Institut für Numerische und Angewandte Mathematik 11.12.2014
FB Mathematik und Informatik der Universität Münster
Prof. Dr. M. Ohlberger, Dipl.-Math. C. Himpe

Übung zur Vorlesung

Numerische Lineare Algebra

Wintersemester 2014/2015 — Blatt 9

Abgabe: 18.12.2014, vor der Vorlesung

Aufgabe 1 (Mehrdimensionales Newton-Verfahrens) (4 Punkte)
Bestimmen Sie approximativ einen Extremwert der Funktion:

$$f(x, y) = x^3 + y^3 - 3xy,$$

mit Hilfe des Newton Verfahrens durch Berechnung der ersten drei Iterationsschritte mit Startvektor $\begin{pmatrix} 2 & 2 \end{pmatrix}^T$.

Aufgabe 2 (Modifizertes Newton-Verfahren) (4 Punkte)
Zur Berechnung von $\sqrt{a}$, $a > 0$ kann man das Newtonverfahren einsetzen, z.B. mit $f_0(x) = x^2 - a$ oder mit $f_1(x)\frac{a}{x^2} - 1$. Sei Φ_i die zugehörige Iterationsfunktion zu $f_i(i = 0, 1)$. Betrachten Sie die Iterationsfunktion Φ_s, die durch Linearkombination von Φ_0, Φ_1 entsteht, d.h. $\Phi_s(x) = (1 - s)\Phi_0 + s\Phi_1(x)$.

a) Zeigen Sie, dass das Verfahren $x^{(k+1)} = \Phi_s(x^{(k)})$ mindestens quadratisch gegen $\sqrt{a}$ konvergiert.

b) Zeigen Sie, dass es ein s_a gibt für das das Verfahren kubisch (von 3. Ordnung) konvergiert. Dabei sollte s_a nur von a, nicht aber von $\sqrt{a}$ abhängen.

Aufgabe 3 (Konvergenz des Sekanten-Verfahren) (4 Punkte)
Sei $f \in C^2([a, b])$ mit $x^* \in (a, b)$, $f(x^*) = 0$ und $m := \min_{x\in[a,b]} |f'(x)| > 0, M := \max_{x\in[a,b]} |f''(x)|$. Sei $q := \frac{M}{2m}\rho < 1$ für ein $\rho > 0$. Sei $x^{(0)}, x^{(1)} \in B_\rho(x^*)$ mit $x^{(0)} \neq x^{(1)}$ und $x^{(k+1)}$, $k \geq 1$ definiert durch das Sekantenverfahren gemäß

$$x^{(k+1)} = x^{(k)} - f(x^{(k)})\frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})}$$

Zeigen Sie:

a) Für alle k gilt $x^{(k)} \in B_\rho(x^*)$ und $\lim_{k\to\infty} x^{(k)} = x^*$.

b) Für alle k gilt die a-priori Fehlerschranke $|x^{(k)} - x^*| \leq \frac{2m}{M} q^{\gamma_k}$, wobei $(\gamma_k)_{k\in\mathbb{N}}$ die Folge der Fibonacci-Zahlen ist, d.h. $\gamma_0 = \gamma_1 = 1, \gamma_{k+2} = \gamma_{k+1} + \gamma_k$.

c) Für alle $k \geq 1$ gilt die a-posteriori Fehlerschranke

$$\left|x^{(k)} - x^*\right| \leq \frac{1}{m}|f(x^{(k)})| \leq \frac{M}{2m}|x^{(k)} - x^{(k-1)}| \cdot |x^{(k)} - x^{(k-2)}|.$$

Hinweis: Zeigen Sie zunächst, dass gilt $\left|\frac{f(y)-f(x)}{y-x} - \frac{f(z)-f(x)}{z-x}\right| \leq \frac{M}{2}|y - z|$.

Aufgabe 4 (Programmieraufgabe: Kombination von Newton-Verfahren und Intervallschachtelung) (4 Punkte)

a) Schreiben Sie ein Programm zur Nullstellensuche einer Funktion f, welches auf der Kombination von Newton-Verfahren und Intervallschachtelung basiert.
Als Eingabeparameter sollte Ihre Routine die Intervallgrenzen a und b eines Startintervalls $[a, b]$, eine Funktion $f : \mathbb{R} \to \mathbb{R}$, deren Ableitung $f' : \mathbb{R} \to \mathbb{R}$ sowie eine Toleranz $tol > 0$ für das Abbruchkriterium $|f(x^{(k)})| \leq tol$ erwarten.
Die Rückgabewerte sollten die berechnete Näherungslösung $\bar{x} \in \mathbb{R}$ sowie die Anzahl der durchgeführten Iterationen (a) des Intervallschachtelungsverfahrens und (b) des Newton-Verfahrens sein.

b) Testen Sie die Funktionalität Ihrer Routine, mit der stetig differenzierbaren Funktion $f : \mathbb{R} \to \mathbb{R}$,

$$f(x) = \begin{cases} -1, & \text{für } x < 0, \\ (12(x-1) - 4(x-1)^3)/8 & \text{für } 0 \leq x \leq 2, \\ 1 & \text{für } x > 2. \end{cases}$$

Sie benötigen dazu Funktionen, welche die Funktion f bzw. deren Ableitung f' implementieren. Plotten Sie die Funktion f im Intervall $[-1, 3]$. Verwenden Sie für das Startintervall $a = 0$ und $b = 100$ und als Toleranz $tol = 10^{-15}$. Geben Sie den Wert der Näherungslösung und die Anzahl der durchgeführten Iterationen (Intervallschachtelung / Newton) aus. Plotten Sie den logarithmischen absoluten Fehler (exakte Lösung: $x = 1$) gegen die Anzahl der Iterationen.

Lösung der Nikolausaufgabe:

```
B = numpy.array([[0,0,11,9.7,9.9],
                 [0,8.7,10.1,10.5,10.4],
                 [0,0,0,0,11],
                 [7,11.4,11.1,10.4,10.1],
                 [0,0,0,10.4,11.6]])

print numpy.array(numpy.abs(numpy.linalg.qr(10*B))[1],dtype='uint8').view('c')
```

Die Wertespanne der Matrix liegt zwischen 7 und 12, multipliziert mit 10 also im Bereich der Buchstaben in der ASCII Zeichenkodierung. Nun muss man den Buchstabensalat noch ordnen. Bei genauem Hinsehen ist die Matrix eine (Zeilen-)permutierte rechte obere Dreiecksmatrix, daher erledigt dies eine QR- oder LR-Zerlegung. Zuletzt muss nur noch der Datentyp von float/double in 8-bit Ganzzahlen konvertiert werden, da diese isomorph zur ASCII Kodierung sind, und als eben diese ausgegeben werden.