Übungsblatt Nr. 6: in der Übungsstunde

1) Setzen Sie `X = [ 0.0, 0.1, 0.2, ... , 1.9, 2.0 ]` und berechnen Sie Sie die Liste der Sinuswerte. Speichern Sie alle Zahlenpaare in der Datei `Sinus.txt`. Die Werte sollten genau untereinanderstehen!

2) Lesen Sie aus der Datei `Sinus.txt` wieder alle Werte in 2 Listen `x, y` ein. Geben Sie am Ende aus, wieviel Paare eingelesen wurden.

3) Numerisch differenzieren! Python kann (wie die meisten Programmiersprachen) eine Funktion nicht mathematisch korrekt differenzieren, sondern kann höchstens einen Näherungswert für die Ableitung berechnen. Als Ausgangspunkt dient die mathematische Definition der Ableitung:

$$f'(x_0) = \lim_{x->x_0} \frac{f(x) - f(x_0)}{x - x_0} = \text{ (setze } x = x_0 + h \text{) } \lim_{h->0} \frac{f(x_0 + h) - f(x_0)}{h}.$$

Man kann den Bruch ganz rechts als Näherungswert für die Ableitung $f'(x_0)$ nehmen, indem man ein ´genügend´ kleines $h > 0$ einsetzt. Man nennt diesen Differenzenquotienten Vorwärtsdifferenzenquotient, weil im Falle einer zeitabhängigen Funktion $x_0 + h$ ein Zeitpunkt in der Zukunft ist. Aber welcher konkrete Wert für `h` ist ´genügend´ klein?

   a) Nehmen Sie die Funktion $f(x) = e^x$ mit der Ableitung $f'(x) = e^x$ an der Stelle `x0 = 1`. Dann berechnen Sie den Wert des Vorwärtsdifferenzenquotienten für verschiedene Werte von `h`, die Sie in einer Schleife einlesen. Geben Sie das Resultat und den Fehler aus.

   b) Jetzt wollen wir den Fehler in Abhängigkeit von `h` plotten. Am besten sieht man den Zusammenhang in einem Loglogplot, d.h. wir plotten `log(Fehler)` über `log(h)`. Um direkt die dezimale Stellenanzahl zu sehen, nimmt am besten den Logarithmus zur Basis 10. Versuchen Sie das optimale `h` aus dem Plot zu ermitteln.

4) a) Parameterplot: Plotten Sie die in Polarkoordinaten gegebene Funktion

$$r(\phi) = 2(1 + \cos(\phi)), \ \phi \in [0, 2\pi].$$

   Die Polarkoordinaten sind mit den kartesischen Koordinaten verbunden durch $x = r\cos(\phi), \ y = r\sin(\phi)$.

   b) Plotten Sie die in kartesischen Koordinaten gegebene Kurve $(\cos(\phi)^3, \sin(\phi)^3)$.

Übungsblatt Nr. 6 Hausübung: Die folgenden Aufgaben sind Pflicht und zählen 1 Punkt!

1) a) Gehen Sie analog zum Demobeisiel 1) vor. Erzeugen Sie diesmal auch eine "leicht gestörte" Werteliste `z`, mit $z_i = y_i + \epsilon_i$, wobei die ϵ_i Stichproben einer Normalverteilung `gauss(0.0, 0.01)` sind. Speichern Sie die Tripel `x[i], y[i], z[i]` in der Datei `Tabelle.txt`. Lesen Sie die Werte wieder aus der Tabelle ein und plotten Sie `(x, y), (x, z)`.

2) a) Lesen Sie Werte für `n > 1`, `mu` und `sigma > 0` ein und schreiben Sie in die Datei `Normalverteilung.txt` `n` Stichproben einer `gauss(mu, sigma)`-Normalverteilung. Wenn $n \leq 1$ oder $\sigma \leq 0$ gilt, brechen Sie das Programm mit einer Fehlermeldung ab.

   b) Lesen Sie die Werte aus der Datei in eine Liste ein. Benutzen Sie das `statistics`-Modul, um den Mittelwert und Standardabweichung zu berechnen.

   c) Geben Sie auch den minimalen und den maximalen Wert aus.

   d) Berechnen Sie ohne externe Funktionen das arithmetische Mittel m_1 und das quadratische Mittel m_2 der Werte:

$$m_1 = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i, \qquad m_2 = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2}$$

3) a) Kompilieren das Programm `arithmetic1.cpp` aus der Vorlesung 1 (im Ordner Demo-Beispiele) und probieren Sie das Programm aus.

   b) Erweitern Sie das Programm analog um folgende Rechnungen:
   `2’000’000’000 + 200’000’000`
   `3’000’000’000 + 200’000’000`

4) a) Kompilieren das Programm `arithmetic2.cpp` aus der Vorlesung 1 (im Ordner Demo-Beispiele) und probieren Sie das Programm aus.

   b) Erweitern Sie das Programm analog um folgende Rechnungen:
   `2’000’000’000 + 200’000’000`
   `3’000’000’000 + 200’000’000`

Extra-Aufgabe(n): Diese Aufgaben sind freiwillig und zählen 2 Punkte!

5) a) Erstellen Sie analog zu Vorführbeispiel 3 einen Fehlerplot für den symmetrischen Differenzenquotienten SDQ als Approximation der ersten Ableitung, wobei Sie diesmal die Cosinusfunktion und $x_0 = 1$ verwenden:

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

Welches ist hier die beste Wahl für `h` und auf wie viele Dezimalstellen Genauigkeit kommen Sie?

b) Dieselbe Fragestellung, aber für die Approximation der 2. Ableitung durch den symmetrischen DQ2:

$$f''(x_0) \approx \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2}$$

6) a) Programmieren Sie mit Python die Funktion $f(x, y) = \dfrac{x + y}{x^2 + y^2}$.

b) Berechnen Sie die Liste L alle 10000 = 100*100 Funktionswerte $f(x, y)$ mit $x, y \in \{0.01, 0.02, 0.03, ..., 1.00\}$.

c) Wie oft kommt der Wert 4.0 in L vor?

d) Wieviele Werte in L sind kleiner als 1.2?

e) Berechnen Sie die Summe dieser Werte aus d).

Challenge-Aufgabe(n): Diese Aufgaben sind freiwillig, zählen 3-8 Punkte und sind bis zum angegeben Termin abzugeben!

7) Die Fibonacci-Zahlen sind die Glieder der Folge [0, 1, 1, 2, 3, 5, 8, ...], die mit 0 und 1 beginnt und in der jede weitere Zahl die Summe ihrer 2 Vorgänger ist. Berechnen Sie die Summe aller geraden Fibonaccizahlen, die nicht größer als 10, als 4 Millionen und als 4 Milliarden sind. Zur Kontrolle: die erste Summe ist 10, die zweite 4613732. Tipp: Berechnen Sie ausreichend viele Fibonacci-Zahlen in einer Liste, bevor Sie die Summen berechnen. (2 Punkte, Abgabe bis Sonntag 28.11.2021)