

Wiederholung: Krylov Unterraum:

$$K_m(\underline{A}, x_1) = \text{span}(\underline{x}_1, \underline{A} \cdot \underline{x}_1, \underline{A}^2 \cdot \underline{x}_1, \dots, \underline{A}^{m-1} \cdot \underline{x}_1)$$

Basis für viele Verfahren.

(Achtung: Wenn sich  $A(\lambda)$  kontinuierlich ändert, dann ist es besser Davidson method zu verwenden)

Arnoldi - Methode

Hessenbergmatrix

$$H = \begin{pmatrix} h_{11} & & & & \\ h_{21} & h_{22} & & & \\ & & \ddots & & \\ 0 & & & h_{mm} & \\ & & & & \ddots \end{pmatrix}$$

Einfachster Arnoldi

Ziel ist es eine Matrix  $A$  auf die Hessenberg-Form zu bringen:

$$AV = VH$$

↑  
Orthogonal  
matrix

← Hessenberg matrix

$V$  wird durch die erste Spalte  $v_1 = \underline{v} = \underline{e}_1$

Teilweise schlägt der Algorithmus fehl, dann gibt es eine  $n \times m$  Matrix  $V_m$

$$AV_m - V_m H_m = 0$$

Es gibt das Residuum nach  $m$  Durchgängen

$$\underline{A} \cdot \underline{V}_m - \underline{V}_m \cdot \underline{H}_m = \underline{r}_m$$

Algorithmus (analog zum Simplex technical report)

Ausgangspunkt Matrix  $A$ , Iterationszahl und Ausgangsvektor  $v_1$  (normiert!)

For  $j = 1, 2, \dots, m-1$

$$w = A v_j$$

$w$  bzgl  $V_j$  orthogonalisiert Resultat  $h_{1,j:j}$

$$h_{j+1,j} = \|w\|_2$$

If  $h_{j+1,j} = 0$  stop

$$v_{j+1} = w / h_{j+1,j}$$

end

$$f = A v_m$$

$f$  bzgl.  $V_m$  orthogonalisiert (Resultat  $h_{1,j:j}$ )

$\Rightarrow$  Resultat  $V_m, H_m$  und  $f$  und  $\beta = \|f\|_2$   
mit  $A V_m - V_m H_m = \beta e_m^x$

Schau wir uns den Algorithmus an:

- Das ist im Prinzip die Konstruktion eines Krylov-Unterraums! (Iterativ Orthogonalisieren)
- Die Spalten von  $V_m$  heißen Arnoldi Vektoren
- Im Falle von  $h_{j+1,j} = 0$  bekommen wir ein exaktes invariante Unterraum von  $A$  ( $A$  verändert die Unterraum nicht, passiert in der Praxis nicht, endliche Genauigkeit)

Da  $f$  per Konstruktion  $\perp$  zu  $V_m$  ist.

$$\text{gilt } A V_m - V_m H_m = f e_m^x$$

$$\Rightarrow V_m^x A V_m = H_m$$

Erlaubt die Berechnung der Eigenpaare über Rayleigh-Ritz Approximation.

Also falls  $(\lambda_i, y_i)$  ein Eigenpaar <sup>von  $H_m$</sup>  ist, so ist der

$$\text{Ritzwert } \lambda_i = \frac{x_i^x \cdot A \cdot x_i}{x_i^x \cdot x_i} \text{ und } x_i = V_m y_i \text{ ein}$$

Approximant des Eigenpaares.

Meist sind nur sehr wenige Approximate brauchbar  
(geringe Prozentzahl).

Die Qualität kann mit der Residual Norm überprüft  
werden:

$$\begin{aligned} \|\underline{A} \cdot \underline{x}_i - \lambda_i \underline{x}_i\|_2 &= \|A V_m y_i - \lambda_i V_m y_i\|_2 \\ &= \|(A V_m - V_m H_m) y_i\|_2 = \beta \|e_m^x y_i\| \end{aligned}$$

Achtung, die Speicheranforderungen wachsen mit  
der Anzahl der Schritte.

Meist ist die Schrittzahl zu hoch für eine gute Konvergenz

⇒ Lösung Algorithmus neustarten.

Idee: Berechne die Arnoldi in Schritt Faktorisierung  
mit immer besser initialen Vektoren!

Ausführung: Wir verwenden den Ritzvektor des dominanten  
Eigenvektors.

Wichtig: Um nicht nur einen Eigenvektor zu bekommen,  
müssen die schon gut approximiert EV gelöst  
werden!  
(Werden nicht mehr modifiziert!)

$$A_{So} \quad V_m = \left[ \underbrace{V_{1:k}^{(k)}}_{\substack{\text{schon} \\ \text{konvergiert} \\ \text{Eigenvektor} \\ \text{(gelöst)}}} \mid \underbrace{V_{k+1:m}^{(n)}}_{\substack{\text{aktive} \\ \text{Vektoren} \\ \text{müssen noch} \\ \text{aktualisiert.}}} \right]$$

↑ ↗

Orthogonalisierung erfolgt bezgl beide Vektorengruppen.  
Arnoldi mit Deflation

Matrix  $A$ , Anzahl der Schritte  $m$ ,  $v_{i:k}$ ,  $H_{kk}$  mit  $k < m$  und  
 $v_{k+1}$  mit Norm 1

