Interdisziplinäres Zentrum
für Wissenschaftliches Rechnen (IWR)
Ruprecht–Karls–Universität Heidelberg

Dr. Peter Zaspel

# Übungsblatt 2

26.04.2016 – 10.05.2016

## Einführung in die Numerik SS 2016

**Aufgabe 1.** Lagrange-Darstellung (3+(1+2+2) Punkte)

**a.** Zeigen Sie, dass zu paarweise verschiedenen Knoten $x_i$, $i = 0, \ldots, n$, die Lagrange'schen Basis-Polynome $\{L_i(x),\, i = 0, \ldots, n\}$ eine Basis des Polynomraums $P_n$ bilden.

**b.** Zeigen Sie, dass für

$$\mathcal{L}_k(x) := \sum_{i=0}^{n} x_i^k L_i(x)\,, \quad k \in \mathbb{N}_0$$

gilt, dass

(i) $\mathcal{L}_0 \equiv 1\,,$

(ii) $\mathcal{L}_k(0) = 0\,, \quad k = 1, \ldots, n\,,$

(iii) $\mathcal{L}_{n+1}(0) = \, (-1)^n \prod_{i=0}^{n} x_i\,.$

**Aufgabe 2.** Dividierte Differenzen (6 Punkte)
Zeigen Sie: Die dividierte Differenz $f\,[x_0, \ldots, x_k]$ ist eine symmetrische Funktion der $x_i$, d.h. für eine Permutation $x_{i_0}, \ldots, x_{i_k}$ der Zahlen $x_0, \ldots, x_k$ gilt

$$f\,[x_{i_0}, \ldots, x_{i_k}] = f\,[x_0, \ldots, x_k]\,.$$

**Hinweis:** $f\,[x_0, \ldots, x_k]$ ist der Koeffizient der höchsten $x$-Potenz des interpolierenden Polynoms $P_{0,\ldots,k}(x)$ durch die Stützstellen $x_0, \ldots, x_k$.

**Aufgabe 3.** Neville-Schema (3 Punkte)
Sei $P \in \Pi_3$ das Interpolationspolynom mit $P(x_i) = f_i$ für

| $x_i$ | $-1$ | $0$ | $1$ | $3$ |
|---|---|---|---|---|
| $f_i$ | $5$ | $-6$ | $-9$ | $33$ |

Berechnen Sie mit Hilfe des Neville-Schemas den Wert $P(2)$.

**Aufgabe 4.** Newton Interpolation (2+1 Punkte)

**a.** Interpolieren Sie die Funktion $f(t) = \sqrt{t}$ mit Hilfe des Newton'schen Interpolationspolynoms vom Grad 2 ($p_2 \in \mathcal{P}_2$) zwischen den Stützstellen $t_0 = \frac{1}{4}$, $t_1 = 1$, und $t_2 = 4$.

**b.** Skizzieren Sie die Graphen von $f$ und $p_2$ (per Hand oder mit Gnuplot).

**Aufgabe 5.** Praktische Aufgabe (2+3+3+2 Punkte)
In dieser Aufgabe werden Sie die Interpolation von Funktionen mit Hilfe der Newton-Polynome schrittweise programmieren. Als Basis für diese Aufgabe können Sie das Programm-Gerüst in der Datei `newton.cpp` benutzen, die ebenfalls im Moodle-Kurs veröffentlicht wird. Bitte beachten Sie, dass der Befehl zum Kompilieren / Übersetzen dieses Quellcodes

```
g++ -o newton newton.cpp -lm
```

lautet, d.h. der C++ - Kompiler zum Einsatz kommt. Unabhängig davon sollten alle Inhalte aus dem Programmier-Crashkurs direkt anwendbar sein.

**a.** Ergänzen Sie die vorgegebene Funktion `void writeDataIntoFile(double* x, double* y, int count, char* filename)`, so dass diese die `count` vielen Aszissen in Array `x` und die Funktionswerte `y` als Zweiertupel zeilenweise in eine Datei mit Name `filename` schreibt. Die Ausgabe soll z.B. so aussehen:

```
1.00000000 1.00000000
2.00000000 4.00000000
3.00000000 9.00000000
4.00000000 16.00000000
...
```

**b.** Vervollständigen Sie die Funktion `void generateNewtonCoeffs(double* c, double* x, double* y, int n)`, die genau die Koeffizienten des Newton-Polynoms in Array `c` durch die $n + 1$ Stützstellen `x`, `y` (natürlich jeweils als Arrays übergeben) liefert. Beachten Sie, dass im gegebenen Programmgerüst bereits ein zweidimensionales Array `table` als Berechnungsbasis bereitgestellt wird. Dessen Einträge $i, j$ können über `table[i][j]` angesprochen werden kann.

**c.** Vervollständigen Sie die Routine `void evaluateNewton(double* y_eval, double* x_eval, double* c, double* x, int n, int eval_count)`, die das Newton-Polynom, das durch die $n + 1$ Koeffizienten in Array `c` und die $n + 1$ Stützstellen in Array `x` gegeben ist, in den `eval_count` Punkten `x_eval` auswertet. Die berechneten Funktionswerte sollen im Array `y_eval` ausgegeben werden. (`c` und `x` haben also die Länge $n + 1$ und `x_eval` und `y_eval` haben die Länge `eval_count`.)

**d.** Verwenden Sie das vorhandene Programmgerüst zur Interpolation der folgenden Funktionen:

(a) $[a, b] = [0, 2\pi]$, $f(x) = \sin(x)$
(b) $[a, b] = [0, 2]$, $f(x) = \exp(x)$

(c) $[a,b] = [-1,1]$, $f(x) = \frac{1}{1+(5x)^2}$

(Diese Funktionalität wird also bereits im Programmgerüst umgesetzt.) Plotten Sie die Funktionen und die Interpolationspolynome jeweils für den Grad $n$ des Interpolationspolynoms ($n \in \{4, 10, 19\}$) mittels Gnuplot. Verwenden Sie für die Auswertung 100 äquidistante Auswertestellen im Intervall $[a,b]$.

**Abgabe: 10.05.2016, 14:00-14:15 Uhr. (Mappen in der Vorlesung)**