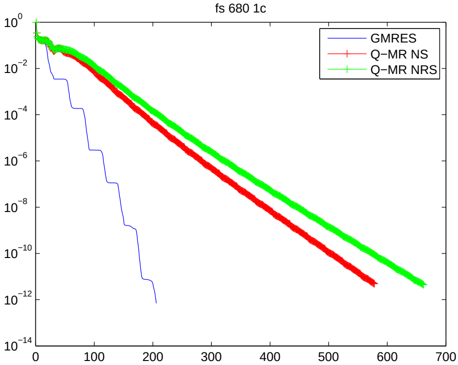
$\Rightarrow$ We just minimize the norm of the quasi-residual



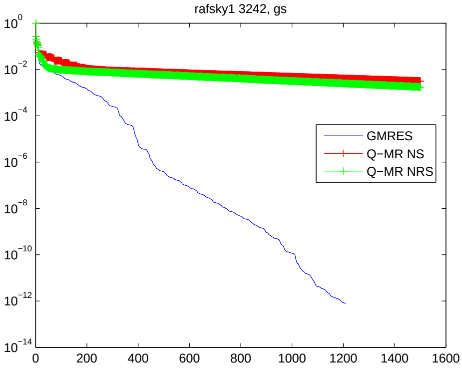

supg001 225, GMRES(30), Q-MR(30) with half-ellipse and spoke sets




supg001 1600, GMRES(30), Q-MR(30) with half-ellipse and spoke sets




fs 680 1c, GMRES(30), Q-MR(30) with half-ellipse and spoke sets




raefsky1 3242, GMRES(30), Q-MR(30) with half-ellipse and spoke sets

# Q-MR with loose orthogonalization

Q-MR is not always working well with the Newton basis

We just need the "R" factor of a QR factorization of $V_{m+1}$

$$AV_m = V_{m+1} T_{m+1,m}$$

where $T_{m+1,m}$ is with at most an upper bandwidth of 2

$$V_{m+1} = Q_{m+1} R_{m+1,m+1}$$

It yields

$$AV_m = Q_{m+1} R_{m+1,m+1} T_{m+1,m}$$

and if $x_m = x_0 + V_m z$, we have

$$\|b - Ax_m\| = \|r_0 - AV_m z\| = \| Q_{m+1}(\|r_0\| e_1 - R_{m+1,m+1} T_{m+1,m} z)\|$$

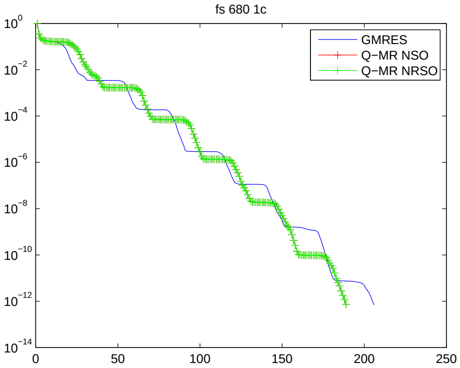
We can use CholQR that is, the Cholesky factorization

$$V_{m+1}^T V_{m+1} = R_{m+1,m+1}^T R_{m+1,m+1}$$

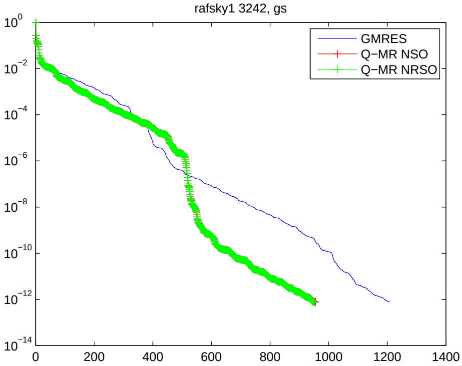
However, when $\sigma_{\min}(V_{m+1})$ is small $V_{m+1}^T V_{m+1}$ might not be numerically positive definite

If this happens, we use CGS (classical Gram-Schmidt)

Note that this is very close to what we did for GMRES except that the orthogonalization is done differently




fs 680 1c, GMRES(30), CholQR Q-MR(30) with half-ellipse and spoke sets




raefsky1 3242, GMRES(30), CholQR Q-MR(30) with half-ellipse and spoke sets

# Conclusion

- Newton bases + QR work well for GMRES($m$)
- Interpolation points and spoke sets yield similar results
- Newton bases do not always work well with Q-MR
- But Newton bases + loose orthogonalization could give nice results

