

Outline

- 1 Introduction
- 2 Projection Methods
- 3 Arnoldi's Methods
- 4 Generalised Minimal Residual Method (GMRES)
- 5 Newton–Krylov Method
- 6 Preconditioning



Newton–Krylov Method

- Consider solving $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ using the Newton iteration:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta\mathbf{x}^{(k)}$$

where the Newton step $\delta\mathbf{x}^{(k)}$ is the solution of the linear system:

$$\underbrace{\mathbf{F} + \mathbf{J}(\mathbf{x}^{(k)})}_{\text{Red}} \delta\mathbf{x}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)}). \quad (14)$$

- In a Newton-Krylov method the solution of the linear system (14) is carried out using a Krylov subspace method.
- Let $\mathbf{w}^{(m)}$ denote the approximate solution of (14) after m iterations of GMRES.

```

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: Matrix A, vector b
N = size(A,1);
H = zeros(N,MaxKrylov+1);
V = zeros(N,MaxKrylov+1);
%Initialise
r = b - A*x0;
beta = norm(r,2);
v(:,1) = r/beta;
m=0;
rnorm=inf;
while rnorm > betatol & m <= MaxKrylov
    %Arnoldi (Modified Gram-Schmidt)
    V(:,m+1) = A*(M\V(:,m));
    for j = 1:m
        H(j,m) = V(:,j)*V(:,m+1);
        H(:,m+1) = V(:,m+1) - H(j,m)*V(:,j);
    end
    V(:,m+1) = V(:,m+1)/H(m+1,m);
    if abs(H(m+1,m)) < 1e-14
        % Check for breakdown
        if abs(H(m+1,m)) \ ([beta; zeros(m-1,1)]) < 1e-14
            fprintf('Invariant Krylov Subspace detected at m=%g\n',m);
            y = H(1:m,1:m) \ rhs;
            rhs=[beta; zeros(m,1)];
            Y = H(1:m+1,1:m) \ rhs;
            norm = norm(rhs-H(1:m+1,1:m)*y);
            % determine residual norm
            % Solve small m dimensional least squares problem for Y
            % Small m dimensional least squares problem for Y
            end
        else
            break;
        end
    end
    if diagnostics, fprintf('m=%g ||r_m||=%g tol=%g\n',m,rnorm,betatol); end
    % Compute approximate solution
    x = x0 + M\((V(:,1:m)*y));
end

```

Newton–Krylov Method

- Termination of the GMRES iterations is often based on the criterion “**forcing term**”.

$$\left\| \mathbf{J}(\mathbf{x}^{(k)})\mathbf{w}^{(m)} + \mathbf{F}(\mathbf{x}^{(k)}) \right\|_2 < \boxed{\eta_k} \left\| \mathbf{F}(\mathbf{x}^{(k)}) \right\|_2$$

- where the left-hand side is the residual norm associated with (14) and the tolerance tol has been replaced with a forcing term η_k .
- The general idea is to avoid *oversolving* the linear system: away from the solution \mathbf{x} the cost associated with obtaining a highly accurate Newton step $\delta\mathbf{x}^{(k)}$ may provide little additional progress toward the solution.

- η_k is called a **FORCING TERM**
- the plan is to vary η_k as Newton iterations progress.
 - choosing a small η_k make the iterations behave more like Newton's method.
 - choosing that small η_k may make the computational cost of computing the Newton step $Sx^{(k)}$ high.
- We construct a sequence η_k that becomes smaller and smaller as we get closer and closer to the solution of the nonlinear system.
- . The formula to determine η_k is complex

Newton–Krylov Method

- One choice for the forcing term is the modified Eisenstat-Walker formula¹:

$$\eta_{max} = 0.9, \gamma = 0.9, \alpha = 2,$$

$$\eta_k^R = \gamma \left(\frac{\|\mathbf{F}(\mathbf{x}^{(k)})\|_\infty}{\|\mathbf{F}(\mathbf{x}^{(k-1)})\|_\infty} \right)^\alpha,$$

$$\eta_k^S = \begin{cases} \eta_{max}, & k = 0 \\ \min(\eta_{max}, \eta_k^R), & k > 0 \text{ and } \gamma \eta_{k-1}^\alpha \leq 0.1, \\ \min(\eta_{max}, \max(\eta_k^R, \gamma \eta_{k-1}^\alpha)), & k > 0 \text{ and } \gamma \eta_{k-1}^\alpha > 0.1, \\ \eta_k = \min \left(\eta_{max}, \max \left(\eta_k^S, \frac{0.5(\tau_a + \tau_r \|\mathbf{F}(\mathbf{x}^{(0)})\|_\infty)}{\|\mathbf{F}(\mathbf{x}^{(k)})\|_\infty} \right) \right). \end{cases}$$

