# Outline

$$K_m(A, \underline{r}^{(0)}) = \text{span}\{\underline{r}^{(0)}, A\underline{r}^{(0)}, \ldots, A^{m-1}\underline{r}^{(0)}\}$$

$\downarrow$

ARNOLDI PROCESS

$$= \text{span}\{\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_m\}$$

ARNOLDI'S RELATION :

upper Hessenberg $m \times m$

$$A V_m = V_{m+1} \bar{H}_m$$

$$= \left(V_m \,\vdots\, \underline{v}_{m+1}\right) \begin{pmatrix} H_m \\ \cdots\cdots\cdots \\ \underline{h_{m+1,m}\, \underline{e}_m^T} \end{pmatrix} (0, 0, \ldots, h_{m+1,m})$$

$$= V_m H_m + h_{m+1,m}\, \underline{v}_{m+1} \underline{e}_m^T .$$

# Arnoldi's Method

- *Arnoldi's method* or the *Arnoldi process* makes use of the Gram-Schmidt process to generate an orthonormal basis $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ for the Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{r}^{(0)})$.
- The *Arnoldi process* produces the following decomposition:

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_m^T \tag{5}$$

or equivalently:

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m\,, \qquad \overline{\mathbf{H}}_m = \left[\begin{array}{c} \mathbf{H}_m \\ \hdashline \underbrace{h_{m+1,m}\mathbf{e}_m^T} \end{array}\right] \tag{6}$$

where the columns of $\mathbf{V}_m = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m] \in \mathbb{R}^{n\times m}$ and $\mathbf{V}_{m+1} = [\mathbf{V}_m, \mathbf{v}_{m+1}] \in \mathbb{R}^{n\times(m+1)}$ form an orthonormal basis for $\mathcal{K}_m(\mathbf{A}, \mathbf{r}^{(0)})$ and $\mathcal{K}_{m+1}(\mathbf{A}, \mathbf{r}^{(0)})$, respectively.

# Arnoldi's Method

- This means that $\mathbf{V}_m^T\mathbf{V}_m = \mathbf{I}$ and $\mathbf{V}_{m+1}^T\mathbf{V}_{m+1} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Note also that $\mathbf{V}_m^T\mathbf{A}\mathbf{V}_m = \mathbf{H}_m$.
- The rectangular matrix $\overline{\mathbf{H}}_m \in \mathbb{R}^{(m+1)\times m}$ is an upper Hessenberg matrix:

m

upper Hessenberg.

$$\overline{\mathbf{H}}_m = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m-1} & h_{2,m} \\ & h_{3,2} & \cdots & h_{3,m-1} & h_{3,m} \\ & & \ddots & \vdots & \vdots \\ & & & h_{m,m-1} & h_{m,m} \\ 0 & 0 & \cdots & 0 & h_{m+1,m} \end{bmatrix} \begin{array}{l} \\ \\ m \\ \\ \\ \end{array} \quad m+1.$$

# Arnoldi's Method

- When the Classical Gram-Schmidt (CGS) Algorithm is implemented in a floating point environment the basis vectors (columns of $\mathbf{V}_m$) are sometimes not quite orthogonal due to rounding errors.
- This can be rectified by the small modification highlighted in blue in the algorithm below. This new version is referred to as Modified Gram-Schmidt (MGS).

## CLASSICAL GRAM-SCHMIDT PROCESS (CGS)

$$\underline{\omega}_{m+1} = A\underline{v}_m - P_{K_m}(A\underline{v}_m)$$

$$= (A\underline{v}_m) - \sum_{j=1}^{m} (\underline{v}_j^T A\underline{v}_m)\underline{v}_j$$

$$\underline{v}_{m+1} = \underline{\omega}_{m+1} / \|\underline{\omega}_{m+1}\|_2$$

## MODIFIED GRAM-SCHMIDT PROCESS (MGS)

$(I - \underline{v}_2\underline{v}_2^T)$

$$\underline{\omega}_{m+1} = (I - \underline{v}_m\underline{v}_m^T)(I - \underline{v}_{m-1}\underline{v}_{m-1}^T)\cdots\underbrace{(I - \underline{v}_1\underline{v}_1^T)A\underline{v}_m}$$

$$\underline{v}_{m+1} = \underline{\omega}_{m+1} / \|\underline{\omega}_{m+1}\|_2$$