# Other bases

Let $\mathcal{X}$ be an arbitrary subspace of dimension $k$ in $\mathbb{C}^n$ with a basis $W = [w_1, \ldots, w_k]$ and let $w \notin \mathcal{X}$
Let $\mathcal{P}$ be the orthogonal projector onto $\mathcal{X}$. Then,

$$\|(I - \mathcal{P})w\|^2 = w^*w - w^*W(W^*W)^{-1}W^*w$$

Let us assume that we have a basis with unit norm vectors $v_j, j = 1, \ldots, k$ with $v_1 = r/\|r\|$ and

$$V_k = \begin{pmatrix} v_1 & \ldots & v_k \end{pmatrix}$$

Let us construct $v_{k+1}$ as

$$\tilde{v} = \begin{pmatrix} AV_k & v_k \end{pmatrix} y, \quad v_{k+1} = \tilde{v}/\|\tilde{v}\|$$

Hence, the new vector is a linear combination of the columns of $AV_k$ and the last known basis vector $v_k$

Clearly, this is in $\mathcal{K}_{k+1}(A, r)$ since $V_k y$ is in $\mathcal{K}_k(A, r)$

Now we have to compute the coefficients in $y$

We consider the distance of $v_{k+1}$ to $\mathcal{K}_k(A, r)$

Let $B_k = \begin{pmatrix} AV_k & v_k \end{pmatrix}$. The square of the distance of $v_{k+1}$ to $\mathcal{K}_k(A, r)$ is

$$d_k = 1 - \frac{y^T B_k^T V_k (V_k^T V_k)^{-1} V_k^T B_k y}{y^T B_k^T B_k y}$$

The squared distance $d_k$ is less than or equal to 1

There is a vector $y$ such that $y^T B_k^T V_k (V_k^T V_k)^{-1} V_k^T B_k y = 0$, $B_k^T B_k$ is positive definite

With $B_k = \begin{pmatrix} AV_k & v_k \end{pmatrix}$ the basis is orthonormal and the matrix $B_k^T V_k (V_k^T V_k)^{-1} V_k^T B_k$ is singular

The vector $y$ can be chosen as an eigenvector corresponding to the zero eigenvalue

# An Arnoldi-like relation

Let $\eta_k = \|B_k y\|$. Then

$$\eta_k v_{k+1} = y_{k+1} v_k + y_1 A v_1 + \cdots + y_k A v_k$$

Therefore, (provided that $y_k \neq 0$),

$$A v_k = \frac{1}{y_k}[\eta_k v_{k+1} - y_{k+1} v_k - y_1 A v_1 - \cdots - y_{k-1} A v_{k-1}]$$

In matrix form

$$A V_m = V_m L_m - A V_m R_m + \frac{\eta_m}{y_m} v_{m+1} e_m^T$$

where $L_m$ is a lower bidiagonal matrix and $R_m$ is a strictly upper triangular matrix

The matrix $I + R_m$ is upper triangular with ones on the diagonal

$$AV_m = V_m L_m [I + R_m]^{-1} + \frac{\eta_m}{y_m} v_{m+1} e_m^T [I + R_m]^{-1}$$

But, $e_m^T [I + R_m]^{-1} = e_m^T$ and we finally get

$$AV_m = V_m H_m + \frac{\eta_m}{y_m} v_{m+1} e_m^T$$

with $H_m = L_m [I + R_m]^{-1}$

Another possible choice for $B_k$ is $B_k = \begin{pmatrix} Av_k & V_k \end{pmatrix}$

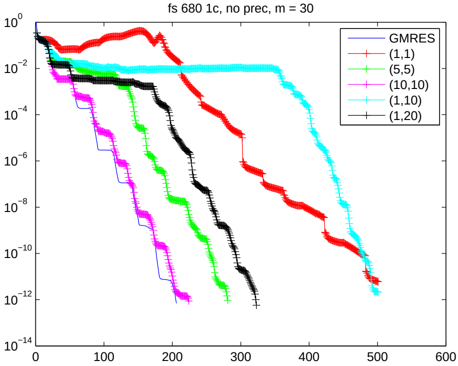
This yields the Arnoldi process (in exact arithmetic) and another orthonormal basis

More generally we can choose

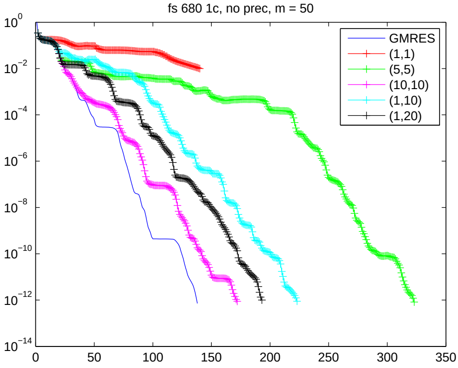
$$B_k = \begin{pmatrix} AV_{\max(1,k-p+1):k} & V_{\max(1,k-q+1):k} \end{pmatrix}$$

where $p$ and $q$ are positive integers
This means that for computing the new vector $v_{k+1}$ we just retain the $p$ and $q$ last vectors

