Abteilung für
Angewandte Mathematik

SoSe 2019
Christian Palus, Steve Wolff-Vorbeck

Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 10 – 08.07.2019

Abgabe: Briefkästen RZ/E-Mail bis Montag, den 15.07.2019, 16:00 Uhr

---

Aufgabe 1 (5 Punkte)**.** Ein bekanntes schachmathematisches Problem ist das N-Damenproblem, bei welchem auf einem quadratischen Schachbrett der Größe $N \times N$ wiederum N Damenfiguren so verteilt werden sollen, dass sie sich jeweils nicht gegenseitig bedrohen, d.h. in jeder Spalte, jeder Zeile und jeder Diagonalen höchstens eine Dame steht. Um das Problem zu lösen werde das Schachbrett als eine $N \times N$ Matrix $(A_{ij})_{i,j=1,..,N}$ aufgefasst. Formulieren Sie einen Algorithmus, welcher bei gegebener Verteilung von Figuren entscheidet, ob die Platzierung einer weiteren Dame in einem Feld A_{ij} zulässig ist, d. h. dieses Feld weder besetzt noch von einer bereits platzierten Dame bedroht ist.

Aufgabe 2 (5 Punkte)**.** Eine häufig verwendete Lösungsstrategie für Aufgabestellungen wie das N-Damenproblem aus Aufgabe 1 ist das *rekursive Backtracking*, das nach dem Versuch-und-Irrtum-Prinzip arbeitet. Das Vorgehen folgt dabei abstrakt dem folgenden Verfahren.

Algorithmus. *Eingabe:* Zustand , Aufruf: Löse(Zustand).

(1) Falls der Zustand eine Lösung ist, so stoppe und setze Rückgabewert **`true`**.
(2) Falls es im aktuellen Zustand einen noch nicht getesteten, zulässigen Teilschritt gibt, so führe diesen aus (Zustand $\rightarrow$ Zustand_neu). Andernfalls gehe zu (5).
(3) Löse (Zustand_neu), falls Rückgabe **`true`**, so stoppe mit Rückgabe **`true`**.
(4) Mache den Teilschritt in (2) rückgängig (Zustand $\leftarrow$ Zustand_neu) und gehe zu (2).
(5) Stoppe mit Rückgabewert **`false`**.

Formulieren Sie einen rekursiven Algorithmus zur Lösung des N-Damenproblems auf einem beliebigen $N \times N$ Schachbrett (Hinweis: Für das N-Damenproblem sind die Zustände gegeben durch die Brettmatrix A sowie die Anzahl N_D der noch zu verteilenden Damen. Der Anfangszustand ist damit gegeben durch (M, N) wobei $M \in \mathbb{R}^{N \times N}$ die Nullmatrix sei. Das Problem ist gelöst, falls man einen Zustand $(A, 0)$ erreicht, also alle Damen platziert sind.)

Aufgabe 3 (2+2+1 Punkte)**.** (i) Schreiben Sie ein MATLAB-Programm, das, wenn man vorher eine Matrix A und einen Vektor x definiert, mittels **`for`**-Schleifen das Matrix-Vektorprodukt $b = Ax$ berechnet (also einen entsprechenden Vektor b erstellt).
(ii) Schreiben Sie das Programm neu mit nur einer **`for`**-Schleife, die über die Zeilen der Matrix iteriert und dann jeweils das Skalarprodukt der Zeilenvektoren mit dem Vektor x (mittels `dot(x,y)`) berechnet und das Ergebnis in den entsprechenden Eintrag von b schreibt.
(iii) Testen Sie die Geschwindigkeit Ihrer Implementierungen und vergleichen Sie diese mit der Geschwindigkeit für das MATLAB-Matrix-Vektorprodukt `A*x` mittels der Befehle `tic` und `toc` für große Matrizen, z.B. `A=rand(1e4)` bzw. `x=rand(1e4,1)`.

Aufgabe 4 (4+1 Punkte)**.** Betrachten Sie das Räuber-Beute Modell aus Aufgabe 2 von Blatt 9 in der diskreten Version

$$B_{i+1} = B_i + \Delta t B_i(\epsilon_B - \gamma_B R_i),$$
$$R_{i+1} = R_i + \Delta t R_i(\epsilon_R - \gamma_R B_i).$$

(i)Implementieren Sie das obige Schema in MATLAB um die Räubern- bzw. Beutepopulation für Zeitpunkte $t \in [0, 60]$ zu bestimmen. Wählen Sie $\Delta t = 0.01$, $\epsilon_R = 0.3, \epsilon_B = 0.6$, $\gamma_R = 0.001$ und $\gamma_B = 0.005$ sowie die Anfangswerte $B_0 = 200$ sowie $R_0 = 300$. Visualisieren Sie Ihre Ergebnisse, indem Sie den zeitlichen Verlauf der einzelnen Populationen jeweils in einem 2d-Plot darstellen. Erstellen Sie auch ein Phasendiagramm, in welchem Sie die Räuber-Beutepopulation in einem 2d-Plot darstellen.
(ii) Was passiert, wenn Sie die Zeitschrittweiten $\Delta t = 0.25$ sowie $\Delta t = 0.5$ wählen?