Winfried Auzinger
Dirk Praetorius

Sommersemester 2013
20. März 2013

Übungen zur Vorlesung Computermathematik

Serie 3

Aufgabe 3.1. Ziel ist die Implementierung des MergeSort-Algorithmus. Schreiben Sie dazu eine Funktion `mergesort` und eine Unterfunktion `merge` (im File `mergesort.m`) mit folgender Funktionalität:

- Die Funktion `merge` übernimmt zwei aufsteigend sortierte Zeilenvektoren $a \in \mathbb{R}^m$ und $b \in \mathbb{R}^n$ und vereinigt diese (mittels geeigneter Schleifen) so, dass der resultierende Zeilenvektor $c \in \mathbb{R}^{m+n}$ ebenfalls sortiert ist, z.B. soll $a = (1, 3, 3, 4, 7)$ und $b = (1, 2, 3, 8)$ als Ergebnis $c = (1, 1, 2, 3, 3, 3, 4, 7, 8)$ liefern. Dabei soll ausgenutzt werden, dass die Vektoren a und b bereits sortiert sind. Es soll insbesondere *kein* Sortierverfahren auf den vereinigten Vektor `[a,b]` angewendet werden.
- Die rekursive Funktion `mergesort` übernimmt einen unsortierten Zeilenvektor $c \in \mathbb{R}^N$ und soll c aufsteigend sortiert zurückgeben. Das Vorgehen ist wie folgt: Für $N \leq 2$ sortiert man c direkt. Für $N > 2$ zerlegt man c in zwei Hälften a und b, ruft `mergesort` rekursiv für a und b auf und verwendet `merge`, um aus den sortierten Hälften a und b den sortierten Vektor c zu erhalten.

Aufgabe 3.2. Schreiben Sie ein Skript, das die Laufzeiten von Mergesort für zufällige Vektoren $x \in \mathbb{R}^N$ und $N = 100 \cdot 2^n$ mit $n = 0, 1, 2, \dots$ geeignet visualisiert. Zeichnen Sie geeignete Vergleichsgeraden in den Plot, um visuell den Aufwand Ihrer Implementierung zu bestimmen.

Aufgabe 3.3. Schreiben Sie eine Funktion `plotPotential`, die eine Funktion $f : \mathbb{R}^2 \to \mathbb{R}$, einen quadratischen Bereich $[a, b]^2$ in Form seiner Grenzen $a, b \in \mathbb{R}$ und eine Schrittweite τ übernimmt und $f(x, y)$ als farbige Projektion auf die Ebene darstellt. Dabei soll $f(x, y)$ auf einem äquidistanten Tensorgitter der Schrittweite τ ausgewertet werden. Sie dürfen davon ausgehen, dass die MATLAB-Realisierung der Funktion f Matrizen $x, y \in \mathbb{R}^{M\times N}$ übernimmt und eine Matrix $\mathbb{R}^{M\times N}$ der Funktionswerte zurückgibt. Falls der Funktion `plotPotential` *optional* ein weiterer Parameter $n \in \mathbb{N}$ übergeben wird, sollen zusätzlich n Potentiallinien (= Konturlinien) in schwarz oder weiß in den Plot eingezeichnet werden. Geben Sie neben den Plot eine `colorbar`. Schreiben Sie ein Skript, das `plotPotential` mit Hilfe der Potentialfunktion $f(x, y) = x \cdot \exp(-x^2 - y^2)$ aus der Vorlesung testet.

Aufgabe 3.4. Gegeben Sei eine C-Funktion mit der Signatur `double fct(double x, double y);` als Implementierung einer Funktion $f : \mathbb{R}^2 \to \mathbb{R}$. Schreiben Sie eine `MEX`-Schnittstelle, sodass der MATLAB-Aufruf `Z = fct(X,Y)` mit Matrizen $X, Y \in \mathbb{R}^{M\times N}$ gleicher Dimension eine Matrix $Z \in \mathbb{R}^{M\times N}$ mit den Funktionswerten $Z_{jk} = f(X_{jk}, Y_{jk})$ liefert. Das Programm soll die übergebenen Parameter X und Y auf gleiche Dimension prüfen und ggf. eine Fehlermeldung ausgeben. Zum Test implementieren Sie die Funktion $f(x, y) = x \cdot \exp(-x^2 - y^2)$ und reproduzieren einige der Plots aus der Vorlesung.

