Dr. R. Käppeli
M. Sprecher

Serie 1

1. Matlab-Aufgabe **Pell–Zahlen**

Die Folge $f_0, f_1, f_2, f_3, \ldots$ der Pell–Zahlen ist wie folgt definiert:

$$f_0 = 0;\ f_1 \ = \ 1;\ f_n = 2f_{n-1} + f_{n-2}, \quad n > 1.$$

Es gelten also $f_2 = 2f_1 + f_0 = 2$, $f_3 = 2f_2 + f_1 = 5$, $f_4 = 2f_3 + f_2 = 12$ usw.

a) Schreiben Sie eine Matlab-Funktion `pell(n)`, die zu einer gegebenen positiven ganzen Zahl n die Pell–Zahlen $f_0, \ldots, f_n$ berechnet und in einem Zeilenvektor zurückgibt. Sie können dazu das Grundgerüst `pell.m` von der Homepage verwenden.

b) Geben Sie die Pell–Zahlen f_0 bis f_9 an.

c) Erstellen Sie einen Plot der Pell–Zahlen f_n über n für $n = 1, 2, \ldots, 50$. Skalieren Sie dabei die f_n-Achse logarithmisch indem Sie den Matlab-Befehl `semilogy` verwenden.

2. Matlab-Aufgabe **Einfache Fixpunktiterationen**

Betrachten Sie für $a \in \mathbb{R}, a \neq 0$ die Fixpunkgleichungen $x = \Phi_i(x), i = 1, 2, 3$ mit

$$\Phi_1(x) = a + x - x^3,\ \Phi_2(x) = 1 + x - \frac{x^3}{a} \text{ und}$$
$$\Phi_3(x) = \frac{1}{3}\left(2x + \frac{a}{x^2}\right) \qquad \forall x \in \mathbb{R}\backslash\{0\}.$$

a) Bestimmen Sie für $i = 1, 2, 3$ die Lösung(en) x der Gleichungen $x = \Phi_i(x)$ (ohne Matlab).

b) Implementieren Sie ähnlich wie im Matlab-Tutorial eine Matlab-Funktion `function x=fixIt(Phi,x0,tol,kmax)`, die die Fixpunkiteration

$$x_{i+1} = \Phi(x_i), \qquad i = 0, 1, 2, \ldots.$$

mit dem Startwert $x_0 =$ `x0` durchführt. Der Rückgabewert `x` ist dabei ein Vektor mit allen Iterationswerten x_i. Brechen Sie dabei die Iteration ab, falls $|x_{i+1} - x_i| <$ `tol` erfüllt ist, oder falls `kmax` Schritte durchgeführt wurden.

Bitte wenden!

c) Verwenden Sie Ihre Implementierung um für $a = 3$ die Fixpunktiterationen zu Φ_1, Φ_2, Φ_3 ausgehend von $x_0 = 1$ durchzuführen. Verwenden Sie als Abbruchkriterien `tol`=10^{-8} und `kmax=1000` und geben Sie für jede der Iterationen die ersten 5 Iterationswerte aus.

d) Welche der Iterationen Φ_1, Φ_2, Φ_3 konvergiert für $a = 3$ und $x_0 = 1$?

e) **(optional)** Verwenden Sie Ihre Implementierung aus Aufgabe 3 um die numerische Konvergenzordnung der 3 Iterationen zu bestimmen.

3. Matlab-Aufgabe **Numerische Konvergenzordnung**

Ein Kommiltone hat ein numerisches Verfahren zur Bestimmung der Quadratwurzel $\sqrt{a}$ einer Zahl a entwickelt, er möchte nun wissen wie schnell das Verfahren konvergiert. Dazu schlagen Sie einen numerischen Test vor, er soll mit seinem Verfahren für $a = 4$ die Wurzel $\sqrt{4} = 2$ bestimmen und Ihnen dann die Werte $e_k = |x_k - 2|, k = 1, 2, \ldots$ geben. Die Werte $e_1, \ldots, e_{14}$ stehen auf der Homepage in der Datei `konvergenzstudie.m` zum Download bereit. Schätzen sie mittels der Formel $\log(e_{k+2}/e_{k+1}) / \log(e_{k+1}/e_k)$ aus der Vorlesung die Konvergenzordnung p für $k = 1, 2, 3, 4, 5$. Welche Konvergenzordnung p hat das Verfahren also vermutlich? Warum ändert sich der Schätzer für p wenn Sie k variieren?

4. Matlab-Aufgabe **Nullstellen des Wilkinson Polynoms**

a) Was ist das Wilkinson Polynom und welche Nullstellen hat dieses Polynom?
Hinweis: Die englische Wikipedia hat einen interessanten Artikel über das "Wilkinson's polynomial".

b) Führen Sie folgenden Matlab-Code zur Bestimmung der Nullstellen des Wilkinson Polynoms durch

```
>> format long
>> r = roots(poly(1:20))
```

c) Wie genau sind die von diesem Matlab-Code berechnete Näherungen der Nullstellen?

Erstellen Sie ein Zip-file ihrer Matlab-Programme und laden Sie dieses unter `www.math.ethz.ch/~grsam/submit` hoch. Die schriftlichen Ergebnisse können Sie separat in den jeweiligen Übungsgruppen oder in den Assistentenfächern im HG G 53 abgeben.
Abgabe: Bis Freitag, den 28.02.2013