

**Interaktives mathematisches Paket WS 11**  
**9. Computerpraktikum zum Abgabetermin 23.01.2012, 23:59**

Hinweis Zur Lösung der folgenden Aufgaben dürfen keine `break`-, `continue`- oder `goto`-Konstrukte benutzt werden.

42. Ein Jogger läuft entlang einer (von Ihnen) vorgegebenen, parametrisierten Kurve  $(X(t); Y(t))$ . Ein Hund verfolgt den Jogger mit einem konstanten Betrag der Geschwindigkeit  $v$ . In jedem Augenblick der Bewegung läuft der Hund auf die Stelle zu, an der sich der Jogger gerade befindet. Schreiben sie ein MATLAB-Programm, das für eine gegebene Bewegungskurve des Joggers die Bewegungskurve des Hundes zum Zeitpunkt  $t$  bestimmt und die beiden Kurven grafisch darstellt. (2 Pkt.)

*Hinweis:* Stellen Sie die Kurve des Joggers (und auch des Hundes) durch dieselben Daten dar, welche für `plot` benötigt werden. Zu jedem diskreten Zeitpunkt  $t_k$  müssen Sie die Laufrichtung des Hundes neu berechnen da sich der Jogger weiterbewegt hat.

43. Schreiben Sie eine Funktion `trace_43`, welche die Bahnpunkte sowohl des Verfolgungsproblems des Joggers-Hund-Paares als auch des entsprechenden Flugzeug-Rakete-Paares für einige Zeitschritte errechnet. Massenträgheit und Massenänderung sollen nicht berücksichtigt werden, wohl aber eine Verfolgung im  $n$ -dimensionalen Raum. (3 Pkt.)

Testen Sie Ihre Verfolgungsfunktion in 2D und in 3D. Visualisieren Sie die Verfolgungen.

44. Schreiben Sie eine Matlab-Funktion `bsp_44_fkt` welche ein Badmintonspiel zweier Spieler simuliert. Die Funktion soll, neben dem Sieger, die in jedem Satz erzielten Punkte an das aufrufende Skript zurückliefern. Wir benutzen die Rallypointzählweise, d.h.:
- Ein Satz muß mindestens bis zu 21 Punkten, aber höchstens bis zu 30 Punkten gespielt werden. Wenn dann einer der Spieler mindestens 2 Punkte zurückliegt (also mindestens 21:19) hat der Spieler mit den Mehrpunkten den Satz gewonnen. Ansonsten muß danach so lange weitergespielt werden, bis ein Spieler 2 Punkte Vorsprung erreicht hat, jedoch bestenfalls bis 30:29. Die Zahl der Sätze hängt vom Spielverlauf ab. (3 Pkt.)
  - Zum Sieg sind zwei Gewinnsätze erforderlich und die Zahl der Sätze hängt natürlich vom Spielverlauf ab. Dies ergibt bei
    - 2 Siegsätzen eines Spielers in Folge (2:0) 2 Sätze und bei
    - 2 Siegsätzen des einen, 1 Siegsatz des anderen (2:1) 3 Sätze.

Ein Spiel kann also höchstens 3 Sätze haben.

- Nutzen Sie zur Simulation der Gewinnpunkte die Funktion `rand`  $\in (0, 1)$  und weisen Sie den beiden Spielern geeignete Teilintervalle von  $(0, 1)$  zu, um den Gewinner des Punktes zu bestimmen.

- 45.\* Simulieren Sie die Spiele eines Badmintonturniers an dem  $n$  Spieler teilnehmen, welche alle genau einmal gegeneinander spielen (Sie müssen keinen Spielplan aufstellen). Nutzen Sie zur Simulation der einzelnen Spiele Ihre Funktion `bsp_44_fkt`. Geben Sie die Ergebnisse der einzelnen Spiele (Sätze, Punkte im Satz) aus und präsentieren Sie zum Schluß die gereihte Endtabelle des simulierten Turniers. Die Reihung soll zuerst nach gewonnenen Spielen, dann (bei diesbzgl. Gleichheit) nach der besseren Satzdiffenz und danach (falls immer noch Gleichheit vorliegt) nach der Anzahl der gewonnenen Sätze erfolgen. (3 Pkt.)

