Prof. Dr. Karsten Urban
Dr. Timo Tonn
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

WiSe 2023/2024
10.12.2023

Angewandte Numerik

Übungsblatt 08

Besprechung in den Tutorien von Montag, 18.12.2023 bis Freitag, 22.12.2023

Für dieses Übungsblatt gibt es 13 Theorie- und 24 Matlab-Punkte, sowie 10 Theorie- und 7 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 70-Prozent-Grenzen liegen aktuell (inklusive Blatt 08) bei 109,2 Theorie- und 93,8 Matlab-Punkten.

Aufgabe 29 *(QR-Aufdatierung: Einfügen einer Zeile)* (4T Punkte)

Seien m und $n \in \mathbb{N}$ und sei $A \in \mathbb{R}^{m\times n}$ eine Matrix, deren QR-Zerlegung durch die orthogonale Matrix $Q \in \mathbb{R}^{m\times m}$ und die rechte obere Dreiecksmatrix $R \in \mathbb{R}^{m\times n}$ gegeben sei.

Der Vektor $x \in \mathbb{R}^n$ soll als i-te Zeile ($1 \leq i \leq m+1$) in die Matrix A eingefügt werden. Dadurch erhalten wir die Matrix $\tilde{A} := \begin{pmatrix} A_1 \\ x^T \\ A2 \end{pmatrix} \in \mathbb{R}^{(m+1)\times n}$ mit $A_1 \in \mathbb{R}^{(i-1)\times n}$, $A_2 \in \mathbb{R}^{(m-i+1)\times n}$ und $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$.

In der Vorlesung haben wir gesehen, dass sich die Matrix $\tilde{A}$ schreiben lässt als

$$\tilde{A} = \begin{pmatrix} A_1 \\ x^T \\ A2 \end{pmatrix} = \begin{pmatrix} 0 & Q_1 \\ 1 & 0 \\ 0 & Q_2 \end{pmatrix} \cdot \begin{pmatrix} x^T \\ R \end{pmatrix}.$$

Zeigen Sie, dass die Matrix $\hat{Q} = \begin{pmatrix} 0 & Q_1 \\ 1 & 0 \\ 0 & Q_2 \end{pmatrix}$ orthogonal ist.

Hinweis: Was können Sie aus der Orthogonalität der Matrix Q folgern, wenn Sie Q in der Form $Q = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$ betrachten?

Aufgabe 30 *(Programmieraufgabe: Eine Ellipse als Lineares Ausgleichsproblem, Einfügen einer Zeile)* (3T+3M+5M+2T*+3M*+2M* Punkte)

Gegeben sind die vier Messwert-Paare

x_i	-1.6	-0.9	0.75	2.7
y_i	1.6	-0.9	1.0	-1.0

Es ist bekannt, dass diese Messwerte zu einer Ellipse der Form

$$\alpha x^2 + \beta xy + \gamma y^2 - 10 = 0 \tag{1}$$

mit unbekannten Parametern α, β und γ gehören.

a) Stellen Sie das lineare Ausgleichsproblem $\|Az - b\|_2 \to \min$ zur Bestimmung der unbekannten Parameter auf. Geben Sie A, z und b explizit an. Welche Dimensionen haben diese Größen?

b) Lösen Sie das lineare Ausgleichsproblem mittels MATLAB. Bestimmen Sie dazu eine QR-Zerlegung der Matrix A und berechnen Sie die Lösung des linearen Ausgleichsproblems durch Rückwärtseinsetzen.

   Fertigen Sie mit MATLAB eine Skizze, in der die Ausgleichsellipse und die Messpunkte eingezeichnet sind. Beschriften Sie Ihre Zeichnung.

   Hinweis: In der Form (1) ist die Ellipse als sogenannte *implizite Funktion*, die von zwei Parametern x und y abhängt, gegeben. Diese als implizite Funktion gegebene Ellipse können Sie beispielsweise mit der MATLAB-Funktion `fimplicit` zeichnen.

c) Schreiben Sie eine MATLAB-Funktion `[Q, R] = myQrInsertRow(Q, R, i, x)`, mit der Sie eine gegebene QR-Zerlegung $A = Q \cdot R$ einer Matrix A effizient aktualisieren können. Dabei soll der Vektor `x` analog zur vorigen Aufgabe 29 als `i`-te Zeile in die ursprüngliche Matrix A eingefügt werden.

