ComputerMathematik WS 20/21
5. Computerpraktikum zum Abgabetermin 30.11.2020, 23:59

23. Nutzen Sie `cylinder` um die folgenden Aufgaben zu lösen:

    (a) Rotation der Kurve $r(x) = x \sin(\pi x)$, $x \in [0.05, 0.8]$ um die x-Achse.

    (b) Die Rotation der Kurve $r(h) = \sqrt{1 - h^2}$ für $h \in [-1, 0.5]$.

    (c) Kegelstumpf der Höhe 4 mit unterem Radius 3 und oberem Radius 1.

    Hinweis: Ist die Höhe in Richtung der Rotationsachse korrekt? Haben Sie `axis equal` eingeschaltet?

24. **Roulette-Simulation**:
    Schreiben Sie eine Matlabfunktion `[gewinn] = bsp_24_fkt(wahl,einsatz)`, die in Abhängigkeit der Eingabeparameter (`wahl,einsatz`) ein Roulettespiel simuliert.
    Folgende vereinfachte Regeln sind zu beachten:

    - Es besteht nur die Wahl zwischen `gerade/ungerade` oder einer Zahl zwischen 0 und 36.
    - Setzt der Spieler erfolgreich auf `gerade/ungerade` erhält er das Doppelte seines Einsatzes zurück. Fällt die Zahl 0, so erhält er die Hälfte seines Einsatzes zurück. Andernfalls verliert er seinen Einsatz.
    - Setzt der Spieler erfolgreich auf eine Zahl, so erhält er das 36-fache seines Einsatzes zurück. Andernfalls verliert er seinen Einsatz.

    Vergleichen Sie das Setzverhalten zweier Spieler A und B in einem scriptfile bsp_24.m.

    - Beide Spieler besitzen ein Guthaben von 50000 Euro.
    - Spieler A setzt immer einen Euro auf `gerade`.
    - Spieler B setzt immer einen Euro auf eine Zahl (z.B. 2).
    - Lassen Sie beide Spieler so lange spielen, bis die jeweiligen Guthaben 1 Euro unterschritten haben.
    - Wer macht mehr Spiele? Erklären Sie Ihre Beobachtung mathematisch.

    Visualisieren Sie den Verlauf der beiden Guthaben in einer Grafik.

    Hinweise: `randi`, `switch`, `case`, `otherwise`

25. **Koch-Schneeflocke**:

Schreiben Sie eine Matlabfunktion `[p] = bsp_25_fkt(n,A,B,C)` die in Abängigkeit eines vorgegebenen gleichseitigen Dreiecks mit den Punkten A,B,C die dazugehörige Koch-Schneeflocke (siehe Koch-Kurve[1]) berechnet.

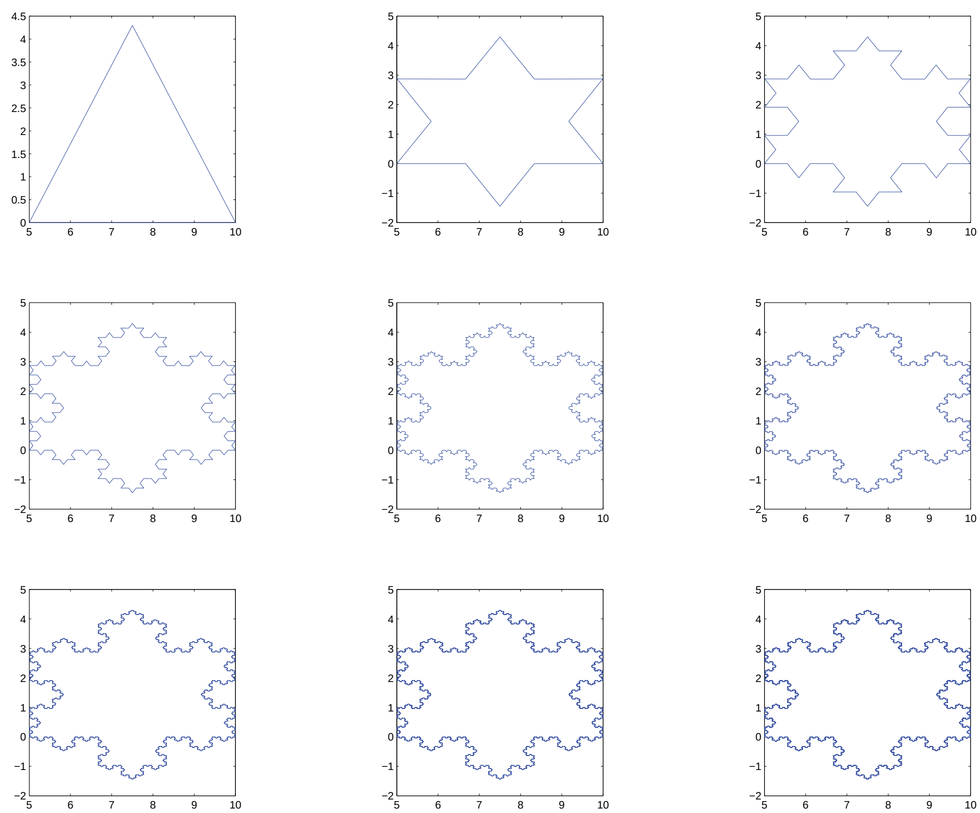
Folgende Hinweise sind zu beachten:

- Der Eingabeparameter n bezeichnet die Anzahl der auszuführenden Iterationen.
- Der Ausgabeparameter p soll die Punkte des entstandenen Polygons enthalten.
- Zeichnen Sie die ersten neun Iterierten in unterschiedliche Grafiken.
- Berechnen Sie den Flächeninhalt des jeweiligen Polygons.

