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

WiSe 2023/2024
08.01.2024

Angewandte Numerik

Übungsblatt 10

Besprechung in den Tutorien von Montag, 15.01.2024 bis Freitag, 19.01.2024

Für dieses Übungsblatt gibt es 2 Theorie- und 23 Matlab-Punkte, sowie 10 Theorie- und 5 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 70-Prozent-Grenzen liegen aktuell (inklusive Blatt 10) bei 119,0 Theorie- und 128,8 Matlab-Punkten.

Aufgabe 36 *(Newton-Verfahren für Systeme)* (8T* 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

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

Lösen Sie das System von nichtlinearen Gleichungen $f(x_1, x_2) = 0$ näherungsweise durch Anwendung des Newton-Verfahrens für Systeme auf diese Funktion.

Beginnen Sie mit dem Startwert $x^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$ 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 $(J_f(x^{(k)}))^{-1}$ der Jacobi-Matrix $J_f(x^{(k)})$, sondern berechnen Sie die Newton-Korrektur $s^{(k)}$ durch Lösen des Gleichungssystems $J_f(x^{(k)})\, s^{(k)} = -f(x^{(k)})$.

Rechnen Sie mit Brüchen.

Aufgabe 37 *(Programmieraufg.: Newton-Verfahren für Systeme oder „Mathe kann auch schön sein")*
(5M+2M*+(9M+2T)+4M Punkte)

a) Erweitern Sie Ihre MATLAB-Funktion `xk = newtonSkalar(f, df, x0, toly, maxIt)` aus Aufgabe 34 von Blatt 09 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 34 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 36 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

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

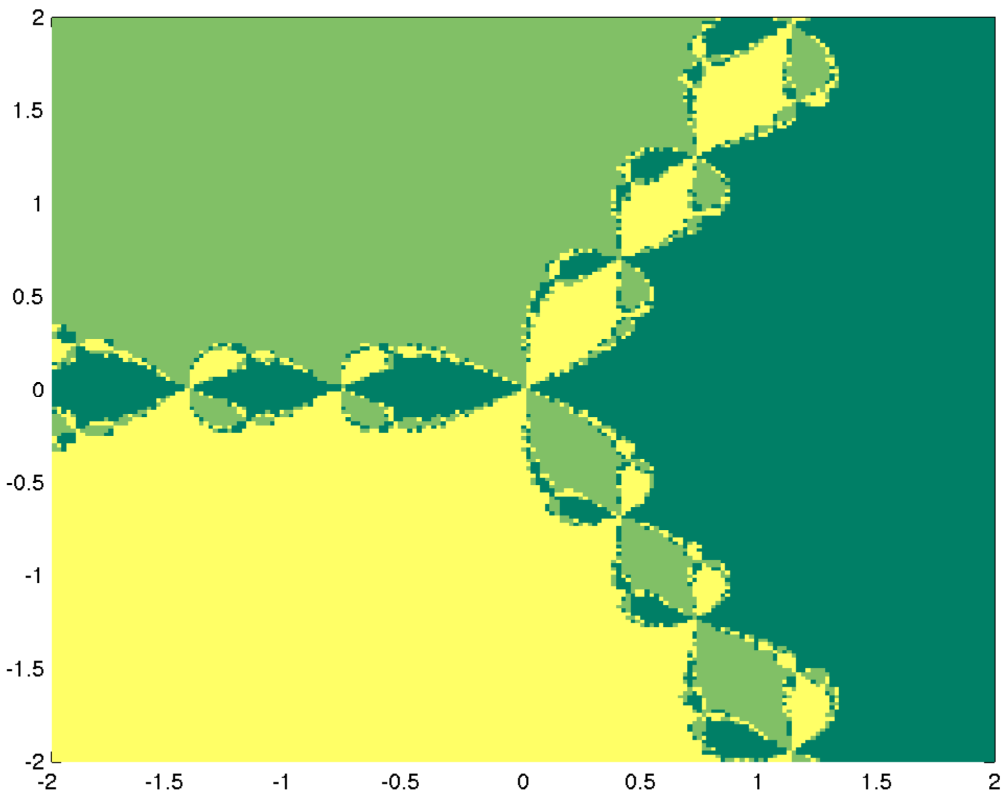
die sich als Nullstellenproblem formulieren lässt. Die Lösungen dieser Gleichung sind gegeben durch $z_1 = 1$, $z_2 = -\frac{1}{2} + \frac{\sqrt{3}}{2}\,i$ und $z_3 = -\frac{1}{2} - \frac{\sqrt{3}}{2}\,i$.