   Achten Sie darauf, dass Ihre MATLAB-Funktion `myQrInsertRow` die neue QR-Zerlegung mit quadratischem Aufwand berechnet.

d) Begründen Sie, warum Ihre MATLAB-Funktion `myQrInsertRow` die neue QR-Zerlegung mit quadratischem Aufwand berechnet.

e) Nun erhalten Sie eine weitere Messung $(x_5, y_5) = (-2.4, 1.0)$. Bestimmen Sie mit Ihrer MATLAB-Funktion `myQrInsertRow` die neue QR-Zerlegung und berechnen Sie die Lösung des erweiterten linearen Ausgleichsproblems mit 5 Messwerten durch Rückwärtseinsetzen.

   Zeichnen Sie auch die neue Ausgleichsellipse und den neuen Messpunkt in Ihre Skizze aus Aufgabenteil b) ein.

f) Testen Sie Ihre MATLAB-Funktion `myQrInsertRow`, indem Sie die aus der neuen Messung (x_5, y_5) resultierende Matrix-Zeile auch als eine andere Zeile als in Aufgabenteil e) in die Ausgangsmatrix A einfügen. Erhalten Sie die gleiche Lösung des erweiterten linearen Ausgleichsproblems wie in Aufgabenteil e)?

Aufgabe 31 *(Intervallschachtelung)* (5T* Punkte)

Das Bisektionsverfahren beruht auf dem Prinzip der Intervallschachtelung:
Sei $I_0 = [a_0, b_0]$ ein Anfangsintervall. Eine Annäherung an einen Punkt $x^* \in I_0$ kann man mit Hilfe der Intervallschachtelung erreichen. Dazu definiert man iterativ $I_{n+1} = [a_{n+1}, b_{n+1}]$ $(n = 0, 1, 2, \ldots)$ mit

$$a_{n+1} = \begin{cases} a_n\,, & \text{falls } x^* \leq \frac{a_n+b_n}{2} \\ \frac{a_n+b_n}{2}\,, & \text{falls } x^* > \frac{a_n+b_n}{2} \end{cases} \qquad \text{sowie} \qquad b_{n+1} = \begin{cases} \frac{a_n+b_n}{2}\,, & \text{falls } x^* \leq \frac{a_n+b_n}{2} \\ b_n\,, & \text{falls } x^* > \frac{a_n+b_n}{2}\,. \end{cases}$$

Wie viele Schritte benötigt das Bisektionsverfahren, um die Nullstelle $x^* \in I_0 = [0, 1]$ einer Funktion $f : \mathbb{R} \to \mathbb{R}$ mit einer Genauigkeit von mindestens 10^{-8} näherungsweise zu berechnen? Bestimmen Sie $n_0 \in \mathbb{N}$ so, dass $|x^* - m| < 10^{-8}$, wobei m der Mittelpunkt des Intervalls I_{n_0} ist.

Anmerkung:
Beim Prinzip der Intervallschachtelung kann der Punkt x^* bekannt oder unbekannt sein. Man muss lediglich entscheiden können, in welcher der beiden Intervallhälften er liegt. Beim Bisektionsverfahren ist x^* die gesuchte Nullstelle der Funktion f, also unbekannt. Die Entscheidung zwischen den beiden Intervallhälften (also zwischen den Fällen $x^* \leq \frac{a_n+b_n}{2}$ und $x^* > \frac{a_n+b_n}{2}$) wird durch die Überprüfung der Bedingung $f(a) \cdot f\left(\frac{a_n+b_n}{2}\right) < 0$ getroffen.

Aufgabe 32 *(Programmieraufgabe: Vergleich der Verfahren für nichtlineare skalare Gleichungen)* (9M+4M+3T+2M*+3M+3T+3T* Punkte)

In dieser Aufgabe wollen wir die drei bisher in der Vorlesung vorgestellten Verfahren zur numerischen Lösung nichtlinearer skalarer Gleichungen vergleichen.

a) Schreiben Sie drei MATLAB-Funktionen

   i) `xk = bisektion(f, a, b, tolx)`,

   ii) `xk = regulaFalsi(f, a, b, toly, maxIt)` und

   iii) `xk = sekanten(f, x0, x1, toly, maxIt)`,

   die die Lösung x^* einer Gleichung $f(x^*) = 0$ mit der Bisektionsmethode, der Regula-Falsi und der Sekantenmethode näherungsweise bestimmen. Dabei sind jeweils

   - `f` die Funktion, deren Nullstelle bestimmt werden soll, als *function handle*,
   - `a` und `b` die linke und rechte Intervallgrenze für die Bisektionsmethode und die Regula-Falsi,
   - `tolx` die Genauigkeit, mit der das Bisektionsverfahren die Nullstelle bestimmen soll,
   - `toly` die Toleranz für den Funktionswert der Funktion f an der näherungsweise berechneten Nullstelle,
   - `maxIt` eine obere Schranke für die Anzahl der durchzuführenden Iterationen sowie
   - `x0` und `x1` die Startwerte für die Sekantenmethode.