Testen Sie Ihr Programm für die Punkte

$$A = (5, 0), B = (10, 0), C = (7.5, \sqrt{18.75})$$

in einem scriptfile bsp_25.m. Das Resultat sollte ähnlich zu nachfolgender Grafik sein.



Hinweise: `polyarea`, `subplot`, `axis square`

[1] `https://de.wikipedia.org/wiki/Koch-Kurve`

26. Benutzen Sie in a)-b) das symbolische Matlab-Paket (z.B., `solve`, `roots`,...) oder das numerische Paket (z.B., `vpasolve`, `fzero`, `roots`), sowie `int`, `diff` in d)-f) um folgende Aufgaben zu bearbeiten:

   (a) Lösen Sie $x^3 + 118 * x^2 - 2029 * x - 4522 = 0$ numerisch (`roots`; `fzero`) und symbolisch (`syms`, `solve`).
   (Tip: Plotten Sie die Funktion, um die Werte der Nullstellen abzuschätzen.)

   (b) Lösen Sie $x^2 = 10 * \sin(x) - 2$ sowohl symbolisch (`solve`) als auch numerisch (`vpasolve` oder `fzero`).
   Haben sie alle Lösungen ermittelt?

   (c) Lösen Sie folgendes Gleichungssystem mittels der symbolischen Funktion `solve` nach x, y auf:

$$\begin{aligned} 4x - 3y &= 12 \\ 3x + 4y &= -9 \end{aligned}$$

   Berechnen Sie ebenfalls die Lösung, wenn die erste Gleichung in $4x - 3y = 12 + c$ geändert wird.

   (d) Berechnen Sie $\displaystyle\int e^x (x^3 - 1) \cos(x) dx$.

   (e) Berechnen Sie $\displaystyle\int_{-7}^{3} e^x (x^3 - 1) \cos(x) dx$.

   (f) Berechnen Sie die erste und zweite Ableitung von

   - $f(x) = \dfrac{\ln(a - x)}{x}$ und von
   - $g(x) = \dfrac{x + x\cos(x)}{2e^x - \ln(x)}$:
   - $h(x) = \dfrac{x + x\tan(x)}{\sin(x) - 4\ln(x)}$: Visualisieren Sie $h''(x)$ für $x \in [0, 10]$.

   (g) Nutzen Sie `limit` und berechnen Sie $\lim_{x\to\infty} g''(x)$.

   (h) Berechnen Sie $\displaystyle\sum_{k=1}^{\infty} \frac{1}{k^2}$ mittels `symsum`.
   Kommen Sie mit einem FOR-Loop (bis zu großen k) auf denselben Wert?

   Hinweis: Mit dem Funktion `matlabFunction` können Sie einen symbolischen Ausdruck in eine Funktion umwandeln und dann, z.B., in `fzero` verwenden.

   Hinweis zu Ocatve: Um das symbolische Paket im Skript nutzen zu können ist

   - das symbolische Paket zu installieren[2] `pkg install -forge symbolic`
   - am Skriptanfang die Anweisung `pkg load symbolic` zu setzen.
   - Ein für Matlab und Octave geeignetes Skript sollte folgende Zeilen einbinden:

```
isOctave = exist('OCTAVE_VERSION', 'builtin') ~= 0;
if (isOctave)
   pkg load symbolic
end
```

   an dessen Anfang die Anweisung `pkg load symbolic` zu setzen.

   Hinweise: `syms`, `solve`, `roots`, `vpasolve`, `fsolve`, `int`, `diff`, `limit`, `sum`, `symsum`.

Abgabe der Lösungen:
Die Abgabe der Lösungen (`*.m`-Files und Grafiken) muß über Kreuzlliste[3] erfolgen.
Die Filenamen **müssen** dem Schema `bsp_nummer`, gefolgt von der Filextension, entsprechen. Andere

[2] `https://octave.sourceforge.io/packages.php`
[3] `http://imsc.uni-graz.at/haasegu/Lectures/CompMath/Modus_WS_18.html`

Filebezeichner zählen nicht als abgegebene Files.

Abzugebende Files (auch als ein zip-File möglich):
`bsp_23.m`, `bsp_23.jpg`
`bsp_24.m`, `bsp_24_fkt.m`, `bsp_24.jpg`
`bsp_25.m`, `bsp_25_fkt.m`, `bsp_25.jpg`
`bsp_26.m`