Iteratives Lösen linearer Gleichungssysteme - Übung 1

Tobias Danczul

17. März 2017

1 Beispiel

a)

Beh.: $\forall x \in \mathbb{C} : \Re(Ax, x) = (\Re Ax, x)$

$$\begin{aligned}
(\Re Ax, x) &= \frac{1}{2}(Ax + A^* x, x) = \frac{1}{2}(Ax, x) + \frac{1}{2}(A^* x, x) \\
&= \frac{1}{2} x^* A^* x + \frac{1}{2} x^* Ax \\
\Re(Ax, x) &= \frac{1}{2}\left((Ax, x) + \overline{(Ax, x)}\right) \\
&= \frac{1}{2}\left((Ax)^* x + \overline{(Ax)^* x}\right) \\
&= \frac{1}{2} x^* A^* x + \frac{1}{2}\overline{x^* A^* x} \\
&= \frac{1}{2} x^* A^* x + \frac{1}{2} x^T A^T \overline{x}
\end{aligned}$$

Insgesamt gilt also:

$$\Re(Ax, x) = (\Re Ax, x) \Longleftrightarrow \overline{x}^T Ax = x^T A^T \overline{x}$$

Transponieren des rechten Ausdruckes führt zur Behauptung.

Beh.: $\forall x \in \mathbb{C} : \Im(Ax, x) = -(\Im Ax, x)$
Sowohl für Matrizen $A \in \mathbb{C}^{n \times n}$ als auch für Skalare $A \in \mathbb{C}$ gilt:

$$\Im A = -i(A - \Re A)$$

Damit folgt einerseits:

$$\Im(Ax, x) = -i(Ax, x) + i\Re(Ax, x) \tag{1}$$

Und andererseits:

$$-(\Im Ax, x) = -(-iAx, x) - (i\Re Ax, x) = -i(Ax, x) + i(\Re Ax, x) = -i(Ax, x) + i\Re(Ax, x) \tag{2}$$

Insgesamt folgt also mit (1) und (2) die Behauptung.
Eine weitere Möglichkeit, die Aussage zu zeigen, liefert schlichtes Nachrechnen:

$$\begin{aligned}
\Im(Ax,x) &= \frac{1}{2i}\Big((Ax,x) - \overline{(Ax,x)}\Big) = \frac{1}{2i}\Big((Ax)^*x - \overline{(Ax)^*x}\Big) \\
&= \frac{1}{2i}\left((x^*A^*x - \overline{x^*A^*x}\right)) \\
&= \underbrace{\frac{1}{2i}}_{-\frac{i}{2}}\left(x^*A^*x - x^TA^T\overline{x}\right) \\
(\Im Ax,x) &= \Big(\underbrace{\frac{1}{2i}}_{-\frac{i}{2}}(A-A^*)x,x\Big) = \frac{i}{2}\Big((Ax,x) - (A^*x,x)\Big) \\
&= \frac{i}{2}(x^*A^*x - x^*Ax)
\end{aligned}$$

Insgesamt gilt also:

$$\Im(Ax,x) = -(\Im Ax,x) \Longleftrightarrow x^TA^T\overline{x} = x^*Ax$$

Transponieren der rechten Ausdrucks führt zur Behauptung.
Was bedeutet diese Identität für reelle Matrizen? In diesem Fall gilt, dass $(Ax,x) = (\Re Ax,x)$ und der Realteil von A genau der symmetrische Anteil der Matrix ist.

b)

Beh.: $A^*A = AA^* \Longleftrightarrow \Re A \cdot \Im A = \Im A \cdot \Re A$

$$\begin{aligned}
\Re A \cdot \Im A = \Im A \cdot \Re A &\Longleftrightarrow \frac{1}{4i}(A+A^*)(A-A^*) = \frac{1}{4i}(A-A^*)(A+A^*) \\
&\Longleftrightarrow A^2 + A^*A - AA^* - (A^*)^2 = A^2 - A^*A + AA^* - (A^*)^2 \\
&\Longleftrightarrow 2A^*A - 2AA^* = 0 \\
&\Longleftrightarrow A^*A - AA^* = 0 \\
&\Longleftrightarrow A^*A = AA^* \\
&\Longleftrightarrow A\,\text{normal}
\end{aligned}$$

c)

