ComputerMathematik-LAK WS 25/26
3. Computerpraktikum zum Abgabetermin 19.01.2026, 23:55

Literatur

[Ueber05] C. Überhuber, S. Katzenbeisser and D. Praetorius. *MATLAB 7: Eine Einführung.* Springer-Verlag, Wien, 2005. E-book[1].

12. Zuerst ein wenig Mathematik:
Die **Drehung** eines Punktes/Ortsvektors $\vec{x} = \binom{x_1}{x_2}$ mit dem Winkel φ um den 2D–Koordinatenursprung (0,0) wird durch die Matrix–Vektor-Multiplikation

$$\vec{x}_{new} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}_{2\times 2} * \vec{x} \tag{1}$$

beschrieben.
Das **Verschieben** eines Punktes $\vec{x}$ um einen Vektor $\vec{v} = \binom{v_1}{v_2}$ wird durch die Vektoraddition

$$\vec{x}_{new} = \vec{x} + \vec{v} \tag{2}$$

ausgedrückt.

——————— nun zur Aufgabe ———————

Gegeben sei das aus den Punkten $(-1.5, -1)$, $(-3, -0.5)$, $(1, 3)$ und $(1.5, 2.5)$ bestehende Viereck.

(a) Zeichnen Sie das gegebene Viereck.

(b) Stellen Sie (in derselben Graphik), ausgehend vom gegebenen Viereck, das zuerst mit dem Vektor $\binom{-2}{4}$ verschobene und dann mit $2/3\ \pi$ Rad um den Koordinatenursprung gedrehte Viereck in derselben Graphik dar.

(c) Stellen Sie (in derselben Graphik) das zuerst mit $2/3\ \pi$ Rad mathematisch positiv um den Koordinatenursprung gedrehte und dann mit dem Vektor $\binom{-2}{4}$ verschobene Viereck graphisch dar.

Hinweis: Die MatLab-Funktion `plot(x,y)` zeichnet stets den Polygonzug

$$(x_1, y_1) \to (x_2, y_2) \to \cdots \to (x_n, y_n)\ .$$

Um einen geschlossenen Polygonzug zu erhalten muß daher der letzte Punkt wieder der erste sein (alternativ kann auch `fill` verwendet werden).
Hinweis: Sie können die Koordinaten so als eine Matrix x speichern, daß die Matrizen in Gleichung (1) kompatibel sind und sich alle Koordinaten mit einer Matrixoperation transformieren lassen. Siehe auch valide MATLAB-Befehlsfolge[2].

13. Berechenen Sie die (sogenannte Mengoli teleskopische) Summe $s = \sum_{k=1}^{n} \frac{1}{k(k+1)}$ der ersten n natürlichen Zahlen (n=11, 60, 101) mittels

(a) eines FOR-Loops

(b) ohne FOR-Loops mit `sum` oder `cumsum`

(c) vergleichen Sie die Ergebnisse, u.a., indem Sie die entsprechende Summenformel beweisen oder im Internet finden.

(d) Berechnen Sie die Mengoli Summe der ersten n ungeraden Zahlen.
Wie lautet diesmal die Summenformel?

[1] `http://search.obvsg.at/primo_library/libweb/action/search.do?vid=UGR`
[2] `https://de.mathworks.com/help/matlab/ref/plus.html#expand_panel_body_bu90zu2`

14. Zwei Primzahlen p und q werden als Primzahlencousins bezeichnet falls $q = p + 4$ gilt. Bestimmen Sie alle Primzahlencousins zwischen 967 und 1783. Geben Sie die Anzahl der gefundenen Paare an.
    Hinweise:
    - Schauen Sie sich die Matlabfunktionen `primes` und `isprime` an.
    - Sie werden einen/mehrere FOR/WHILE-Loops zusammen mit der Alternative (IF) benötigen.
    - Man kann die Primzahlencousins auch mit reinen Vektoroperationen bestimmen (fakultativ).

