CompMath-Vorlesung 25. Okt. 2024

Zu 2.

- **Funktionen für Matrizen:** v_3_a
  - size, length, numel
  - zeros(n), zeros(n,m), ones, rand, randi, randn, diag,
  - sum, prod, diff, cumsum, cumprod:
    - Bsp.: x = 1:10
    - Summe der Quadratzahlen: sum(x.^2)
    - Fakultät 10!; prod(x); factorial(10)
    - cumsum(x)
- meshgrid, surf zur Visualisierung von Matrixeinträgen; spy → Besetztheitsmuster
  - A = gallery('poisson',5); spy(a)

Höherdim. Matrizen: view_jpg.m

- RGB-Darstellung von Images
  - imread
  - imwrite
    - verlustbehaftet: jpg-File
    - ohne Verluste: bmp-File (Bitmap)
  - image
  - 3 Dimensionen img(:,:,[1,2,3]) <==> RGB (Red-Green-Blue), Values in [0,255]

Array of cells of C-strings

ss = {'Deep','Purple'} % Wozu !?

- a = 'Deep'; b = 'Purple'; % Vector von char: 1x10 char
- s = [a,b] % nur noch ein langer String
- s = [a;b] % Spalten von a ungleich Spalten von b ==> Fehler
- s = {a,b} % Einzige Möglichkeit zwei Strings in einer Variablen zu speichern ==> array of cells of srings
- Zugriff:
  - a1 = s(1) % Cell
  - a2 = s{1} % C-String

Vektor von C++-Strings

vv = ["Deep","Purple"] % array 1×2 string

vv(1) % 1x1 string

vv{1} % 1x4 char (C-String)

Interne Speicherung von Zahlen

- dec2hex(55) % Dezimalzahl als 1 Byte int8, hexadezimal gespeichert
- num2hex(55) % Gleitkommazahl als 8 Byte double, hexadezimal gespeichert

3. Strukturierte Programmierung *v_4_a.m*

3.1. Sequenz

3.2 Funktion

- Deklaration:
- Input, Output
- FUNCTION-END
- Scope (Güligkeitsbereich) von Variablen

3.3. Alternative (IF-END) [Matlab-Editor unterstützt Einrückungen!]

- einseitig
- zweiseitig
- mehrseitig
- Klammerung des Scopes durch KEYWORD-END

3.4. Zählzyklus (FOR-END) % Range-For

- lesend/schreibender Zugriff auf Vektorelemente
    - Wdh.: Produkt der ungeraden Zahlen bis n
    - Wdh.: Berechnung Fibonaccizahlen F(1) = F(2) = 1; F(k) = F(k-1)+F(k-2) für k>=3
    - Hilbertmatrix: Symmetrie (2 Loops vs. 1.5 Loops)

3.5. abweisender Zyklen (WHILE-END; [DO-WHILE]) v_5_a.m

- einfache Eingabeabfrage via WHILE
    - Wdh.: Eingabe einer durch 7 teilbaren Zahl
- Umwandlung FOR- in WHILE-Zyklus: sum k
- Umwandlung FOR- in WHILE-Zyklus: sum 1/k^2 → pi^2/6
    1. sum bzw. cumsum
    2. FOR-Loop
    3. WHILE-Loop (äquivalent zu FOR)
    4. WHILE-Loop mit Genauigkeitsbbruch (`abs(sum(k-1)-realval) >= seps`)
- Wann nimmt man FOR, WHILE, DO-WHILE, IF ??
    - Ich kenne die Anzahl der Zyklendurchläufe bevor ich den Zyklus starte, d.h., diese Anzahl wird nicht durch Berechnungen im Zyklus beeinflußt ==> FOR
    - Die Anzahl der Zyklendurchläufe hängt von den Berechnungen im Zyklus ab ==> WHILE

3.6. sonstiges (CASE)