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

SoSe 2017
Blatt 11
30.06.2017

Angewandte Numerik 1

Besprechung in den Tutorien in der Woche vom 10.07.2017 bis 14.07.2017

Für dieses Übungsblatt gibt es 14 Theorie- und 15 Matlab-Punkte, sowie 6 Theorie- und 17 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 60-Prozent-Grenzen liegen aktuell (inklusive Blatt 11) bei 114,6 Theoriepunkten und 105,6 Matlabpunkten.

Aufgabe 42 *(Lagrange-Darstellung, Aitken-Neville)* (2T+4T+4T+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.

a) An welcher Stelle x müssen Sie dieses Interpolationspolynom P auswerten, um einen Näherungswert für $\sqrt{3}$ zu erhalten? Begründen Sie Ihre Aussage.

b) Stellen Sie das Interpolationspolynom P mit Hilfe der Lagrange'schen Interpolationsformel explizit dar und werten Sie anschließend das Polynom P an der richtigen Stelle x aus.

c) Berechnen Sie den Wert des Interpolationspolynoms P an der Stelle x 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) Was müssten Sie tun, wenn Sie auch Ihre Berechnung mit der Lagrange'schen Interpolationsformel um x_4 erweitern wollten?

Hinweis: Rechnen Sie in der ganzen Aufgabe mit Brüchen.

Aufgabe 43 *(Programmieraufgabe: Schema von Aitken-Neville)* (6M+4M+5M* Punkte)

a) Schreiben Sie eine Matlab-Funktion `values = aitkenNeville(xk, fk, t)`, welche den Funktionswert des Iterpolationspolynoms $P(f|x_0, \dots, x_n)$ von f zu den Stützstellen $x_0, \dots, x_n$ an mehreren Stellen $t \in \mathbb{R}$ mit Hilfe des Schemas von Aitken-Neville berechnet.

   `xk` $= (x_0, \dots, x_n)$ ist dabei der Vektor der Stützstellen, `fk` $= (f_0, \dots, f_n)$ der Vektor der zugehörigen Daten (also der Funktionswerte) und `t` $= (t_1, \dots, t_m)$ ist ein Vektor, der die Stellen t enthält, an denen das Interpolationspolynom ausgewertet werden soll. Der Rückgabewert `values` $= (v_1, \ldots, v_m)$ ist der Vektor mit den zugehörigen Funktionswerten, also $v_k = P(f|x_0, \ldots, x_n)(t_k)$ für $k = 1, \ldots, m$.

Hinweis: Das Verfahren von Aitken-Neville wird in der Regel nur verwendet, um den Wert des Interpolationspolynoms P an einer oder höchstens „wenigen“ Stellen t zu berechnen. Da in der Vorlesung bisher noch kein weiteres Verfahren vorgestellt wurde, verwenden wir hier Aitken-Neville auch, um P an „vielen“ Stellen auszuwerten.

b) Testen Sie Ihre Funktion am Beispiel aus Aufgabe 42 mit 4 und mit 5 Stützstellen.

c) Verkürzen Sie die von Ihrer Matlab-Funktion `aitkenNeville` benötigte Rechenzeit, indem Sie möglichst wenige Schleifen verwenden (Stichwort *Vektorisieren*). Wie viele Schleifen benötigen Sie mindestens?

Aufgabe 44 *(Programmieraufgabe: Tschebyscheff-Interpolation)*

((4M+1T)+(1M+1T)+(6M*+2T*) Punkte)

Testen Sie Ihr Programm aus Aufgabe 43 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 `stuetzstellen`.

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 $n = 2, 4, 8, 16, 31$. Was stellen Sie fest? Überlegen Sie sich anhand der Schaubilder, wie groß jeweils der Interpolationsfehler ist und an welcher Stelle er maximal ist.

b) Geben Sie für die verschiedenen n jeweils den (maximalen) Interpolationsfehler aus. Wie entwickelt sich der Interpolationsfehler mit größer werdender Anzahl der Stützstellen (also $n + 1$)?

c) Verwenden Sie jetzt andere Stützstellen, die sogenannten *Tschebyscheff-Stützstellen*: Berechnen Sie zunächst $y_i = \cos\left(\frac{2i+1}{2(n+1)}\pi\right)$ für $i = 0, \ldots, n$. Die so berechneten y_i liegen alle im Intervall $(-1, 1)$. Durch Transformation des Intervalls $[-1, 1]$ auf den jeweiligen Definitionsbereich der Funktion f erhalten Sie aus den y_i Stützstellen x_i $(i = 0, \ldots, n)$ im Definitionsbereich von f.

Plotten Sie für obige Funktionen jeweils wiederum mindestens die Interpolationspolynome für $n = 2, 4, 8, 16, 31$. Was beobachten Sie jetzt? Geben Sie für die verschiedenen n jeweils wieder den Interpolationsfehler aus. Wie entwickelt sich der Interpolationsfehler jetzt mit größer werdender Anzahl der Stützstellen?

Aufgabe 45 *(Programmieraufgabe: Gegenbeispiel von Runge mit Aitken-Neville)*

(3M*+(3M*+3T*) Punkte)

In Aufgabe 44 haben wir an drei verschiedenen Beispielen die Interpolationsfehler für äquidistante Stützstellen einerseits und Tschebyscheff-Stützstellen andererseits betrachtet. In dieser Aufgabe wollen wir die Entwicklung des Interpolationsfehlers bei zunehmender Anzahl n der Stützstellen anhand des dritten Beispiel aus Aufgabe 44, dem sogenannten Gegenbeispiel von Runge, genauer untersuchen:

a) Schreiben Sie ein Matlab-Skript `Aufgabe45`, in dem Sie für $n = 1, \ldots, 200$ äquidistante Stützstellen $x = (x_0, \ldots, x_n)^T$ jeweils das Interpolationspolynom der Funktion $f : [-5, 5] \to \mathbb{R}$, $f(x) = \frac{1}{1+x^2}$ mit dem Verfahren von Aitken-Neville an mindestens 500 Stellen auswerten. Sie dürfen dazu Ihre Matlab-Funktion `values = aitkenNeville(xk, fk, t)` aus Aufgabe 43 verwenden. Berechnen Sie jeweils den maximalen Interpolationsfehler. Plotten Sie diesen maximalen Interpolationsfehler in Abhängigkeit von der Anzahl der Stützstellen. Verwenden Sie dabei eine logarithmische Darstellung der y-Achse.

b) Erweitern Sie Ihr Matlab-Skript `Aufgabe45`. Verwenden Sie zusätzlich die auf das Intervall $[-5, 5]$ transformierten Tschebyscheff-Knoten, also die Stützstellen aus Aufgabe 44 c), und plotten Sie wiederum den Interpolationsfehler in Abhängigkeit von der Anzahl der Stützstellen. Was fällt Ihnen auf? Haben Sie dieses Ergebnis erwartet? Interpretieren Sie das Ergebnis und erklären Sie Ihre Beobachtungen!

Achtung: Die Berechnungen dieser Aufgabe können durchaus etwas länger dauern. Die benötigte Rechenzeit hängt auch davon ab, wie weit Sie in Aufgabe 43 c) vektorisiert haben.

Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Der Source Code muss strukturiert und dokumentiert sein. Senden Sie **spätestens 24 Stunden vor Ihrem Tutorium** alle Matlab-Files und alle Ergebnisse in einer E-mail mit dem Betreff **Loesung-Blatt11** an **angewandte.numerik@uni-ulm.de**.