MAIN OBSERVATION

CGS:

$$\underline{w}_{m+1} = \left[ I - \sum_{j=1}^{m} \underline{v}_j \cdot \underline{v}_j^T \right] A\underline{v}_m$$

$$= \left[ I - \underline{v}_1 \underline{v}_1^T - \underline{v}_2 \underline{v}_2^T - \dots - \underline{v}_m \underline{v}_m^T \right] A\underline{v}_m$$

MGS: DUE TO ORTHOGONALITY

$$(I - \underline{v}_m \underline{v}_m^T)(I - \underline{v}_{m-1} \underline{v}_{m-1}^T)$$

$$= I - \underline{v}_{m-1} \underline{v}_{m-1}^T - \underline{v}_m \underline{v}_m^T + \underline{v}_m \underbrace{\cancel{(\underline{v}_m^T \underline{v}_{m-1})}}_{=0} \underline{v}_{m-1}^T$$

$\therefore$ CONTINUING THE PROCESS

$$\underline{w}_{m+1} = (I - \underline{v}_m \underline{v}_m^T)(I - \underline{v}_{m-1} \underline{v}_{m-1}^T) \cdots (I - \underline{v}_1 \underline{v}_1^T) A\underline{v}_m$$

$$= \left[ I - \underline{v}_m \underline{v}_m^T - \underline{v}_{m-1} \underline{v}_{m-1}^T - \dots - \underline{v}_1 \underline{v}_1^T \right] A\underline{v}_m$$

a bases for $K_m$ is $\{\underline{r}^{(0)}, A\underline{r}^{(0)}, \ldots A^{m-1}\underline{r}^{(0)}\}$

$AK_m$ $\{A\underline{r}^{(0)}, A^2\underline{r}^{(0)}, \ldots, A^m\underline{r}^{(0)}\}$.

provided $A$ is nonsingular then this is a basis for $AK_m$.

The only solution of

$$c_1\underline{r}^{(0)} + c_2 A\underline{r}^{(0)} + \ldots + c_m A^{m-1}\underline{r}^{(0)} = \underline{0}$$

is the trivial solution $c_1 = c_2 = \ldots = c_m = 0$.

Consider $b_1 A\underline{r}^{(0)} + b_2 A^2\underline{r}^{(0)} + \ldots + b_m A^m\underline{r}^{(0)} = \underline{0}$

$$A\underbrace{\left(b_1\underline{r}^{(0)} + b_2 A\underline{r}^{(0)} + \ldots + b_m A^{m-1}\underline{r}^{(0)}\right)}_{\underline{z}} = \underline{0}$$

If $A$ is nonsingular, the only solution is

$$\underline{z} = b_1\underline{r}^{(0)} + b_2 A\underline{r}^{(0)} + \ldots + b_m A^{m-1}\underline{r}^{(0)} = \underline{0}$$

$\therefore$ $b_1 = b_2 = \ldots = b_m = 0$ because $\{\underline{r}^{(0)}, \ldots, A^{m-1}\underline{r}^{(0)}\}$ is linearly indep. $\square$

# Generalised Minimal Residual Method (GMRES)

- For the Generalised Minimal Residual Method (GMRES) the constraint space is chosen as $\mathcal{W}_m = \mathbf{A}\mathcal{K}_m(\mathbf{A}, \mathbf{r}^{(0)})$.
- An approximate solution $\mathbf{x}^{(m)}$ is sought from the affine space $x^{(m)} \in \mathbf{x}^{(0)} + \mathcal{K}_m(\mathbf{A}, \mathbf{r}^{(0)})$ by imposing the *Petrov-Galerkin condition*:

$$\mathbf{r}^{(m)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(m)} \perp \overbrace{\mathbf{A}\mathcal{K}_m(\mathbf{A}, \mathbf{r}^{(0)})}^{\{A\underline{v}_1, A\underline{v}_2, \ldots, A\underline{v}_m\}}\,. \tag{7}$$

