Prof. Dr. Karsten Urban
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

SoSe 2014
Blatt 5
27.06.2014

Angewandte Numerik 1

Abgabetermin: Freitag, 11.07.2014, **vor der Übung**

Dieses Übungsblatt hat eine Bearbeitungszeit von zwei Wochen. Beachten Sie, dass nicht alle Aufgaben mit dem Stoff der Vorlesung vom 26.06.2014 bearbeitet werden können.

Für dieses Übungsblatt gibt es 27 Theorie- und 24 Matlab-Punkte, sowie 7 Theorie- und 5 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 50-Prozent-Grenzen liegen aktuell (inklusive Blatt 5) bei 59 Theoriepunkten und 57,5 Matlabpunkten.

Aufgabe 32 *(Lagrange-Basispolynome)* (3T+3T+3T+4T* Punkte)

Seien $x_0, ..., x_n \in \mathbb{R}$ paarweise verschiedene Knoten. Die Lagrange-Basispolynome von Grad n sind definiert durch

$$L_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \qquad i = 0, \ldots, n.$$

Weiter sei auf $\mathbb{P}_n$, dem Raum der Polynome vom Grad kleiner oder gleich n, die folgende Abbildung definiert:

$$(p, q) := \sum_{k=0}^{n} p(x_k)q(x_k).$$

a) Zeigen Sie, dass $L_i \in \mathbb{P}_n$ für alle $i = 0, \ldots, n$.

b) Zeigen Sie, dass $(\cdot, \cdot)$ auf $\mathbb{P}_n$ ein Skalarprodukt bildet.

c) Zeigen Sie, dass $\{L_i \,:\, i = 0, \ldots, n\}$ eine Orthonormalbasis von $\mathbb{P}_n$ bzüglich $(\cdot, \cdot)$ bildet.

d) Seien nun die Stützstellen äquidistant verteilt, d. h. $x_k = x_0 + kh$ mit $h > 0$ für alle $k = 0, \ldots, n$. Dann lässt sich x darstellen durch $x = x_0 + th$ mit einem $t \in \mathbb{R}$. Zeigen Sie

$$L_i(x) = \frac{1}{i!} \frac{(-1)^{n-i}}{(n-i)!} \prod_{\substack{j=0 \\ j\neq i}}^{n} (t - j).$$

Aufgabe 33 *(Lagrange, Newton, Aitken-Neville)* (4T+4T+4T+2T+2T*+1T* Punkte)

Sie wollen $\sqrt{3}$ näherungsweise berechnen. Dazu haben Sie die folgende Idee: Sie wählen die Funktion $f(x) = 3^x$ und interpolieren diese Funktion an den Stützstellen

x_0	x_1	x_2	x_3
-2	-1	0	1

durch ein Interpolationspolynom P. Dieses Interpolationspolynom wollen Sie an der Stelle $x = \frac{1}{2}$ auswerten.

a) Stellen Sie das Interpolationspolynom P mit Hilfe der Lagrange'schen Interpolationsformel explizit dar und werten Sie anschließend das Polynom P an der Stelle $x = \frac{1}{2}$ aus.

b) Stellen Sie P als Newton'sches Interpolationspolynom mit Hilfe der Methode der dividierten Differenzen explizit dar und werten Sie anschließend das Polynom an der Stelle $x = \frac{1}{2}$ aus.

c) Berechnen Sie den Wert an der Stelle $x = \frac{1}{2}$ des Interpolationspolynoms P mit Hilfe des Aitken-Neville-Schemas.

d) Nun stellen Sie fest, dass Ihnen die Genauigkeit der näherungsweisen Berechnung von $\sqrt{3}$ noch nicht genügt. Daher nehmen Sie eine weitere Stützstelle $x_4 = 2$ hinzu. Erweitern Sie Ihre Berechnung mit dem Aitken-Neville-Schema um diese zusätzliche Stützstelle.

e) Erweitern Sie auch Ihre Berechnung mit den dividierten Differenzen um die zusätzliche Stützstelle x_4.

f) Was müssten Sie tun, wenn Sie auch Ihre Berechnung mit der Lagrange'schen Interpolationsformel um x_4 erweitern wollten?

Aufgabe 34 *(Programmieraufgabe: Horner-Schema)* (3M*+2M* Punkte)

Polynome $p(x) = \sum_{k=0}^{n} a_k x^k = a_0 + a_1 x + \cdots + a_n x^n$ (mit $x^0 := 1$) können mit Hilfe des Horner-Schemas effizient ausgewertet werden. Dazu wird das Polynom $p(x) = \sum_{k=0}^{n} a_k x^k$ umgeformt zu

$$p(x) = \sum_{k=0}^{n} a_k x^k = (\ldots((x a_n + a_{n-1})x + a_{n-2})x + \ldots a_1)x + a_0.$$

a) Schreiben Sie eine Matlab-Funktion `hornerSchema(a,x)` zur effizienten Auswertung des Polynoms $p(x) = \sum_{k=0}^{n} a_k x^k$ mit den Koeffizienten $a = (a_0, a_1, a_2, \ldots, a_n)^T \in \mathbb{R}^{n+1}$ an einer Stelle $x \in \mathbb{R}$.

b) Vergleichen Sie Ihre Funktion mit der Matlab-Funktion `polyval`. Diese Funktion verwendet auch das Hornerschema. Schreiben Sie hierzu ein Matlab-Skript `testHornerSchema`, das Ihre Funktion `hornerSchema(a,x)` und die Matlab-Funktion `polyval` für verschiedene Polynome aufruft und die Ergebnisse vergleicht.
Tipp: Die Matlab-Funktion `flipud` könnte Ihnen beim Aufruf der beiden Funktionen helfen.

