Technische Universität München
Institut für Informatik
Prof. Dr. Thomas Huckle
Dipl.-Tech. Math. Stefanie Schraufstetter
Michael Lieb, M. Sc.

SS 2010

Numerisches Programmieren, Übungen

9. Übungsblatt: Fixpunktiteration

1) Algorithmus zur Quadratwurzelberechnung

Wie bestimmt man möglichst genau den Zahlenwert von $\sqrt{a}$? Diese Frage stellten sich schon die alten Babylonier. Ihre Antwort war ein iteratives Verfahren, das heute als *babylonisches Wurzelziehen* bekannt ist und immer noch die Grundlage für Taschenrechner-Algorithmen darstellt. Die Idee hinter dem Verfahren ist geometrischer Natur: Wenn ich ein Quadrat mit Flächeninhalt a hätte, dann wäre seine Seitenlänge $\sqrt{a}$. Um ein solches Quadrat zu erhalten, startet man mit einem Rechteck mit einer Breite von $b_0 = a$ und einer Höhe von $h_0 = \frac{a}{b_0} = 1$. Dieses hat also einen Flächeninhalt der Größe a. Dieses Rechteck wird nun sukzessive, einer quadratischen Form angenähert. Dabei bleibt der Flächeninhalt aber konstant a. Zu diesem Zweck wählt man als neue Breite des Rechtecks genau das arithmetische Mittel von alter Breite und alter Höhe und passt anschließend die neue Höhe an (vgl. Abbildung 1). In Formeln ausgedrückt liefert dies die Iterationsvorschrift

$$\begin{aligned} & b_0 := a \; ; \; h_0 := 1 \\ & b_{k+1} := \frac{1}{2}\left(b_k + \frac{a}{b_k}\right) \; ; \; h_{k+1} := \frac{a}{b_{k+1}} \quad k \in \mathbb{N}_0 \; . \end{aligned} \tag{1}$$

Die Rechteckbreiten und Rechteckhöhen werden sich durch diese Iterationsvorschrift immer ähnlicher und nähern sich beide tatsächlich dem Grenzwert $\sqrt{a}$.

i) Die obige Abbildung zeigt, dass man schon nach wenigen Iterationen gute Näherungswerte erhält. Sie zeigt aber auch, dass die gewählten Startwerte noch weit vom gesuchten Wert entfernt sind. Bei modernen Taschenrechner-Algorithmen ist vor allem die Wahl des Startwertes verbessert. Entwickeln Sie eine Formel, die für eine beliebige Zahl a aus dem Intervall $[1; 4[$ einen besseren Startwert b_0 für die Breite des Rechtecks liefert!

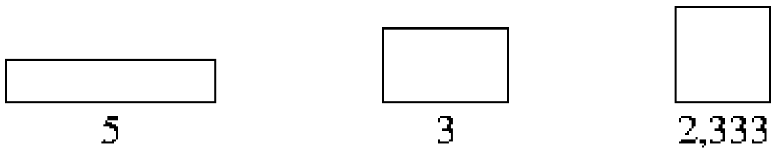

Abbildung 1: Approximation von $\sqrt{5}$ durch babylonisches Wurzelziehen

ii) Bei heutigen Taschenrechnern hat sich die Iterationsvorschrift nicht geändert. Es ist allerdings ausreichend, alleine die Rechteckbreite zu iterieren. Wir wollen zeigen, dass die Iterationsvorschrift

$$b_{k+1} := \frac{1}{2}\left(b_k + \frac{a}{b_k}\right) \tag{2}$$

auch mit Hilfe des Newton-Verfahrens hergeleitet werden kann. Dazu stellen wir fest, dass $\sqrt{a}$ Nullstelle der Funktion $f(x) := x^2 - a$ ist. Stellen Sie nun die Iterationsvorschrift für das Newtonverfahren zur Nullstellenbestimmung von f auf! Diese sollte Formel (5) gleichen.

iii) Um eine taugliche Taschenrechner-Funktion zu erhalten, müssen wir uns noch Gedanken machen, wie man für Zahlen außerhalb des Intervalls $[1;4[$ vorzugehen hat. Geben Sie eine Möglichkeit an, wie man das Problem *"Finde $\sqrt{a}$ für $a \notin [1;4[$ "* auf das Problem *"Finde $\sqrt{a}$ für $a \in [1;4[$ "* zurückführen kann!

iv) Betrachten Sie nun die Iterationsfunktionen

$$\Phi_1(x) := \frac{x^3}{a} \quad \text{und} \quad \Phi_2(x) := \frac{a}{x} \tag{3}$$

und zeigen Sie, dass die beiden Funktionen ebenfalls den Fixpunkt $\sqrt{a}$ haben. Eignen sich diese ebenfalls zur iterativen Berechnung von $\sqrt{a}$? Begründen Sie Ihre Antwort!

2) Banach'scher Fixpunktsatz

Gegeben sei die auf $\mathbb{R}$ definierte Funktion

$$\Phi(x) := x(x^2 - 1) + x.$$

i) Bestimmen Sie alle Fixpunkte von Φ und entscheiden Sie, ob es sich um anziehende oder abstoßende Fixpunkte handelt.

ii) Geben Sie ein Intervall I an, so dass alle Bedingungen des Banach'schen Fixpunktsatzes für die Funktion Φ erfüllt sind. Zeigen Sie, dass auch tatsächlich alle Bedingungen erfüllt sind. Was lässt sich daraus folgern?