- Since we seek $\mathbf{x}^{(m)} \in \mathbf{x}^{(0)} + \mathcal{K}_m(\mathbf{A}, \mathbf{r}^{(0)})$ we assume the following form:

$$\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + \mathbf{V}_m\mathbf{y}_m \quad = [\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_m] \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}.$$

where $\mathbf{y}_m \in \mathbb{R}^m$ is a vector to be determined and the columns of $\mathbf{V}_m$ are the orthonormal basis vectors of $\mathcal{K}_m(\mathbf{A}, \mathbf{r}^{(0)})$ generated using Arnoldi's method.

$$\underline{r}^{(m)} = \underline{b} - A\{\underline{x}^{(0)} + V_m \underline{y}_m\}$$
$$= \underline{r}^{(0)} - (AV_m)\underline{y}_m$$
$$= \underline{r}^{(0)} - V_{m+1}\overline{H}_m \underline{y}_m$$
$$= V_{m+1}\{\beta_0 \underline{e}_1 - \overline{H}_m \underline{y}_m\}.$$

ARNOLDI RELATION

$$AV_m = V_{m+1}\overline{H}_m$$

$$\underline{v}_1 = \frac{\underline{r}^{(0)}}{\beta_0}, \quad \beta_0 = \|\underline{r}^{(0)}\|_2$$

$$\boxed{\beta_0 V_{m+1}\underline{e}_1 = \underline{r}^{(0)}}$$

APPLY PETROV-GALERKIN CONDITION

$$(AV_m)^T \underline{r}^{(m)} = \underline{0}$$

$$\boxed{A\underline{x} = \underline{b}}$$

"$n \times n$"

Projection

SUBSTITUTION GIVES

$$\overline{H}_m^T \underbrace{(V_{m+1}^T V_{m+1})}_{= I_{m+1}}\{\beta_0 \underline{e}_1 - \overline{H}_m \underline{y}_m\} = \underline{0}$$

(NORMAL EQUATIONS ??)

HENCE "$m \times m$"

$$\boxed{\overline{H}_m^T \overline{H}_m \underline{y}_m = \beta_0 \overline{H}_m^T \underline{e}_1}$$

$m \times m{+}1 \; m{+}1 \times m$

$\therefore$ $\underline{y}_m$ is therefore a solution of the Least Squares problem

# Generalised Minimal Residual Method (GMRES)

- The orthgonality condition (7) yields the following $m$-dimensional system to be solved for $\mathbf{y}_m$:

$$\overline{\mathbf{H}}_m^T \overline{\mathbf{H}}_m \mathbf{y}_m = \overline{\mathbf{H}}_m^T \beta \mathbf{e}_1 \,. \tag{8}$$

where $\beta = \|\mathbf{r}^{(0)}\|$ and $\mathbf{e}_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^{m+1}$.

- Note that $\overline{\mathbf{H}}_m \in \mathbb{R}^{(m+1)\times m}$ so $\overline{\mathbf{H}}_m^T \overline{\mathbf{H}}_m \in \mathbb{R}^{m \times m}$.
- The system of linear equations (8) are the *normal equations* associated with the $(m+1) \times m$ linear system:

$$\boxed{\overline{\mathbf{H}}_m \mathbf{y}_m \approx \beta \mathbf{e}_1 .} \tag{9}$$

$\underline{x}^{(m)} = \underline{x}^{(0)} + V_m y_m$

# Generalised Minimal Residual Method (GMRES)

- Hence the solution of (8) is equivalent to the *least squares solution* of (9), which be computed in MATLAB using the *backslash operator* $\mathbf{y}_m = \overline{\mathbf{H}}_m \backslash (\beta_0 \mathbf{e}_1)$. $= \overline{H}_m^{+}(\beta_0 \underline{e}_1)$
- The GMRES approximate solution is therefore given by:

$$\boxed{\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + \beta_0 \mathbf{V}_m \overline{\mathbf{H}}_m^{\dagger} \mathbf{e}_1} \tag{10}$$

where $\overline{\mathbf{H}}_m^{\dagger}$ is the *pseudoinverse* of $\overline{\mathbf{H}}_m$, which can be determined from the singular value decomposition of $\mathbf{A}$. The residual vector simplifies to:

$$\|\mathbf{r}^{(m)}\|_2^2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}^{(m)}\| = \|\mathbf{V}_{m+1}\big(\beta_0 \mathbf{e}_1 - \overline{\mathbf{H}}_m \mathbf{y}_m\big)\|_2^2 = \underline{z}^T \underbrace{V_{m+1}^T V_{m+1}}_{= I_{m+1}} \underline{z} = \underline{z}^T \underline{z} \tag{11}$$

and therefore:

$$\left\|\mathbf{r}^{(m)}\right\|_2 = \|\beta_0 \mathbf{e}_1 - \overline{\mathbf{H}}_m \mathbf{y}_m\|_2 . \tag{12}$$

