



Numerische Analysis - Matlabblatt 1

(Besprechung: Freitag, 9. Mai 2014)

Hinweise

- Ihren Tutoriumstermin entnehmen Sie der aktuellen Liste auf der Homepage.
- Abgabe der Lösungen nur per Email und **nur** an den **jeweiligen Matlabtutor**, also entweder an
 - `katharina.schachmatov@uni-ulm.de` oder
 - `christoph-1.maier@uni-ulm.de` oder
 - `markus-2.schuster@uni-ulm.de`

mit Betreff

[NumAna]: Blatt0x - NachnameVorname

wobei $x \in \{1, 2, 3, 4, 5, 6\}$ die Nummer des Übungsblattes ist.

- Ihre Lösungen werden dann in den Matlabtutorien diskutiert und bewertet. Fehlerhafte Programme führen zu Punktabzug.

Aufgabe 1 (Nullstellen kubischer Polynome)

(6 Punkte)

Die Nullstellen von kubischen Polynome der Form

$$p(x) = ax^3 + bx^2 + cx + d$$

kann man mit Hilfe der Cardanischen Formeln¹ analytisch berechnen.

- Recherchieren Sie im Internet und/oder in Fachliteratur nach den Cardanischen Formeln und versuchen Sie, die Herleitung der verschiedenen Fälle zu verstehen.
- Implementieren Sie eine Funktion

```
x = cubicRoots(a,b,c,d)
```

die als Parameter die Koeffizienten $a, b, c, d \in \mathbb{R}$ mit $a \neq 0$ übergeben bekommt und die Nullstellen des Polynoms p berechnet und diese in einem Vektor $x \in \mathbb{C}^3$ zurückliefert. Achten Sie darauf, dass alle Fälle abgedeckt werden.

- Schreiben Sie ein Testskript `test_cubicRoots.m`, welches die Nullstellen für folgende Polynome berechnet

- $p(x) = 2x^3 - 12x^2 + 24x - 16$
- $p(x) = x^3 + 3x^2 - 6x - 1$
- $p(x) = x^3 + 2x^2 + 3x - 2$
- $p(x) = 3x^3 + 3x^2 - 99x + 189$
- $p(x) = -x^3 - 6x + 20$.

¹Siehe z.B. <http://www.matheplanet.com/matheplanet/nuke/html/article.php?sid=483>

Aufgabe 2 (*Newton- und Halley-Verfahren*)

(8 Punkte)

Das Newton-Verfahren und das Halley-Verfahren² berechnen jeweils für genügend glatte Funktionen eine zugehörige Nullstelle. Der Unterschied der beiden Verfahren besteht darin, dass beim Halley-Verfahren die Information der zweiten Ableitung genutzt wird. Ziel der Aufgabe wird es sein, beide Verfahren zu implementieren und miteinander zu vergleichen.

a) Schreiben Sie eine Funktion

```
[x,nit] = newton(f,df,x0,tol)
```

welche für die Eingabeparameter

- **f**: die Funktion f als *function handle*
- **df**: die Jacobimatrix J_f bzw. die Ableitungsfunktion f' als *function handle*
- **x0**: der Startwert
- **tol**: die Toleranz für den Abbruch der Iteration

einer gegebenen Funktion f eine Nullstelle mittels Newton-Verfahren berechnet und sowohl die Nullstelle \mathbf{x} als auch die Anzahl der benötigten Iterationen **nit** zurückliefert. Beachten Sie hierbei, dass dies auch **für Systeme**, d.h. $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ funktionieren sollte.

b) Schreiben Sie eine Funktion

```
[x,nit] = halley(f,df,d2f,x0,tol)
```

welche für die Eingabeparameter

- **f**: die Funktion f als *function handle*
- **df**: die Ableitungsfunktion f' als *function handle*
- **d2f**: die Ableitungsfunktion f'' als *function handle*
- **x0**: der Startwert
- **tol**: die Toleranz für den Abbruch der Iteration

einer gegebenen Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$ eine Nullstelle mittels Halley-Verfahren berechnet und sowohl die Nullstelle \mathbf{x} als auch die Anzahl der benötigten Iterationen **nit** zurückliefert.

c) Schreiben Sie ein Testskript `testRoots.m`, welches beide Verfahren für folgende Funktionen

(i) $f(x) = \cos x - x, x_0 = 0.1$

(ii) $f(x) = e^x - 1, x_0 = 0.5$

testet und diese inklusive der berechneten Nullstellen dann in einem Plot darstellt. Verwenden Sie **tol = 1e-8**.d) Schreiben Sie ein Testskript `compareNewtonHalley.m`, welches für $f(x) = \arctan x$ und 200 Startwerte $x_0 \in [-10, 10]$ die Anzahl der Iterationen beider Verfahren bestimmt und diese Beobachtung visualisiert (Plot - x-Achse: Startwert, y-Achse: Anzahl der Iterationen). Verwenden Sie **tol = 1e-10**. Was fällt auf?