Aufgabe 3.5. Schreiben Sie eine Funktion `saveMatrix`, die eine gegebene Matrix $A \in \mathbb{R}^{M\times N}$ mittels formatierter Ausgabe in eine Textdatei `matrix.dat` schreibt. Optional soll der Funktion auch ein Dateiname übergeben werden können, sodass die Matrix in die Textdatei `name.dat` geschrieben wird. Schreiben Sie zum Test ein Skript, das eine zufällige Matrix $A \in \mathbb{R}^{M\times N}$ in eine Textdatei schreibt, diese dann mittels `load` als B wieder einliest und A und B auf Gleichheit prüft.

Aufgabe 3.6. Es sei $f : [a,b] \to \mathbb{R}$ eine integrierbare Funktion. Für $N \in \mathbb{N}$ und $x_j := a + j\,(b-a)/N$ für $j = 0, \dots, N$ definieren wir die **summierte Mittelpunktsregel**

$$I_N := \frac{b-a}{N} \sum_{j=1}^{N} f\big((x_{j-1} + x_j)/2\big). \tag{1}$$

Da I_N eine Riemann-Summe ist, gilt

$$\lim_{N\to\infty} I_N = \int_a^b f\,dx.$$

Man kann sogar zeigen, dass für $f \in C^2[a,b]$

$$\Big|\int_a^b f\,dx - I_N\Big| = \mathcal{O}(N^{-2})$$

gilt. Schreiben Sie eine MATLAB-Funktion

```
int = mittelpunktsregel(a,b,f,n)
```

die für eine Folge $N = 2^k$ und $k = 0, \dots, n$ die Werte I_N berechnet und als Vektor `int` zurückgibt.

Aufgabe 3.7. Betrachten Sie das Integral $I := \int_0^5 \exp(x)\,dx$. Berechnen Sie mithilfe der vorausgegangenen Aufgabe eine Folge I_N von Approximationen. Plotten Sie den Fehler $E_N = |I - I_N|$ und den Fehlerschätzer $\delta_N = |I_{2N} - I_N|$ in einem doppellogarithmischen Plot. Verifizieren Sie das Konvergenzverhalten $\mathcal{O}(N^{-2})$, indem Sie eine Gerade mit Gefälle 2 in den Plot einzeichnen. Welches Konvergenzverhalten beobachten Sie, wenn Sie die Konvergenz der Folge $x_n = I_N$ mittels Aitkens Δ^2-Verfahren beschleunigen? Welches Konvergenzverhalten beobachten Sie, wenn Sie in der summierten Mittelpunktsregel (1) die f-Auswertung am Mittelpunkt $f\big((x_{j-1}+x_j)/2\big)$ durch $f(x_{j-1})$ ersetzen?

Aufgabe 3.8. Gegeben seien Dimensionen $m, n \in \mathbb{N}$ und Vektoren $I, J, Aij \in \mathbb{R}^N$, die eine schwachbesetzte Matrix in naiver Speicherung (Koordinatenformat) repräsentieren. Schreiben Sie eine Funktion

```
[II,JJ,AA] = naive2ccs(I,J,Aij,m,n)
```

die als Ergebnis die entsprechenden Vektoren des CCS-Formats zurückliefert. Schreiben Sie ferner eine Matrix-Vektor-Multiplikation

```
Ax = mvm(II,JJ,AA,m,n,x)
```

die den Produkt-Vektor $Ax \in \mathbb{R}^m$ für eine im CCS-Format gespeicherte Matrix $A \in \mathbb{R}^{m\times n}$ und einen Spaltenvektor $x \in \mathbb{R}^n$ zurückliefert. Sie können Ihre Funktion testen, indem Sie für eine `sparse`-Matrix A mittels `[I,J,Aij] = find(A)` das Koordinatenformat auslesen und das Ergebnis von `A*x` mit Ihrer eigenen Matrix-Vektor-Multiplikation vergleichen.