Prof. Dr. Helmut Abels
Andreas Schöttl
Josef Weber

Fakultät für Mathematik
Universität Regensburg
26. November 2015

2. Programmierübungsblatt

zur Vorlesung: *Numerik I* im WiSe 15/16
Abgabe bis 14.12.2015 um 12:00 Uhr im Übungskasten 34 der Mathematik sowie in GRIPS.

Laden Sie den Quellcode für die von Ihnen bearbeitete Programmieraufgabe bis zum Abgabetermin in GRIPS hoch und geben Sie einen Ausdruck dieses Quellcodes bis 14.12.2015 um 12:00 Uhr im Übungskasten 34 der Mathematik ab. Die Datei soll dabei den Namen $Projekt2\_Nachname\_Vorname(\_Nachname\_Vorname).c$ (oder auch Dateiendung .cpp) haben mit den Vor- und Nachnamen der Personen, die die Aufgabe bearbeitet haben. Laden Sie nicht das ausführbare Programm in GRIPS hoch. In den ersten beiden Zeilen Ihres Quellcodes müssen als Kommentar die Nummer der von Ihnen bearbeiteten Aufgabe, also Aufgabe 1 oder Aufgabe 2, sowie die Namen der Personen, die die Aufgabe bearbeitet haben, vermerkt sein.
Die Programmieraufgabe gilt als bestanden, wenn die Lösungen abgegeben wurden, einem Assistenten in den Programmierübungen vorgestellt wurden und Fragen zur Lösung hinreichend beantwortet werden konnten. Dabei können die Lösungen einzeln oder in Zweiergruppen abgegeben werden. Bei Abgabe in Zweiergruppen müssen beide Personen die Fragen zur Lösung beantworten können.
Die Terminvergabe für die Vorstellung Ihrer Lösung erfolgt über GRIPS. Bitte tragen Sie sich dazu in die dortige Terminliste ein.
Versehen Sie Ihren Quellcode mit hinreichend vielen Kommentarzeilen, so dass Sie ihn problemlos lesen und erklären können. Deklarieren Sie die zu programmierenden Funktionen so, wie es in der Aufgabenstellung vorgegeben ist. Weiterhin soll in beiden Aufgaben der Speicherplatz dynamisch erzeugt werden.

Lösen Sie eine der beiden Aufgaben.

Aufgabe 1: (Bisektions- und Newtonverfahren)

i) Programmieren Sie das Bisektionsverfahren und das Newtonverfahren in einer Dimension. Hierbei sollen die beiden Funktionen die Form

```
double Bisektionsverfahren(double (*f)(double x), double a, double b, int maxiter,
                           double tol, FILE *file)
double Newtonverfahren(double (*f)(double x), double (*df)(double x), double x0, int maxiter,
                           double tol, FILE *file)
```

haben, wobei die benötigten Funktionen f und deren Ableitung df als Funktionspointer übergeben werden. Der Rückgabewert sei jeweils die gefundene Nullstelle.
Beim Bisektionsverfahren seien a und b die untere bzw. obere Intervallgrenze, *maxiter* die maximale Anzahl an Iterationen, nach denen die Funktion abbrechen soll, sowie *tol* der Wert, bei dem das Bisektionsverfahren abbrechen soll, sobald der Funktionswert von f diesen betragsmäßig unterschreitet.
Beim Newtonverfahren seien $x0$ der Startwert, *maxiter* die maximale Anzahl an Iterationen, nach denen die Funktion abbrechen soll, sowie *tol* der Wert, bei dem das Newtonverfahren abbrechen soll, sobald die Newtonkorrektur diesen Wert betragsmäßig unterschreitet.
Testen Sie beide Verfahren, indem Sie die Nullstelle $x = 2$ folgender Funktionen numerisch bestimmen:

$$f_1(x) = (x-2)\,x$$
$$f_2(x) = (x-2)^5.$$

Verwenden Sie für das Newtonverfahren den Startwert $x0 = 100$ und für das Bisektionsverfahren die Intervallgrenzen $a = 1.5$ und $b = 100$. Verwenden Sie beide Funktionen mit $maxiter = 100$

sowie $tol = 1.0e-7$.
Schreiben Sie den Fehler in Abhängigkeit von der Anzahl der Iterationen für beide Verfahren jeweils in die Datei, welche den Funktionen als Parameter übergeben wird.