# Generalised Minimal Residual Method (GMRES)

In summary, GMRES involves the following steps:

1. Choose $\mathbf{x}^{(0)} = 0$ and compute $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$, $\beta = \|\mathbf{r}^{(0)}\|$ and $\mathbf{v}_1 = \mathbf{r}^{(0)}/\beta$.
2. Generate $\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m$ using the Arnoldi algorithm. (MGS).
3. Solve the least squares problem $\min_{\mathbf{y}\in\mathbb{R}^m} \|\beta\mathbf{e}_1 - \overline{\mathbf{H}}_m\mathbf{y}\|_2$ for $\mathbf{y}_m$.
4. Compute residual norm $\|\mathbf{r}^{(m)}\|_2 = \|\beta\mathbf{e}_1 - \overline{\mathbf{H}}_m\mathbf{y}_m\|_2$.
   $y_m = \underset{y\in\mathbb{R}^m}{\arg\min} \|\beta e_1 - \overline{H}_m y\|_2$
   Check if
   $$\left\|\mathbf{r}^{(m)}\right\|_2 < tol \left\|\mathbf{r}^{(0)}\right\|_2$$
   and if so, go to step 5 else go back to step 2. A typical value of the tolerance *tol* is $10^{-8}$.
5. Compute approximate solution $\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + \mathbf{V}_m\mathbf{y}_m$.

# SOLUTION OF LEAST SQUARES PROBLEM

$$\min_{\underline{y} \in \mathbb{R}^m} \| \beta_0 \underline{e}_1 - \bar{H}_m \underline{y} \|_2$$

## USING SINGULAR VALUE DECOMPOSITION (S.V.D.)

$$\underset{m+1 \times m}{\bar{H}_m} = Q_{m+1} \Sigma_m \overbrace{(P_m)^T}^{m \times m} \quad ; \quad \underset{m+1 \times m+1}{Q_{m+1}} = (Q_m \,\vdots\, \underline{q}_{m+1})$$

$$\underset{m+1 \times m}{\Sigma_m} = \begin{pmatrix} \sigma_1 & & 0 \\ & \sigma_2 \ddots & \\ 0 & & \sigma_m \\ 0 & \cdots & 0 \end{pmatrix} = \begin{pmatrix} S_m \\ \underline{0}^T \end{pmatrix}$$

$$y_m = \bar{H}_m^{+} (\beta_0 \underline{e}_1) = P_m S_m^{-1} (Q_m^T (\beta_0 \underline{e}_1))$$

$$= \sum_{j=1}^{m} \left[ \frac{\beta_0 (\underline{q}_j^T \underline{e}_1)}{\sigma_j} \right] \underline{p}_j = \underline{y}_m$$

$$\| \underline{r}^{(m)} \|_2 = \| \beta_0 \underline{e}_1 - \bar{H}_m (P_m S_m^{-1} Q_m^T \beta_0 \underline{e}_1) \|_2$$

$$= \| \beta_0 \underline{e}_1 - \underbrace{Q_{m+1} \Sigma_m}_{Q_m S_m} \underbrace{P_m^T P_m}_{I_m} S_m^{-1} Q_m^T \beta_0 \underline{e}_1 \|$$

$$= \| \underline{(I - Q_m Q_m^T)} \beta_0 \underline{e}_1 \|$$

Note: $$Q_{m+1}^T Q_{m+1} = \begin{pmatrix} Q_m^T \\ q_{m+1}^T \end{pmatrix} (Q_m \vdots q_{m+1}) = I$$

$$Q_{m+1} Q_{m+1}^T = (Q_m \vdots q_{m+1}) \begin{pmatrix} Q_m^T \\ q_{m+1}^T \end{pmatrix} = I$$