For  $j = k+1, \dots, m-1$

$$w = Av_j$$

Orthogonalisierung von  $w$  bezgl.  $v_j$  (ergibt  $h_{j+1,j}$ )

$$h_{j+1,j} = \|w\|_2$$

falls  $h_{j+1,j} = 0$  stopp

$$v_{j+1} = w / h_{j+1,j}$$

and

$f = Av_m$   
 $f$  bezgl.  $v_m$  orthogonalisiert (ergibt  $h_{1,m}$ )

$$\beta = \|f\|_2$$

[Explizit neustartende Arnoldi]

Eingang: Matrix  $A$ , initialvektor  $v_1$   
 Dimension des Unterraums  $m$

Normalisieren  $v_1$

$$\text{Setze } V_m = [v_1], k=0$$

Restart Schleife

$m-k$  Schritte Arnoldi mit Deflation

Reduziere  $H_m$  auf (quasi) Dreieckform  $H_m \leftarrow U_1^* H_m U_1$  (~~z.B. QR~~  
 z.B. QR)

Bei Entartung

Sortiere  $1 \times 1, 2 \times 2$  diagonale Blöcke:  $H_m \leftarrow U_2^* H_m U_2$

$$U = U_1 U_2$$

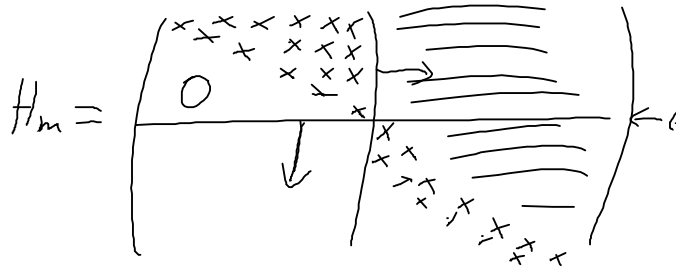
Bestimme Eigenvektoren von  $H_m$ ,  $H_m y_i = y_i \theta_i$

Brech residuel von  $\varepsilon_i = \beta \text{ klein } |y_i|$

Locke konvergente Eigenvektoren

$$V_m \leftrightarrow V_m U$$

and



Wir sehen die Verfahren sind sehr kompliziert.

⇒ Wichtiges Verfahren Krylov sehr (Konvergenz sehr schnell), ähnlich Arnoldi.

Grundidee: Diese Verfahren kann für große Matrizen kein Alternative, die die Eigenwerte gut approximieren.

⇒ Wichtiges Resultat, Forschungsfeld für sich, benutze fortgeschrittene Solver für eigene wissenschaftliche Projekte!

## II. 6) Preconditioning von Matrizen

Problem bei iterativen Verfahren

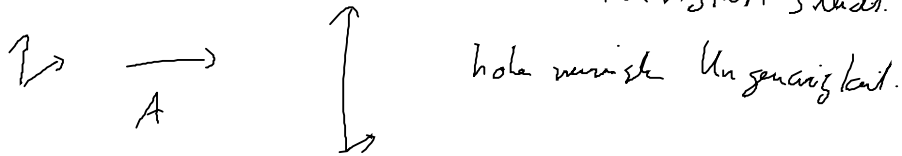
Die Konvergenz kann sehr langsam sein.

Meist hängt die Konvergenz von der Konditionszahl ab.

$$\kappa(A) = \left| \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \right| \left\{ \begin{array}{l} \leftarrow \text{größte Eigenwert} \\ \leftarrow \text{kleinste Eigenwert} \end{array} \right.$$

Falls  $\kappa(A) \gg 1$ , schlecht konditioniertes Problem (Kontext

Problem: schlecht konditioniertes System konvergiert schlecht. Gleichungssystem



$\Rightarrow$  Lösung: Preconditioning (Vorconditioning)  
geschieht jeweils problemabhängig und  
verändert Spektrum.

Die Idee ist es, dass z.B. ein Gleichungssystem

$$\underline{A} \cdot \underline{x} = \underline{b} \quad \text{mit einer regulären Matrix (invertierbar)} \quad \underline{M} \cdot \underline{A} \cdot \underline{x} = \underline{M} \cdot \underline{b}$$

umgeformt werden kann.

$$\underline{A} \cdot \underline{M} \cdot \underline{y} = \underline{b} \quad \text{mit } \underline{y} = \underline{M}^{-1} \cdot \underline{x}$$

Wichtig,  $M$  wird so konstruiert, dass sich  
ihr Inverses gut approximieren lässt.

$\Rightarrow$  Ziel schnelle Konvergenz

### Beispiele

- Jacobi Preconditioning
- Falls die Diagonale dominant ist (z.B. per DGL)  
 $M = \text{diag}(A)$  und  $M_{ij}^{-1} = \frac{\delta_{ij}}{A_{ij}}$  ( $A_{ij} \neq 0!$ )
- Unvollständige Cholesky Faktorisierung
- Unvollständige LU Faktorisierung ...

$\Rightarrow$  Wenn Konvergenz zu langsam  $\Rightarrow$  Scheitert  
nach der Preconditioning.