ii) Programmieren Sie das Newton-Verfahren im Mehrdimensionalen. Die zu programmierende Funktion soll dabei von der Form

int Newton_Multidim(int N, double (*f) (double *, double *, int),
double (*df)(double*, double**, int), double* x0, double* fx,
int maxiter, double tol)

sein. Beachten Sie hierbei, dass Sie eine Funktion $f : \mathbb{R}^N \to \mathbb{R}^N$ als Funktionspointer mit der Form $double\ (^*f)(double^*,\ double^*, int)$ übergeben sollen, wobei der erste Parameter $x \in \mathbb{R}^N$, der zweite $f(x) \in \mathbb{R}^N$ sowie der dritte die Dimension N sein sollen. Analog erfolgt die Übergabe für das Differential. Weiterhin seien $x0$ der Startwert, welcher in jedem Newtonschritt mit der neuen Newtoniterierten überschrieben wird, fx der Funktionswert von f an der Stelle der neuen Newtoniterierten sowie *maxiter* und *tol* wie im eindimensionalen Newtonverfahren.
Nutzen Sie zum Testen die Funktion

$$f : \mathbb{R}^2 \to \mathbb{R}^2 \text{ mit } f(x, y) = \begin{pmatrix} \frac{1}{10}x^2 + \sin(y) \\ \cos(x) + \frac{1}{10}y^2 \end{pmatrix}$$

mit den Startwerten $x0 = (0.5, 0.5)$, $x0 = (-0.5, -0.5)$ und $x0 = (0.5, 1.5)$. Falls eine Nullstelle ermittelt werden konnte, geben Sie diese auf dem Bildschirm aus.

Hinweis: Auf der GRIPS-Seite finden Sie die Headerdatei LRmP.h sowie einen Link zur dazugehörigen Objektdatei LRmP.o. Mit diesen können Sie die LR-Zerlegung mit Zeilenpivotisierung aufrufen, falls Sie die 1. Aufgabe des 1. Programmierübungsblatts nicht bearbeitet haben. Weiterhin befindet sich auf der GRIPS-Seite ein kurzes Beispiel, wie man Objektdateien zu einem ausführbaren Programm linkt.

Aufgabe 2: (Lineare Ausgleichprobleme und QR-Zerlegung)

i) Schreiben Sie ein Programm, das Ihnen zu einer gegebenen $m \times n$-Matrix A mit $m \geq n$ die QR-Zerlegung $A = QR$ mit einer orthogonalen Matrix $Q \in \mathbb{R}^{m\times m}$ und einer rechten oberen Dreiecksmatrix $R \in \mathbb{R}^{m\times n}$ mit Hilfe von Householder-Reflexionen berechnet. Programmieren Sie dazu eine Funktion

int Householder(double** A, double* alpha, int n, int m).

Die Rückgabe von Q und R erfolgt in kompakter Speicherweise in der Matrix A wie in der Vorlesung, wobei statt Q die Householder-Vektoren gespeichert werden. Speichern Sie die Diagonalelemente der Matrix R im Vektor *alpha*. Die Funktion *Householder* gibt den Wert 0 zurück, wenn die QR-Zerlegung berechnet wurde, andernfalls den Wert 1.
Testen Sie Ihr Programm mit der Matrix

$$A = \begin{pmatrix} 1 & -2 & 1 \\ 1 & 6 & -2 \\ 1 & 0 & -2 \\ 1 & 6 & -2 \\ 0 & \sqrt{7} & \frac{11}{\sqrt{7}} \end{pmatrix}$$

und geben Sie R sowie die Householder-Vektoren aus.

ii) Erweitern Sie Ihr Programm zur QR-Zerlegung, so dass der Rechner Ihnen die Lösung von linearen Ausgleichsproblemen $\min_x \|Ax - b\|_2$ mit $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ und $m \geq n$ bestimmt. Schreiben Sie dazu eine Funktion

int Solve_QR(double** A, double* alpha, int n, int m, double* b), wobei im Vektor $alpha$ die Diagonaleinträge von R gespeichert seien.

Lösen Sie mit Hilfe Ihres Programms das folgende Ausgleichsproblem:
Die Qualität von Weizenkörnern ist durch den Proteingehalt und den Stärkegehalt bestimmt. In der folgenden Tabelle sind die Ergebnisse der Untersuchung von 20 Proben zusammengestellt. Im Gegensatz zum Stärkegehalt x lässt sich der Proteingehalt y nur sehr aufwendig bestimmen, so dass man diesen in Abhängigkeit von x annähern möchte. Versuchen Sie den Proteingehalt durch eine quadratische Funktion $y(x) = ax^2 + bx + c$ des Stärkegehalts zu approximieren, indem Sie $a, b, c \in \mathbb{R}$ bestimmen, so dass die Summe der Quadrate der Abweichungen minimal ist. Geben Sie R, die Householder-Vektoren sowie die Lösung für a, b und c aus.

Stärkegehalt	Proteingehalt	Stärkegehalt	Proteingehalt
6	11.8	36	11.7
75	13.7	97	18.5
87	16.0	74	15.3
55	12.6	24	11.6
34	12.4	85	15.9
98	19.6	96	17.3
91	15.5	92	17.1
45	12.3	94	16.5
51	12.9	84	14.8
17	12.5	99	20.5

Die Tabelle „Messdaten.c“ können Sie sich von der GRIPS-Seite herunterladen.