Prof. Dr. Bernd Simeon
Christian Gobert
Thomas März

TECHNISCHE UNIVERSITÄT MÜNCHEN

Numerik

SS 2009

Musterlösung zu Hausaufgabenblatt 2
Abgabe: in den Übungen vom 17.6. oder 18.6.

Aufgabe 1: *(Sylvestergleichung)*

Es seien $A, B, C \in \mathbb{C}^{n\times n}$ gegeben. Die Sylvestergleichung für $X \in \mathbb{C}^{n\times n}$ lautet

$$AX - XB = C \tag{1}$$

und kann mit den Schurzerlegungen von A, B

$$A = URU^H \quad , \quad B = VSV^H$$

auf die Form

$$RY - YS = D \tag{2}$$

gebracht werden, wobei $D = U^H CV$ und $Y = U^H XV$.

a) Finden Sie einen Algorithmus zur Berechnung von Y.

   Hinweis: Schreiben Sie Gleichung (2) als n Gleichungen für die Spaltenvektoren.

b) Zeigen Sie: Gleichung (1) ist genau dann eindeutig lösbar, falls

$$\sigma(A) \cap \sigma(B) = \emptyset$$

   gilt, also die Spektren der Matrizen A und B disjunkt sind.

c) **Dieser Teil ist optional !**
   Implementieren Sie einen Algorithmus zur Lösung von Gleichung (1). Wie viele Operationen benötigt er ?

   Hinweis: Matlab bietet die Funktion `schur`, Sie benötigt $O(n^3)$ Operationen.

Lösung Aufgabe 1: $\boxed{\text{10 Punkte}}$

a) Seien $Y = [y_1|y_2|\ldots|y_n]$, analog $S = [s_1|s_2|\ldots|s_n]$ und analog $D = [d_1|d_2|\ldots|d_n]$ in Spalten unterteilt (Spalteneinteilen $\boxed{\text{1 P.}}$) dann erhält man für y_k das Gleichungssystem

$$Ry_k - Ys_k = d_k \,, \qquad \boxed{\text{1 P.}}$$

   Weil S eine obere Dreiecksmatrix ist gilt für die Spalte s_k in Komponeten

$$s_k = (s_{1k}, s_{2k}, \ldots, s_{kk}, 0, \ldots, 0)^T \,. \qquad \boxed{\text{1 P.}}$$

Das Matrixvektorprodukt schreibt man als

$$Ys_k = \sum_{i=1}^{n} s_{ik} y_k = \sum_{i=1}^{k} s_{ik} y_k \,, \qquad \boxed{1 \text{ P.}}$$

um eine Rekursionsgleichung zu erhalten

$$(R - s_{kk} I) y_k = d_k + \sum_{i=1}^{k-1} s_{ik} y_k \qquad \boxed{2 \text{ P.}}$$

mit Startsituation

$$(R - s_{11} I) y_1 = d_1 \,. \quad \boxed{1 \text{ P.}}$$

Nachdem R auch eine obere Dreiecksmatrix ist, können die n Systeme der Rekursion jeweils mit Rückwärtssubstitution gelöst werden.

b) Gleichung (1) ist eindeutig lösbar genau dann, wenn Gleichung (2) es ist $\boxed{1 \text{ P.}}$.
Damit jede obere Dreiecksmatrix $(R - s_{kk} I)$ invertierbar ist, müssen für jedes k alle Diagonalenelemente von $(R - s_{kk} I)$ von Null verschieden sein, also $r_{jj} - s_{kk}$ für alle k und j $\boxed{1 \text{ P.}}$.
Da aber R, S aus Schurzerlegungen stammen stehen auf den Diagonalen jeweils die Eigenwerte von A, B und die letzte Bedingung ist äquivalent zu

$$\sigma(A) \cap \sigma(B) = \emptyset \,. \qquad \boxed{1 \text{ P.}}$$

Aufgabe 2: *(Rayleighquotient)*

Es sei $A \in \mathbb{C}^{n \times n}$ und $(\lambda, x) \in \mathbb{C} \times \mathbb{C}^n$ ein Rechtseigenpaar von A. Zeigen Sie für den Rayleighquotienten

$$r(x, A) = \frac{x^H A x}{x^H x}$$

folgende Störungseigenschaften:

a) $r(x+h, A) - r(x, A) = O(\|h\|)$, für $\|h\| \to 0$,

b) $r(x+h, A) - r(x, A) = O(\|h\|^2)$, für $\|h\| \to 0$, falls A normal ist.