Beh.: $\forall x \in \mathbb{C}^n : \|x\|_2 = \sup_{y \in \mathbb{C}^n \setminus \{0\}} \frac{|(x,y)|}{\|y\|_2}$

Es gilt:

$$\forall x,y \in \mathbb{C}^n : \quad |(x,y)| \overset{CS}{\leq} \|x\|_2\|y\|_2 \Longrightarrow \frac{|(x,y)|}{\|y\|_2} \leq \|x\|_2$$

Diese Ungleichung bleibt auch unter Bildung des Supremums erhalten. Mit der Wahl von $y = x$ folgt sogar Gleichheit. Damit folgt die Behauptung.

d)

Beh.: $A \in \mathbb{R}^{n\times n}$ positiv definit $\Longrightarrow A$ invertierbar mit $\|A^{-1}\|_2 \leq \frac{1}{\lambda_{min}(\Re A)}$

$$\begin{aligned}
\|Ax\|_2^2 &\overset{(*)}{=} \|\Re Ax\|_2^2 + \|\Im Ax\|_2^2 \\
&\geq (\Re Ax, \Re Ax) = (QD\underbrace{Q^Tx}_{:=y}, QD\underbrace{Q^Tx}_{=y}) \\
&= (D^2y,y) \geq \lambda_{min}^2\|y\|_2^2 = \lambda_{min}^2\|x\|_2^2
\end{aligned}$$

wobei λ_{min} den betragskleinsten Eigenwert von $\Re A$ bezeichne. Die letzte Gleichheit gilt aufgrund der unitären Eigenschaft von Q. Die Gleichheit mit $(*)$ gilt aufgrund von Pythagoras.
Damit folgt unmittelbar nach Ziehung der Wurzel:

$$\frac{\|Ax\|_2}{\|x\|_2} \geq \lambda_{min}$$

Unter Berücksichtigung der Definition von $y := Ax$ folgt

$$\frac{\|y\|_2}{\|A^{-1}y\|_2} \geq \lambda_{min}$$

und damit:

$$\|A^{-1}\|_2 = \sup_{y \in \mathbb{C}^n \setminus \{0\}} \frac{\|A^{-1}y\|_2}{\|y\|_2} \leq \frac{1}{\lambda_{min}}$$

Unter Verwendung von c) kann diese Ungleichung auch einfacher gezeigt werden. Sei dafür $x \in \mathbb{C}^n$ fest aber beliebig, dann gilt:

$$\|Ax\|_2 \overset{c)}{=} \sup_{y \in \mathbb{C}^n \setminus \{0\}} \frac{|(Ax, y)|}{\|y\|_2} \geq \frac{|(Ax, x)|}{\|x\|_2} = \frac{|(\Re Ax, x)|}{\|x\|_2} \geq \frac{\lambda_{min}(\Re A)\|x\|_2^2}{\|x\|_2} = \lambda_{min}(\Re A)\|x\|_2$$

Einsetzen von $x = A^{-1}y$ liefert:

$$\|AA^{-1}y\|_2 = \|y\|_2 \geq \lambda_{min}(\Re A)\|A^{-1}y\|_2 \iff \frac{\|A^{-1}y\|_2}{\|y\|_2} \leq \frac{1}{\lambda_{min}(\Re A)}$$

Da unter Anwendung des Supremums die Gleichheit bestehen bleibt, folgt:

$$\|A^{-1}\|_2 \leq \frac{1}{\lambda_{min}(\Re A)}$$

e)

Geg.: Betrachte $A \in \mathbb{R}^{n \times n}$ beliebig und das dazugehörige System von Differentialgleichungen der Form $x'(t) = Ax(t)$.
Beh.: Für alle Lösungen $x(t)$ gilt: $\frac{d}{dt}\|x(t)\|_2 \leq \|A\|_2\|x(t)\|_2$
Es gilt:

$$\frac{d}{dt}\|x(t)\|_2 = \frac{2(x'(t), x(t))}{2\sqrt{(x(t), x(t))}} = \frac{(x'(t), x(t))}{\|x\|_2}$$