$$\therefore \quad Q_m Q_m^T + q_{m+1} q_{m+1}^T = I$$

i.e. $$I - Q_m Q_m^T = \underline{q_{m+1} q_{m+1}^T}$$

$$\therefore \quad \| \underline{r}^{(m)} \| = \| q_{m+1} (q_{m+1}^T \beta_0 \underline{e}_1) \|_2$$

$$= \beta_0 | q_{m+1}^T \underline{e}_1 | \longleftarrow$$

# Generalised Minimal Residual Method (GMRES)

- Note that the above result is only true for the 2-norm. For a general norm, the result is:

$$\left\|\mathbf{r}^{(m)}\right\| = \left\|\mathbf{V}_{m+1}\left(\beta\mathbf{e}_1 - \overline{\mathbf{H}}_m\mathbf{y}_m\right)\right\|. \tag{13}$$

- To compute the residual norm there is no need for the matrix-vector product $\mathbf{A}\mathbf{x}^{(m)}$.

**Theorem (Minimum residual)**

*The GMRES approximate solution $\mathbf{x}^{(m)}$ minimises the 2-norm of the residual vector over the affine space $\mathbf{x}^{(0)} + \mathcal{K}_m(\mathbf{A}, \mathbf{r}^{(0)})$.*

Another way to view the derivation of GMRES.

$$\min_{\underline{x} \in \underline{x}^{(0)} + \mathcal{K}_m} \| \underline{b} - A\underline{x} \|_2$$

$$
\begin{aligned}
\Longleftrightarrow \quad & \min_{\underline{y} \in \mathbb{R}^m} \| \underline{b} - A\{\underline{x}^{(0)} + V_m \underline{y}\} \|_2 \\
= \quad & \min_{\underline{y} \in \mathbb{R}^m} \| \underline{b} - A\underline{x}^{(0)} - A V_m \underline{y} \|_2 \\
= \quad & \min_{\underline{y} \in \mathbb{R}^m} \| \underline{r}^{(0)} - V_{m+1} \overline{H}_m \underline{y} \|_2 \\
= \quad & \min_{\underline{y} \in \mathbb{R}^m} \| \underline{V_{m+1}} \{ \beta_0 \underline{e}_1 - \overline{H}_m \underline{y} \} \|_2 \\
= \quad & \boxed{\min_{\underline{y} \in \mathbb{R}^m} \| \beta_0 \underline{e}_1 - \overline{H}_m \underline{y} \|_2}
\end{aligned}
$$

# Preconditioning

- The basic idea is to choose a *preconditioning matrix* or *preconditioner* $\mathbf{M}$ and solve a preconditioned version of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$. $\longleftrightarrow$ $(\mathbf{A}\mathbf{M}^{-1})(\underbrace{\mathbf{M}\mathbf{x}}_{\tilde{x}}) = \underline{b}$ i.e. $\boxed{\tilde{A}\,\tilde{\underline{x}} = \underline{b}}$
- Here we consider only right preconditioning:
  $\boxed{\textbf{Right preconditioning:}}$ *The matrix* $\mathbf{M}^{-1}$ *is actioned on the right of* $\mathbf{A}$. *The following right preconditioned system is then solved for* $\widetilde{\mathbf{x}}$:

$$(\mathbf{A}\mathbf{M}^{-1})\widetilde{\mathbf{x}} = \mathbf{b}, \qquad \widetilde{\mathbf{x}} = \mathbf{M}\underline{\mathbf{x}}. \tag{16}$$

  *The required solution of the original linear system* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *is then computed by solving* $\mathbf{M}\mathbf{x} = \widetilde{\mathbf{x}}$ *for* $\mathbf{x}$. Very easy compute.
- The preconditioner $\mathbf{M}$ is chosen to be close to $\mathbf{A}$ (in some sense) so that the matrix $\mathbf{A}\mathbf{M}^{-1}$ is close to the identity matrix, which produces faster convergence of the GMRES iterations.