   Ihre Funktionen sollen jeweils einen Vektor `xk` mit den Iterationswerten x_k aller durchgeführten Iterationen zurückgeben. Die Intervallgrenzen a und b sowie die Startwerte x_0 und x_1 sollen jeweils im Vektor `xk` enthalten sein.

b) Schreiben Sie ein MATLAB-Skript `testKonvergenz`, in dem Sie mit Hilfe Ihrer drei MATLAB-Funktionen die Nullstelle der Funktion

   $$f : \mathbb{R} \to \mathbb{R},\ x \mapsto f(x) \quad \text{mit} \quad f(x) = x^2 - \cos x.$$

   näherungsweise berechnen. Wählen Sie als Startintervall für die Bisektionsmethode und die Regula-Falsi das Intervall $[0, 3]$. Für das Sekantenverfahren verwenden Sie die Startwerte 0 und 3.

   Visualisieren Sie für alle drei Verfahren in einem gemeinsamen Schaubild, wie sich jeweils der Fehler $|x_k - x^*|$ während des jeweiligen Verfahrens entwickelt. Plotten Sie dazu zu jedem Iterationsschritt (x-Achse) den jeweiligen Fehler (y-Achse). Wählen Sie zur besseren Darstellung für den Fehler eine logarithmische Skala (MATLAB-Befehl `semilogy`).

   Sie können für die exakte Lösung x^* das Ergebnis der MATLAB-Anweisung `fzero(f, 1)` verwenden, wobei `f` ein function handle für die Funktion f ist.

c) Interpretieren Sie das Ergebnis. Welche Konvergenz im Sinne der Definition 5.0.1 (Konvergenzgeschwindigkeit) auf Folie 8 (Definition 5.0.5 auf Seite 65 im Skript) liegt jeweils vor?

   Begründen Sie Ihre Aussage.

d) Zeichnen Sie in Ihr Schaubild auch die jeweiligen Intervalllängen des Bisektionsverfahrens ein. Welche Aussagen können Sie treffen?

e) Untersuchen Sie nun auch das Konvergenzverhalten Ihrer drei Verfahren für die Funktion

   $$f : \mathbb{R} \to \mathbb{R},\ x \mapsto f(x) \quad \text{mit} \quad f(x) = (x - 1)^2.$$

   Plotten Sie dazu in ein neues Schaubild wieder zu jedem Iterationsschritt (x-Achse) den jeweiligen Fehler (y-Achse). Verwenden Sie für den Fehler eine logarithmische Skala. Wählen Sie als Startintervall für das Bisektionsverfahren und die Regula-Falsi das Intervall $[-0.1, 3]$. Verwenden Sie für das Sekantenverfahren die Startwerte -0.1 und 3. Den Parameter `tol` für die geforderte Genauigkeit der Näherungslösung können Sie auf 10^{-14} setzen.

f) Interpretieren Sie das Ergebnis. Liegt Konvergenz vor? Falls ja, welche Konvergenzordnung haben die jeweiligen Verfahren? Begründen Sie Ihre Aussage.

g) Wie passen die beobachteten Ergebnisse zu den Aussagen aus der Vorlesung, insbesondere zu Lemma 5.1.1 (Konvergenz des Bisektionsverfahrens, Folie 11 bzw. Seite 68 im Skript), zu Satz 5.1.3 (Konvergenz Regula Falsi, Folie 14 bzw. Seite 69 im Skript) und zu Satz 5.1.4 (Konvergenz Sekantenmethode, Folie 16 bzw. Seite 71 im Skript)?

Hinweis: Sie können *anonyme Funktionen* verwenden:
Die Anweisung <Funktionsname> = @(<Argumentliste>) Funktionsbeschreibung legt eine Variable vom Typ function handle an, die dann als Parameter an eine andere Funktion, beispielsweise Ihre Funktion bisektion, übergeben werden kann. Für die erste Funktion mit $f(x) = x^2 - \cos x$ legt f1 = @(x) x.^2 - cos(x) die Variable f1 an, die Sie beispielsweise mit xk = bisektion(f1, a, b, tolx)) an Ihre MATLAB-Funktionen übergeben können.

