

Letzte Vorlesung

gibt  $\underline{X} = \underline{Q} \cdot \underline{R}$

QR Algorithmus

Die QR Zerlegung kann genutzt werden, um Eigenwerte und Eigenvektoren einer Matrix zu bestimmen.

$$\underline{A} \rightarrow \underline{Q}^T \cdot \underline{A} \cdot \underline{Q}$$

↑  
Eine unitäre Transformation ändert die Eigenwerte nicht („Ähnliche Matrix“)

Da  $\underline{A} \cdot \underline{x} = \lambda \underline{x}$   
 $\Leftrightarrow \underline{Q}^T \cdot \underline{A} \cdot \underline{Q} \cdot \underline{Q}^T \underline{x} = \lambda \underline{Q}^T \underline{x}$

(Einheitsform invariant)

Algorithmus

$$\underline{A}_1 = \underline{A}$$

for  $k = 1, \dots, d$

Führe eine QR Zerlegung, so dass  $\underline{A}_k := \underline{Q}_k \cdot \underline{R}_k$   
 $\underline{A}_{k+1} := \underline{R}_k \cdot \underline{Q}_k$

end

Damit  $\underline{A}_{k+1} = \underline{Q}_k^+ \cdot \underline{A}_k \cdot \underline{Q}_k$

Man kann zeigen, dass

$$\underline{A}_k = \underline{Q}_k^+ \cdot \underline{Q}_{k-1}^+ \cdot \underline{Q}_{k-2}^+ \dots \underline{Q}_1^+ \cdot \underline{A}_1 \cdot \underline{Q}_1 \dots \underline{Q}_{k-2} \cdot \underline{Q}_{k-1} \cdot \underline{Q}_k$$

Konvergenz zu

$$\begin{pmatrix} \lambda_1 & x & \dots & x \\ & \ddots & & \vdots \\ & & \ddots & \vdots \\ & & & \lambda_n \end{pmatrix}$$

auf der Diagonale stehen die Eigenwerte

o.B. kann man zeigen.

Verallgemeinerung:

- ① Varietät dieses Algorithmus (z.B. mit Shift  $(A_k - \lambda_0 I)$ ...) sind die Basis viele Eigenwertprobleme für dicht besetzte Matrizen. (Existieren viele Bibliotheken, die das übernehmen)
- ② Dünnbesetzte Matrizen, Verfahren zu  $\lambda$  und  $x$ , skaliert schlecht mit der Anzahl der Elemente.  
 $\Rightarrow$  spezielle Verfahren für dünn besetzte Matrizen (Braucht tw. auch das QR Verfahren)

Ziel ist es bei dünn besetzten Matrizen nicht alle Eigenwerte zu bestimmen, sondern meist die mit dem niedrigsten Wert z.B.

$\Rightarrow$  iterative Methode, meist im Krylov Raum.

Power method (Richtung Krylov Method)

Wir haben eine Matrix  $A$  (symmetrisch) mit Eigenwerten  $|\lambda_1| \geq |\lambda_2| > |\lambda_3| > \dots > 0$   
 Eigenvektoren  $e_1, e_2, e_3, \dots$

Initial guess Vektor  $v_0 = c_1 e_1 + c_2 e_2 + c_3 e_3 + \dots + c_n e_n$

$$A \cdot v_0 = c_1 \lambda_1 e_1 + c_2 \lambda_2 e_2 + c_3 \lambda_3 e_3 + \dots + c_n \lambda_n e_n$$

$$A^k \cdot v_0 = c_1 \lambda_1^k e_1 + c_2 \lambda_2^k e_2 + c_3 \lambda_3^k e_3 + \dots + c_n \lambda_n^k e_n$$

Wenn  $k$  groß genug ist,

$$A^k \cdot v_0 \approx c_1 \lambda_1^k e_1$$

Gute Methode um niedrigste EW zu bestimmen.

Rayleigh Algorithm

$$y = v_0$$

For  $k=1, 2, \dots$

$$\theta = \|y\|_{\infty}$$

$$x = y / \theta$$

$$y = A \cdot x$$

end

Konvergenz hängt von der Rate  $\left(\frac{\lambda_1}{\lambda_2}\right)$  ab, falls eng benachbart sehr langsam.

oft wird auch eine inverse Iteration durchgeführt mit  $A^{-1}$ , dann  
 mit einem Shift  $(A - \sigma I)^{-1}$  um die Eigenwerte in der Nähe  
 von  $\sigma$  zu bekommen.

Dann braucht man ein Löser für lineare Gleichungssysteme (iterativ oder direkt)

$$y = v_0$$

$$\text{Für } k = 1, 2, \dots$$

$$\theta = \|y\|_{\infty}$$

$$v = y / \theta$$

$$\text{Löse } (A - \sigma I) y = v \quad \left\{ \begin{array}{l} y = (A - \sigma I)^{-1} v \end{array} \right.$$

end

Die Methode kann verallgemeinert werden und auf einen Unterraum  
 zu werten statt Vektoren.

Überblick über Methoden für dünnbesetzte Matrizen

Arnoldi (u.a. Varianten Rostark, Detlation), CG (conjugate gradient)  
 Lanczos (Spezialfall von Arnoldi)

Krylov-Subraum

Power Methode

Davidson Type Subspace Expansion

Viele Methoden benutzen Krylov Unterräume

Krylov Unterräume

Die Konstruktion zeigt Verbindung zur Power-Algorithmus.