15. Entwerfen und schreiben Sie **zwei** Funktionen, welche die Reihenentwicklung für $\sin x$, $x > 0$

$$\begin{aligned} \sin x \approx t(x,n) \quad &:= \quad \sum_{k=0}^{n} \frac{(-1)^k\, x^{2k+1}}{(2k+1)!} \\ &= \quad \left[x - \frac{x^3}{6} + \frac{x^5}{120} + \cdots + \frac{(-1)^n\, x^{2n+1}}{(2n+1)!} \right] \end{aligned} \tag{3}$$

    in einem gegebenen Punkt x berechnen sollen.
    (a) Funktion `bsp_15_fkt_a(x,n)` zur Berechnung des Wertes $t(x,n)$ für Input-Parameter x, n mittels eines FOR-Loops.
    Output: approximierter Funktionwert
    (b) Funktion `bsp_15_fkt_b(x,delta)` zur Berechnung des Wertes $T(x,\delta)$ für Input-Parameter x, δ mittels eines WHILE-Loops. Hierbei sollen Sie die Summation in (3) solange fortführen wie der Absolutbetrag des neuen Summanden größer als δ ist.
    Output: approximierter Funktionwert und erreichter Berechnungsindex n.
    Die Konstrukte `break`, `continue`, `goto`, `return` sind nicht erlaubt.
    Wählen Sie selbständig Testdaten aus ($\delta <= 10^{-4}$) und kontrollieren Sie die Korrektheit der Berechnungen.
    Hinweis: Der Matlab-Octave Befehl für die Faktorielle von einer Zahl x, d.h. $x!$, ist $factorial(x)$.

16. Die Reihenentwicklungen in Aufg. 15 sollen visualisiert werden.
    (a) Grafik 1 (`bsp_16_a.jpg`): **Visualisieren** Sie für x im Intervall $[0.01, 10]$ die Funktion $\sin(x)$ und deren Approximationen $t(x,n)$ für $n = 10, 11, 12, 13, 14$ (Verwendung von `bsp_15_fkt_a` mit Vektoren?).
    (b) Grafik 2 (`bsp_16_b.jpg`): Berechnen Sie für fixes $x = 6.5$ den Fehler $err(n) = |\sin(x) - t(x,n)|$ der Reihenentwicklung $t(x,n)$ zur exakten Funktion $\sin(x)$ für $n = 10, 11, 12, 13, 14$.
    Visualisieren Sie diesen Fehler $err(n)$ mit n als Abszisse und dem Fehler als Ordinate. Warum ist die Verwendung des speziellen Matlab-Befehls `semilogy` zum Plotten hier von Vorteil?
    (c) Grafik 3 (`bsp_16_c.jpg`): Berechnen Sie für fixes δ = 1e-6 mittels Funktion $T(x,\delta)$ (`bsp_15_fkt_b`) den Abbruchindex n für $x \in [0.01, 10]$.
    Visualisieren Sie diesen Abbruchindex $n(x)$ als Funktion von x.

Abgabe der Lösungen:
Die Abgabe der Lösungen (`*.m`-Files und Grafiken) muß über Moodle[3]. erfolgen.

Die Filenamen **müssen** dem Schema `bsp_nummer`, gefolgt von der Filextension, entsprechen. Andere Filebezeichner zählen nicht als abgegebene Files.

Abzugebende Files:

`bsp_11.m`,	`bsp_11_fkt.m`		
`bsp_12.m`,	`bsp_12.jpg`		
`bsp_13.m`			
`bsp_14.m`			
`bsp_15.m`,	`bsp_15_fkt_a.m`,	`bsp_15_fkt_b.m`	
`bsp_16.m`,	`bsp_16_a.jpg`,	`bsp_16_b.jpg`,	`bsp_16_c.jpg`

[3] `http://moodle.uni-graz.at`