

# Einführung in die Numerik

## 1. Programmierübung



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Fachbereich Mathematik  
Prof. Dr. Herbert Egger, Dr. Mirjam Walloth, Thomas Kugler

WS 2015/16  
3.12.2015

### Programmierübung

**Aufgabe P1** (Auswertung des Newtonpolynoms)

**eval\_newton\_poly.m:** Schreiben Sie eine Funktion

```
function r=eval_newton_poly(ci,xi,x)
```

zur Auswertung des Interpolationspolynoms  $p(x) = c_0 + c_1(x - x_0) + \dots + c_n(x - x_0) \cdots (x - x_{n-1})$  in Newton-Darstellung an den Stellen des Vektors  $x$ .

**testp1.m:** Überprüfen Sie Ihr Programm mit den Eingaben

- (a) `ci=[1]; xi=[0]; x=[0:0.1:5];`
- (b) `ci=[1,2]; xi=[0,1]; x=[0:0.1:5];`
- (c) `ci=[1,3,4]; xi=[0,1,2]; x=[0:0.1:5];`

indem Sie jeweils  $p(x)$  und das Resultat `r=eval_newton_poly(ci,xi,x)` plotten. Welche Komplexität besitzt der Algorithmus (Kommentar).

**Aufgabe P2** (Aufstellen des Newtonpolynoms)

**compute\_newton\_poly.m:** Implementieren Sie eine Funktion

```
function [ci,xi]=compute_newton_poly(xi,yi)
```

zum Berechnen der Koeffizienten des Interpolationspolynoms in Newtonform. Verwenden Sie hierzu das Schema der dividierten Differenzen. Welche Komplexität besitzt der Algorithmus (Kommentar).

**testp2.m:** Überprüfen Sie Ihr Programm mit den Eingaben

- (a) `xi=[0]; yi=[1];`
- (b) `xi=[0,1]; yi=[1,2];`
- (c) `xi=[0,1,2]; yi=[1,2,1];`

und vergleichen Sie mit der Lösung von Hand (Kommentar).

**Aufgabe P3** (Konvergenzbetrachtung)

**testp3.m:** (a) Stellen Sie mit Hilfe obiger Programme die Interpolationspolynome  $p_n$ ,  $n = 1$ , für die Funktion  $f(x) = e^x$  zu den Stützstellen  $x_i = ih$ ,  $0 \leq i \leq n$  mit  $h = 2^{-k}$ ,  $k = 0 : 10$  auf.

(b) Berechnen Sie jeweils Näherungen für die Interpolationsfehler  $e(h) = \max_{x \in [0, x_n]} |f(x) - p_n(x)|$ .  
Bemerkung: Es genügt, den Fehler auf einem feineren Gitter `x=[0:h/10:n*h]` zu betrachten!

(c) Tragen Sie die Fehler in einer Tabelle (je eine Zeile für  $h$ ) auf. Fügen Sie auch die estimated order of convergence (eoc) an und vergleichen Sie mit den theoretischen Konvergenzaussagen (Kommentar!)

(d) Wiederholen Sie die Experimente mit Polynomordnung  $n = 2, \dots, 5$ . Erklären Sie die Ergebnisse (Kommentar!)

(e) Machen Sie einen Konvergenzplot ( $h$  vs  $e(h, n)$ ) in logarithmischer Skala und tragen Sie die Ergebnisse der vorhergehenden Experimente ein. Tragen Sie zum Vergleich auch "Kurven" für  $h^k$  ein. Erstellen Sie eine Legende. Aus der Grafik sollten am Ende die Konvergenzraten leicht ersichtlich sein!

---

#### Aufgabe P4 (Basiswechsel)

**newton\_to\_monomial.m:** Schreiben Sie eine Funktion

```
function a = newton_to_monomial(ci,xi)
```

welche die Koeffizienten der Newton-Darstellung in die der Monomdarstellung umrechnet.

**testp4.m:** Überprüfen Sie Ihr Programm anhand der Beispiele aus Aufgabe 1. Rechnen Sie hierzu die Monomdarstellung per Hand aus.

#### Aufgabe P5 (Polynomauswertung)

**eval\_monom\_poly.m:** Schreiben Sie eine Funktion

```
function r=eval_monom_poly(ai,x)
```

zur Auswertung eines Polynoms  $p(x) = a_0 + a_1x + \dots + a_nx^n$  in Monomdarstellung an den Stellen des Vektors  $x$  mit dem Horner Schema. Welche Komplexität besitzt der Algorithmus (Kommentar!)

**testp5.m:** Überprüfen Sie Ihr Programm mit den Polynomen aus der vorhergehenden Aufgabe. Vergleichen Sie mit den grafischen Ausgaben aus Aufgabe 1.

#### Aufgabe P6 (Lagrange-Polynome)

**testp6.m:** Plotten Sie die sechs Lagrange-Polynome (Fundamentalpolynome) zu den Stützstellen  $xi=[0:5]$ . Verwenden Sie hierzu die Funktionen aus Aufgabe 1 und 2. Überprüfen Sie anhand der Grafik die Richtigkeit Ihrer Ergebnisse.

#### Organisatorische Hinweise:

- a) Legen Sie ein Verzeichnis pp1 an und speichern Sie alle Funktionen und Skripte für diese Übung darin ab. Der Inhalt des Verzeichnisses sollte hier also sein:

```
testp1.m    eval_newton_poly.m
testp2.m    compute_newton_poly.m
testp3.m
testp4.m    newton_to_monomial.m
testp5.m    horner.m
testp6.m
```

- b) Zippen Sie das Verzeichnis; in Linux: `zip -r pp1.zip pp1` von außerhalb des Verzeichnisses. In Windows kann WinZip oder ein ähnliches Tool verwendet werden.
- c) Überprüfen Sie, dass alle Skripte und Funktionen lauffähig sind!
- d) Kommentieren Sie die Funktionen geeignet. Insbesondere sollten Eingabe und Ausgabe Parameter beschrieben werden. `help <funktionsname>` sollte Information über die Routinen liefern.
- e) Antworten auf die Fragen sollten in den `testp*.m` files als Kommentare hinterlegt werden.