

Notation wie in Skriptum (Abschnitt 7–9). Bei einigen der Aufgaben spielen die Identitäten zwischen A , H_m , V_m und v_{m+1} eine wesentliche Rolle (Abschnitt 9.1). Im symmetrischen Fall schreiben wir T_m (tridiagonal) statt H_m (Hessenberg).

1. Sei $A > 0$, $\phi(x) = \frac{1}{2}(Ax, x) - (b, x)$, und x_* die exakte Lösung von $Ax = b$.

a) Zeigen Sie

$$\phi(x) - \phi(x_*) = \frac{1}{2} \|x - x_*\|_A^2$$

b) Sei \hat{x} das Ergebnis eines SD-Iterationsschrittes (mit Suchrichtung $0 \neq d = r = b - Ax$) ausgehend von x . Das SD-Verfahren ist strikt monoton konvergent in der Energienorm, da ja gilt

$$\|\hat{x} - x_*\|_A \leq \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1} \right) \|x - x_*\|_A \tag{1}$$

(siehe den Beweis von Theorem 7.1).

Sei nun (x_0, x_1, \dots) eine Folge von CG-Iterierten. Wir betrachten eines der x_k ; dann gilt für das Ergebnis $\hat{x} = \hat{x}_k$ eines SD-Iterationsschrittes ausgehend von $x = x_k$ die Ungleichung (1).

Argumentieren Sie, dass gilt

$$\|x_{k+1} - x_*\|_A \leq \|\hat{x}_k - x_*\|_A$$

d.h. der CG-Schritt ausgehend von x_k ist ‘mindestens so gut’ wie der SD-Schritt.

Hinweis: Beachten, Sie, welcher Minimaleigenschaft sich x_{k+1} im Vergleich zu \hat{x}_k erfreut.

Anmerkung: Die globale Fehlerschranke für CG aus Theorem 8.2 kann man aus dieser lokalen Überlegung nicht herleiten.

2. *Verwendung von pcg.*

Verwenden Sie MATLAB/`pcg`, um ein lineares Gleichungssystem $Ax = b$ mit einer Matrix $A > 0$ aus der Sammlung `MatrixMarket` zu lösen. Plotten Sie den Verlauf des Residuums (2-Norm) über die Iterationsschritte hinweg, ggf. auf einer logarithmischen Skala.

3. [Exercise 9.1:]

Zeigen Sie mittels eines Induktionarguments über die Potenzen von A :

$$p(A)y = V_m p(H_m) V_m^T y = \|y\|_2 V_m p(H_m) e_1 \tag{2}$$

für beliebiges $y \in \mathbb{R}^n$ und alle $p \in \mathcal{P}_{m-1}$. Dabei bezeichnet $e_1 = (1, 0, \dots, 0)^T$ den ersten kartesischen Einheitsvektor im \mathbb{R}^m .

(Die Arnoldi-Iteration startet bei y , mit $v_1 = y/\|y\|_2$.)

4. *Ein Lanczos-Code + Krylov-Approximation für $x_* = A^{-1}b$.*

a) Programmieren Sie die Lanczos-Iteration für eine gegebene symmetrische **sparse** Matrix A und gegebenes r_0 in MATLAB.

Input: A, r_0, m

Output: V_m, T_m

b) Relation (2) motiviert die Approximation (für allgemeines A)

$$x_m = x_0 + V_m H_m^{-1} e_1$$

für $x_* = x_0 + A^{-1}r_0$. Überlegen Sie, dass mit A auch H_m invertierbar ist. Stellen Sie das Residuum $r_m = b - Ax_m$ dar und zeigen Sie insbesondere, dass r_m auf $\mathcal{K}(A, r_0)$ orthogonal steht (Galerkin-Orthogonalität), nämlich $r_m \in \text{span}\{v_{m+1}\}$. Was folgern Sie daraus für den Fall, dass A SPD ist?

- c) Sei A symmetrisch, V_m Ihre Basismatrix aus a), und $H_m = T_m$. In die Berechnung von x_m geht die Basismatrix $V_m \in \mathbb{R}^{n \times m}$ ein. Angenommen, die Speicherung der vollen Basismatrix V_m ist ‘kritisch teuer’ im Vergleich zu den Kosten der Lanczos-Iteration. Überlegen Sie, wie man diesen Speicheraufwand unter Inkaufnahme redundanter Berechnungen vermeiden kann. Modifizieren Sie Ihren Code aus a) entsprechend und realisieren Sie ein numerisches Beispiel mit einer SPD-Matrix A . Vergleichen Sie mit `pcg`.