Da x die Differentialgleichung löst, gilt für den Zähler:

$$(x', x) = (Ax, x) \leq \|Ax\|_2\|x\|_2 \leq \|A\|_2\|x\|_2^2$$

Insgesamt folgt damit:

$$\frac{d}{dt}\|x(t)\|_2 \leq \frac{\|A\|_2\|x\|_2^2}{\|x\|_2} = \|A\|_2\|x(t)\|_2$$

Im Folgenden wollen wir diese Aussage weiter verschärfen.
Beh.: $\frac{d}{dt}\|x(t)\|_2 \leq \lambda_{max}(\Re A)\|x(t)\|_2$

$$\frac{d}{dt}\|x(t)\|_2 = \frac{(Ax, x)}{\|x\|_2} = \frac{(\Re Ax, x)}{\|x\|_2} \leq \max_{x(t)\text{Lsg.}} \frac{(\Re Ax, x)}{\|x\|_2^2}\|x\|_2 = \lambda_{max}(\Re A)\|x\|_2$$

Beim zweiten Gleichheitszeichen wurde benutzt, dass es sich bei A um eine reelle Matrix handelt.
Um zu zeigen, dass es sich bei der zweiten Abschätzung tatsächlich um eine Verschärfung der ersteren handelt, betrachten wir folgendes Beispiel: Sei A eine reelle Matrix mit:

$$A := \begin{pmatrix} a & b \\ -b & a \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

sodass gemäß unserer Überlegungen aus a) der Realteil der Matrix genau der symmetrische Anteil der Matrix A ist, also:

$$\Re A = aI = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix}$$

Es gilt: $\sigma(AA^T) = a^2 + b^2$. Da A insbesondere normal ist, entspricht die Spektralnorm von A genau dem betragsgrößten Eigenwert von AA^T, also:

$$\|A\|_2 = \sqrt{\rho(AA^T)} = \sqrt{a^2 + b^2} > a = \sigma(\Re A)$$

für hinreichend große Rotationen b, womit die Aussage bewiesen wäre.

Erstellt von Daniel Herold

Beispiel 2

a)

Beh.: Jede beliebige Matrix $A \in \mathbb{C}^{n\times n}$ erfüllt $\chi(A) = \det(A - \lambda I) = 0$.

Wir betrachten den allgemeinen Fall, wo A lediglich eine Jordannormalform besitzt, demnach gilt also $A = XJX^{-1}$, wobei die Jordan Matrix J die folgende Blockdiagonalgestalt besitzt:

$$J = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_k \end{pmatrix}$$

und die einzelnen Jordan-Blöcke J_i für alle $i = 1...k$ mit $k \leq n$ die folgende Gestalt besitzen:

$$J_i := \begin{pmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix} = \underbrace{\begin{pmatrix} \lambda_i & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \lambda_i \end{pmatrix}}_{EW-Diagonalmatrix} + \underbrace{\begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{pmatrix}}_{Nilpotenter Anteil} \in \mathbb{C}^{n_i \times n_i}$$

mit den Eigenwert $\lambda_i \in \mathbb{C}$ auf der Diagonalen und der zugehörigen algebraischen Vielfachheit $n_i \in \mathbb{N}$. Da ähnliche Matrizen das gleiche Charakteristische Polynom besitzen, gilt:

$$\chi(A) = \chi(XJX^{-1}) = \chi(J)$$

Aufgrund der Blockdiagonalgestalt von J, gilt gemäß Definition:

$$\chi(J) = \begin{pmatrix} \chi(J_1) & & \\ & \ddots & \\ & & \chi(J_k) \end{pmatrix}$$

Betrachten wir zu einem fixen $i \in \{1...k\}$ den zugehörigen Jordan-Block J_i, so erfüllt das zugehörige charakteristische Polynom $\chi_i(\lambda) = (\lambda - \lambda_i)^{n_i}$. Einsetzen des Jordan-Blocks führt zu

$$\chi_i(J_i) = (J_i - \lambda_i I)^{n_i} = 0$$

da $J_i - \lambda_i I$ eine obere Dreiecksmatrix mit 0er auf der Diagonale. Damit bleibt nur noch der nilpotente Anteil der Matrix erhalten, welcher mittels n_i-maliger Potenzierung in die Nullmatrix übergeführt wird.