*Hinweis: Verwenden Sie als Abbruchkriterium der Iteration $\mathbf{f}(\mathbf{x}_n) \leq \mathbf{tol} * \max\{\mathbf{f}(\mathbf{x}_0), 1\mathbf{e}-10\}$*

²<http://de.wikipedia.org/wiki/Halley-Verfahren>

Aufgabe 3 (*Fractal Sets*)

(6 Punkte)

Julia sets are famous fractal shapes in the complex plane. They are constructed from the basins of attraction of zeros of complex functions when the Newton method is applied to find them. In the space \mathbb{C} of complex numbers the equation

$$z^3 = 1$$

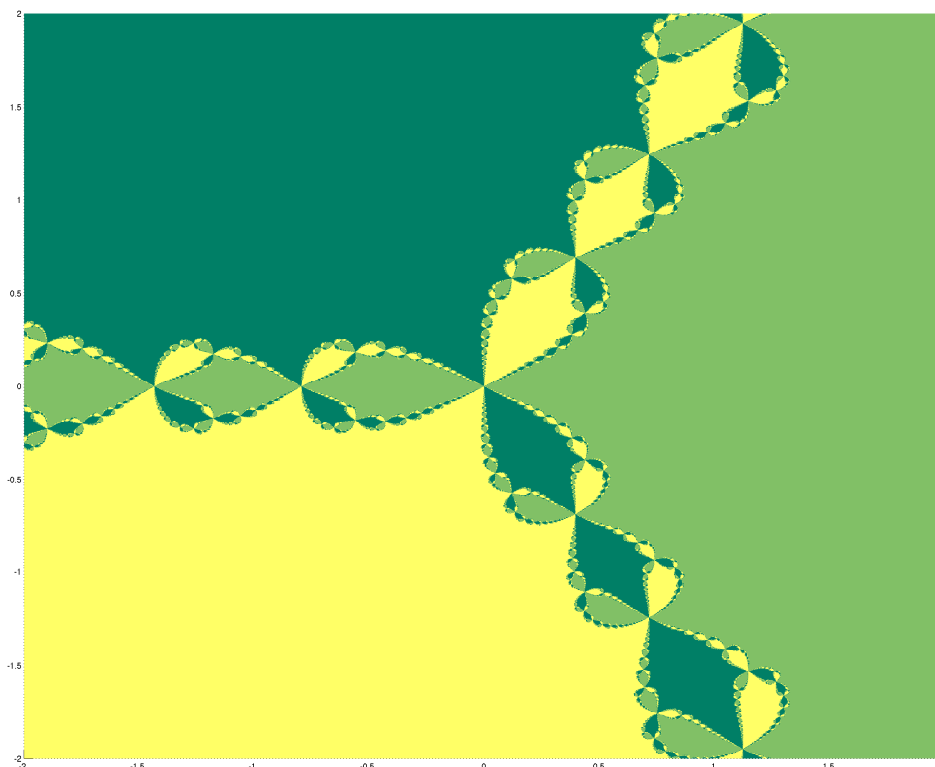
has three solutions $z_1 = 1$, $z_2 = -\frac{1}{2} + \frac{\sqrt{3}}{2}i$ und $z_3 = -\frac{1}{2} - \frac{\sqrt{3}}{2}i$. They're also known as the cubic roots of unity. As you know from the analysis course, the complex plane \mathbb{C} can be identified with \mathbb{R}^2 and using this identification this equation can be converted into a system of equations $F(x, y) = 0$ for a suitable function $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$. Using the formulation of the Newton iteration for the non-linear iteration $F(\mathbf{x}) = 0$ with $\mathbf{x} = (x, y)^T$ from exercise sheet 2, denote by \mathbf{x}_k the iterates produced by the Newton method with some initial vector \mathbf{x}_0 this sequence will either diverge or converge to one of the three cubic roots of unity.

Write a script `drawSet.m`, which analyzes the behavior of the Newton iterates using the following procedure

- i) use equally spaced points (e.g. 200 x 200 points gives a "nice" picture) on the domain $[-2, 2] \subset \mathbb{R}^2$ as starting points of the Newton iterations,
- ii) color the starting points differently depending on which of the three roots is the limit of the sequence \mathbf{x}_k
- iii) You may stop the iteration once you are closer in distance to one of the three roots of unity than 10^{-4} .

The three (not connected) sets of points whose iterations are converging to the different z_i are called Fatou domains, their boundaries are the Julia sets.

Hint: Useful Matlab commands are `meshgrid`, `surf`, `view(2)`, `colormap`, `shading`. The result should look like



Mehr Informationen zur Vorlesung und den Übungen finden Sie auf

<http://www.uni-ulm.de/mawi/mawi-numerik/lehre/sommersemester-2014/numana.html>