Anmerkung: Man kann es noch etwas effizienter machen (Abschnitt 9.5).

5. Krylov-Approximation für die Matrix-Exponentialfunktion.

Die Hessenberg-Matrizen H_m aus der Arnoldi[Lanczos]-Iteration dienen in verschiedenem Kontext als Approximationen für A . Will man z.B. die Anwendung der Matrix-Exponentialfunktion¹ $\exp(A)$ auf einen Vektor y numerisch approximieren, so wird dazu oft

$$\exp(A)y_0 \approx y_m := \|y_0\|_2 V_m \exp(H_m) e_1 \quad (3)$$

verwendet (eine Motivation dafür ist wiederum durch Exercise 9.1 gegeben). Dabei wird die ‘kleine’ Matrix $\exp(H_m)$ ($m \ll n$) mittels einer Standard-Methode berechnet (z.B. mittels MATLAB/`expm`).

Sei A die symmetrisch negative definite Matrix, die aus der FD-Approximation von $-u''$ entsteht (wie beim 1D-Poisson-Modellproblem). Berechnen Sie² $\exp(A)y_0$ für irgend einen Startvektor y_0 mittels (3), und protokollieren Sie den Fehler der Approximation in der $\|\cdot\|_2$ -Norm, für $m = 1, 2, \dots$. Variieren Sie auch die Dimension von A (z.B. $n = 10, 100, 1000$).

Zum Vergleich berechnen Sie $\exp(A)y_0$ basierend auf der bekannten Spektralzerlegung von A mittels der diskreten Sinustransformation `[i]dst`.

6. Fortsetzung von Aufgabe 5.

Bei einem linearen Gleichungssystem $Ax = b$ ist das Residuum ein natürliches Maß für die Genauigkeit einer Approximation $x \approx x_*$. Im Kontext von Aufgabe 5 versuchen wir nun eine ähnliche Überlegung.

Die Lösung des Systems von linearen Differentialgleichungen mit Anfangswert y ,

$$y'(t) = Ay(t), \quad y(0) = y_0$$

ist $y(t) = \exp(tA)y_0$. Auswertung mittels (3) (mit tA anstatt A) liefert also eine Approximation³ $y_m(t) \approx y(t)$. Stellen wir uns t als kontinuierliche Variable vor, dann ist die ‘Defektfunktion’ (eine Art Residuum)

$$d_m(t) := y'_m(t) - Ay_m(t)$$

ein Maß für die Genauigkeit von $y_m(t)$.

- a) Auf den ersten Blick scheint $d_m(t)$ keine berechenbare Größe zu sein, da ja t in der Praxis eine fest gewählte Schrittlänge (und keine kontinuierliche Variable) ist. Man kann jedoch die Definition von $y_m(t)$ gemäß (3) dazu verwenden, um eine berechenbare Darstellung für $d_m(t)$ (für den festen Zeitschritt t) zu erhalten.⁴ Geben Sie diese an. Zeigen Sie insbesondere, dass gilt $d_m(t) \in \text{span}\{v_{m+1}\}$.
- b) Erweitern Sie Ihren Code aus Aufgabe 5 in dem Sinn, dass $y_m(t) = \exp(tA)y_0$ für eine vorgegebene Schrittlänge t berechnet wird und $\|d_m(t)\|_2$ mit ausgegeben wird. Betrachten Sie $m = 1, 2, 3, \dots$, wählen dann jeweils Sie t ausreichend klein, und wiederholen den Schritt mit $t/2$ (ev. auch $t/4, \dots$) anstelle von t . Damit können Sie für jedes m die experimentelle Konvergenzordnung von $\|d_m(t)\|_2$ für $t \rightarrow 0$ bestimmen. Wie hängt diese von m ab? (Machen Sie eine Tabelle.)

¹ bzw. irgendeine Matrixfunktion $f(A)$

² Analoge Vorgangsweise wie in Aufgabe 4. Beachten Sie insbesondere 4c).

³ Man denke an einen Schritt mit kleinem t , als Basis für eine Iteration $0 \rightarrow t_1 \rightarrow t_2 \dots$ im Sinne eines Einschrittverfahrens.

⁴ Dann ist $\|d_m(t)\|_2$ ein grobes Maß für die Qualität der Approximation.