Prof. Dr. Dirk Lebiedz
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

Angewandte Numerik 1
SoSe 2020
03.07.2020

Übungsblatt 11

Besprechung in den Tutorien in der Woche vom 13.07.2020 bis 17.07.2020

Für dieses Übungsblatt gibt es 21 Theorie- und 20 Matlab-Punkte, sowie 16 Theorie- und 38 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 70-Prozent-Grenzen liegen aktuell (inklusive Blatt 11) bei 163,1 Theorie- und 112,7 Matlabpunkten.

Aufgabe 47 *(Lagrange-Darstellung, Neville-Aitken, Newtonsche Interpolationsformel)*
(2T+4T+4T+4T+2T+2T+1T*+3T 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 $P = P(f \mid x_0, x_1, x_2, x_3)$. In dieser Aufgabe werden wir den Näherungswert für $\sqrt{3}$ auf verschiedene Arten berechnen.

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 nach Satz 8.3 explizit dar und werten Sie es anschließend an der richtigen Stelle x aus.

c) Berechnen Sie den Wert des Interpolationspolynoms P an der Stelle x auch mit Hilfe des Neville-Aitken-Schemas.

d) Stellen Sie P als Newton'sches Interpolationspolynom mit Hilfe der Methode der dividierten Differenzen explizit dar und werten Sie es anschließend an der Stelle x aus.

e) 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 Neville-Aitken-Schema um diese zusätzliche Stützstelle.

f) Erweitern Sie auch Ihre Berechnung mit der Methode der dividierten Differenzen um die zusätzliche Stützstelle x_4.

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

h) Welche Vor- und Nachteile haben die Darstellung des Interpolationspolynoms mit Hilfe der Lagrange'schen Interpolationsformel, das Neville-Aitken-Schema und die Methode der dividierten Differenzen? Für welche Fragestellungen sind diese drei Verfahren jeweils geeignet, für welche nicht?

Hinweis: Rechnen Sie in der ganzen Aufgabe mit Brüchen.

Aufgabe 48 *(Programmieraufgabe: Schema von Neville-Aitken)* (6M+4M+5M* Punkte)

a) Schreiben Sie eine MATLAB-Funktion `values = nevilleAitken(xk, fk, t)`, die die Funktionswerte des Iterpolationspolynoms $P(f \mid x_0, \ldots, x_n)$ von f zu den Stützstellen $x_0, \ldots, x_n$ an mehreren Stellen $t \in \mathbb{R}$ mit Hilfe des Schemas von Neville-Aitken berechnet.

   `xk` $= (x_0, \ldots, x_n)$ ist dabei der Vektor der Stützstellen, `fk` $= (f_0, \ldots, f_n)$ der Vektor der zugehörigen Daten (also der Funktionswerte) und `t` $= (t_1, \ldots, t_m)$ 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 $v_k = P(f \mid x_0, \ldots, x_n)(t_k)$ für $k = 1, \ldots, m$.

   Hinweis: Beachten Sie, dass Sie zur Berechnung der j-ten Spalte ($j = 1, \ldots, n$) des Schemas von Neville-Aitken nur die Werte der vorigen Spalte (also der Spalte $0, \ldots, n-1$) benötigen. Das Schema von Neville-Aitken benötigt daher für die Speicherung der schon berechneten Werte keine Matrix, sondern kann mit einem Spaltenvektor realisiert werden.

   Hinweis: Das Verfahren von Neville-Aitken wird in der Regel nur verwendet, um den Wert des Interpolationspolynoms P an einer oder höchstens „wenigen“ Stellen t zu berechnen.

b) Testen Sie Ihre Funktion am Beispiel aus Aufgabe 47 mit 4 und mit 5 Stützstellen.

c) Versuchen Sie möglichst wenige Schleifen zu verwenden (Stichwort *Vektorisieren*). Wie viele Schleifen benötigen Sie mindestens?

Aufgabe 49 *(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 `pt = hornerSchema(a, t)` zur effizienten Auswertung des Polynoms $p(x) = \sum_{k=0}^{n} a_k x^k$ mit den Koeffizienten `a` $= (a_0, a_1, a_2, \ldots, a_n)^T \in \mathbb{R}^{n+1}$. Dabei enthält der Vektor `t` $= (t_1, t_2, \ldots, t_m)^T$ mit $t_k \in \mathbb{R}$, $(k = 1, \ldots, m)$ die Stellen, an denen das Polynom p ausgewertet werden soll. Der Ergebnisvektor `pt` $= (pt_1, pt_2, \ldots, pt_m)^T$ soll die Werte des Polynoms an den Stellen `t` enthalten, also $pt_k = p(t_k)$, $(k = 1, \ldots, m)$.

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 50 *(Programmieraufgabe: Newton-Interpolationspolynom und Schema der Dividierten Differenzen)* (4M+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, c, 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 in den Punkten `t` $= (t_1, \ldots, t_m)^T$, $m \in \mathbb{N}$ auswertet.

c) Testen Sie Ihre Funktionen am Beispiel aus Aufgabe 47 mit 4 und mit 5 Stützstellen.

Aufgabe 51 *(Programmieraufgabe: Tschebyscheff-Interpolation)*

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

Testen Sie Ihre MATLAB-Funktionen aus Aufgabe 50 oder, falls Sie Aufgabe 50 nicht bearbeitet haben, Ihre MATLAB-Funktion aus Aufgabe 48 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 $i = 0, \dots, n$ 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?

