

Anmerkung: Beginnend mit diesem Blatt dürfen Sie die Techniken der *prozeduralen Programmierung* verwenden, sofern diese in der Vorlesung behandelt wurden. Verzichten Sie daher vor allem auf Rekursion und die `cond`-Funktion und lösen Sie die gestellten Aufgaben durch die Verwendung von Schleifen und bedingten Anweisungen.

ÜBUNG 4.1 ZAHLENDARSTELLUNG

a) Die Determinante einer 2×2 -Matrix A berechnet man durch

$$\det(A) = \det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc.$$

Schreiben Sie eine C++-Funktion

```
float determinante(float a, float b, float c, float d)
```

die die Determinante berechnet. Berechnen Sie damit die Determinante $\det(A)$ der Matrix

$$A = \begin{pmatrix} 100 & 0.01 \\ -0.01 & 100 \end{pmatrix}.$$

Die exakte Lösung ist 10000.0001. Warum stimmt das Ergebnis nicht? Was passiert, wenn Sie anstatt des Datentyps `float` den Typ `double` verwenden?

Bemerkung: Der C++-Compiler rundet bei der Ausgabe von `float` und `double`-Zahlen normalerweise automatisch, um das Ergebnis lesbarer zu machen. Die Ausgabefunktion `print` aus dem von uns bereitgestellten Header `fcpp.h` schaltet die Rundung ab und gibt alle Stellen aus.

b) Berechnen Sie mit einem C++-Programm $(a + b) + c$ und $a + (b + c)$ mit $a = 10^n$, $b = -10^n$ und $c = 10^{-n}$ und $n = 6, 7, 8, \dots, 14$. Für welches n ist die Addition auf Ihrem Computer nicht mehr assoziativ wenn Sie `float` verwenden? Warum passiert das?

5 Punkte

ÜBUNG 4.2 EFFEKTIVER ZINSSATZ

Sie sind der Programmierer des neuen Onlinesystems für kurzfristige Einlagen der Bank *Schnell & Reich*. Bei einer jährlichen Verzinsung von z und einer Abrechnung der Zinsen in Schritten von $1/n$ Jahren ergibt sich der *effektive* Zinssatz zu

$$z_{\text{eff}} = \left(1 + \frac{z}{n}\right)^n - 1$$

Der effektive Zinssatz steigt mit n , da bereits gewährte Zinsen aus vorherigen Abrechnungszeiträumen ebenfalls verzinst werden (sogenannter *Zinseszins*). Das Supremum entspricht einer Abrechnung in infinitesimal kleinen Schritten:

$$\lim_{n \rightarrow \infty} z_{\text{eff}} = \exp(z) - 1$$

Schreiben Sie ein Programm zur Berechnung des effektiven Zinssatzes, wobei Sie einmal den Datentyp `float` und dann `double` zur Berechnung verwenden. Schreiben Sie dazu Funktionen

```
float zins(float z, int n) und double zins(double z, int n).
```

Verwenden Sie $z = 0.06$ und lassen Sie n folgende Werte durchlaufen:

1	(jährliche Abrechnung)	365 * 24	(stündlich)
4	(quartalsweise)	365 * 24 * 60	(minütlich)
12	(monatlich)	365 * 24 * 60 * 2	(alle 30 Sekunden)
365	(täglich)	365 * 24 * 60 * 60	(sekündlich)

Erzeugen Sie für jedes n eine Ausgabe, die folgende Informationen enthält:

- Anzahl der Abrechnungsvorgänge n
- Zugehöriger Zinssatz unter Verwendung von `float`
- Differenz zu $\exp(z) - 1$ für `float`
- Zugehöriger Zinssatz unter Verwendung von `double`
- Differenz zu $\exp(z) - 1$ für `double`

Binden Sie den Header `fcpp.hh` ein. Dieser bindet weitere Header ein, so dass Ihnen Funktionen wie `exp()` zur Verfügung stehen. Verwenden Sie außerdem für Ein- und Ausgabe die dort definierten Funktionen.

Welche Ergebnisse wären bei exakter Rechnung zu erwarten, und in welchen Punkten weichen die von Ihnen erhaltenen davon ab? Können Sie mögliche Ursachen angeben?

5 Punkte

ÜBUNG 4.3 MULTIPLIKATION IM ZWEIERKOMPLEMENT

Zeigen Sie, dass für $n \in \mathbb{N}$ unter der Voraussetzung

$$a, b, a \cdot b \in [-2^{n-1}, \dots, 2^{n-1} - 1]$$

gilt, dass die Multiplikation im Zweierkomplement sich folgendermaßen darstellen lässt:

$$d_n(a \cdot b) = s_n(d_n(a) \cdot d_n(b))$$

Dabei bezeichnen d_n und s_n die Zweierkomplementdarstellungs- und Abschneidefunktion aus der Vorlesung. *Hinweis:* Rekapitulieren Sie das vergleichbare Resultat zur Addition aus der Vorlesung.

5 Punkte

ÜBUNG 4.4 METHODE DER INTERVALLHALBIERUNG ZUR NULLSTELLENSUCHE

Aus der Vorlesung kennen Sie das Newtonverfahren zur Bestimmung von Quadratwurzeln. Eine andere Methode ergibt sich aus der Methode der Intervallhalbierung. Diese Methode wird eigentlich zur Bestimmung einer Nullstelle einer stetigen Funktion $f(x)$ verwendet. Wenn man aber $f(x) = x^2 - w$ ansetzt, erhält man dadurch \sqrt{w} .

Die grundlegende Idee der Intervallhalbierung ist: Wenn a und b gegeben sind und $f(a) < 0 < f(b)$ gilt, dann muß mindestens eine Nullstelle zwischen a und b liegen. Um die Nullstelle zu finden, berechnet man den Mittelwert m von a und b . Wenn $f(m) > 0$ gilt, so muß f eine Nullstelle zwischen a und m haben. Wenn $f(m) < 0$ gilt, so muß f eine Nullstelle zwischen m und b haben. Durch wiederholtes Anwenden wird das Intervall, in dem der gesuchte Wert liegt, immer kleiner. Implementieren Sie dieses Verfahren analog zu dem Programm `newton.cc` aus der Vorlesung, d.h. schreiben Sie Funktionen analog zu `double wurzelIter (double xn, double a)` und `double wurzel (double a)`.

Bestimmen Sie experimentell, wie viele Iterationen die Intervallhalbierung braucht, um die Werte für $\sqrt{2}$, $\sqrt{3}$ und $\sqrt{4}$ bis auf eine Genauigkeit von 10^{-12} zu berechnen. Beginnen Sie mit den Startwerten 0 und 4. Vergleichen Sie das mit dem Newtonverfahren.

5 Punkte