*Hinweise*

- Nutzen Sie die Matlab-Funktion `sort`, am besten in der Form `[B,IX] = sort(A, ...)`.
- Der notwendige Vergleich für die Reihung der Spieler geht am einfachsten über eine Bewertungsfunktion der Ergebnisse bei welcher beispielsweise Spielgewinne wesentlich höher gewichtet werden als die Satzdiffenz.

*Testdaten* ( $n$ ): (4), (10)

46. Nach den europäischen Regeln, auch als französische Regeln bekannt, besteht das Roulette aus den Feldern 0 bis 36. Fällt die Kugel auf die 0 dann bekommt die Bank Ihren Einsatz; außer Sie haben auf eine einfache Chance gesetzt (gerade/ungerade, rot/schwarz, untere/obere Hälfte). In diesem Falle können Sie sich entscheiden, ob die Bank die Hälfte des Einsatzes einzieht, oder ob Sie den Gesamteinsatz liegen lassen (*en prison*). (4 Pkt.)

Simulieren (Funktion `randi`) Sie mit MatLab das europäische Roulette im Casino und setzen Sie immer auf *ungerade Zahl*. Wenn Sie gewinnen, erhalten Sie einen Gewinn in der gleichen Höhe wie ihr Einsatz, wenn sie verlieren, verfällt ihr Einsatz. Bei 0 wenden sie *en prison* an, d.h., Sie erzielen weder Gewinn noch Verlust.

Als Spielstrategie wenden Sie das *Martingale System* an, welches folgendermaßen funktioniert:

Sie starten mit einem Grundeinsatz  $E_0$ . Wenn Sie verlieren, verdoppeln Sie im nächsten Spiel Ihren Einsatz. Sobald Sie gewinnen, stecken Sie ihren Gewinn ein und starten erneut mit dem Grundeinsatz. Spielen Sie solange, bis Sie den 5-fachen Grundeinsatz  $E_{end}$  in der Hand haben.

In jedem Schritt soll die Nachricht 'gewonnen', 'verloren' oder 'en prison', der aktuelle Kontostand  $K$  (der natürlich bei längerem Verlust auch negativ sein kann) und die Einsätze für das aktuelle und das nächste Spiel ausgegeben werden. Beenden sie die Berechnung, wenn der Kontostand den Wert  $E_{end}$  erreicht hat. Am Anfang beträgt der Kontostand  $E_0 = 10$  EUR.

- (a) Die Bank gewährt ihnen unbeschränkt Kredit, d.h. sie können einen beliebigen negativen Kontostand während des Spiels haben.
- (b) Wie ändert sich Ihr Gewinn/Verlust, wenn Sie maximal 100 Euro pro Spiel setzen dürfen?

Simulieren Sie einige hundert bis tausende Casinobesuche. Protokollieren und Visualisieren Sie für (a) und für (b) den erzielten Gewinn/Verlust, die Anzahl der zum Gewinn notwendigen Spiele und den minimalen (negativen) Kontostand während jedes Spieles.

Speichern Sie die erzielten Roulettezahlen für sämtliche Spiele und werten Sie diese bzgl. Gleichverteilung der Zahlen aus.

Wieviel Rendite erzielen Sie bzgl. des eingesetzten Geldes?

Hinweise: Matlab-Funktionen: `mod`, `rand`, `randi`, `hist`, `bar`, `figure`, `sign`, `abs`, `&&`, `num2str`

Abgabe der Lösungen:

Zu jeder Aufgabe muß (mind.) das File `bsp_nummer.m` abgegeben werden.

Die Abgabe der Lösungen (\*.m-Files und Grafiken) erfolgt ausschließlich durch das Kopieren dieser Files in Ihr (beim erstmaligen anzulegende) Verzeichnis `Nachname_Vorname` im globalen Abgabepfad, siehe LV-Homepage.

Die Filenamen **müssen** dem Schema `bsp_nummer`, gefolgt von der Fileextension, entsprechen. Z.B. sind in Beispiel 2 die Files `bsp_2.m` und `bsp_2.jpg` abzugeben. Andere Filebezeichner zählen nicht als abgegebene Files.