$$J_i - \lambda_i = \begin{pmatrix} 0 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & \ddots & 0 \\ & & & & 0 \end{pmatrix} \mapsto \cdots \mapsto \begin{pmatrix} 0 & 0 & 0 & & 1 \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 0 \\ & & & \ddots & 0 \\ & & & & 0 \end{pmatrix} \mapsto \mathbf{0}$$

Unter Berücksichtigung eines passenden Restpolynoms $\pi_i(\lambda)$, welches die Identität

$$\chi(\lambda) = \chi_i(\lambda)\pi_i(\lambda) = (\lambda - \lambda_i)^{n_i}\pi_i(\lambda)$$

erfüllt, folgt: $\chi(J_i) = \chi_i(J_i)\pi_i(J_i) = 0$. Obige Überlegungen auf jeden Block einzeln angewandt, liefern, dass $\chi(J) = 0$ gilt und damit auch die Behauptung.

b)

Geg.: $f : D \longrightarrow \mathbb{C}$ Funktion mit $\sigma(A) \subset D$
Beh.: $f(A) := Xf(\Lambda)X^{-1}$ ist ein Matrixpolynom p in A mit $\deg p \leq n-1$
Wir wollen im Folgenden zeigen, dass p dem Langrange-Interpolationspolynom von $f(z)$ zu den Knoten $z = \lambda_i$ entspricht, also

$$f(A) = \sum_{j=0}^{n-1} \prod_{\substack{i=0 \\ i \neq j}}^{n-1} \frac{A - \lambda_i I}{\lambda_j - \lambda_i} f(\lambda_j) =: p(A)$$

Dafür betrachten wir: $f(A) = Xf(\Lambda)X^{-1} \overset{!}{=} p(A) = Xp(\Lambda)X^{-1}$
Also gilt:

$$f(A) = p(A) \Longleftrightarrow f(\Lambda) = p(\Lambda) \Longleftrightarrow \forall i \in \{0...n-1\} : f(\lambda_i) = p(\lambda_i)$$

Letzte Gleichheit ist aufgrund der Interpolationeigenschaft von p erfüllt. Das Polynom ist aufgrund seiner Definition maximal vom Grad $n-1$. Damit folgt die Behauptung.

c)

- **Geg.:** $A \in \mathbb{R}^{3\times 3}$ via:

$$A := \begin{pmatrix} -1 & 6 & -8 \\ -7 & 16 & -19 \\ -4 & 8 & -9 \end{pmatrix}$$

Ges.: Die beiden Matrizen A^{-5} und e^A ohne Verwendung der Eigenbasis von A.
Wir gehen wie in a) vor und berechnen das zu $f(A) := A^{-5}$ eindeutig zugeordnete Polynom p, sodass $f(A) = p(A)$ erfüllt ist. Die Berechnung der Eigenwerte fürt auf $\sigma(A) = \{1, 2, 3\}$. Ferner gilt:

$$\begin{aligned} p(x) &= \sum_{i=1}^{3} \prod_{\substack{i=1 \\ i \neq j}}^{3} \frac{A - \lambda_i I}{\lambda_j - \lambda_i} f(\lambda_j) \\ &= \frac{x - 2I}{1-2}\frac{x-3I}{1-3} f(1) + \frac{x-1I}{2-1}\frac{x-3I}{2-3} f(2) + \frac{x-1I}{3-1}\frac{x-2I}{3-2} f(3) \end{aligned}$$

Für $f(x) := \frac{1}{x^5}$ führt uns das in den ersten 4 Nachkommastellen zu:

$$p_1(X) = 0.4708X^2 - 2.3812X + 2.91I$$

Setzt man A in das Matrixpolynom ein, erhält man das auf 2 Nachkommastellen gerundete Ergebnis:

$$A^{-5} = p_1(A) = \begin{pmatrix} 1.05 & -2.05 & 3.04 \\ 3.01 & -6 & 8.99 \\ 1.99 & -3.98 & 5.98 \end{pmatrix}$$

Analoge liefert $f(x) := e^x$:

$$p_2(X) = 4.0129X^2 - 7.3678X + 6.0732I$$

Erneutes einsetzen resultiert in:

$$e^A = p_2(A) = \begin{pmatrix} -22.67 & 60.13 & -77.49 \\ -64.8 & 136.99 & -169 \\ -34.73 & 69.47 & -84.19 \end{pmatrix}$$