Für einen gegebenen Vektor  $x_1$  ist der Krylov Unterraum der Ordnung  $m$

$$\mathcal{K}_m(A, x_1) = \text{span} \left( x_1, A \cdot x_1, A^2 \cdot x_1, \dots, A^{m-1} \cdot x_1 \right)$$

Es fällt sofort auf, dass spaltenweise Shifts  $(A - \sigma I)$  den Krylov Raum  
 nicht verändern. Solche Optimierungen sind nichts zu bringen oder notwendig  
 zu sein.

Die Idee von dem Verfahren ist wenn das Problem nicht für die gesamte  
 Matrix zu lösen, sondern nur auf dem Teil des Krylov Unterrums gelöst.

Wir haben schon bei der Power Methode gesehen, dass die dominante ELe  
 im Krylov Raum dominiert. Idee ist, dass dies auch hier der Fall ist  
 (Potenzen in den Eigenwerten / Polynom Matrix)

Beispiel Arnoldi Methode

Hessenbergmatrix

$$H = \begin{pmatrix} h_{11} & & & & \\ h_{21} & h_{22} & & & \\ 0 & & \ddots & & \\ & & & h_{m-1,m-1} & \\ & & & & h_{mm} \end{pmatrix}$$

Einheitsform

Ziel ist es eine Matrix  $A$  auf die Hessenberg zu bringen.

$$\underline{A} \cdot \underline{V} = \underline{V} \cdot \underline{H}$$

orthogonale Matrix

Hessenbergmatrix

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

Teilweise schlägt der Algorithmus fehl, dann gibt es  $m \times m$  Matrix  $V_m$  mit orthogonalen Spalten

$$\underline{A} \cdot \underline{V}_m - \underline{V}_m \cdot \underline{H}_m = 0$$

Es gibt es doch ein Residuum  $\underline{A} \cdot \underline{V}_m - \underline{V}_m \cdot \underline{H}_m = \underline{r}_m^x$

Algorithmus (analog Störtechnikreparat)

Ausgangspunkt Matrix  $A$ , Iterationszahl und Ausgangsvektor  $\underline{v}_1$

Für  $j = 1, 2, \dots, m-1$

$$\underline{w} = \underline{A} \cdot \underline{v}_j$$

$w$  bzgl.  $V_j$  orthogonalisiert Resultat  $h_{i,j}, j$   
 $\overbrace{1 \dots j}^{1 \dots j}$

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

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

$$\underline{v}_{j+1} = \underline{w} / h_{j+1,j}$$

end

$$\underline{f} = \underline{A} \cdot \underline{v}_m$$

$\underline{f}$  bzgl.  $v_m$  orthogonalisiert. (Resultat  $h_{i,j}$ )

$\Rightarrow$  Resultat  $V_m, H_m$  und  $f$  und  $\beta = \|\underline{f}\|_2$

$$m+1 \quad \underline{A} \cdot \underline{V}_m - \underline{V}_m \cdot \underline{H} = \underline{\epsilon}_m^T$$

Schauen wir uns den Algorithmus genauer!

- Das ist die Konstruktion eines Krylov Unterraums! (Wichtig, iterative Orthogonalisierung)
- Die Spalten von  $\underline{V}_m$  heißen Arnoldi Vektoren.
- Im Fall von  $h_{j+1,j} = 0$ , bekommen wir eine exakte inverse des Kerns von  $A$  ( $A$  verliert den Unterraum nicht, passiert in der Praxis nicht an dieser Genauigkeit)

- Da  $\underline{\epsilon}$  per Konstruktion  $\perp$  zu  $\underline{V}_m$  ist, gilt:

$$\underline{A} \cdot \underline{V}_m - \underline{V}_m \cdot \underline{H} = \underline{\epsilon}_m^T$$

$$\Rightarrow \underline{V}_m^T \cdot \underline{A} \cdot \underline{V}_m = \underline{H}_m$$

Upper Hessenberg Form, EW, EV (nicht  
sich besten über QR!  
(Klein Matrix! im Vergleich zu  $A$ )

Darstellung von  $A$  im Krylov Unterraum.

Dies erlaubt die Bestimmung der Eigenwerte über die Rayleigh-Ritz Approximation.

Also falls  $(\lambda_i, \underline{x}_i)$  Eigenpaar von  $\underline{H}_m$  ist, so ist der Ritzwert  $\lambda_i = \frac{\underline{x}_i^T \underline{A} \underline{x}_i}{\underline{x}_i^T \underline{x}_i}$

und Ritzvektor  $\underline{x}_i = \underline{V}_m \underline{y}_i$  eine Approximation des Eigenpaars!

Meist sind nur wenige dieser Approximationen brauchbar (geringe Prozentzahl)!  
Die Qualität kann mit der Residual Norm überprüft werden:

$$\| \underline{A} \underline{x}_i - \lambda_i \underline{x}_i \|_2 = \| \underline{A} \underline{V}_m \underline{y}_i - \lambda_i \underline{V}_m \underline{y}_i \|_2 = \| (\underline{A} \underline{V}_m - \underline{V}_m \underline{H}_m) \underline{y}_i \|_2 = \rho | \underline{\epsilon}_m^T \underline{y}_i |$$

$\uparrow$  Das haben wir schon.

Achtung, die Speicheranforderung wachsen mit der Anzahl der Schritte. Meist ist die Schrittanzahl für gute Konvergenz ist meist zu hoch für den Speicher / Effizienz!

$\Rightarrow$  Lösung Algorithmen neu starten!