Zum Abschluss des Kapitels über Fixpunktiteration gibt es auch dieses Mal eine Wiederholungsaufgabe. Die Aufgabe ist der Semestralklausur des SS 2008 entnommen.

Wiederholung: Fixpunktiteration

Gegeben sei eine Fixpunktiteration mit Startwert $x_0 \in \mathbb{R}$, definiert durch die Funktion

$$\Phi(x) = \frac{1}{2}\left(x + \frac{1}{x}\right). \tag{4}$$

i) Geben Sie die Iterationsvorschrift zur Berechnung der Folge x_k, $k = 1, 2, \ldots$, an!

ii) Welche Fixpunkte hat die Funktion Φ? Geben Sie jeweils an, ob es sich um einen anziehenden oder abstoßenden Fixpunkt handelt.

iii) Skizzieren Sie den Graphen der Funktion $\Phi(x)$ für $x > 0$. Bestimmen Sie dazu zunächst das Verhalten von $\Phi(x)$ für $x \to 0$ und $x \to +\infty$ sowie die Extremstellen und deren Art.

iv) Im Folgenden sollen weitere Eigenschaften der Iterationsfunktion Φ bzw. der zugehörigen Iterationswerte x_k, $k = 1, 2, \ldots$, gezeigt werden:

   (1) Man begründe: $\Phi(x) \geq 1 \quad \forall\, x > 0$

   (2) Man zeige: $x_0 > 0 \quad \Rightarrow \quad x_k \geq 1 \quad \forall\, k = 1, 2, \ldots$

   (3) Man zeige: $x_0 > 0 \quad \Rightarrow \quad x_k \geq x_{k+1} \quad \forall\, k = 1, 2, \ldots$

   Was folgt daraus für die Konvergenz von x_k, falls $x_0 > 0$?

v) Nun betrachten wir die Funktion

$$f(x) = x - \frac{1}{x}. \tag{5}$$

   Zeigen Sie die Äquivalenz

$$f(x) = 0 \quad \Leftrightarrow \quad \Phi(x) = x.$$

   Welche Nullstellen hat f?

vi) Formulieren Sie das Newton-Verfahren zur Bestimmung einer Nullstelle der Funktion f aus Teilaufgabe v)!

Auch zum Kapitel über Fourier-Transformation gibt es noch eine Wiederholungsaufgabe, die wieder zum Selbststudium gedacht ist. Hier können Sie Ihre bisherigen Kenntnisse über komplexe Zahlen und die Fourier-Transformation anwenden und den FFT-Algorithmus an einem Beispiel nachvollziehen.

Wiederholung: Fourier-Transformation

In dieser Aufgabe wollen wir die schnelle Fouriertransformation (FFT) im Detail nachvollziehen und mit der direkten Formel der inversen diskreten Fourier-Transformation ($IDFT$) vergleichen. Der rekursive FFT-Algorithmus lautet:

```
FUNCTION [v[0],...,v[n-1]]=IDFT(c[0],...,c[n-1],n)
if n==1
   v[0] = c[0];
else
   m = n/2;
   z1 =IDFT(c[0],c[2],...,c[n-2],n/2);
   z2 = IDFT(c[1],c[3],...,c[n-1],n/2);
   omega = exp(2*pi*i/n);
   for j=0,...,m-1
      v[j]    = z1[j] + omega^j * z2[j];
      v[m+j]  = z1[j] - omega^j * z2[j];
   end
end
return;
```

Der Algorithmus kann wie in der Vorlesung durch eine Sortierphase und eine Kombinationsphase (sukzessive Anwendung des Butterfly-Operators) visualisiert werden (vgl. Abb. 2 und Abb. 3).

Der Einfachheit halber soll nun $n = 4$ sein, d.h. wir wollen mit der $IDFT$ aus den vier Koeffizienten c_0, c_1, c_2, c_3 die zugehörigen vier Funktionswerte v_0, v_1, v_2, v_3 berechnen.

Die direkte Formel dazu lautet ja

$$v_j \;=\; \sum_{k=0}^{n-1} c_k \omega^{jk}, \quad j = 0, \ldots, n-1. \tag{6}$$

i) Berechnen Sie die v_j zunächst nach der direkten Formel (6)!

ii) Verwenden Sie nun den FFT-Algorithmus, um zu zeigen, dass man damit tatsächlich dasselbe Ergebnis wie in i) erhält!

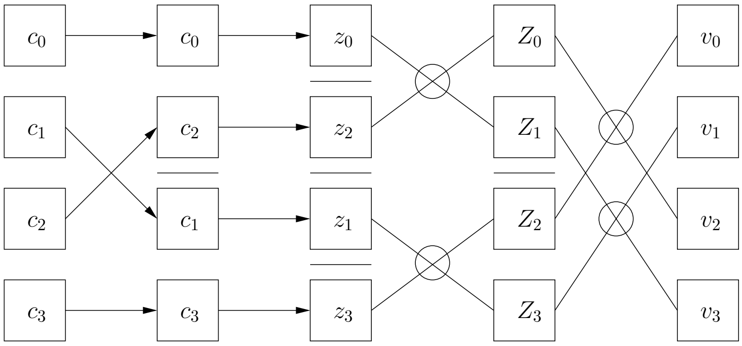

Abbildung 2: Schematischer Ablauf des FFT-Algorithmus.

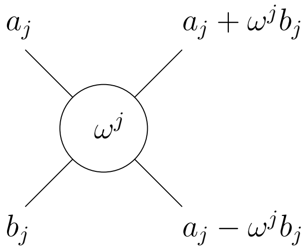

Abbildung 3: Der Butterfly-Operator.