- The idea is to use

$$\gamma_k^* = \gamma \frac{\|F(\underline{x}^{(k)})\|^2}{\|F(\underline{x}^{(k)})\|^2}$$

- If this value is bounded away from 1 then using $\gamma_k = \gamma_k^*$ works well (provided a good choice of γ_0)
- otherwise limit to its maximum size
- A good safeguarding strategy applied to protect against "overshooting" to avoid too rapid changes in γ_k .
- If γ_{k-1} is sufficiently large then do not decrease γ_k by too much.

Newton–Krylov Method

- One of the major computational costs associated with Krylov subspace methods is computing the matrix-vector product $\mathbf{A}\mathbf{v}_m$ in Arnoldi's method.
- When applied to the linear system (14) this matrix-vector product takes the form $\mathbf{J}(\mathbf{x}^{(k)})\mathbf{v}_m$, which can be approximated using the finite difference approximation described in Chapter 2:

$$\mathbf{J}(\mathbf{x}^{(k)})\mathbf{v}_m \approx \frac{\mathbf{F}(\mathbf{x}^{(k)} + h\mathbf{v}_m) - \mathbf{F}(\mathbf{x}^{(k)})}{h} \quad (15)$$

where the shift parameter is chosen as

$$h = \begin{cases} \sqrt{\epsilon} \|\mathbf{x}^{(k)}\|_2 & \mathbf{x}^{(k)} \neq \mathbf{0} \\ \sqrt{\epsilon} & \mathbf{x}^{(k)} = \mathbf{0}. \end{cases}$$



$$\underline{F}(\underline{x}^{(k)} + h \underline{v_m}) = \underline{F}(\underline{x}^{(k)}) + \frac{\int_0^1 J(\underline{x}^{(k)}_t + t h \underline{v_m}). \underline{h} dt}{J(\underline{x}^{(k)}). (\underline{h})^2}$$

Solve $M\bar{u} = \bar{v}_m$ (preconditioner)

$$T(\underline{x}^{(k)}) M^{-1} \bar{v}_m \approx \frac{F(\underline{x}^{(k)} + h \underline{u}) - F(\underline{x}^{(k)})}{h}$$

where shift parameter :

$$h = \begin{cases} \frac{\sqrt{\epsilon} \|\underline{x}^{(k)}\|_2}{\|\underline{u}\|_2}, & \underline{x}^{(k)} \neq 0 \\ \frac{\sqrt{\epsilon}}{\|\underline{u}\|_2}, & \underline{x}^{(k)} = 0 \end{cases}.$$

A Basic Algorithm

```
Evaluate  $\mathbf{F}(\mathbf{x}^{(0)})$ , set  $\tau = \tau_r \|\mathbf{F}(\mathbf{x}^{(0)})\| + \tau_a$      $k=0$ ; Initialise  $\eta_o$ 
while  $\|\mathbf{F}(\mathbf{x}^{(k)})\| > \tau$  loop
    Find  $\mathbf{s}^{(k)}$  such that  $\|\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} + \mathbf{F}(\mathbf{x}^{(k)})\| \leq \eta_k \|\mathbf{F}(\mathbf{x}^{(k)})\|$ 
    If no such  $\mathbf{s}^{(k)}$  can be found, terminate with failure.
    Begin parabolic two-point line search strategy to determine  $\lambda^{(k)}$ 
         $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)} \mathbf{s}^{(k)}$ 
    End parabolic two-point
    Adjust forcing term  $\eta_k$  as per Eisenstat-Walker formula
end while
```



TIME STEPPING

— NEWTON (OUTER) ITERATIONS

— GMRES (INNER) ITERATIONS