Lösung Aufgabe 2: $\boxed{10 \text{ Punkte}}$

a) Hier gibt es 2 Lösungsansätze, ein (einfacherer) Ansatz i) und ein Ansatz ii), der bereits auf Aufgabenteil b) vorbereitet. Deshalb gibt Ansatz i) nur 5 Punkte und Ansatz ii) 8 Punkte.

i) Für $\delta \in \mathbb{R}$, $\delta \to 0$ gilt (Taylorentwicklung)

$$\frac{1}{1 + \mathcal{O}(\delta)} = 1 + \mathcal{O}(\delta) \qquad \boxed{1 \text{ P.}}$$

Damit folgt:

$$\begin{aligned} r(x+h, A) &= \frac{x^H A x + h^H A x + x^H A h + h^H A h}{x^H x + 2h^H x + h^H h} = \frac{x^H A x}{x^H x + 2h^H x + h^H h} \\ &\quad + \frac{h^H A x + x^H A h}{x^H x + 2h^H x + h^H h} + \frac{h^H A h}{x^H x + 2h^H x + h^H h} \\ &= \frac{x^H A x}{x^H x + \mathcal{O}(\|h\|)} + \mathcal{O}(\|h\|) + \mathcal{O}(\|h\|^2) = r(x, A) + \mathcal{O}(\|h\|) \qquad \boxed{4 \text{ P.}} \end{aligned}$$

ii) Nachdem (λ, x) ein Rechtseigenpaar von A ist gilt

$$r(x,A) = \frac{x^H A x}{x^H x} = \frac{\lambda x^H x}{x^H x} = \lambda\,. \qquad \boxed{1 \text{ P.}}$$

Für die Störung betrachten wir erst $(x+h)^H A(x+h)$

$$\begin{aligned}(x+h)^H A(x+h) &= x^H A x + h^H A x + x^H A h + h^H A h\\ &= \lambda x^H x + h^H A x + x^H A h + h^H A h\,. \qquad \boxed{1 \text{ P.}}\end{aligned}$$

Um links $r(x+h,A)$ per Division zu erhalten , wäre es günstiger wenn rechts $\lambda(x+h)^H(x+h)$ stünde. Wir ergänzen entsprechend und fassen zusammen:

$$\begin{aligned}(x+h)^H A(x+h) &= \lambda(x+h)^H(x+h) - \lambda(x+h)^H(x+h)\\ &\quad + \lambda x^H x + h^H A x + x^H A h + h^H A h\\ &= \lambda(x+h)^H(x+h) - \lambda(x^H x + h^H x + x^H h + h^H h)\\ &\quad + \lambda x^H x + h^H A x + x^H A h + h^H A h\\ &= \lambda(x+h)^H(x+h)\\ &\quad + h^H(A-\lambda I)x + x^H(A-\lambda I)h + h^H(A-\lambda I)h\\ &= \lambda(x+h)^H(x+h) + x^H(A-\lambda I)h + h^H(A-\lambda I)h\,, \qquad \boxed{1 \text{ P.}}\end{aligned}$$

wobei wir im letzten Schritt nochmal ausgenutzt haben, dass (λ, x) Rechtseigenpaar ist und somit

$$h^H(A-\lambda I)x = 0 \qquad \boxed{1 \text{ P.}}$$

gilt.

Division mit $(x+h)^H(x+h)$ liefert

$$\begin{aligned}r(x+h,A) &= \lambda + \frac{x^H(A-\lambda I)h + h^H(A-\lambda I)h}{(x+h)^H(x+h)}\\ &= r(x,A) + \frac{x^H(A-\lambda I)h + h^H(A-\lambda I)h}{x^H x + h^H x + x^H h + h^H h}\,. \qquad \boxed{1 \text{ P.}}\end{aligned}$$

Wir trennen nun h in Norm und Richtung

$$n := \frac{h}{\|h\|} \quad \Rightarrow \quad h = \|h\| \cdot n$$

somit gilt

$$\begin{aligned}r(x+h,A) - r(x,A) &= \frac{(x^H(A-\lambda I)n)\|h\| + (n^H(A-\lambda I)n)\|h\|^2}{x^H x + (n^H x + x^H n)\|h\| + \|h\|^2} \qquad \boxed{1 \text{ P.}}\\ &= \frac{O(\|h\|)}{1+O(\|h\|)} = O(\|h\|)(1+O(\|h\|)) = O(\|h\|)\,. \qquad \boxed{2 \text{ P.}}\end{aligned}$$

b) Falls A normal ist, ist (λ, x) auch Linkseigenpaar und somit zusätzlich

$$x^H(A-\lambda I)n = 0\,, \qquad \boxed{1 \text{ P.}}$$

damit gilt hier

$$\begin{aligned}r(x+h,A) - r(x,A) &= \frac{(n^H(A-\lambda I)n)\|h\|^2}{x^H x + (n^H x + x^H n)\|h\| + \|h\|^2}\\ &= \frac{O(\|h\|^2)}{1+O(\|h\|)} = O(\|h\|^2)\,. \qquad \boxed{1 \text{ P.}}\end{aligned}$$

