Dirk Praetorius, Samuel Ferraz-Leite
Christoph Überhuber

Sommersemester 2009
17. Juni 2009

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 11

Die Aufgaben mit Stern (*) sind bis zur nächsten Übung vorzubereiten und werden dort abgeprüft. Kopieren Sie bitte den Source-Code in ein Unterverzeichnis `serie11` *Ihres Home-Verzeichnisses. Überprüfen Sie bitte vor der Übung, ob Ihre Source-Codes mit* `matlab` *auf der* `lva.student.tuwien.ac.at` *interpretiert werden können.*

Aufgabe 101*. Eine effiziente Implementierung des einseitigen Differenzenquotienten $\Phi(h)$ aus Aufgabe 93 verwendet die vorherigen Werte $\Phi(h_0), \dots, \Phi(h_n)$, indem man (theoretisch!) das Interpolationspolynom p_n vom Grad $n-1$ zu den Punkten $(h_j, \Phi(h_j))$ für $j = 1, \dots, n$ betrachtet, d.h. $p_n(h) \approx \Phi(h)$, und dieses mit dem Neville-Verfahren bei $h = 0$ auswertet. Man bezeichnet dieses Vorgehen als *Richardson-Extrapolation des einseitigen Differenzenquotienten.* (Einen Konvergenzbeweis für dieses Verfahren sehen Sie in der Vorlesung zur Numerischen Mathematik.) Mit $h_n := 2^{-n} h_0$ betrachten wir die Folge der $y_n := p_n(0)$. Man schreibe eine Funktion `richardson`, die neben dem Funktionshandle einer Funktion f, den Auswertungspunkt x, die erste Schrittweite $h_0 > 0$ sowie die Toleranz $\tau > 0$ übernimmt und $y_{n+1} \approx f'(x)$ zurückliefert, sobald gilt

$$|y_n - y_{n+1}| \leq \begin{cases} \tau, & \text{falls } |y_{n+1}| \leq \tau, \\ \tau\,|y_{n+1}| & \text{anderenfalls.} \end{cases}$$

Verwenden Sie bei der Realisierung die Funktion `neville` aus Aufgabe 91. Speichern Sie den Source-Code unter `richardson.m` in das Verzeichnis `serie11`.

Aufgabe 102*. Vergleichen Sie die Konvergenz des einseitigen Differenzenquotienten aus Aufgabe 93 mit der Konvergenz der durch Richardson-Extrapolation gemäß Aufgabe 101 berechneten Approximationen an $f'(x)$. Schreiben Sie dazu ein Skript, dass folgende Aufgaben erfüllt:

- Für eine Funtkion $f \in \mathcal{C}^\infty(\mathbb{R})$ sollen Folgen der Differenzenquotienten mit und ohne Extrapolation berechnet werden.
- Es soll eine Tabelle der absoluten Fehler $|\Phi(h) - f'(x)|$ bzw. und relativen Fehler $|\Phi(h) - f'(x)|/|f'(x)|$ für beide Verfahren ausgegeben werden.
- Es soll ein (doppelt logarithmischer) Konvergenzgraph erstellt werden, in dem beide beide Verfahren direkt miteinander verglichen werden. Fügen Sie eine Legende, Achsenbeschriftungen und eine Überschrift ein.

Aufgabe 103*. Gegeben seien Dimensionen $m, n \in \mathbb{N}$ und Vektoren $I, A \in \mathbb{R}^N$ sowie $J \in \mathbb{R}^{n+1}$, die eine schwach besetzte $(m \times n)$-Matrix in CCS-Speicherung repräsentieren. Schreiben Sie eine Funktion `ccs2naive(I,J,A,m,n)`, die als Ergebnis die entssprechenden Vektoren bei naiver Speicherung zurückgibt.

Aufgabe 104*. Gegeben seien Dimensionen $m, n \in \mathbb{N}$ und 3 Vektoren $I, A \in \mathbb{R}^N$ und $J \in \mathbb{R}^{n+1}$, die eine schwachbesetzte $(m \times n)$-Matrix in CCS-Speicherung repräsentieren, sowie ein Vektor $x \in \mathbb{R}^n$. Schreiben Sie mittels geeigneter (möglichst weniger) Schleifen eine Matrix-Vektor-Multiplikation `mvmsparse(I,J,A,m,n,x)`.

Aufgabe 105. Gegeben seien Dimensionen $m, n \in \mathbb{N}$ und 3 Vektoren $I, J, A \in \mathbb{R}^N$ mit $N \leq mn$, die eine schwachbesetzte Matrix in naiver Speicherung repräsentieren. Schreiben Sie eine Funktion `naive2ccs(I,J,A,m,n)`, die als Ergebnis die entsprechenden Vektoren des CCS-Formats zurückliefert.

