Institut für Numerische und Angewandte Mathematik
FB Mathematik und Informatik der Universität Münster
Prof. Dr. Mario Ohlberger
Tobias Leibner

17. September 2019

Übung zum Kompaktkurs

Einführung in die Programmierung mit C++

Sommersemesterferien 2019 — Blatt 2

Aufgabe 1 (Ableitung)

Schreiben Sie ein Programm `diffquot`, welches die Ableitung der Funktion $f(x) = x^2$ mit Hilfe des Vorwärtsdifferenzenquotienten

$$f_h^+(x) := \frac{f(x+h) - f(x)}{h}$$

berechnet. Da $f_h^+(x) = f'(x) + \mathcal{O}(h)$ lässt sich der Vorwärtsdifferenzenquotient als Approximation für $f'(x)$ für kleines $h \in \mathbb{R}_+$ verwenden.
Gehen Sie folgendermaßen vor:

(a) Schreiben Sie eine Funktion namens `evaluate`, welche die Funktionswerte für $f(x) = x^2$ berechnet.

(b) Schreiben Sie eine weitere Funktion `differentiate`, welche den Differenzenquotienten für festes $h > 0$ berechnet (unter Nutzung von `evaluate`).

(c) Schreiben Sie ein Haupt-Programm, das als Kommandozeilen-Parameter die Intervallgrenzen a und b und die Anzahl N von Gitterpunkten übergeben bekommt. Die Schrittweite h wird auf $h = \frac{b-a}{N}$ gesetzt. D.h.

$$a = x_0 < x_1 < \ldots < x_{N-1} < x_N = b, \quad \text{wobei } x_n = a + n \cdot h.$$

Das Programm soll die Ableitung der Funktion in den Punkten x_n mittels obigem Differenzenquotienten approximieren und diese zusammen mit den x_n zeilenweise in eine Textdatei `diffquot.txt` schreiben.

(d) Das Programm soll außerdem eine Textdatei `func.txt` schreiben, welche die entsprechenden Werte der Funktion selber enthält. Plotten Sie die Ableitung sowie die Originalfunktion auf dem Intervall $[0, 3]$ mit $N = 300$ Unterteilungen mit Hilfe von `gnuplot`.

(e) Tauschen sie $f(x) = x^2$ durch $f(x) = x^3 + x^2 + 2x + 1$ aus, in dem sie nur die `evaluate`-Methode verändern.

Aufgabe 2 (Ein einfacher ODE-Löser)

Schreiben Sie ein Programm `euler`, welches die Anfangswertaufgabe

$$y'(x) = f(x, y(x)), \quad y(0) = y_0 \tag{1}$$

mit $f(x,y) = \frac{\sqrt{1-y(x)^2}}{1+x^2}$ und $y_0 = 0$ im Intervall $[0,1]$ mit dem Eulerverfahren löst.
Gehen Sie folgendermaßen vor:

(a) Schreiben Sie eine Funktion namens `evaluate`, welche die Funktionswerte für

$$f(x,y) = \frac{\sqrt{1-y^2}}{1+x^2}$$

berechnet.

(b) Schreiben Sie eine weitere Funktion `euler`, welche einen Schritt des Euler-Verfahrens durchführt (siehe unten).

(c) Schreiben Sie ein Haupt-Programm, das als Kommandozeilen-Parameter die Intervallgrenzen a und b, den Anfangswert y_0 und die Anzahl der Gitterpunkte N. Das Programm soll die Anfangswertaufgabe mit dem Eulerverfahren lösen und die Textdatei `euler.txt` erzeugen, die zeilenweise die Lösung $y_n \approx y(x_n)$ zu den jeweiligen Werten x_n (siehe Aufgabe 1) enthält.

(d) Plotten Sie die Lösung zu einem Startwert $y_0 \in (-1, 1)$ auf dem Intervall $[0, 1]$ mit $N = 100$ mit Hilfe von `gnuplot`.

(e) Ändern Sie die `evaluate`-Methode, um das Problem

$$y'(x) = \sqrt{1 - y(x)^2}, \quad y(0) = 0$$

im Intervall $[0, 1]$ zu lösen. Vergleichen Sie das Ergebnis mit der exakten Lösung $y(x) = \sin(x)$.

(f) (Bonusaufgabe) Implementieren Sie den zweistufigen Adams-Bashforth Löser (Beachten Sie das Sie für dieses Verfahren 2.ter Ordnung auch einen Startwert entsprechender Ordnung benötigen).

Bemerkung 1

Das allgemeine Eulerverfahren wird durch

$$\begin{aligned} y_0 &:= y(0), \\ y_{n+1} &= y_n + h \cdot f(x_n, y_n) \qquad \text{für alle } 0 \le n < N \end{aligned}$$

realisiert. In unserem Fall ist $f(x, \bar{y}) = \frac{\sqrt{1-\bar{y}^2}}{1+x^2}$ bzw. $f(x, \bar{y}) = \sqrt{1 - \bar{y}^2}$.

Aufgabe 3 (ODE-Löser mit Hilfsfunktionen)

Schreiben Sie ein Programm, welches die Anfangswertaufgabe (1) mit f wie in $1a$) und $y_0 = 0$ auf dem Interval $[0, 1]$ mit dem Runge-Kutta-Verfahren vierter Ordnung löst.
Zur Erinnerung: Ein Schritt des Runge-Kutta-Verfahrens vierter Ordnung wird durch

$$y_{n+1} = y_n + \frac{h}{6}(f_1 + 2 \cdot f_2 + 2 \cdot f_3 + f_4)$$

realisiert, wobei

$$\begin{aligned} f_1 &= f(x_n, y_n), \\ f_2 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2} f_1), \\ f_3 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2} f_2), \\ f_4 &= f(x_n + h, y_n + h \cdot f_3). \end{aligned}$$

Erweitern Sie dazu das Programm aus Aufgabe 2:

(a) Ändern Sie entsprechend die Funktion `evaluate`.

(b) Schreiben Sie eine weitere Funktion `rungekutta`, welche einen Schritt des Runge-Kutta-Verfahrens durchführt.

(c) Schreiben Sie die mit dem Runge-Kutta-Verfahren berechnete Lösung in eine Datei `rk.txt` und vergleichen Sie diese Lösung mit derjenigen, die sie mit Hilfe des Euler-Verfahrens berechnet haben (z.B. für $N = 1000$).

Aufgabe 4 ((Bonusaufgabe) Komponenten eines Dateipfades)

(a) Schreiben Sie ein Programm, das bei einem via Kommandozeilenargument gegebenen Pfad wie z.B.:

`/this/is/my/special/path/to/a/file/myfile.txt`

die Komponenten

- Pfad (hier: `/this/is/my/special/path/to/a/file/`),
- Dateiname (hier: `myfile`) und
- Erweiterung (hier: `txt`)

extrahiert und ausgibt.

(b) Beachten Sie, dass nicht jeder Pfad mit `/` beginnt und auch nicht jede Datei eine Endung hat.

(c) Falls kein Kommandozeilenargument vorhanden ist soll eine interaktive Abfrage erfolgen.

(d) Welche Signatur (Rückgabewert und Argumente) hätten Funktionen, die diese Komponenten zur Verfügung stellen?

(e) Welche Funktionalität aus der Standardbibliothek kann hier hilfreich sein?