

Praktikum:

## Einführung in die Programmierung zur Numerik mit C++

Mittwoch, 16.03.2011

---

### Aufgabe 1 (ODE-Löser mit Hilfsfunktionen)

Schreiben Sie ein Programm `rk.cc`, welches die Anfangswertaufgabe

$$y'(x) = f(x, y(x)) = 1 + y^2(x), \quad y(0) = 0$$

im Intervall  $[0,1]$  mit dem Runge-Kutta-Verfahren vierter Ordnung löst. Testen sie Ihr Programm mit der Schrittweite  $h=0.01$  und visualisieren Sie Ihre Lösung.

Zur Erinnerung: Das Runge-Kutta-Verfahren 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

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

Gehen Sie hierbei wie bei Blatt 2, Aufgabe 2 vor:

1. Schreiben Sie eine Funktion `names evaluate`, welche die Funktionswerte für

$$f(x, y) = 1 + y^2$$

berechnet.

2. Schreiben Sie eine weitere Funktion `rungekutta`, welche den Runge-Kutta-Schritt

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

durchführt.

3. Dies sind die einzigen Änderungen im Vergleich zu Blatt 2, Aufgabe 2. Deshalb lässt sich das alte Programm leicht ergänzen, in dem man einfach die zusätzliche Methode `rungekutta` hinzufügt, `evaluate` ändert und sonst alles beim Alten lässt.
4. Schauen Sie sich die Lösung auf dem Intervall  $[0, 1]$  mit  $N = 100$  (d.h. Schrittweite  $h = 0.01$ ) mit Hilfe von `gnuplot` an.

5. Lösen Sie das Problem

$$y'(x) = \frac{\sqrt{1 - y(x)^2}}{1 + x^2}, \quad y(0) = y_0 = 0$$

in  $[0, 1]$  und vergleichen Sie die Lösung mit der aus dem Euler-Verfahren.

### Aufgabe 2 (Klassen)

Gegeben sind die Header-Datei `matrix.h` und das Programm `matrix.cc` auf der Webseite der Veranstaltung. Hier wird eine `Matrix`-Klasse mit verschiedenene Funktionen implementiert. Ergänzen Sie die fehlenden Implementierungen für Konstruktoren und Methoden in der `matrix.cc`. Insbesondere:

1. Ergänzen Sie den Konstruktor `Matrix( int m, int n )`.
2. Fügen Sie der Klasse Funktionen zum Abfragen und Ändern von Matrixeinträgen hinzu.
3. Fügen Sie der Klasse Funktionen zur Addition und Multiplikation von Matrizen hinzu.
4. Fügen Sie der Klasse eine Funktion namens `skaliere` hinzu, welche ein Objekt der Matrix-Klasse mit einem `double` multipliziert.
5. Fügen Sie der Klasse eine Funktion namens `transponiere` hinzu, welche die Matrix transponiert.
6. Testen Sie Ihr Programm, indem sie das 9-fache von  $\begin{pmatrix} 1 & 4 & 7 & 4 \\ 2 & 6 & 7 & 3 \\ 8 & 3 & 5 & 1 \\ 4 & 8 & 7 & 9 \end{pmatrix}$  mit der Transponierten von  $\begin{pmatrix} 3 & 1 & 7 & 3 \\ 8 & 3 & 4 & 5 \end{pmatrix}$  multiplizieren.