Aufgabe 3: *(Rayleighquotienteniteration)*

Betrachten Sie folgenden Algorithmus zur Eingabe $A \in \mathbb{C}^{n\times n}$, $0 \neq v_0 \in \mathbb{C}^n$, $\mu_0 \in \mathbb{C}$:

$$
\begin{array}{l}
\texttt{for } k = 1, 2, \ldots \\
\quad \texttt{solve for } w : (\mu_{k-1} I - A) w = v_{k-1} \\
\quad v_k := w / \|w\| \\
\quad \mu_k := v_k^H A v_k \\
\texttt{end}
\end{array}
$$

a) Zeigen Sie, dass für normales $A \in \mathbb{C}^{2\times 2}$ mit den reellen Eigenwerten $\lambda_1 > \lambda_2$ gilt: Falls obiger Algorithmus konvergiert, dann konvergiert v_k kubisch gegen einen Eigenvektor von A.

   Hinweise:

   (a) Nutzen Sie die Diagonalisierbarkeit von A um die Iteration in eine einfachere für die Vektoren $\hat{v}_k$, $\hat{w}$ zu transformiern.

   (b) Sie dürfen ohne Einschräkung annehmen, dass die Folge der v_k gegen einen Eigenvektor zum Eigenwert λ_1 konvergiert.
   Wogegen konvergiert dann die Folge der $\hat{v}_k$?

   (c) Bezeichne α_k, β_k die Komponenten von $\hat{v}_k$:

   $$\hat{v}_k =: (\alpha_k, \beta_k)^T \quad \text{und} \quad h_k := \frac{|\beta_k|}{|\alpha_k|}$$

   bezeichne das Verhältnis der Komponenten. Stellen Sie h_k in Abhängigkeit von h_{k-1} dar indem Sie α_k, β_k, μ_{k-1} durch α_{k-1}, β_{k-1} ausdrücken. Treffen Sie geeignete Annahmen über α_0, β_0 und μ_0.

   (d) Zeigen Sie, dass die Konvergenzgeschwindigkeit des Algorithmus' durch die Konvergenzgeschwindigkeit der h_k gegeben ist.

b) **Dieser Teil ist optional !**
   Implementieren Sie obigen Algorithmus. Testen Sie Ihre Implementierung z.B. an einer normalen Matrix $A \in \mathbb{C}^{100\times 100}$ und stellen Sie die Konvergenz der Folgen μ_k für verschiedene Startwerte μ_0 graphisch dar.

Lösung Aufgabe 3: $\boxed{\text{20 Punkte}}$

Nachdem A normal ist gibt es eine unitäre Transformation U so, dass

$$A = UDU^H \quad \text{mit} \quad D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \qquad \boxed{\text{1 P.}}$$

damit ist die erste Anweisung des Algorithmus äquivalent zu

$$(\mu_{k-1} I - A) w = v_{k-1} \quad \Leftrightarrow \quad (\mu_{k-1} I - D)(U^H w) = (U^H v_{k-1}) \qquad \boxed{\text{1 P.}}$$

und wir setzen $\hat{v}_{k-1} = U^H v_{k-1}$, $\hat{w} = U^H w$.

Wegen der Unitarität von U gilt weiterhin

$$\hat{w}/\|\hat{w}\| = U^H w / \|U^H w\| == U^H w / \|w\| = U^H v_k = \hat{v}_k \,, \qquad \boxed{\text{1 P.}}$$

also transformiert sich automatisch die ganze Folge v_k zu $\hat{v}_k$.

Schließlich gilt auch

$$\mu_k := v_k^H A v_k = v_k^H (UDU^H) v_k = (U^H v_k)^H D (U^H v_k) = \hat{v}_k^H D \hat{v}_k \;, \qquad \boxed{1\text{ P.}}$$

also die Folge $\hat{v}_k$ erfüllt die gleiche Iteration wie die Folge v_k nur mit D anstatt A $\boxed{1\text{ P.}}$.