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 $i = 0, \dots, n$. 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 $(i = 0, \dots, n)$ 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 52 *(Programmieraufgabe: Gegenbeispiel von Runge)*

(3M*+(3M*+3T*)+3M*+2T*+(3M*+2T*)+(2M*+2T*)+(3M*+2T*) Punkte)

In Aufgabe 51 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 51, dem sogenannten Gegenbeispiel von Runge, genauer untersuchen:

a) Schreiben Sie ein MATLAB-Skript `Aufgabe52`, in dem Sie für äquidistante Stützstellen $x = (x_0, \dots, x_n)^T$ $(n = 1, \dots, 200)$ jeweils das Interpolationspolynom der Funktion $f : [-5, 5] \to \mathbb{R}$, $f(x) = \frac{1}{1+x^2}$ mit dem Verfahren von Neville-Aitken an mindestens 500 Stellen auswerten. Sie dürfen dazu Ihre Matlab-Funktion `values = nevilleAitken(xk, fk, t)` aus Aufgabe 48 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 `Aufgabe52`. Verwenden Sie zusätzlich die auf das Intervall $[-5, 5]$ transformierten Tschebyscheff-Knoten, also die Stützstellen aus Aufgabe 51 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!

c) Betrachten Sie nun auch das Newton-Interpolationspolynom und das Verfahren der Dividierten Differenzen. Sie dürfen dazu Ihre Matlab-Funktionen `c = divDiff(x, f)` und `v = evalNewtonpolynom(x, c, t)` aus Aufgabe 50 verwenden. Berechnen Sie also für äquidistante Stützstellen $x = (x_0, \ldots, x_n)^T$ ($n = 1, \ldots, 200$) jeweils das Interpolationspolynom der Funktion $f : [-5, 5] \to \mathbb{R}$, $f(x) = \frac{1}{1+x^2}$ mit dem Verfahren der Dividierten Differenzen. Werten Sie dieses Interpolationspolynom jeweils mit Ihrem modifizierten Horner-Schema an mindestens 500 Stellen aus und berechnen Sie jeweils den maximalen Interpolationsfehler. Ergänzen Ihr Schaubild um diese maximalen Interpolationsfehler.

d) Vergleichen Sie Ihren Plot der maximalen Interpolationsfehler bei äquidistanten Stützstellen mit dem entsprechenden Plot für das Schema von Neville-Aitken. Was fällt Ihnen auf? Interpretieren Sie Ihr Schaubild. Können Sie Ihre Beobachtung erklären?

e) Erweitern Sie Ihr MATLAB-Skript auch um das mit dem Verfahren der Dividierten Differenzen berechnete Interpolationspolynom zu den auf das Intervall $[-5, 5]$ transformierten Tschebyscheff-Knoten. Plotten Sie wiederum den Interpolationsfehler in Abhängigkeit von der Anzahl der Stützstellen und vergleichen Sie diesen mit dem entsprechenden durch das Schema von Aitken-Neville berechneten Interpolationsfehler. Was fällt Ihnen auf? Haben Sie dieses Ergebnis erwartet? Interpretieren Sie das Ergebnis und erklären Sie Ihre Beobachtungen!

f) Mit Hilfe der Leja-Sortierung der Stützstellen kann der bei Verwendung des Verfahrens der Dividierten Differenzen aufgetretene Interpolationsfehler reduziert werden.

Erweitern Sie Ihr MATLAB-Skript erneut und verwenden Sie zusätzlich sowohl die mit Hilfe der Leja-Sortierung umsortierten äquidistanten Stützstellen als auch die mit Hilfe der Leja-Sortierung umsortierten Tschebyscheff-Stützstellen (Aufruf jeweils `x = leja(x);`). Was stellen Sie nun fest? Woran liegt das?

g) Vergleichen Sie die benötigten Laufzeiten des Schemas von Neville-Aitken mit den Laufzeiten des Schemas der Dividierten Differenzen einschließlich der anschließenden Auswertung mit dem modifizierten Hornerschema aus Aufgabe 50. Plotten Sie die Laufzeiten der Verfahren jeweils in Abhängigkeit von der Anzahl der Stützstellen und achten Sie dabei auf eine geeignete Skalierung der Achsen.

Berücksichtigen Sie bei Ihrem Laufzeitenvergleich sowohl die nicht vektorisierte als auch Ihre vektorisierten Varianten des Schemas von Neville-Aitken.

Was fällt Ihnen auf? Haben Sie dieses Ergebnis so erwartet? Welches Verfahren ist das für diese Problemstellung geeignetere? Warum?

Achtung: Die Berechnungen dieser Aufgabe können je nach verwendeter Hardware durchaus etwas länger dauern.

Hinweise:
Die Lösungen der Theorieaufgaben und Ihren Text zu den Programmieraufgaben können Sie in LaTeX erstellen oder handschriftlich aufschreiben und einscannen. Die Programmieraufgaben sind in Matlab zu lösen. Der Source Code muss strukturiert und dokumentiert sein.
Falls Sie die Aufgaben im Team lösen, geben Sie bitte auf allen Lösungen alle an der Aufgabe beteiligten Teammitglieder an. Jedes Teammitglied muss eine Lösung abgeben.
Speichern Sie Ihre Lösungen und Ihre Ergebnisse in einem Directory mit dem Namen **Blatt11_*Vorname*_*Nachname*** und verpacken Sie dieses in eine `.zip`-Datei. Laden Sie **spätestens 48 Stunden vor Ihrem Tutorium** diese `.zip`-Datei in Moodle hoch.