Aufgabe 106. Bei der Richardson-Extrapolation des einseitigen Differenzenquotienten aus Aufgabe 101 kann man sich folgende Beobachtung zum Neville-Verfahren zu Nutze machen: Bei einer effizienten Implementierung des Neville-Verfahrens gemäß Aufgabe 92 überschreibt man den Vektor y durch die Diagonale $(p_{1,n}, p_{2,n-1}, \ldots, p_{n,1})$, und $p_{1,n}$ ist der gesuchte Wert. Fügt man nun einen weiteren Interpolationsknoten (x_{n+1}, y_{n+1}) hinzu, so muss man nicht noch einmal das vollständige Schema rechnen, sondern lediglich die „neue Diagonale“ $(p_{1,n+1}, p_{2,n}, \ldots, p_{n+1,1})$. Mit dieser Beobachtung muss man in jedem Schritt der Richardson-Extrapolation nur noch eine Schleife durchlaufen, nicht mehr zwei!

Aufgabe 107. Für einen stetigen Integranden $f : [a, b] \to \mathbb{R}$ berechnet man das Integral $I := \int_a^b f\,dx$ numerisch über geeignete Summen. Bei der *summierten Rechteckregel* berechnet man für gegebenes $n \in \mathbb{N}$ und $h := (b-a)/n$ zum Beispiel

$$I_n := h \sum_{j=1}^{n} f(a + jh). \tag{1}$$

Man schreibe eine Funktion `rechteckregel(f,a,b,tau)`, die die Folge der Approximationen I_n berechnet, bis gilt

$$|I_n - I_{n-1}| \leq \begin{cases} \tau & \text{für } |I_n| \leq \tau, \\ \tau\,|I_n| & \text{anderenfalls.} \end{cases}$$

In diesem Fall gebe man die vollständige Folge $(I_1, \ldots, I_n)$ der Approximationen zurück. Man teste die numerische Integration am Beispiel $f(x) = \exp(x)$ auf dem Intervall $[0, 10]$ und gebe abhängig von n den Fehler $|I - I_n|$ tabellarisch aus.

Aufgabe 108. Eine weitere Quadraturregel ist die *summierten Trapezregel*. Hier berechnet man für gegebenes $n \in \mathbb{N}$ und $h := (b-a)/n$ die Approximation

$$I_n := \frac{h}{2}\Big(f(a) + 2\sum_{j=1}^{n-1} f(a + jh) + f(b)\Big). \tag{2}$$

Dies ist gerade das Integral über die stetige und stückweise affine Funktion p mit $p(a + jh) = f(a + jh)$. Man schreibe eine Funktion `trapezregel(f,a,b,tau)`, die die Folge der Approximationen I_n berechnet, bis gilt

$$|I_n - I_{n-1}| \leq \begin{cases} \tau & \text{für } |I_n| \leq \tau, \\ \tau\,|I_n| & \text{anderenfalls.} \end{cases}$$

In diesem Fall gebe man die vollständige Folge $(I_1, \ldots, I_n)$ der Approximationen zurück. Man teste die numerische Integration am Beispiel $f(x) = \exp(x)$ auf dem Intervall $[0, 10]$ und gebe abhängig von n den Fehler $|I - I_n|$ tabellarisch aus.

Aufgabe 109. Vergleichen Sie die Konvergenz der Verfahren aus Aufgabe 107–108. Schreiben Sie dazu eine Funktion `compare_quadrature(f,a,b,tau,trueint)`, die folgende Aufgaben erfüllt:

- Für die gegebene Funtkion f soll die Folge der Approximationen an das Integral mit beiden Methoden berechnet werden.
- Es soll eine Tabelle der absoluten Fehler $|I_n - \int_a^b f(x)|$ bzw. und relativen Fehler $|I_n - \int_a^b f(x)|/|\int_a^b f(x)|$ für beide Verfahren ausgegeben werden.
- Es soll eine Tabelle der Experimentellen Konvergenzordnung beider Verfahren ausgegeben werden.
- Es soll ein (doppelt logarithmischer) Konvergenzgraph erstellt werden, in dem beide beide Verfahren direkt miteinander verglichen werden. Fügen Sie eine Legende, Achsenbeschriftungen und eine Überschrift ein. Was beobachten Sie?

Testen Sie beide Verfahren mit unterschiedlichen Funktionen und insbesondere mit Funktionen unterschiedlicher Differenzierbarkeitsordnung.

Aufgabe 110. Gegeben sei ein sortierter Vektor x der Länge n. Man schreibe eine Matlab-Funktion `findbisection(x,y)`, die einen Index i zurückgibt, für den $x_i = y$ gilt. Falls y nicht in x vorkommt soll 0 zurückgegeben werden. Naive Realisierung führt auf Aufwand $\mathcal{O}(n)$ im worst case, d.h. man durchsucht den Vektor von vorne nach hinten und das gesuchte y steht gerade an der letzten Stelle in x bzw. kommt überhaupt nicht in x vor. Wie kann man den Bisektionsalgorithmus geeignet modifizieren, um einen Algorithmus mit worst case Aufwand $\mathcal{O}(log(n))$ zu erhalten?