

Computergestützte Mathematik zur Linearen Algebra – 5. Übungsblatt

Aufgabe 17: Befehle: `for`, `if`, `error`, `ceil`

Die Folge a_n sei folgendermaßen definiert:

$$a_1 = 0, a_2 = 1, a_n = \lceil \frac{5}{3}a_{n-1} - a_{n-2} \rceil, \quad n \in \mathbb{N} \setminus \{1, 2\}$$

($\lceil x \rceil$ ist das Zeichen für Aufrunden) Schreiben Sie eine Funktion `an=Folge(n)` die das n -te Folgenglied mit Hilfe einer Schleife berechnet. Für $n \notin \mathbb{N}$ soll mit `error` eine Fehlermeldung ausgegeben werden.

Aufgabe 18: (Sudoku)

Sudokuregeln: Bei einem Sudokuspiel der Größe N müssen in eine $N \times N$ Matrix S natürliche Zahlen z zwischen 1 und N eingetragen werden. Es ist zulässig die Zahl z in die Position $S_{i,j}$ einzutragen, falls die vier folgenden Bedingungen erfüllt sind:

- (i) Die Position $S_{i,j}$ ist unbesetzt.
 - (ii) Die Zahl z steht nicht bereits in der j -ten Spalte von S .
 - (iii) Die Zahl z steht nicht bereits in der i -ten Zeile von S .
 - (iv) Die Zahl z steht nicht bereits in einem der Unterblöcke von S , wie sie in der Abbildung für $N = 4$ und $N = 9$ markiert sind.
- (a) Schreiben Sie eine MATLAB-Funktion `[flag] = sudokutest(S,i,j,z)`, die für eine $N \times N$ Sudokumatrix S , eine Zahl z und Indices $i, j \in \{1, 2, \dots, N\}$ testet, ob die Zahl z nach den Regeln (i) – (iii) in die Position $S_{i,j}$ eingetragen werden darf. Falls z in der Position $S_{i,j}$ eingetragen werden darf, soll `flag = 1` zurückgegeben werden, andernfalls `flag = 0`.
- (b) Schreiben Sie zwei MATLAB-Skripte, die die MATLAB-Funktion `[flag] = sudokutest(S,i,j,z)` für die beiden unten angegebenen Beispielen mit $i = j \in \{1, 2, \dots, N\}$ und $z = 4$ testet.

		3	2
2	1	4	
4			

			1			2		
		2			3			
			9	5				
		8					7	
			7					5
	7	4				6		
					4			
6	2						4	8
3								

Aufgabe 19: (Sudoku)

Erweitern Sie Ihre Funktion `sudokutest` aus Aufgabe 19 um die Regel (*iv*) und führen Sie die Beispiele erneut aus.

Aufgabe 20: Befehle: `for`, `if`, `tic`, `toc`

Schreiben Sie mit Hilfe von `for` und `if` Schleifen zwei MATLAB-Funktionen, die die Befehle `any` und `all` umsetzen. Testen Sie Ihre Funktionen an einer $n \times n$ Zufallsmatrix mit $n = 10^2, 10^3, 10^4$ und vergleichen Sie mit Hilfe der Befehle `tic` und `toc` die Geschwindigkeiten Ihres `any` und `all` Befehles mit den MATLAB-Befehlen.