# Right–preconditioning

- In other words, **M** should be some approximation to the matrix **A** (or equivalently $\mathbf{M}^{-1}$ should be some approximation to the inverse of the matrix **A**). The preconditioner **M** must also be cheap to generate and easy to invert. ↓↓ computations
- Some common choices include:
  - Jacobi: $\mathbf{M} = \mathbf{D}$ $\quad D^{-1} = \left( \frac{1}{a_{11}}\ \frac{1}{a_{22}} \ \cdots\ \frac{1}{a_{nn}} \right)$ $\quad A = L + D + U$
  - Gauss-Seidel: $\mathbf{M} = \mathbf{D} + \mathbf{L}$ ← forward subst.
  - SOR: $\mathbf{M} = \frac{1}{\omega}\mathbf{D} + \mathbf{L}$
  - Incomplete LU Factorisation: $\boxed{\mathbf{M} = \mathtt{ilu}(\mathbf{A})}$ ←

  where **D** is the diagonal portion of **A**, **L** is the lower triangular portion of **A** and $\omega$ is a chosen parameter.
- The incomplete LU factorisation produces sparse factors **L** and **U** such that $\mathbf{LU} \approx \mathbf{A}$.

# Right–preconditioning

- Despite the knowledge of what a good preconditioner should do, the choice of an optimal **M** for a particular system is still very much a trial and error process.
- In your group projects, **M** is chosen as the full Jacobian matrix of **F** but is held constant over several Newton iterations/time steps with heuristics used to determine when the preconditioner is updated.
- Note that for right preconditioning the residual vector does not change:

$$\tilde{\mathbf{r}}^{(m)} = \mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\tilde{\mathbf{x}}^{(m)} = \mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\left(\mathbf{M}\mathbf{x}^{(m)}\right) = \mathbf{b} - \mathbf{A}\mathbf{x}^{(m)} = \mathbf{r}^{(m)}. \tag{17}$$

# Right–preconditioning GMRES

- For right-preconditioned GMRES an approximate solution $\mathbf{x}^{(m)}$ is sought from the affine space $\mathbf{x}^{(0)} + \mathbf{M}^{-1}\mathcal{K}_m(\mathbf{AM}^{-1}, \mathbf{r}^{(0)})$ by imposing the *Petrov-Galerkin condition*:

$$\mathbf{r}^{(m)} = \mathbf{b} - \mathbf{Ax}^{(m)} \perp \mathbf{AM}^{-1}\mathcal{K}_m(\mathbf{AM}^{-1}, \mathbf{r}^{(0)}) . \qquad (18)$$

Right-preconditioned GMRES involves the following steps:

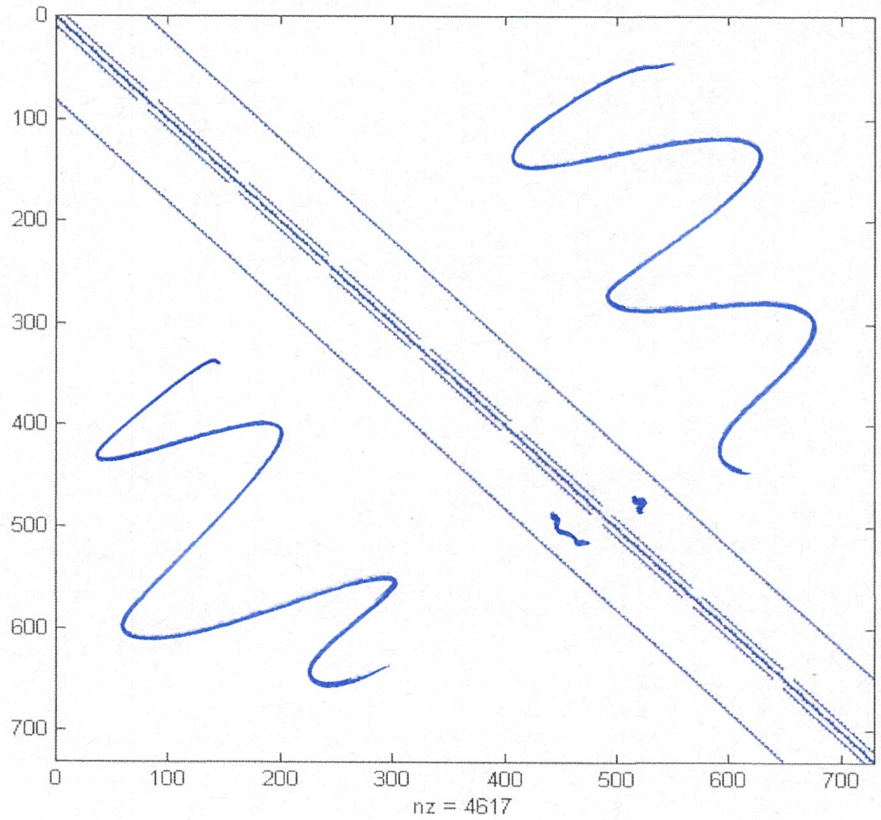
1. Choose $\mathbf{x}^{(0)}$ and compute $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{Ax}^{(0)}$, $\beta = \|\mathbf{r}^{(0)}\|_2$ and $\mathbf{v}_1 = \mathbf{r}^{(0)}/\beta$.
   $AM^{-1}\underline{v}_m = A\underline{\omega}$ ; $\underline{\omega} = M^{-1}\underline{v}_m \Longleftrightarrow M\underline{\omega} = \underline{v}_m$
2. Generate $\boxed{\mathbf{AM}^{-1}}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m$ using the Arnoldi algorithm:
   *Solve* $\mathbf{Mu} = \mathbf{V}(:, m)$ *for* $\mathbf{u}$ *and then compute* $\mathbf{V}(:, m+1) = \mathbf{Au}$.
3. Solve the least squares problem $\min_{\mathbf{y}\in\mathbb{R}^m} \|\beta\mathbf{e}_1 - \overline{\mathbf{H}}_m\mathbf{y}\|_2$ for $\mathbf{y}_m$.
4. Compute residual norm $\left\|\mathbf{r}^{(m)}\right\|_2 = \left\|\beta\mathbf{e}_1 - \overline{\mathbf{H}}_m\mathbf{y}_m\right\|_2$. If $\left\|\mathbf{r}^{(m)}\right\|_2 < tol \left\|\mathbf{r}^{(0)}\right\|_2$ go to step 5 else go back to step 2. A typical value of the tolerance *tol* is $10^{-8}$.
   $= \underline{\omega}_m$ $\quad M\underline{z}_m = \underline{\omega}_m$
5. Compute approximate solution $\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + \underbrace{\mathbf{M}^{-1}\boxed{\mathbf{V}_m\mathbf{y}_m}}_{\underline{z}_m}$:
   *Solve* $\mathbf{Mu} = \mathbf{V}_m\mathbf{y}_m$ *for* $\mathbf{u}$ *then* $\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + \mathbf{u}$.