- Selbiges Vorgehen wiederholen wir für die Matrix

$$A := \begin{pmatrix} 1 & 2 & -3 \\ -3 & 8 & -9 \\ -2 & 4 & -4 \end{pmatrix}$$

Rechnungen ergeben, dass $\sigma(A) = \{1, 2\}$, wobei 2 sich als doppelte Nullstelle des zugehörigen charakteristischen Polynoms erweist. In diesem Fall lauten die Polynome:

$$p_1(X) = -0.9688X + 1.9688I$$
$$p_2(X) = 4.6708X - 1.9525I$$

mit den zugehörigen Matrizen:

$$A^{-5} = \begin{pmatrix} 1 & -1.94 & 2.91 \\ 2.91 & -5.78 & 8.72 \\ 1.94 & -3.88 & 5.85 \end{pmatrix} \quad \text{und} \quad e^A = \begin{pmatrix} 2.72 & 9.34 & -14.01 \\ -14.01 & 35.41 & -42.04 \\ -9.34 & 18.68 & -20.64 \end{pmatrix}$$

Lukas Kogler (1026606); Übung 1

4. April 2017

1 Aufgabe 4

1.1 Angabe

Sei $A \in \mathbb{R}^{n \times n}$ beliebig und p ein Polynom. Zu zeigen ist, dass

$$\sigma(p(A)) = p(\sigma(A))$$

D.h, dass das Spektrum von $p(A)$ genau das Bild von $\sigma(A)$ unter p ist.
p soll dabei die Koeffizienten $a_0 \dots a_m$ haben: $p(x) = \sum_{i=0}^{m} a_i x^i$.

1.2 Beweis

Wir werden dies zuerst für den Fall, dass A genau ein Jordan-Block ist, zeigen und das Resultat dann mit Hilfe der Jordan-Normalform auf allgemeine Matrizen ausweiten. Zuerst werde ich aber ein wenig Notation einführen, die die weitere Rechnung einfacher machen wird.

1.2.1 Notation

Sei $J_n^k \in \mathbb{R}^{n \times n}$ sodass $(J_n^k)_{ij} = \delta_j^{i+k}$, bzw:

$$J_n^k = \begin{pmatrix} 0 & \dots & 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & \ddots & \ddots & 0 & 1 & 0 & \dots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 & 1 & 0 \\ \vdots & & & & \ddots & \ddots & 0 & 1 \\ \vdots & & & & & \ddots & \ddots & 0 \\ \vdots & & & & & & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & 0 \end{pmatrix}$$

J_n^k ist also eine Einheitsmatrix der Größe n, bei der die Diagonale k-mal nach rechts geshiftet wurde. Die Einheitsmatrix der Größe n sei als I_n geschrieben. Dann gilt

$$J_n^k \cdot J_n^j = J_n^{k+j}$$

und insbesondere

$$J_n^0 = I_n \quad \text{and } J_n^k = 0 \text{ for } k \geq n$$

Außerdem gilt für einen Vector v: $(J_n^k v)_j = v_{j+k}$ (wobei wir $v_{n+j} = 0$ schreiben). Multiplikation mit J_n^k ist also ein k-facher Links-Shift (wobei rechts nur Nuller eingeschoben werden).
Da ein Jordan-Block der Größe m zum Eigenwert λ als $\lambda I_m + J_m^1$ geschrieben werden kann, wird uns diese Notation das Rechnen mit Potenzen von Jordan-Blöcken erleichtern.

