Prof. Dr. Dirk Lebiedz
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

Angewandte Numerik 1
SoSe 2020
26.06.2020

Übungsblatt 10

Besprechung in den Tutorien in der Woche vom 06.07.2020 bis 10.07.2020

Für dieses Übungsblatt gibt es 11 Theorie- und 17 Matlab-Punkte, sowie 19 Theorie- und 33 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 70-Prozent-Grenzen liegen aktuell (inklusive Blatt 10) bei 148,4 Theorie- und 98,7 Matlabpunkten.

Aufgabe 41 *(Programmieraufgabe: Konvergenz des Bisektionsverfahrens)* (4M+3M+2T Punkte)

In dieser Aufgabe wollen wir zunächst die Konvergenz des Bisektionsverfahrens untersuchen.

a) Schreiben Sie eine MATLAB-Funktion `xk = bisektion1(f, a, b, tolx)`, die die Lösung einer Gleichung $f(x) = 0$ im Intervall $[a, b]$ mit dem Bisektionsverfahren näherungsweise bestimmt. Dabei soll die Nullstelle mit der Genauigkeit `tolx` bestimmt werden. Ihre MATLAB-Funktion `bisektion1` soll einen Vektor `xk` mit den Iterationswerten x_k aller durchgeführten Iterationen zurückgeben.

b) Schreiben Sie ein MATLAB-Skript `testKonvergenz`, in dem Sie mit Hilfe Ihrer Funktion `bisektion1` die Nullstelle der Funktion $f : \mathbb{R} \to \mathbb{R}$ mit $f(x) = x^2 - \cos x$ auf 10 Nachkommastellen genau berechnen. Wählen Sie als Startintervall das Intervall $[0, 3]$.

   Visualisieren Sie, wie sich der Fehler $|x_k - x^*|$ während des Bisektionsverfahrens 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. Liegt lineare Konvergenz im Sinne der *Definition 5.14* vor? Begründen Sie Ihre Aussage.

Aufgabe 42 *(Programmieraufgabe: Newton-Verfahren)* (3T*+3M+2M* Punkte)

a) Lösen Sie das nichtlineare Gleichungssystem $x^2 = 5$ näherungsweise, indem Sie von Hand drei Iterationen des Newton-Verfahrens durchführen. Verwenden Sie als Startwert $x_0 = 1$.

b) Schreiben Sie eine MATLAB-Funktion `xk = newton1D(f, df, x0, toly, maxIt)`, die eine Nullstelle einer Funktion `f` mit Hilfe des Newton-Verfahrens näherungsweise berechnet. `df` ist die Ableitung der Funktion f als *function handle* und `x0` der Startwert. `toly` ist die Toleranz für den Funktionswert der Funktion f an der näherungsweise berechneten Nullstelle und `maxIt` ist eine obere Schranke für die Anzahl der durchgeführten Iterationen. `xk` soll wieder alle Iterationswerte enthalten.

c) Testen Sie Ihre MATLAB-Funktion `newton1D` an der Funktion aus Aufgabenteil a).

Aufgabe 43 *(Programmieraufgabe: Konvergenzordnungen)*

(2M+2T+2T+4M*+2T*+5M+2T+3T Punkte)

In Aufgabe 41 haben wir die Konvergenzordnung des Bisektionsverfahren untersucht. Diese Untersuchung wollen wir jetzt auf das Newton- und auf das Sekantenverfahren erweitern.

a) Erweitern Sie Ihr MATLAB-Skript `testKonvergenz` aus Aufgabe 41 um das Newton-Verfahren. Wählen Sie als Startwert $x_0 = 3$. Ergänzen Sie Ihr Schaubild um die Darstellung der Fehler des Newton-Verfahrens.

b) Interpretieren Sie das Ergebnis. Liegt beim Newton-Verfahren Konvergenz im Sinne der Definition 5.14 vor?

c) Variieren Sie den Startwert für das Newton-Verfahren. Testen Sie beispielsweise auch den Startwert $x_0 = 0$. Wie erklären Sie sich das Ergebnis?