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:

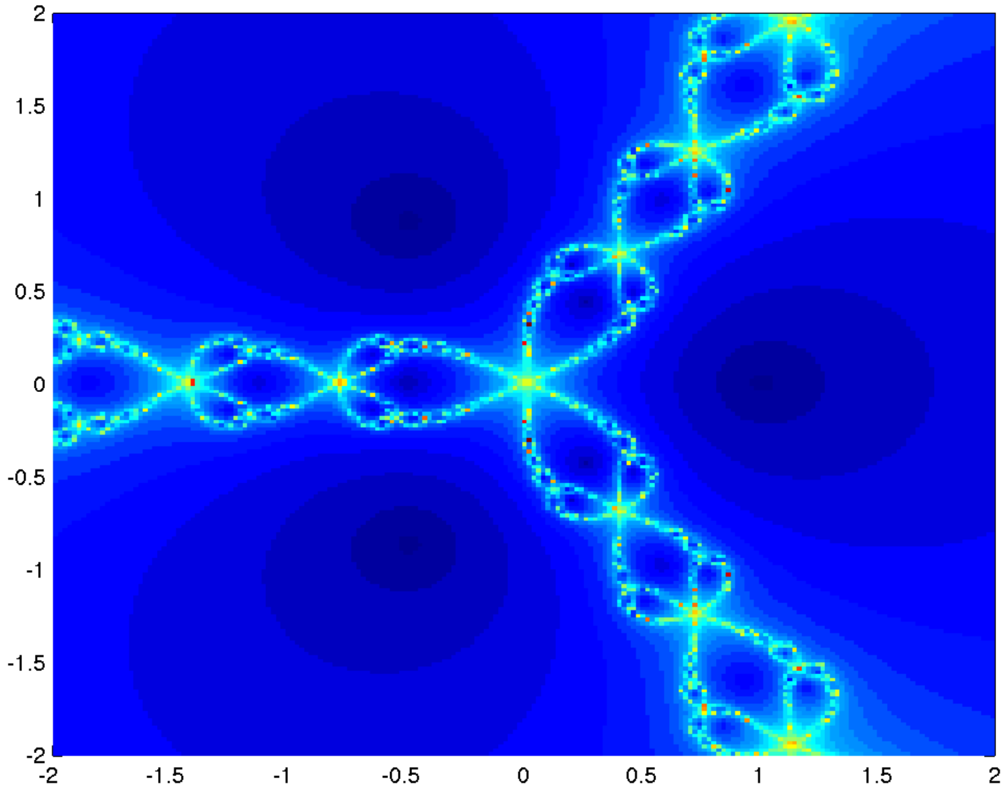
Schreiben Sie ein MATLAB-Skript `juliaMenge`. Verwenden Sie äquidistant verteilte Punkte im Intervall $[-2, 2] \times [-2, 2] \subset \mathbb{R}^2$ als Startwerte für das Newton-Verfahren. Die folgenden Bilder wurden mit 200×200 Punkten und den Parametern `toly` $= 10^{-10}$ und `maxIt` $= 1000$ erzeugt. 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.



Aufgabe 38 *(Programmieraufgabe: Konvergenzbereich des gedämpften Newton-Verfahrens)* (5M+3M*+2T* Punkte)

In Aufgabe 35 vom letzten Übungsblatt 09 haben wir für das Newton-, das Sekanten- und das Dekkerverfahren jeweils die Abhängigkeit der Konvergenz von den Startwerten untersucht. Dabei haben wir festgestellt, dass sowohl das Newton- als auch das Sekantenverfahren nur lokal konvergieren und dass das Dekkerverfahren auf dem gesamten Intervall konvergiert.

In dieser Aufgabe wollen wir analog die Konvergenz des gedämpften Newtonverfahrens untersuchen.

a) Schreiben Sie eine Funktion `xk = newtonGedaempft(f, df, x0, tol, maxIt, lambdaMin)`, die eine Nullstelle einer gegebenen Funktion $f : \mathbb{R}^n \to \mathbb{R}^n$ mit Hilfe des gedämpften Newtonverfahrens berechnet. Die Parameter und die Rückgabewerte sind gleich wie bei Ihrer Funktion `xk = newtonSys(f, df, x0, tol, maxIt)` aus Aufgabe 37 für das Newton-Verfahren für Systeme. Der zusätzliche Parameter `lambdaMin` gibt den kleinstmöglichen Dämpfungsparameter λ an.

b) Ergänzen Sie Ihre MATLAB-Lösung von Aufgabe 35 um das gedämpfte Newtonverfahren. Betrachten Sie wieder die Funktion f mit $f(x) = \arctan x$ und wählen Sie wieder 200 Startwerte $x_0 \in [-10, 10]$.

c) Interpretieren Sie Ihre Ergebnisse. Für welche Startwerte konvergiert das gedämpfte Newtonverfahren?

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 **Blatt10_*Vorname_Nachname*** und verpacken Sie dieses in eine `.zip`-Datei. **Spätestens bis Sonntag, 14.01.2024, 23:59 Uhr** müssen Sie diese `.zip`-Datei in Moodle hochladen und für die Aufgabenteile, die Sie vorstellen können, votieren.

Aufgabe 37

c) $z^3 - 1 = (x + iy)^3 - 1$

$= (x^2 + 2xiy - y^2)(x + iy) - 1$

$= x^3 + 2x^2iy - xy^2 + x^2iy - 2xy^2 - iy^3 - 1$

$= x^3 - xy^2 - 2xy^2 - 1 + 2x^2iy + x^2iy - iy^3$

$= x^3 - 3xy^2 - 1 + 3x^2iy - iy^3$

$\Rightarrow$ real: $x^3 - 3xy^2 - 1$

imaginär: $-y^3 + 3x^2y$

$$\Rightarrow \quad f(x,y) = \begin{pmatrix} x^3 - 3xy^2 - 1 \\ -y^3 + 3x^2y \end{pmatrix}$$