

**ComputerMathematik WS 20/21**  
**3. Computerpraktikum zum Abgabetermin 9.11.2020, 23:59**

15. Zuerst ein wenig Mathematik:

Die **Drehung** eines Punktes/Ortsvektors  $\vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$  mit dem Winkel  $\varphi$  um den 2D-Koordinatenursprung (0,0) wird durch die Matrix-Vektor-Multiplikation

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

beschrieben.

Das **Verschieben** eines Punktes  $\vec{x}$  um einen Vektor  $\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$  wird durch die Vektoraddition

$$\vec{x}_{new} = \vec{x} + \vec{v} \quad (2)$$

ausgedrückt.

nun zur Aufgabe

---

Gegeben sei das aus den Punkten (2.5, 0.5), (3, -2), (1.5, 0.5) und (-1, 3) bestehende Viereck.

- (a) Zeichnen Sie das gegebene Viereck.
- (b) Stellen Sie (in derselben Graphik) das zuerst mit 180 Grad mathematisch positiv um den Koordinatenursprung gedrehte und dann mit dem Vektor  $\begin{pmatrix} 3 \\ -3 \end{pmatrix}$  verschobene Viereck graphisch dar.
- (c) Stellen Sie (in derselben Graphik), ausgehend vom gegebenen Viereck, das mit dem Vektor  $\begin{pmatrix} 3 \\ -3 \end{pmatrix}$  verschobene und dann mit 180 Grad um den Koordinatenursprung gedrehte Viereck in derselben Graphik dar.

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

$$(x_1, y_1) \rightarrow (x_2, y_2) \rightarrow \dots \rightarrow (x_n, y_n) .$$

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<sup>1</sup>.

16. Visualisieren Sie eine Drehgruppe vom Grad 7 analog zum Beispiel in der Wikipedia<sup>2</sup>. Die Art der Visualisierung steht Ihnen frei, Sie können auch ein kleines Movie in Betracht ziehen, siehe `getframe`.

17. Zwei Primzahlen  $p$  und  $q$  werden als Primzahlzwillinge<sup>3</sup> bezeichnet falls  $q = p + 2$  gilt.

- (a) Bestimmen Sie alle Primzahlzwillinge zwischen 1000 und  $10^7$  (FOR/WHILE-Loop).
- (b) Speichern Sie hierbei die gefundenen Werte für  $p$  in einem Vektor.
- (c) Visualisieren Sie die Häufigkeit der Primzahlzwillinge (in 1000er Intervallen). Hierfür ist `histogram(x,xbins)` empfehlenswert, Sie können aber auch selbst zählen und visualisieren.

Hinweise:

- Schauen Sie sich die Matlabfunktionen `primes` und `isprime` an.
  - Sie werden einen/mehrere FOR/WHILE-Loops zusammen mit der Alternatice (IF) benötigen.
- (\*) Man kann die Primzahlzwillinge auch mit reinen Vektoroperationen bestimmen.  
Vergleichen Sie die Laufzeiten bei deaktivierten Ausgaben für beide Versionen (`tic,toc` bzw. "Run and Time").

---

<sup>1</sup>[https://de.mathworks.com/help/matlab/ref/plus.html#expand\\_panel\\_body\\_bu90zu2](https://de.mathworks.com/help/matlab/ref/plus.html#expand_panel_body_bu90zu2)

<sup>2</sup>[http://de.wikipedia.org/wiki/Zyklische\\_Gruppe](http://de.wikipedia.org/wiki/Zyklische_Gruppe)

<sup>3</sup><https://de.wikipedia.org/wiki/Primzahlzwilling>

18. Laden Sie das Bildfile *ssrb.jpg* mittels der MatLab-Funktion `imread` in die dabei angelegte Intensitätsmatrix mit 3 Dimensionen [Pixel\_vertikal][Pixel\_horizontal][3]. Die 3 Komponenten der höchsten Dimension indizieren die RGB-Komponenten<sup>4</sup> (Red-Green-Blue) des Farbraumes für jedes Pixel des Bildes. Zum Beispiel stellt das Element `img(723,509,1)` die Rotintensität des Bildpixels  $(x,y) = (509, 723)$  dar, die Intensitäten sind natürliche Zahlen aus dem Intervall  $[0, 255]$ .

Fassen Sie die nachfolgenden Bilddarstellungen in einer Grafik zusammen (Matlab-Funktion `subplot`) und beschriften Sie diese hinreichend. Sämtliche Bildmanipulationen gehen immer wieder vom Originalbild aus.

- Zeichnen Sie das Originalbild mittels `imshow`.
- Zeichnen Sie das Bild kopfstehend.
- Halbieren Sie den Rotanteil des Bildes und zeichnen Sie es.
- Verdoppeln Sie den Grünanteil des Bildes und zeichnen Sie es.
- Zeichnen Sie den Blauanteil des Bildes separat (Wie separieren!?).
- Zeichnen Sie nur jede dritte Zeile und Spalte des Bildes.

Denkaufgabe, keine Pflicht! → Können Sie die Qualität dieser Bildverkleinerung verbessern, z.B. durch Mittelung der Farbwerte?

Speichern Sie die Gesamtgrafik als **\*.png**-File.

Hinweise: Operatoren für Matrizen („ \* “, „ \ “, „ , “, „ / “), elementweise Operatoren für Matrizen („ .\* “, „ ./ “, „ .^ “), `plot`, `histogram`, `bar`, `primes`, `isprime`, `factor`, `imread`, `imshow`, `subplot`, `figure`, `semilogy`, `NaN`, `Inf`, `plot`, `linspace`

Abgabe der Lösungen:

Die Abgabe der Lösungen (\*.m-Files und Grafiken) muß über Kreuzliste<sup>5</sup> erfolgen.

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

Abzugebende Files (auch als ein zip-File möglich):

`bsp_15.m`,    `bsp_15.jpg`  
`bsp_16.m`,    `bsp_16.jpg`  
`bsp_17.m`,    `bsp_17.jpg`  
`bsp_18.m`,    `bsp_18.png`

<sup>4</sup><http://de.wikipedia.org/wiki/RGB-Farbraum>

<sup>5</sup>[http://imsc.uni-graz.at/haasegu/Lectures/CompMath/Modus\\_WS\\_18.html](http://imsc.uni-graz.at/haasegu/Lectures/CompMath/Modus_WS_18.html)