d) Erweitern Sie Ihr Testprogramm und Ihre Grafik auch um das Sekanten-Verfahren. Schreiben Sie dazu eine MATLAB-Funktion `xk = sekanten(f, a, b, toly, maxIt)`, die das Sekanten-Verfahren implementiert. Ihre Funktion soll die gleichen Rückgabewerte und Parameter wie Ihre Funktionen `bisektion1` bzw. `newton1D` haben.

e) Interpretieren Sie die Ergebnisse. Liegt beim Sekanten-Verfahren Konvergenz im Sinne der *Definition 5.14* vor? Wie würde sich die Zahl der benötigten Iterationen der einzelnen Verfahren ändern, wenn Sie die gewünschte Genauigkeit `tol` variieren würden.

f) Untersuchen Sie nun auch das Konvergenzverhalten Ihrer drei Verfahren (also Bisektions-, Sekanten- und Newtonverfahren) 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 das Intervall $[-0.1, 3]$. Verwenden Sie für das Sekantenverfahren die Startwerte -0.1 und 3, sowie den Startwert 3 für das Newtonverfahren. Den Parameter `tol` für die geforderte Genauigkeit der Näherungslösung können Sie auf 10^{-14} setzen.

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

h) Wie passen die beobachteten Ergebnisse zu den Aussagen aus der Vorlesung, insbesondere zum Algorithmus 5.20, zu Satz 5.22 und zur Aussage über die Konvergenzordnung des Sekantenverfahrens auf Folie 175?

Aufgabe 44 *(Programmieraufgabe: Konvergenzbereich Newtonverfahren)*

(4M*+2T*+3M*+4M*+1M*+2T* Punkte)

In dieser Aufgabe wollen wir für das Newton- und das Sekanten-Verfahren die Abhängigkeit der Konvergenz von den Startwerten untersuchen.

a) Schreiben Sie ein Matlab-Skript `konvergenzBereiche.m`, welches für das Newton-Verfahren und für die Funktion f mit $f(x) = \arctan x$ die Abhängigkeiten der Anzahl der durchgeführten Iterationen vom Startwert veranschaulicht. Wählen Sie dazu 200 Startwerte $x_0 \in [-10, 10]$ und bestimmen Sie für jeden Startwert die Anzahl der durchgeführten Iterationen. Visualisieren Sie die Ergebnisse mittels einer Grafik (x-Achse: Startwert, y-Achse: Anzahl der Iterationen). Verwenden Sie `tol = 1e-10.`

b) Was fällt auf? Wie interpretieren Sie die Anzahl der Iterationen für betragsmäßig große Startwerte. Erklären Sie beispielhaft an einigen Startwerten den jeweiligen Verlauf der Iteration.

c) Überlegen Sie sich, wie Sie für jeden Startwert überprüfen können, ob das Newtonverfahren mit der vorgegebenen Toleranz und innerhalb der maximalen Iterationen konvergiert sowie einen Näherungswert für die Nullstelle liefert. Visualisieren Sie den Konvergenzbereich in Ihrem Schaubild.

d) Untersuchen Sie analog zu Aufgabenteil a) auch das Sekanten-Verfahren. Wählen Sie als zweiten Startwert $x_1 = -5$, falls $x_0 \geq 0$, und $x_1 = +5$, falls $x_0 < 0$. Visualisieren Sie den Konvergenzbereich des Sekanten-Verfahrens analog zu Aufgabenteil c).

e) Vertauschen Sie die Startwerte des Sekanten-Verfahrens. Wählen Sie also $x_0 = \pm 5$ und $x_1 \in [-10, 10]$.

f) Interpretieren Sie Ihre Ergebnisse. Für welche Startwerte konvergiert das Sekantenverfahren? Unterscheiden sich die Ergebnisse des Aufgabenteils d) und des Aufgabenteils e)? Falls ja, warum?