Aufgabe 35 *(Programmieraufgabe: Schema der Dividierten Differenzen)* (4M+4M+2M+4M+2M Punkte)

a) Schreiben Sie eine Matlab-Funktion `c = divDiff(x,f)`, die mittels dividierter Differenzen den Vektor $c = (c_0, \ldots, c_n)^T$ der Koeffizienten des Newton-Interpolationspolynoms zu den Daten (x_j, f_j), $j = 0, \ldots, n$, berechnet. $x = (x_0, \ldots, x_n)^T$ sind dabei die Stützstellen, $f = (f_0, \ldots, f_n)^T$ die Funktionswerte an diesen Stützstellen.

b) Schreiben Sie eine Matlab-Funktion `v = evalNewtonpolynom(x,d,t)`, die das Newton-Polynom zu den Stützstellen $x = (x_0, \ldots, x_n)^T$ und den Koeffizienten $c = (c_0, \ldots, c_n)^T$ mittels eines modifizierten Horner-Schemas im Punkt t auswertet.

c) Testen Sie Ihre Funktionen am Beispiel aus Aufgabe 33 mit 4 und mit 5 Stützstellen.

d) Erweitern Sie Ihre Matlab-Funktion `c = divDiff(x,f)` aus Aufgabenteil a) so, dass Sie effizient weitere Stützstellen hinzu nehmen können. Schreiben Sie hierzu eine Matlab-Funktion `[c,dd] = divDiff2(x,f,c,dd)`, die bei jedem Aufruf die für einen weiteren Aufruf benötigten Daten zurück gibt. Überlegen Sie sich, welche Daten Sie in `dd` zurück geben müssen. Beim ersten Aufruf können dann für die Parameter `c` und `dd` die leeren Vektoren `[]` übergeben werden.

e) Testen Sie Ihre Funktion wiederum am Beispiel aus Aufgabe 33. Schreiben Sie hierzu ein Matlab-Skript `testDivDiff2`. Wählen Sie auch weitere Stützstellen, um $\sqrt{3}$ noch genauer zu berechnen.

Aufgabe 36 *(Hermite-Interpolation)* (4T Punkte)

Berechnen Sie mit Hilfe dividierter Differenzen das Newton'sche Interpolationspolynom mit den Eigenschaften

$$P(1) = -2, \quad P'(1) = -2, \quad P''(1) = 0, \quad P(2) = -3 \quad \text{und} \quad P'(2) = 1.$$

Aufgabe 37 *(Programmieraufgabe: Tschebyscheff-Interpolation)* (3M+4M Punkte)

Testen Sie Ihr Programm aus Aufgabe 35 an weiteren Beispielen:

(i) $f : [0, \pi] \to \mathbb{R}$, $f(x) = \sin(x)$,

(ii) $f : [0, 1] \to \mathbb{R}$, $f(x) = \sqrt{x}$,

(iii) $f : [-5, 5] \to \mathbb{R}$, $f(x) = \frac{1}{1+x^2}$.

Schreiben Sie hierzu ein Matlab-Skript `tschebyscheff`.

a) Verwenden Sie zunächst äquidistante Stützstellen $x_i = a + ih$ für $i = 0, \ldots, n$ mit $h = \frac{b-a}{n}$. Plotten Sie jeweils die Funktion f im entsprechenden Intervall und mindestens die Interpolationspolynome für $n = 2, 4, 8, 16$. Was stellen Sie fest? Wie groß ist jeweils der Interpolationsfehler?

b) Verwenden Sie jetzt die Nullstellen der Tschebyscheff-Polynome als Stützstellen. Beachten Sie, dass Sie die Nullstellen auf das entsprechende Intervall transformieren müssen. Plotten Sie jeweils wiederum mindestens die Interpolationspolynome für $n = 2, 4, 8, 16$, entweder jeweils in die Grafiken aus Aufgabenteil a) oder in getrennte Grafiken. Achten Sie aber auf eine übersichtliche Darstellung. Was beobachten Sie jetzt?

Hinweise:

Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Files in einer E-mail mit dem Betreff **Loesung-Blatt05** an **angewandte.numerik@uni-ulm.de** (Abgabetermin jeweils wie beim Theorieteil). Drucken Sie zusätzlich allen Programmcode sowie die Ergebnisse aus und geben Sie diese vor der Übung ab. Der Source Code sollte strukturiert und, wenn nötig, dokumentiert sein.