1.2.2 Spezialfall: Jordan-Block

Sei also fürs erste $A = \lambda I_n + J_m^1$ ein einzelner Jordan-Block. Wir müssen nun zeigen, dass

$$\mu \in \sigma(p(A)) \Leftrightarrow \mu = p(\lambda)$$

Unter Verwendung des Binomischen Lehrsatzes lässt sich $p(A)$ nun schreiben als:

$$p(A) = \sum_{i=0}^{m} A^i = \sum_{i=0}^{m} a_i (\lambda I_n + J_n^1)^i = \sum_{i=0}^{m} \sum_{k=0}^{i} a_i \binom{i}{k} \lambda^{i-k} J_n^k = \sum_{k=0}^{m} \left(\sum_{i=k}^{m} a_i \binom{i}{k} \lambda^{i-k} \right) J_n^k$$

Sei $\mu \in \sigma(p(A))$ und $v \in \mathbb{R}^n$ der Eigenvektor von $p(A)$ zu μ. Falls $v_n \neq 0$, schauen wir uns $(p(A)v)_n$ an und sehen:

$$\left(p(A)v\right)_n = \sum_{k=0}^{m} \left(\sum_{i=k}^{m} a_i \binom{i}{k} \lambda^{i-k} \right) \left(J_n^k v\right)_n$$

Wie oben erwähnt, ist Multiplikation mit J_n^k ein k-facher Links-Shift, daher gilt $\forall k > 0 : (J_n^k v)_n = 0$ und uns fallen aus dieser Summe alle Terme außer denen für $k = 0$ weg:

$$\left(p(A)v\right)_n = \ldots = \sum_{i=0}^{m} a_i \lambda^i v_n = p(\lambda) v_n$$

Falls andererseits $v_l = 0$ für $l > j$ aber $v_j \neq 0$, schauen wir uns $(p(A)v)_j$ an, wobei uns diesmal aufgrund der Shifts nur die Terme für $k \leq n - j$ übrigbleiben.

$$(p(A)v)_j = \sum_{i=0}^{m} a_i \lambda^i v_j + \sum_{k=1}^{n-j} \left(\sum_{i=k}^{m} a_i \binom{i}{k} \lambda^{i-k} \right) \underbrace{v_{j+k}}_{=0} = p(\lambda) v_j$$

Falls also $\mu \in \sigma(p(A))$, also $\mu v = p(A)v$ folgt direkt $\mu = p(\lambda)$, also $\sigma(p(A)) \subseteq p(\sigma(A))$.
Falls andererseits $\lambda v = Av$ gilt:

$$p(A)v = \sum_{i=0}^{m} a_i A^i v = \sum_{i=0}^{m} a_i \lambda^i v = p(\lambda) v \Rightarrow p(\lambda) \in \sigma(p(A))$$

Damit ist auch $p(\sigma(A)) \subseteq \sigma(p(A))$ gezeit.

1.2.3 Der allgemeine Fall

Nun müssen wir diesen Satz noch auf allgemeine Matrizen A erweitern. Sei dafür $A = Q^T \cdot J \cdot Q$ die Jordan-Normalform von A. Q ist bei eine orthogonale Matrix, dh $QQ^T = Q^T Q = I$ und J ist block-diagonal, wobei jeder Block ein Jordan-Block ist.

$$A^2 = Q^T J \underbrace{QQ^T}_{=I} JQ = Q^T JQ \Rightarrow A^k = Q^T J^k Q \Rightarrow p(A) = Q^T p(J) Q$$

J sei block-diagonal mit $J = \mathrm{diag}(B_1, \ldots, B_m)$, also $J^k = \mathrm{diag}(B_1^k, \ldots, B_m^k)$ und damit

$$p(A) = Q^T \mathrm{diag}(p(B_1), \ldots, p(B_m)) Q$$

Jeder der Blöcke B_l ist ein Jordan-Block (für den wir das Resultat bereits gezeigt haben), und da das Spektrum einer Block-Diagonal-Matrix genau die Verinigung der Spektren ihrer Blöcke ist, $\sigma(\mathrm{diag}(B_1 \ldots B_m)) = \bigcup_{i=1}^{m} \sigma(B_i)$, gilt die Aussage des Satzes nun für alle Matrizen, die eine Jordan-Darstellung besitzen. Als Matrizen in $\mathbb{C}^{n \times n}$ sind das alle.