Aufgabe 45 *(Newton-Verfahren für Systeme)* (6T* Punkte)

Mit dem Newton-Verfahren können auch Systeme nichtlinearer Gleichungen gelöst werden.
Betrachen Sie die Funktion $f : \mathbb{R}^2 \to \mathbb{R}^2$, definiert durch

$$f\left(x_1, x_2\right) := \begin{pmatrix} 2\,x_1^2\,x_2 \;+\; 1 \\ 2\,x_1(1 + 2\,x_2^2) \;-\; 2 \end{pmatrix}.$$

Lösen Sie das System von nichtlinearen Gleichungen $f(x, y) = 0$ näherungsweise durch Anwendung des Newton-Verfahrens für Systeme auf diese Funktion.
Beginnen Sie mit dem Startwert $(x_1^{(0)}, x_2^{(0)})^T = (1, 0)^T$ und führen Sie auf dem Papier zwei Newton-Iterationen durch. Bestimmen Sie in jedem Schritt k ($k \in \{0, 1\}$) die Newton-Korrektur $s^{(k)}$. Verzichten Sie dabei auf die Berechnung der Inversen $(f'(x^{(k)}))^{-1}$ der Jacobi-Matrix $f'(x^{(k)})$, sondern berechnen Sie die Newton-Korrektur $s^{(k)}$ durch Lösen des Gleichungssystems $f'(x^{(k)})\, s^{(k)} = -f(x^{(k)})$.

Aufgabe 46 *(Programmieraufgabe: Newton-Verfahren für Systeme)*
(4M*+2M*+(8M*+2T*)+3M* Punkte)

a) Erweitern Sie Ihre MATLAB-Funktion `xk = newton1D(f, df, x0, toly, maxIt)` aus Aufgabe 42 zu einer Funktion `xk = newtonSys(f, df, x0, toly, maxIt)`, die eine Nullstelle eines Systems nichtlinearer Gleichungen mit Hilfe des Newton-Verfahrens näherungsweise berechnet.

   Der Parameter `f` soll wieder die (jetzt mehrdimensionale) Funktion f als *function handle*, `df` die Jacobimatrix der Funktion f als *function handle* und der Parameter `x0` der Startwert als Spaltenvektor sein. Die weiteren Parameter `toly` und `maxIt` sollen die gleiche Bedeutung wie in Aufgabe 42 haben.

   Der Rückgabewert `xk` soll jetzt eine Matrix mit den Iterationswerten aller Iterationen sein. Jede Spalte soll dabei einen Iterationswert enthalten.

b) Verifizieren Sie Ihre Ergebnisse aus Aufgabe 45 mit Hilfe Ihrer Funktion `newtonSys`.

c) In dieser Teilaufgabe wollen wir mit Hilfe des Newton-Verfahrens die Einzugsbereiche von Nullstellen komplexwertiger Funktionen in der komplexen Zahlenebene darstellen. Betrachten wir dazu die Gleichung

$$z^3 = 1, \tag{1}$$

   die sich als Nullstellenproblem formulieren lässt. Die Lösungen dieser Gleichung sind gegeben durch $z_1 = 1$, $z_2 = -\frac{1}{2} + \frac{\sqrt{3}}{2}\, i$ und $z_3 = -\frac{1}{2} - \frac{\sqrt{3}}{2}\, i$.

   Die komplexe Zahlenebene $\mathbb{C}$ kann mit $\mathbb{R}^2$ identifiziert werden. Eine Zahl $z = x + iy \in \mathbb{C}$ entspricht dabei dem Punkt $(x, y)^T \in \mathbb{R}^2$ und umgekehrt. Dadurch kann Gleichung (1) in ein System nichtlinearer Gleichungen $F(x, y) = 0$ mit einer geeigneten Funktion $F : \mathbb{R}^2 \to \mathbb{R}^2$ umgeschrieben werden.