## Case Studies:
## Selected from the Matrix Market

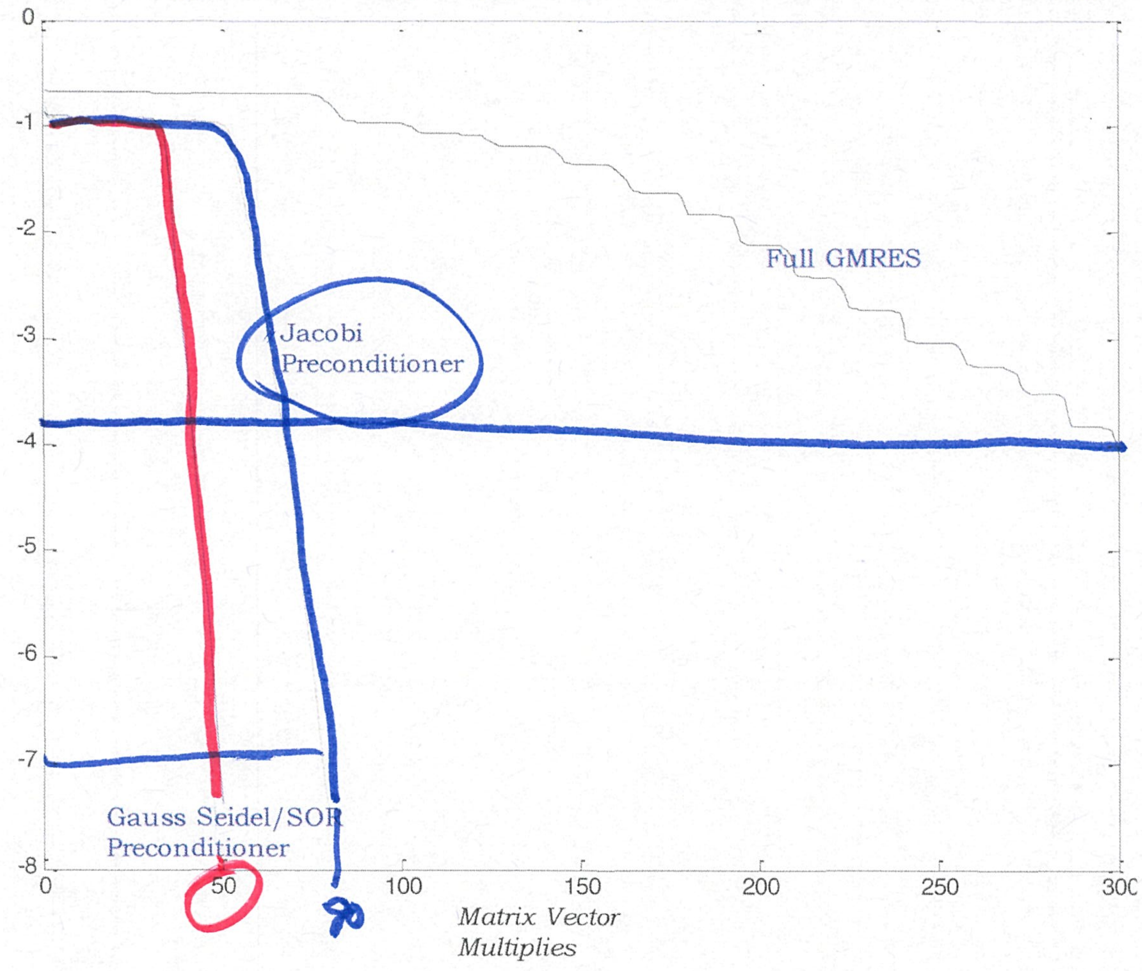
*http://math.nist.gov/MatrixMarket*

# Nos7

$\|\mathbf{b}\|_2 = 0.2313,\ n = 729, \|\mathbf{A}\|_2 = 9.864\times10^6, cond(\mathbf{A}) = 2.3745\times10^9$



# The Impact of (Right) Preconditioning



```
function [x,m] = pGMRESv01(A,b,x0,M,tol,MaxKrylov,diagnostics)
%Calculates the General Minimal Residual Method (GMRES) for a given matrix
%A and solution vector b.
%Input: Matrix A, vector b
%output: solution vector x and number of loops m

%Initialise
N = size(A,1);
H = zeros(MaxKrylov+1,MaxKrylov);
V = zeros(N,MaxKrylov+1);
```

$$\frac{\|r^{(m)}\|}{\|r^{(0)}\|} < tol$$

```
r = b - A*x0;
beta = norm(r,2);
V(:,1) = r/beta;
m=0;
rnorm=inf;

while rnorm > beta*tol && m <= MaxKrylov

    m=m+1;
    %Arnoldi (Modified Gram-Schmidt)
    V(:,m+1) = A*(M\V(:,m));
    for j = 1:m
        H(j,m) = V(:,j)'*V(:,m+1);
        V(:,m+1) = V(:,m+1) - H(j,m)*V(:,j);
    end
    H(m+1,m) = norm(V(:,m+1),2);

    % Check for breakdown
```

"Happy"

```
    if abs(H(m+1,m)) < 1e-14
        fprintf('Invariant Krylov Subspace detected at m=%g\n',m);
        y = H(1:m,1:m) \ ([beta; zeros(m-1,1)]); % Invariant space detected
        break;
    else
        V(:,m+1) = V(:,m+1)/H(m+1,m);
    end
    % Solve small m dimensional least squares problem for y
    rhs= [beta; zeros(m,1)];
    y = H(1:m+1,1:m) \ rhs;
```

$y_m = \bar{H}_m^{+}(\beta e_1)$

```
    % determine residual norm
    rnorm = norm(rhs-H(1:m+1,1:m)*y);

    if diagnostics, fprintf('m=%g ||r_m||=%g tol=%g\n',m,rnorm,beta*tol); end
end

%% Compute approximate solution
x = x0 + M\(V(:,1:m)*y);

end
```

SVD