Sommersemester 2010
27. April 2010

Übungen zur Vorlesung

Mathematische Aspekte der Modellierung und Simulation in den Neurowissenschaften

Dr. S. Lang, D. Popović

Abgabe: 04. Mai 2010 in der Übung

Übung 3 Matrix-Vektor-Multiplikation mit Octave (5 Punkte)

Eine Hilbert-Matrix $A(n)$ des Rangs n ist definiert über $a_{ij} := \frac{1}{i+j-1}$. Die Hilbert-Matrix des Ranges 3 ist zum Beispiel

$$A(3) = \begin{pmatrix} 1.00000 & 0.50000 & 0.33333 \\ 0.50000 & 0.33333 & 0.25000 \\ 0.33333 & 0.25000 & 0.20000 \end{pmatrix}.$$

Wir wollen in dieser Übung das Gleichungssystem $A(n)\cdot\vec{x} = \vec{b}$ lösen, wobei und $\vec{x}$ und $\vec{b} := (1, 0, \dots, 0)^t$ Vektoren des $\mathbb{R}^n$ sind. Eine Methode, ein Gleichungssystem zu lösen, haben wir mit der *Octave*-Funktion `x1 = A\b` in der letzten Übung kennengelernt. Eine andere ist das Verwenden der Gauss-Elimination. In *Octave* geschieht dies mit der Funktion `C = rref(A,b)`, die in der $(n \times n+1)$-Matrix C die Matrix A nach Gauss-Elimination und in der letzten Spalte von C das Ergebnis der entsprechenden Operationen angewendet auf $\vec{b}$ speichert. Mit `x2 = C(1:n,n+1)` wird also die Lösung des Gleichungssystems ausgegeben. Durch `[x1, x2]` werden die Lösungsvektoren nebeneinander auf dem Bildschirm ausgegeben.

1. Berechnen Sie „von Hand“ die Lösung für $n = 3$.
2. Lösen Sie für $n = 1, 2, \dots 10$ das Gleichungssystem $A(n) \cdot \vec{x} = \vec{ba}$ mit beiden oben genannten Funktionen und vergleichen Sie die Ergebnisse miteinander und für $n = 3$ mit der analytischen Lösung aus (1).
3. Vergleichen Sie die Lösungen aus (2) mit den Ergebnissen der *octave*-Funktion `x3 = invhilb( n)* b`.

Die Hilbert-Matrizen sind ein Beispiel für schlecht konditionierte Matrizen, die zu numerischen Fehlern in Algorithmen führen können. Dieser Effekt sollte in dieser Aufgabe auftreten und beobachtet werden können. Die letzte Methode mit der Funktion `invhilb` sollte zu numerisch stabilen Ergebnissen führen.

Übung 4 Lösen von linearen Gleichungssystemen mit octave (5 Punkte)

Octave besitzt einige built-in Integratoren zum Lösen von partiellen Differentialgleichungen. Einer der bekanntesten ist *lsode* von Alan C. Hindmarsh.

Machen Sie sich mit der Hilfe-Funktion oder anderer Dokumentation mit der Funktion `lsode` vertraut und lösen Sie anschließend das folgende System gewöhnlicher Differential-Gleichungen:

$$\begin{aligned} \partial_t x_1 &= 5 \cdot (x_2 - x_1 \cdot x_2 + x_1 - 10^{-6} \cdot x_1^2) \\ \partial_t x_2 &= (x_3 - x_1 \cdot x_2 - x_2)/5 \\ \partial_t x_3 &= 0.1 \cdot (x_1 - x_3) \end{aligned}$$

mit den Startwerten $(x_1(0), x_2(0), x_3(0))^t = (0.1, 3, 0.3)^t$. Der Aufruf von `lsode` geschieht über `y = lsode ("lsode_example1", x0, t);` wobei `x0` der Startwert und `t` ein Zeitinterval ist. Der Parameter `lsode_example1` ist der Name einer Funktion, die die Differentialgleichung implementiert. Am Besten wird diese Funktion in einer Datei gleichen Namens gespeichert (also `lsode_example1.m`). Die Datei kann für dieses Beispiel dann wie folgt aussehen:

```
 1 function xdot = lsode_example1 (x, t)
 2
 3   xdot = zeros (3,1);
 4
 8   xdot(1) = x(1)*x(2);
 8   xdot(2) = ...
 8   xdot(3) = ...
11
12 endfunction
```

Die Funktion `lsode_example1` berechnet das Differentialgleichungssystem und gibt die Lösung des Systems in einem Vektor zurück. Für diese Übungen müssen Sie natürlich noch die passenden Differentialgleichungen einbauen.

Plotten Sie alle drei Komponenten der Lösung im Bereich $x = [0, 500]$ mit einem Spacing von 2. Bei ungefähr $t = 0, 125, 250$ und $t = 365$ sollten Sie Peaks der Lösung erkennen. Mit Hilfe des `logspace` -Befehls können Sie den Plotbereich um die Peaks herum verfeinern. Exportieren Sie einen Plot der Lösung, falls möglich, mit angepasstem Bereich (ansonsten mit dem uniformen).

Übung 5 Informationsgehalt des Izhikevič-Modells (5 Punkte)

In der Vorlesung haben Sie das Punkt-Neuronen-Modell von Izhikevič kennengelernt. In dieser Aufgabe wird für das Modell der Informationsgehalt nach Shannon mit Octave numerisch ermittelt.

Dem Informationsgehalt nach der Shannonschen Informationstheorie liegt folgende Idee zu Grunde:

- Eine Nachricht ist eine Folge z von Zeichen (ein *Wort*) aus der Potenzmenge (alle möglichen Wörter) A^* eines Alphabets A.
- Zugrundegelegt wird eine Wahrscheinlichkeitsverteilung für das Auftreten der Zeichen aus A in einer Nachricht z.
- Es ist ein Versuch, den Begriff der Information statistisch zu erfassen.

Für den Informationsgehalt $I(a)$ eines einzelnen Zeichens a gilt nun

- $I(a)$ um so höher, je unwahrscheinlicher das Auftreten von a ist.
- $I(a) = 0$, falls das Zeichen sicher auftritt, d.h. Wahrscheinlichkeit $w(a) = 1$.
- Definiere $I(a) := -log_2(w(a))$, wobei $w(a)$ die Wahrscheinlichkeit des Auftretens von $a \in A$ ist.

Wenn eine Quelle zum Beispiel immer das gleiche Zeichen a sendet, gilt $w(a) = 1$ und $I(a) = 0$. Für das binäre Alphabet $A = \{0, 1\}$ ist bei der Wahrscheinlichkeit $w(0) = w(1) = 0.5$ für eine Nachricht $z = 010001001$ der Informationsgehalt $I(z) = 9$.

Um für das Izhikevič-Modell den Informationsgehalt zu messen, gehen wir davon aus, daß das Alphabet nur aus den Werten 0 und 1 (Spike detektiert) besteht, und die Länge der Nachricht die Simulationszeit Tgeteilt durch die Schrittweite dt ist. Die Warscheinlichkeit $w(1)$ des Zeichens 1 ist dann die Anzahl der detektierten Spikes durch Nachrichtenlänge und $w(0) = 1 - w(1)$.

1. Ermitteln Sie nun für das Izhikevič-Modell den Informationsgehalt $I(1)$ mit Octave für vier verschiedene Kombinationen der Parameter a, b, c, d und listen Sie das Ergebnis in einer Tabelle auf. Für welche Werte erhalten Sie den höchsten Informationsgehalt?