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

SoSe 2017
Blatt 10
26.06.2017

Angewandte Numerik 1

Besprechung in den Tutorien in der Woche vom 03.07.2017 bis 07.07.2017

Für dieses Übungsblatt gibt es 4 Theorie- und 20 Matlab-Punkte, sowie 6 Theorie- und 3 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 60-Prozent-Grenzen liegen aktuell (inklusive Blatt 10) bei 106,2 Theoriepunkten und 96,6 Matlabpunkten.

Aufgabe 39 *(Newton-Verfahren für Systeme)* (6T* Punkte)

Anders als beim Bisektionsverfahren, der Regula Falsi und dem Sekantenverfahren können mit dem Newton-Verfahren 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 40 *(Programmieraufgabe: Newton-Verfahren für Systeme)* (4M+2M+(6M+2T)+3M* Punkte)

a) Erweitern Sie Ihre Matlab-Funktion `xk = newton(f, df, x0, tol, maxIt)` aus Aufgabe 35 von Übungsblatt 08 zu einer Funktion `xk = newtonSys(f, df, x0, tol, 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*, der Parameter `df` die Jacobimatrix der Funktion f als *function handle* und der Parameter `x0` der Startwert als Spaltenvektor sein. Die weiteren Parameter `tol` und `maxIt` sollen die gleiche Bedeutung wie in Aufgabe 35 haben.

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

b) Verifizieren Sie Ihre Ergebnisse aus Aufgabe 39 mit Hilfe Ihrer Funktion `xk = newtonSys(f, df, x0, tol, maxIt)`.

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.

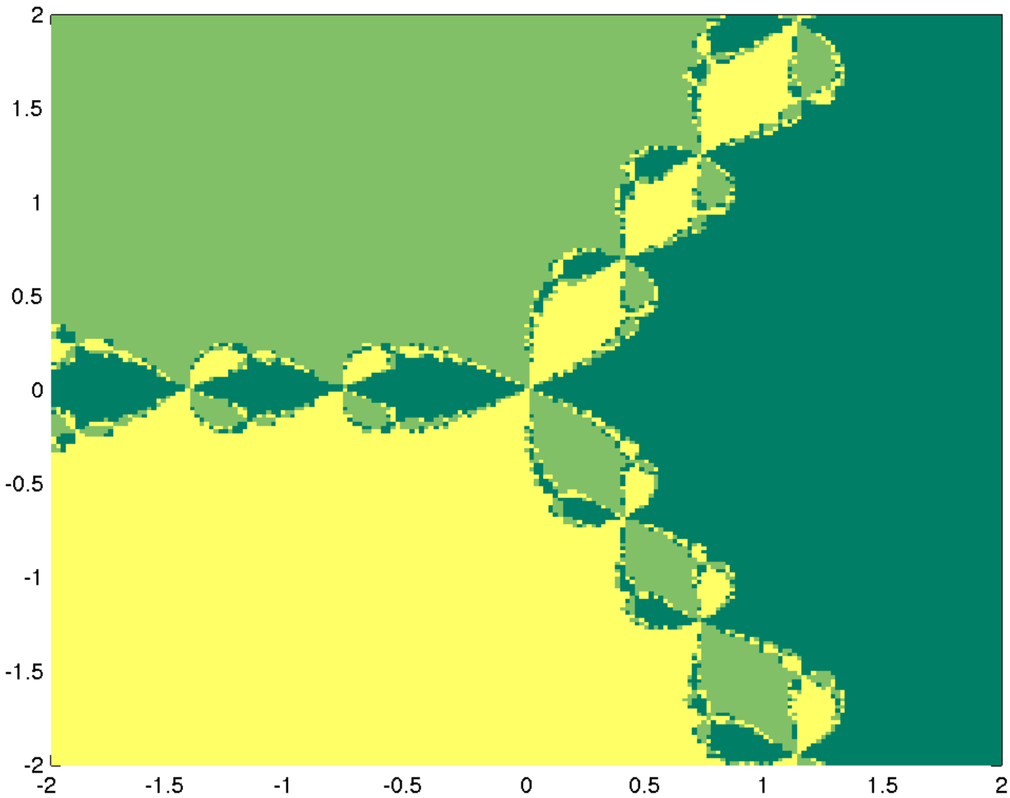
Hinweis: Überlegen Sie sich die Darstellung der Zahl $z^3 = (x + iy)^3 \in \mathbb{C}$ in $\mathbb{R}^2$.