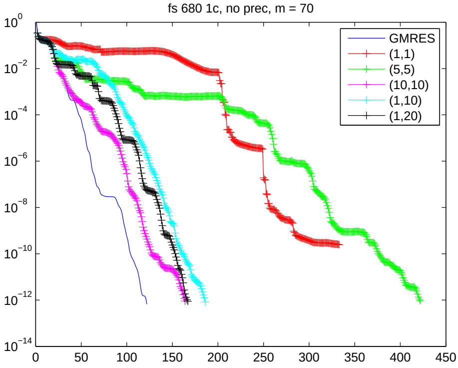
With different choices of $p$ and $q$ we cannot always obtain orthonormal bases, but we can use them with Q-MR



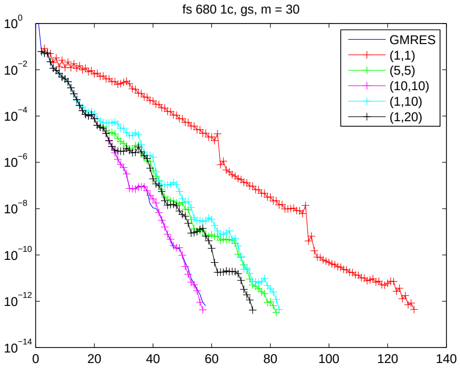

fs 680 1c, GMRES(30), Q-MR(30) with different values of $p$ and $q$



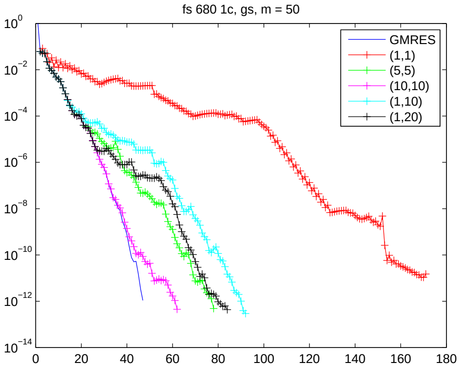

fs 680 1c, GMRES(50), Q-MR(50) with different values of $p$ and $q$



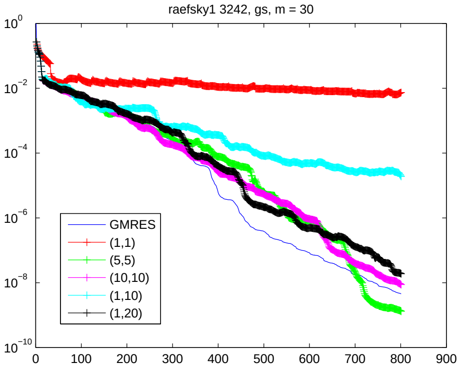

fs 680 1c, GMRES(70), Q-MR(70) with different values of $p$ and $q$



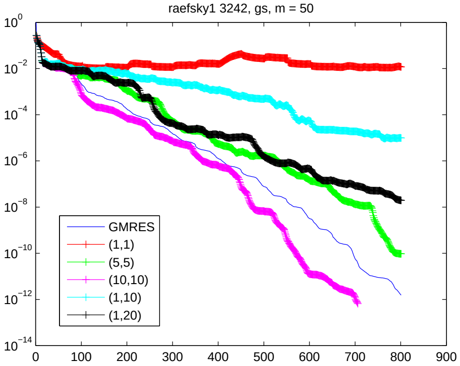

fs 680 1c, GMRES(30), Q-MR(30) with different values of $p$ and $q$, Gauss-Seidel preconditioner



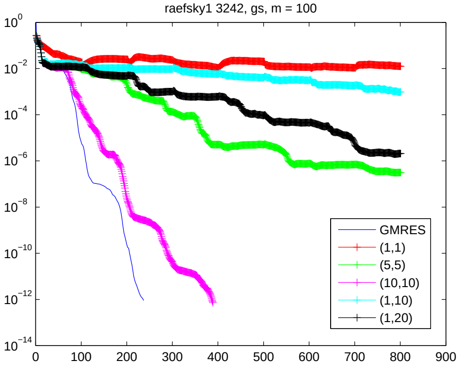

fs 680 1c, GMRES(50), Q-MR(50) with different values of $p$ and $q$, Gauss-Seidel preconditioner




raefsky1 3242, GMRES(30), Q-MR(30) with different values of $p$ and $q$, Gauss-Seidel preconditioner




raefsky1 3242, GMRES(50), Q-MR(50) with different values of $p$ and $q$, Gauss-Seidel preconditioner




raefsky1 3242, GMRES(100), Q-MR(100) with different values of $p$ and $q$, Gauss-Seidel preconditioner

# Conclusion

- Q-MR works well for some small values of $p$ and $q$
- But, the storage is worse than with GMRES($m$)
- There are more dot products
- But, they occur as matrix-vector or matrix-matrix products
- So, there is room for parallelization

What about the stability of these methods?

How to choose $p$ and $q$ in practice?