Iteratives Lösen linearer Gleichungssysteme
Übung 1 - Beispiel 5

Michael Neunteufel

19. März 2017

Das **Compressed Sparse Column Storage Format (CSC)** besteht aus drei Vektoren:

- $\mathrm{AA} \in \mathbb{R}^{nnz}$: besteht aus den nnz Nicht-Null-Elementen der Matrix A in spaltenweiser Sortierung
- $\mathrm{IR} \in \mathbb{N}^{nnz}$: besteht aus den zugehörigen Zeilenindizes
- $\mathrm{PC} \in \mathbb{N}^{n+1}$: besteht aus Pointern, an welcher Stelle in AA eine neue Spalte beginnt

Die Matrix-Vektor-Multiplikation $y = Ax$ welche definiert ist als

$$y = \sum_{i=1}^{n} \mathbf{e}_i \sum_{j=1}^{n} A_{ij} x_j$$

wird dazu in die dazu *duale* Formulierung

$$y = \sum_{j=1}^{n} x_j \sum_{i=1}^{n} A_{ij} \mathbf{e}_i$$

gebracht, wobei $A \in \mathbb{C}^{n\times n}$, $x, y \in \mathbb{C}^n$ und $\mathbf{e}_i$ den i-ten Einheitsvektor bezeichnet.
Diese Darstellung legt bereits nahe, dass bei jeder Iteration über eine Spalte von A nur ein Eintrag von x verwendet wird, dafür aber in jede Komponente von y geschrieben werden könnte.

Eine Implementierung des Matrix-Vektor-Produkts einer Matrix $A \in \mathbb{R}^{n\times n}$, gespeichert im CSC-Format mit einem Vektor $x \in \mathbb{R}^n$ in *Python*:

```
def MatrixVectorCSC(AA,IR,PC,x):
    n = np.size(x)
    y = np.zeros(n)
    for j in range(n):
        for i in range(PC[j],PC[j+1]):
            y[IR[i]] += AA[i]*x[j]
    return y
```

Zum Testen der Funktion werden folgende zwei Matrizen verwendet:

$$A := \begin{pmatrix} 2 & 0 & 0 & 1 & -3 \\ -1 & 0 & 3 & -1 & 0 \\ 9 & 0 & 2 & 0 & 0 \\ 0 & 0 & -2 & 6 & 0 \\ 0 & 0 & -1 & 0 & 2 \end{pmatrix} \quad B := \begin{pmatrix} 5 & 1 & 1 & 0 & 2 \\ 0 & 0 & 0 & 3 & 0 \\ -1 & 3 & 0 & 4 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 6 & 0 & -1 & 3 & 0 \end{pmatrix},$$

mit den dazugehörigen Vektoren:

$$\begin{aligned} \mathrm{AA}_a &= [2,-1,9,3,2,-2,-1,1,-1,6,-3,2] & \mathrm{AA}_b &= [5,-1,6,1,3,1,-1,-1,3,4,3,2,1] \\ \mathrm{IR}_a &= [0,1,2,1,2,3,4,0,1,3,0,4] & \mathrm{IR}_b &= [0,2,4,0,2,0,3,4,1,2,4,0,3] \\ \mathrm{PC}_a &= [0,3,3,7,10,12] & \mathrm{PC}_b &= [0,3,5,8,11,13] \end{aligned}$$

Zu beachten ist, dass in *Python* die Indizierung bei Null startet.

Zum Testen wird bei beiden Multiplikationen der Vektor

$$x := \begin{pmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 1 \end{pmatrix}$$

verwendet.
Der vollständige Code und die entsprechenden Ausgabe:

```
import numpy as np

def MatrixVectorCSC(AA, IC, PR, x):
    n = np.size(x)
    y = np.zeros(n)
    for j in range(n):
        for i in range(PR[j], PR[j+1]):
            y[IC[i]] += AA[i]*x[j]
    return y

AA_a = [2,-1,9,3,2,-2,-1,1,-1,6,-3,2]
IR_a = [0,1,2,1,2,3,4,0,1,3,0,4]
PC_a = [0,3,3,7,10,12]

x = [1,2,3,2,1]

print(MatrixVectorCSC(AA_a, IR_a, PC_a, x))

AA_b = [5,-1,6,1,3,1,-1,-1,3,4,3,2,1]
IR_b = [0,2,4,0,2,0,3,4,1,2,4,0,3]
PC_b = [0,3,5,8,11,13]

print(MatrixVectorCSC(AA_b, IR_b, PC_b, x))
```

```
[  1.   6.  15.   6.  -1.]
[ 12.   6.  13.  -2.   9.]
```