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:

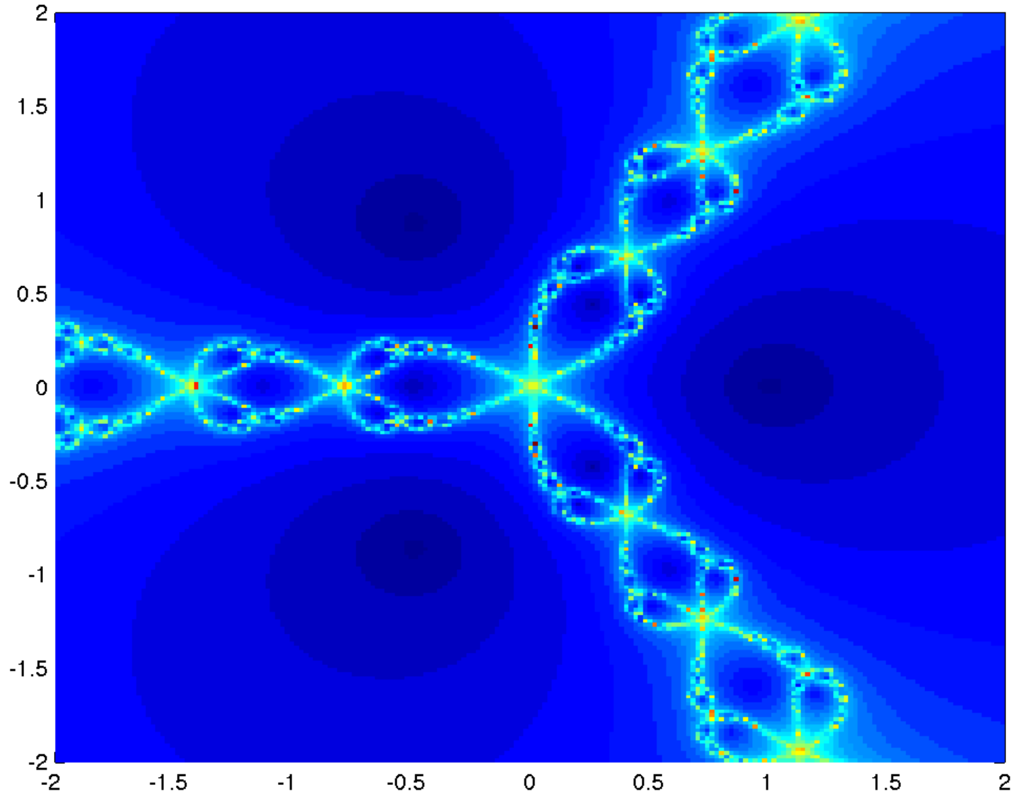
Schreiben Sie ein Matlab-Skript `juliaMenge`. Verwenden Sie äquidistant verteilte Punkte im Intervall $[-2, 2] \times [-2, 2] \subset \mathbb{R}^2$ als Startwerte für das Newton-Verfahren. Die folgenden Bilder wurden mit 200×200 Punkten und den Parametern `tol` $= 10^{-10}$ und `maxIt` $= 1000$ erzeugt.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



Aufgabe 41 *(Programmieraufgabe: gedämpftes Newton-Verfahren)* (4M+4M+2T Punkte)

In Aufgabe 37 vom letzten Übungsblatt 09 haben wir die Konvergenzbereiche des Newton-, des Sekanten- und des Dekker-Verfahrens am Beispiel der Funktion $f(x) = \arctan x$ im Intervall $[-10, 10]$ untersucht. In dieser Aufgabe wollen wir überprüfen, ob das gedämpfte Newton-Verfahren für diese Funktion konvergiert.

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 Newton-Verfahrens berechnet. Die Parameter und die Rückgabewerte sind gleich wie bei Ihrer Funktion `xk = newtonSys(f, df, x0, tol, maxIt)` aus Aufgabe 40 für das Newton-Verfahren für Systeme. Der zusätzliche Parameter `lambdaMin` gibt den kleinstmöglichen Dämpfungsparameter λ an.

b) Ergänzen Sie Ihr Testskript `konvergenzBereiche.m` aus Aufgabe 37 um das gedämpfte Newton-Verfahren.

c) Interpretieren Sie Ihre Ergebnisse. Für welche Startwerte konvergiert das gedämpfte Newton-Verfahren?

Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Der Source Code muss strukturiert und dokumentiert sein. Senden Sie **spätestens 24 Stunden vor Ihrem Tutorium** alle Matlab-Files und alle Ergebnisse in einer E-mail mit dem Betreff **Loesung-Blatt10** an **angewandte.numerik@uni-ulm.de**.