D-ITET, D-MATL
Dr. R. Käppeli
B. Fitzpatrick, D. Devaud

Serie 11

1. *Adaptive Schrittweitensteuerung*

a) Modifizieren Sie die MATLAB Funktionen `RKF45.m` und `adaptHeun.m` von Serie 10 gemäss Algorithmus auf Seite 12 von Kapitel III. Beachten Sie folgende Punkte:

- Falls die Schrittweite `h` kleiner als eine gegebene Toleranz `hmin` wird, soll der Algorithmus enden.
- Für den Algorithmus basierend auf dem Heun-Verfahren und der Intervall-Halbierungs-Methode, benutzen Sie den Fehlerschätzer $\hat{\epsilon}_{j+1}$ gegeben auf Seite 7 von Kapitel III.

b) Wiederholen Sie Aufgabe **1.b)** von Serie 10 mit den modifizierten Algorithmen. Was beobachten Sie? Erklären Sie warum die modifizierte Algorithmen besser sind.

2. *Mehrkörpersimulation*

Lösen Sie mithilfe ihres RKF45-Lösers das folgende Problem:

Wir betrachten im Weltraum p Körper, die in einer Ebene liegen, und Koordinaten $r_1, \ldots, r_p$, mit $r_p \in \mathbb{R}^2$, und Masse $m_1, \ldots, m_p$ haben. Laut Newton's Gravitationsgesetz wirkt Körper k auf den Körper j mit der Kraft

$$Gm_j m_k \frac{r_k - r_j}{|r_k - r_j|^3}.$$

Die Kraft auf Körper r_j zeigt in die Richtung von r_k mit Stärke proportional zu $|r_k - r_j|^{-2}$. Wir wählen jetzt die Zeitskala so, dass die universale Gravitationskonstant gleich 1 ist. Die Bewegung der Körper, die über Gravitation aufeinander einwirken, ist gegeben durch

$$\ddot{r}_j = A_j(r_1, \ldots, r_p) = \sum_{k \neq j} m_k \frac{r_k - r_j}{|r_k - r_j|^3}.$$

Bitte wenden!

Mithilfe von Anfangswerten $r_j(0)$ und $\dot{r}_j(0)$, kann man das Problem auf ein Anfangswertproblem für Systeme erster Ordnung zurückführen:

$$\begin{aligned} x_{4(j-1)+1} &= r_{x,j} \quad \text{(die } x\text{-Komponente von } r_j\text{)} \\ x_{4(j-1)+2} &= r_{y,j} \quad \text{(die } y\text{-Komponente von } r_j\text{)} \\ x_{4(j-1)+3} &= \dot{r}_{x,j} \quad \text{(die } x\text{-Komponente von } \dot{r}_j\text{)} \\ x_{4(j-1)+4} &= \dot{r}_{y,j} \quad \text{(die } y\text{-Komponente von } \dot{r}_j\text{)} \end{aligned}$$

Auf diesem Wege erhalten wir $x_1 = r_{x,1}$ und $x_8 = \dot{r}_{y,2}$. Das ODE System für $x(t)$ ist dann

$$\begin{aligned} \frac{d}{dt}\begin{pmatrix} x_{4(j-1)+1} \\ x_{4(j-1)+2} \end{pmatrix} &= \begin{pmatrix} x_{4(j-1)+3} \\ x_{4(j-1)+4} \end{pmatrix} \\ \frac{d}{dt}\begin{pmatrix} x_{4(j-1)+3} \\ x_{4(j-1)+4} \end{pmatrix} &= \begin{pmatrix} A_{x,j}(r_1, \ldots, r_p) \\ A_{y,j}(r_1, \ldots, r_p) \end{pmatrix} \end{aligned}$$

Dieses Problem ist in der Datei `body_problem.m` für 2 Körper und eine Wahl von Anfangsdaten vorimplementiert. Schreiben Sie eine Funktion `twobodyrhs.m` und rufen Sie das Template `body_problem.m` auf, um die Simulation zu sehen. Wie verhalten sich die adaptiven Schrittweiten?

3. *Fixpunkte*

Geben Sie alle Fixpunkte der Fixpunktfunktion $\phi(x) = 2x^3 - 7x$ an.

4. *Fixpunktiteration*

Wir betrachten das Nullstellenproblem

$$f(x) = xe^x - 1 = 0$$

auf dem Intervall $[0, 1]$ und die Fixpunktiterationen

$$x^{(k+1)} = \phi_i(x^{(k)}) \quad , \, i = 1, 2, 3$$

mit den folgenden drei Fixpunktfunktionen

- $\phi_1(x) = e^{-x}$
- $\phi_2(x) = \frac{x^2 e^x + 1}{e^x(x+1)}$
- $\phi_3(x) = x + 1 - xe^x$.

Siehe nächstes Blatt!

a) Zeigen Sie für ϕ_i ($i = 1, 2, 3$), dass das Fixpunktproblem konsistent mit dem Nullstellenproblem ist.

b) Schreiben Sie eine MATLAB Funktion

```
x = fixpunkt(phi,x0,niter)
```

welche eine Fixpunktiteration `phi` implementiert. Weiter sind `x0` ein Startwert, d.h. $x^{(0)}$, und `niter` die Anzahl auszuführende Iterationen. Die Funktion soll im Vektor `x` alle Glieder $x^{(k)}$ zurückgeben.

c) Wenden Sie die `fixpunkt.m` Funktion an den Fixpunktfunktionen ϕ von **a)** an. Benutzen Sie $x_0 = 0$ und $niter = 1000$. Was beobachten Sie?

Abgabe: Bis Freitag, den 19.05.2017.