Algorithmische Mathematik I

Wintersemester 19/20
Prof. Dr. J. Gedicke
Johannes Rentrop und Jannik Schürg

UNIVERSITÄT BONN

Übungsblatt 5.

Abgabedatum: **18.11.2019**

Aufgabe 1. (Konvergenzordnung)

In Programmieraufgabenblatt 4, Aufgabe 1 haben wir die Konvergenzordnung zweier iterativer Algorithmen untersucht. Hier wollen wir weitere Beispiele betrachten.
Allgemein lautet die Definition wie folgt. Sei $(x_n)_{n\in\mathbb{N}}$ eine konvergente Folge reeller Zahlen mit $x_n \to x^*$ für $n \to \infty$.

- Existiert eine Konstante $0 < c < 1$ mit

  $$|x_{n+1} - x^*| \leq c|x_n - x^*| \quad , \quad \forall n > n_0,$$

  so hat die Folge mindestens Konvergenzordnung 1 oder *lineare Konvergenz*.

- Gilt $|x_{n+1} - x^*| \leq c_n|x_n - x^*|$ mit $c_n \to 0$ für $n \to \infty$, so spricht man von *superlinearer Konvergenz*.

- Sei $p > 1$. Existiert eine Konstante $c > 0$, sodass

  $$|x_{n+1} - x^*| \leq c|x_n - x^*|^p,$$

  so hat die Folge mindestens Konvergenzordnung p.

Geben Sie für diese Folgen die Konvergenzordnung an.

a) $x_n := (1+\epsilon)^{-n}$ mit festem $\epsilon > 0$ (geometrische Folge)

b) $x_n := 2^{-2^{n/2}}$

c) $x_n := 1/n^2$

d) $x_{n+1} := \phi(x_n)$ mit $x_0 = 1/3$ und $\phi(x) = x/\ln(1/x)$

e) $x_{n+1} := 2 + (x_n - 1)^{1/(2x_n-4)}(x_n - 2)^2$ mit $x_0 = 5/2$

(1+1+1+2+2 Punkte)

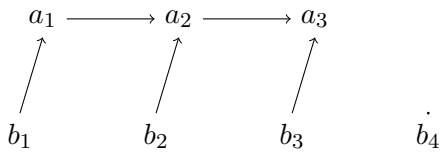
Aufgabe 2. (Optimale Suchalgorithmen)

Wir wollen Suchalgorithmen hinsichtlich der Anzahl der benötigten Vergleiche untersuchen. Sei $S(n)$ die minimale, im schlimmsten Fall benötigte Anzahl von Vergleichen unter allen möglichen Sortieralgorithmen zum sortieren einer n-elementigen Liste. In der Vorlesung wurde gezeigt, dass

$$S(n) \geq \lceil \log_2 n! \rceil.$$

Für $n = 7$ ergibt sich also $S(7) \geq 13$. Wir nehmen der Einfachheit halber an, dass alle Elemente paarweise verschieden sind.

a) Zeigen Sie per expliziter Konstruktion, dass ein Algorithmus existiert, der im schlimmsten Fall genau 13 Vergleiche benötigt. Gehen Sie dabei wie folgt vor: Zunächst werden aus den ersten sechs Elementen drei Paare gebildet und diese intern verglichen (3 Vergleiche insgesamt). Danach werden die drei größeren Elemente in die richtige Reihenfolge gebracht, indem das erste mit dem zweiten, das erste mit dem dritten, und wenn nötig das zweite mit dem dritten Element verglichen wird (im schlimmsten Fall 3 Vergleiche). Dadurch erhält man eine Situation, die durch den Graphen



beschrieben werden kann, wobei eine gerichtete Kante von Element e nach f bedeutet, dass $e < f$. Wie muss weiter vorgegangen werden, um selbst im schlimmsten Fall mit 7 zusätzlichen Vergleichen alle Elemente in die richtige Reihenfolge zu bringen?

Bemerkung: Dieser Algorithmus ist also für den schlimmsten Fall optimal hinsichtlich der benötigten Anzahl an Vergleichen, für $n = 7$. Es gibt n, für die bekannt ist, dass eine allgemeine Version dieses Algorithmus immernoch schlechter als die theoretische untere Schranke ist, man aber nicht weiß, ob ein besserer Algorithmus existiert.

b) Betrachten Sie die unsortierte Liste $(4, 1, 2, 5, 7, 6, 3)$. Nutzen Sie einmal den Algorithmus aus a) und einmal Mergesort (wie im Beispiel aus der Vorlesung), um die Liste zu sortieren. Bestimmen Sie dabei die tatsächlich benötigte Anzahl an Vergleichen. Kommentieren Sie das Ergebnis.

(4+2 Punkte)

Aufgabe 3. (Fehlerrechnung)

In dieser Aufgabe geht es um klassische Fehlerrechnung.

a) Sei $\varepsilon_1, \ldots, \varepsilon_n \in \mathbb{R}$ mit $|\varepsilon_i| \leq \varepsilon < 1/n$ und seien $\sigma_1, \ldots, \sigma_n \in \{-1, 1\}$. Definiere θ_n durch

$$1 + \theta_n \coloneqq \prod_{k=1}^{n} (1 + \varepsilon_k)^{\sigma_k}.$$

Zeigen Sie, dass dann gilt

$$|\theta_n| \leq \frac{n\varepsilon}{1 - n\varepsilon} =: \gamma_n.$$

Evtl. könnten die Ungleichungen

$$x \in \mathbb{R} \Rightarrow 1 + x \leq e^x \quad \text{und} \tag{1}$$

$$x \in [0, 1) \Rightarrow e^x - 1 \leq \frac{x}{1 - x} \tag{2}$$

hilfreich sein.

Wenn Sie bis einschließlich 13.11. die Exponentialfunktion noch nicht in Analysis I kennengelernt haben, müssen Sie die Ungleichungen bei Verwendung nicht zeigen.

b) Zeigen Sie, dass für γ_n aus a) gilt $\gamma_n \leq \gamma_{n+1}$, falls $(n + 1)\varepsilon < 1$.

c) Sei F ein Maschinenzahlbereich mit Maschinenzahlgenauigkeit ε. Algorithmus 1, SUMME($\boldsymbol{a}$), auf diesem Blatt berechnet die Summe eines Vektor $\boldsymbol{a} = (a_1, \ldots, a_n) \in F^n$ in Fließkommaarithmetik. Zeigen Sie, dass für n nicht zu groß

$$\left| \text{SUMME}(\boldsymbol{a}) - \sum_{i=1}^{n} a_i \right| \leq \gamma_{n-1} \sum_{i=1}^{n} |a_i|$$

gilt, wobei γ_{n-1} definiert ist wie in a).

Algorithmus 1 Berechnet in Fließkommarithmetik die Summe eines Vektors

```
1: function SUMME(a)
2:     s ← a_1
3:     for i ← 2, ..., n do
4:         s ← rd(s + a_i)
5:     end for
6:     return s
7: end function
```

(3+1+3 Punkte)