Fachbereich Mathematik und Informatik
Freie Universität Berlin
Prof. Dr. Ralf Kornhuber, Maren-Wanda Wolf

11. Übung zur Vorlesung

Computerorientierte Mathematik I

WS 2015/2016

Abgabe: 28.1.2016

1. Aufgabe (2 Zusatz-TP)
Auf einer Familienfeier werden Sie von Ihrer Tante gefragt, was man denn heute so alles in der Uni lerne. Sie antworten Ihr (unter anderem), dass Sie gelernt haben, dass man mit reellen Zahlen nicht rechnen kann. Da Sie ihre etwas ungläubig schauende Tante nicht mit diesem Statement alleine stehen lassen wollen, erklären Sie Ihr in kurzen (aber eben für Nicht-Mathematiker verständlichen Worten), warum diese Aussage stimmt.
Tipp: Machen Sie es vielleicht anhand von Beispielen deutlich!

2. Aufgabe (3 Zusatz-TP)

a) Rechnen Sie folgende Zahlen aus $G(5, 6)$, also Gleitkommazahlen zur Basis $q = 5$ mit Mantissenlänge 6, in Gleitkommazahlen aus $G(10, 5)$ um:
$x = 13,2441_5$
$x = 1,32144_5$
$x = 0,01332_5$

b) Geben Sie für die obigen Werte jeweils die nächstgrößere und nächstkleinere Zahl in $G(5, 6)$ und $G(10, 5)$ an. Warum geht das?

3. Aufgabe (3 Zusatz-TP)

a) Bestimmen Sie jeweils die absolute Kondition der Abbildung $f : \mathbb{R} \to \mathbb{R}$:

1. $f(x) = |x|$
2. $f(x) = \frac{\pi}{2}$
3. $f(x) = \sin(x)$

b) Geben Sie jeweils ein Beispiel für eine gut bzw. schlecht konditionierte Funktionsauswertung an (jeweils Funktion f und Argument x). Verwenden Sie keines der Beispiele aus a).

4. Aufgabe (3 Zusatz-TP + 1 Zusatz-PP)

a) Betrachten Sie die Funktion

$$f(x) = \frac{1}{1+2x} - \frac{1-x}{1+x} \,,$$

welche in der Nähe des Nullpunktes ausgewertet werden soll. Berechnen Sie zunächst die relative Kondition κ_{rel} von f im Punkt $x_0 > 0$. Diskutieren Sie anschließend die relative Stabilität σ_{rel} der Auswertung des Ausdrucks auf der rechten Seite.

b) Finden Sie für f einen stabileren Algorithmus.

c) Vergleichen Sie die Auswertung von f an der Stelle $x_1 = 0{,}01$ bezüglich der Darstellung in b) mit der Auswertung Ihres neuen Algorithmus, indem Sie das MATLAB-Programm `taschenrechner(L,x,y,op)` von Aufgabenblatt 3 mit dem Wert $L = 3$ verwenden.

5. Aufgabe (3 Zusatz-PP)
In der Vorlesung haben Sie zwei Summationsalgorithmen kennengelernt, rekursive und hierarchische Summation. Implementieren Sie diese in MATLAB und testen sie bei der Berechnung der N-ten Partialsumme der harmonischen Reihe

$$H_n = \sum_{n=1}^{N} \frac{1}{n}.$$

Rechnen Sie in MATLAB mit einer Genauigkeit von 4 Stellen, d.h. benutzen Sie in jedem Zwischenschritt die Funktion `runden(x,4)` vom 3. Übungszettel. Führen Sie für $N = 2^j, j = 1, \ldots, 16$ Testläufe mit beiden Algorithmen durch und plotten Sie die Resultate H_N gegen j. Wie erklären Sie sich die Ergebnisse?

6. Aufgabe (2 Zusatz-TP)
Berechnen Sie unter Verwendung der Grundrechenarten die n-te Potenz von x, $x \in \mathbb{R}$, $n \in \mathbb{N}$, mit einem Aufwand von $\mathcal{O}(\log n)$.

7. Aufgabe (3 Zusatz-TP)
Gegeben sei das lineare Gleichungssystem

$$\begin{aligned} 3.121x_1 + 9.183x_2 + 4.437x_3 &= 1.2 \\ -1.051x_1 - 2.113x_2 + 1.372x_3 &= 2.3 \\ 0.76x_1 + 1.88x_2 + 0.04x_3 &= 4.732 \end{aligned}$$

im Dezimalsystem. Existiert für dieses eine eindeutig bestimmte Lösung $x = (x_1, x_2, x_3)$? Ihre Rechenmaschine habe $l = 3$ Stellen (in Gleitkommadarstellung) zur Verfügung und löse lineare Gleichungssysteme durch Invertieren der Matrix und Multiplikation mit der rechten Seite. Können Sie das System mit dem Rechner lösen? Begründen Sie Ihre Antwort.

8. Aufgabe (5 + 2 Zusatz-PP)

a) Implementieren Sie den Gaußschen Algorithmus (LR–Zerlegung ohne Spaltenpivoting) für quadratische Matrizen in MATLAB und überlegen Sie sich, wie Sie dabei möglichst speichersparend vorgehen können. Stellen Sie dazu drei Routinen zur Verfügung, und zwar die eigentliche LR–Zerlegung, die Vorwärts- und die Rückwärtssubstitution. Testen Sie den von Ihnen geschriebenen Algorithmus an linearen Gleichungssystemen der Dimension $n = 1, 5, 20, 100, 1000$ oder soweit, wie es Ihre Geduld zulässt, und protokollieren Sie die Laufzeiten.

b) Bei Tridiagonalmatrizen, d.h. Matrizen $A = (a_{ij})_{i,j=1}^n$ mit $a_{ij} = 0$ für $|i - j| \geq 2$, kann der Aufwand des Gaußschen Algorithmus gegenüber dem allgemeinen Fall deutlich reduziert werden. Überlegen Sie sich, wie man Ihr Programm aus Teil a) modifizieren kann, um der Struktur von Tridiagonalmatrizen Rechnung zu tragen, und implementieren Sie diese Änderungen. Berechnen Sie wie in Teil a) die Laufzeiten für verschiedene n.

Hinweis: Mit `A=rand(n,n)` können Sie quadratische Zufallsmatrizen aus dem $\mathbb{R}^{n,n}$ erzeugen, die meist auch vollen Rang haben, was Sie mit `rank(A)` testen können. Die Kondition einer Matrix lässt sich mit Hilfe von `cond` bestimmen. Spannend ist auch die Hilbertmatrix `hilb(n)`, die schon für sehr kleine n sehr schlecht konditioniert ist. Für Teil b) sind außerdem die Befehle `diag` oder `spdiags` hilfreich.