Wenn v_k gegen einen (nach Konstruktion) normierten Eigenvektor v zum Eigenwert λ_1 konvergiert, dann konvergiert $\hat{v}_k$ wegen selbiger Normierung und der Wahl der Zerlegung von $A = UDU^H$ gegen $e^{i\varphi}e_1$ $\boxed{1\text{ P.}}$, wobei sich φ in Abhängigkeit von v einstellt. Und wiederum die Unitarität von U liefert

$$\|v_k - v\| = \|U^H(v_k - v)\| = \|\hat{v}_k - e^{i\varphi}e_1\| \;.$$

Die Frage ist also: Wie schnell konvergiert $\hat{v}_k$ gegen $e^{i\varphi}e_1$? $\boxed{1\text{ P.}}$

Mit

$$\hat{v}_k =: (\alpha_k, \beta_k)^T \quad \text{und} \quad h_k := \frac{|\beta_k|}{|\alpha_k|}$$

und dem vorher gesagten, gilt $\alpha_k \to e^{i\varphi}$ also $|\alpha_k| \to 1$ $\boxed{1\text{ P.}}$, also kann man schreiben

$$\hat{v}_k =: \frac{1}{|\alpha_k|}(e^{i\varphi_k}, e^{i\psi_k}h_k)^T \qquad \boxed{1\text{ P.}}$$

und die Konvergenzgeschwindigkeit ergibt sich aus der Geschwindigkeit mit der h_k gegen Null geht $\boxed{1\text{ P.}}$.

Ab $k \geq 2$ gilt

$$\mu_{k-1} = \lambda_1|\alpha_{k-1}|^2 + \lambda_2|\beta_{k-1}|^2 \qquad \boxed{1\text{ P.}}$$

und wegen Normiertheit von $\hat{v}_k$ noch $|\alpha_{k-1}|^2 + |\beta_{k-1}|^2 = 1$ $\boxed{1\text{ P.}}$. Damit folgt

$$\begin{aligned} \mu_{k-1} - \lambda_1 &= -(\lambda_1 - \lambda_2)|\beta_{k-1}|^2 \\ \mu_{k-1} - \lambda_2 &= (\lambda_1 - \lambda_2)|\alpha_{k-1}|^2 \qquad \boxed{1\text{ P.}} \end{aligned}$$

Der Hilfsvektor $\hat{w}$ kann damit geschrieben werden als

$$\hat{w} = \left(\frac{\alpha_{k-1}}{\mu_{k-1} - \lambda_1}, \frac{\beta_{k-1}}{\mu_{k-1} - \lambda_2}\right)^T = \frac{1}{\lambda_1 - \lambda_2}\left(\frac{-\alpha_{k-1}}{|\beta_{k-1}|^2}, \frac{\beta_{k-1}}{|\alpha_{k-1}|^2}\right)^T \qquad \boxed{1\text{ P.}}$$

mit Norm

$$\|\hat{w}\| = \frac{1}{\lambda_1 - \lambda_2}\frac{1}{|\beta_{k-1}|^2|\alpha_{k-1}|^2}\sqrt{|\alpha_{k-1}|^6 + |\beta_{k-1}|^6} \;. \qquad \boxed{1\text{ P.}}$$

Schließlich erhält man $\hat{v}_k$

$$(\alpha_k, \beta_k)^T = \hat{v}_k = \hat{w}/\|\hat{w}\| = \frac{1}{\sqrt{|\alpha_{k-1}|^6 + |\beta_{k-1}|^6}}(-\alpha_{k-1}|\alpha_{k-1}|^2, \beta_{k-1}|\beta_{k-1}|^2)^T \;, \qquad \boxed{1\text{ P.}}$$

und es folgt

$$h_k = \frac{|\beta_k|}{|\alpha_k|} = \frac{|\beta_{k-1}|^3}{|\alpha_{k-1}|^3} = h_{k-1}^3 \qquad \boxed{1\text{ P.}}$$

bzw. nach Auflösung der h_k-Rekursion

$$h_k = (h_1)^{3(k-1)} \;, \qquad \boxed{1\text{ P.}}$$

was die kubische Konvergenz zeigt falls $h_1 < 1$.

h_1 gewinnen wir aus den Startdaten

$$h_1 = \frac{|\beta_0|}{|\alpha_0|} \frac{|\mu_0 - \lambda_1|}{|\mu_0 - \lambda_2|} \qquad \boxed{1 \text{ P.}}$$

und daraus diverse Möglichkeiten für eine geeignete Startdatenwahl, damit $h_1 < 1$ gilt. Da man die Transformation U nicht kennt, nehmen wir an, dass $\frac{|\beta_0|}{|\alpha_0|}$ zufällig gegeben ist, dann reicht es den initialen Shift μ_0 nur nahe genug an λ_1 gewählt zu haben $\boxed{1 \text{ P.}}$ für irgendein sinnvollen Einstellen von h_1.