

Übungsserie 3

Abgabe: gemäss Angaben Dozent

Scannen Sie Ihre Lösung zu den Aufgaben 1 und 2 in die Dateien *Name_Vornamen_Gruppe_S3_Aufg1.pdf* und *Name_Vornamen_Gruppe_S3_Aufg2.pdf* und fassen Sie diese zusammen mit der Lösung *Name_Vorname_Gruppe_S3_Aufg3.m* in eine ZIP-Datei *Name_Vorname_Gruppe_S3.zip* zusammen. Laden Sie dieses File vor der nächsten Übungsstunde nächste Woche auf OLAT hoch. Das m-File muss ausführbar sein und in den Kommentarzeilen (beginnen mit %) soll ein Beispiel eines funktionierenden Aufrufs angegeben werden.

Aufgabe 1 (ca. 20 Min.):

Ist das Potenzieren ($f(x) = x^n$, $n \in \mathbb{N}$) bzw. das Wurzelziehen ($f(x) = x^{\frac{1}{n}}$, $n \in \mathbb{N}$) einer reellen Zahl x gut oder schlecht konditioniert? Begründen Sie! Was hat das für Auswirkungen auf die Auswertung von Polynomen für grosse n ?

Aufgabe 2 (ca. 20 Min.):

Betrachten Sie einen Rechner, der im Dezimalsystem arbeitet mit einer zehnstelligen Gleitpunktarithmetik (also $n = 10$ für die Mantisse) und einem beliebig grossen Exponenten. Erklären Sie anhand einer kurzen konkreten Berechnung, weshalb für eine positive Zahl $x \neq 0$, die kleiner als die Maschinengenauigkeit ϵ ist, der Rechner $1+x$ nicht mehr korrekt berechnen kann (bekanntlich wird er $1+x = 1$ ausgeben), wohingegen er keine Probleme hat, z.B. \sqrt{x} oder $x/10^9$ richtig zu berechnen.

Tipp: nehmen Sie für x eine konkrete Zahl an, berechnen Sie die obigen Grössen und normieren Sie sie wie in Kap. 2 des Skriptes.

Aufgabe 3 (ca. 120 Min.):

Schreiben Sie in MATLAB eine Funktion `[y,value,abs_err,rel_err] = Name_Vorname_Gruppe_S3_Aufg3(x,B,nmax)`, die eine beliebige Dezimalzahl x inklusive Nachkommastellen (Input) umrechnet in eine Basis B (Input, $1 < B < 10$) mit wählbarer Anzahl Nachkommastellen $nmax$ (Input). Zurückgegeben werden die Zahl y in der neuen Basis inklusive Vorzeichen und $nmax$ Nachkommastellen, der Wert (value) von y sowie der absolute und relative Fehler, der durch das allfällige Abschneiden passiert. Die Zahl y muss nicht normiert werden, also z.B. generiert der Aufruf `[y,value,abs_err,rel_err] = Serie3_Aufg3(1006.687,2,13)` das Resultat

```

y      = +1111101110.1010111111011
value  = 1.0067e + 03
abs_err = 1.1035e - 04
rel_err = 1.0962e - 07

```

Tipp: behandeln Sie y als String (char). Funktionen, die Ihnen helfen werden (eine genaue Beschreibung erhalten Sie, wenn Sie im Eingabefenster den Befehl `doc funktionsname` eingeben):

- `fix(x)`: gibt den ganzzahligen Anteil einer Zahl, z.B. `fix(5.98) = 5`

- `abs(x)`: berechnet den Absolutbetrag, z.B. `abs(-3.4) = 3.4`
- `sign(x)`: gibt das Vorzeichen, z.B. `sign(-2) = -1`
- `mod(x,n)`: Modulo, berechnet den Rest bei einer ganzzahligen Division x/n , z.B. `mod(5,3) = 2`
- `num2str(x)`: wandelt eine Zahl in einen String um
- `str2num(s)`: wandelt einen String in eine Zahl um