Hinweise:

Teamarbeit ist bei der Bearbeitung der Übungsblätter erlaubt und erwünscht. Falls Sie die Aufgaben im Team lösen, muss jedes Teammitglied eine eigene Lösung votieren und abgeben. Geben Sie dann bitte auf Ihrer Lösung alle an der Aufgabe beteiligten Teammitglieder an.

Die Programmieraufgaben sind in MATLAB zu lösen. Der Source Code muss strukturiert und dokumentiert sein.

Speichern Sie Ihre Lösungen der Theorieaufgaben, Ihre MATLAB-Programme und Ihre Ergebnisse in einem Directory mit dem Namen **Blatt08_*Vorname_Nachname*** und verpacken Sie dieses in eine .zip-Datei. **Spätestens bis Sonntag, 17.12.2023, 23:59 Uhr** müssen Sie diese .zip-Datei in Moodle hochladen und für die Aufgabenteile, die Sie vorstellen können, votieren.

$$\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} \begin{pmatrix} Q_1^T & Q_2^T \end{pmatrix} = \left(\begin{matrix} Q_1 Q_1^T & Q_1 Q_2^T \\ & \end{matrix} \right.$$

Aufgabe 29

Zeigen Sie, dass die Matrix $\hat{Q} = \begin{pmatrix} 0 & Q_1 \\ 1 & 0 \\ 0 & Q_2 \end{pmatrix}$ orthogonal ist.

$$\hat{Q}\hat{Q}^T = \begin{pmatrix} 0 & Q_1 \\ 1 & 0 \\ 0 & Q_2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ Q_1^T & 0 & Q_2^T \end{pmatrix} = \begin{pmatrix} Q_1 Q_1^T & 0 & Q_1 Q_2^T \\ 0 & 1 & 0 \\ Q_2 Q_1^T & 0 & Q_2 Q_2^T \end{pmatrix} = \begin{pmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I \end{pmatrix} = I$$

$$\hat{Q}^T \hat{Q} = \begin{pmatrix} 0 & 1 & 0 \\ Q_1^T & 0 & Q_2^T \end{pmatrix} \begin{pmatrix} 0 & Q_1 \\ 1 & 0 \\ 0 & Q_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & Q_1 Q_1^T + Q_2^T Q_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & I \end{pmatrix} = I$$

$$\begin{pmatrix} Q_1^T & Q_2^T \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} = \begin{pmatrix} Q_1^T Q_1 + Q_2^T Q_2 \end{pmatrix} = I$$

$\Rightarrow$ $\hat{Q}$ ist orthogonal

Aufgabe 30

a)

x_i	$-1,6$	$-0,9$	$0,75$	$2,7$
y_i	$1,6$	$0,9$	1	-1

$$\alpha x^2 + \beta x y + \gamma y^2 - 10 = 0$$

$$A z - b = 0$$

$$\Rightarrow \begin{pmatrix} x_1^2 & x_1 y_1 & y_1^2 \\ x_2^2 & x_2 y_2 & y_2^2 \\ x_3^2 & x_3 y_3 & y_3^2 \\ x_4^2 & x_4 y_4 & y_4^2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} - \begin{pmatrix} 10 \\ 10 \\ 10 \\ 10 \end{pmatrix} = 0$$

$$A \in \mathbb{R}^{4 \times 3} \qquad z \in \mathbb{R}^3 \qquad b \in \mathbb{R}^4$$

Aufgabe 32

c)
Man kann schnell erkennen, dass das Sekantenverfahren mit Abstand am besten ist.
Dieses besitzt eine Konvergenz der Ordnung p. Das Regula-Falsi Verfahren besitzt eine lineare Konvergenz.
Da die Fehlerwerte des Bisektionsverfahren immer wieder springen, kann hier keine genaue Konvergenzdefinition festgestellt werden.

f)
Das Bisektionsverfahren und das RegulaFalsi Verfahren konvergieren nicht, da die Annahme der Verfahren, dass die Funktion f im Intervall [a,b] folgendes erfüllt: f(a) * f(b) < 0, nicht gilt. Dadurch nähern sie sich nicht der Lösung an.

Das Sekantenverfahren konvergiert hier linear.

g)
Die Fälle zu dem Bisektionsverfahren und Regula-Falsi Verfahren stimmen wie im Skript beschrieben.
Nur die Sekantenmethode scheint in diesem Sonderfall nicht wie erwartet zu funktionieren.