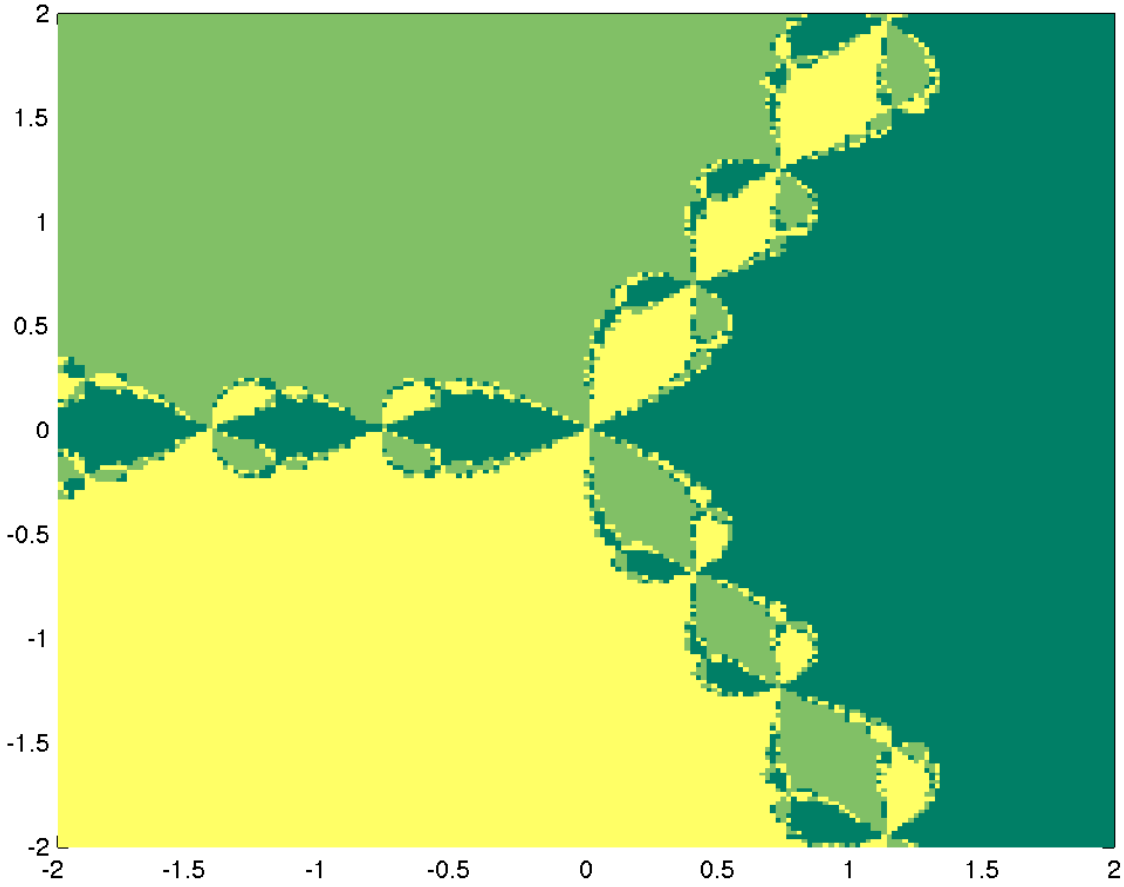
   Hinweis: Überlegen Sie sich die Darstellung der Zahl $z^3 = (x + iy)^3 \in \mathbb{C}$ in $\mathbb{R}^2$.

Nun können wir die Gleichung $F(x, y) = 0$ unter Verwendung eines Startwerts z_0 mit Hilfe des Newton-Verfahrens näherungsweise lösen. Das Newton-Verfahren wird dann entweder divergieren oder gegen eine der drei Lösungen z_1, z_2 oder z_3 konvergieren. Im Folgenden wollen wir das Verhalten des Newton-Verfahrens für verschiedene Startwerte genauer untersuchen:

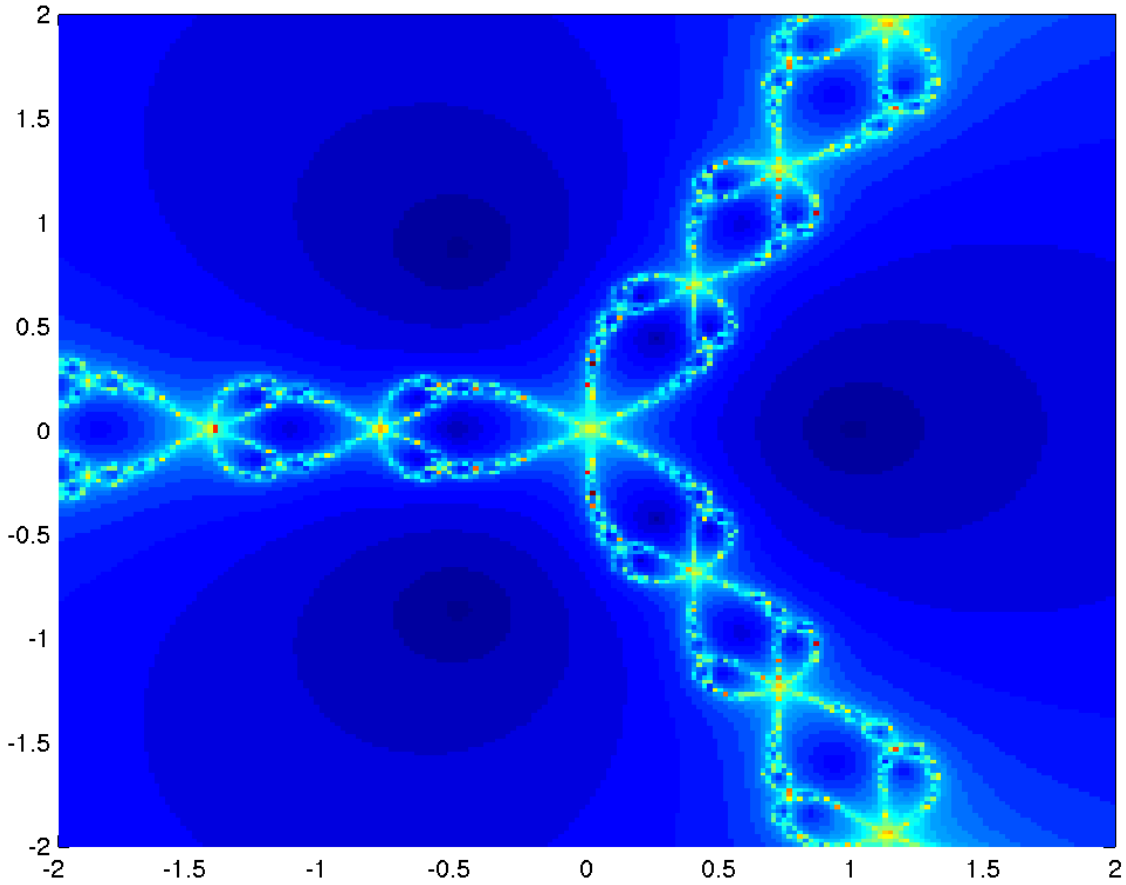
Schreiben Sie ein Matlab-Skript `juliaMenge`. Verwenden Sie äquidistant verteilte Punkte im Intervall $[-2, 2] \times [-2, 2] \subset \mathbb{R}^2$ als Startwerte für das Newton-Verfahren. Die folgenden Bilder wurden mit 200×200 Punkten und den Parametern `tol` $= 10^{-10}$ und `maxIt` $= 1000$ erzeugt. Färben Sie die Startwerte unterschiedlich ein, und zwar in Abhängigkeit davon, ob das Newton-Verfahren mit dem jeweiligen Startwert gegen eine der drei Lösungen konvergiert (also drei verschiedene Farbwerte und zwar für jede der Lösungen einen Farbwert) oder nicht konvergiert (ein vierter Farbwert).

Die drei (nicht verbundenen) Mengen von Startwerten, für die das Newton-Verfahren gegen die drei verschiedenen Lösungen konvergiert, werden als *Fatou-Menge*, ihre Ränder als *Julia-Menge* bezeichnet.

Hinweis: Die folgenden Matlab-Befehle könnten hilfreich sein: `meshgrid`, `pcolor` oder `surf` in Verbindung mit `view(2)`, `shading` und `colormap`.



d) Färben Sie in einem weiteren Schaubild die Startwerte entsprechend der vom Newton-Verfahren durchgeführten Anzahl an Iterationen ein.



Hinweise:
Die Lösungen der Theorieaufgaben und Ihren Text zu den Programmieraufgaben können Sie in LaTeX erstellen oder handschriftlich aufschreiben und einscannen. Die Programmieraufgaben sind in Matlab zu lösen. Der Source Code muss strukturiert und dokumentiert sein.
Falls Sie die Aufgaben im Team lösen, geben Sie bitte auf allen Lösungen alle an der Aufgabe beteiligten Teammitglieder an. Jedes Teammitglied muss eine Lösung abgeben.
Speichern Sie Ihre Lösungen und Ihre Ergebnisse in einem Directory mit dem Namen **Blatt10_*Vorname*_*Nachname*** und verpacken Sie dieses in eine `.zip`-Datei. Laden Sie **spätestens 48 Stunden vor Ihrem Tutorium** diese `.zip`-Datei in Moodle hoch.