

# Algorithmen im Mathematikunterricht

## 0 Literatur

- [Ba] A.B. Barth: *Algorithmik für Einsteiger*. Vieweg, 2003.
- [Be] Behrends: *Introduction to Markov-Chains*. Vieweg, 2003.
- [BSW] A. Beutelspacher, J. Schwenk, K-D. Wolfenstetter: *Moderne Verfahren der Kryptographie*, 4. Auflage. Vieweg, 2001.
- [Devaney] **Hier müssen seine Bücher rein!!!**
- [Er] H. Ernst: *Grundkurs Informatik*, 3. Auflage. Vieweg, 2003.
- [Fi] G.A. Fink: *Mustererkennung mit Markov-Modellen*. Teubner 2003.
- [Fis] G. Fischer: *Lineare Algebra*, 14. Auflage. Vieweg, 2003.
- [Ga] F. Garvan: *The Maple Book*. Chapman & Hall / CRC, 2002.
- [Ha] R. Haggarty: *Diskrete Mathematik für Informatiker*. Pearson Studium, 2004.
- [Hä] O. Häggström: *Finite Markov Chains and Algorithmic Applications*. Oxford University Press, 2002.
- [Le] A.M.Lesk: *Bioinformatik*, eine Einführung. Spektrum Akademischer Verlag, 2003.
- [Kno] M. Knorrenschild: *Numerische Mathematik*. Fachbuchverlag Leipzig, 2003.
- [Kü] H.R. Künsch: *Stochastische Simulation*. Skript zur Vorlesung im WS 03/04. ETH Zürich, 2003.
- [Me] M.L. Mehta: *Random Matrices*, Second Edition. Academic Press 1991.
- [Moe] O. Moeschlin et al: *Experimental Stochastics*. Second Edition. Springer 2003.
- [MRRIT] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller: *Equation of state calculations by fast computing systems*. *Journal of Chemical Physics*, vol. 21, pp.1087–1092.
- [Na] I.T. Nabney: *Netlab, Algorithms for Pattern Recognition*. Springer, 3rd printing 2004.
- [Ne] R.M. Neal: *Bayesian Learning for Neural Networks*. Springer, 1996.
- [P] H.-O. Peitgen, H. Jürgens, D. Saupe, E.M. Maletsky, T. Perciante, L. Yunker: *Chaos*. Springer, Klett, 1992.
- [Re] P. Rechenberg: *Was ist Informatik?* Hanser, 2000.
- [Sch] J. Schwank: *Sicherheit und Kryptographie in Informatik*. Vieweg, 2002.
- [StGr] H. Stoppel, B. Griese: *Übungsbuch zur Linearen Algebra*, 4. Auflage. Vieweg, 2003.
- [Stp] H. Stoppel: *Mathematik anschaulich, Brückenkurs mit Maple*. Oldenbourg Wissenschaftsverlag, 2002.
- [Stp2] H. Stoppel: *Nichtlineare Dynamik und Mandelbrotmenge*. Skriptum zu Mathematik AG, 2002.
- [Ti] G. Tischel: *Analysis Leistungskurs*. Diesterweg 1994.

# 1 Einführung in die Algorithmik

## 1.1 Einleitung

Der vermutlich bekannteste Algorithmus ist der Algorithmus der Addition zweier natürlicher Zahlen, der hier an einem Beispiel dargestellt werden soll:

$$\begin{array}{r} 18 \\ 33 \\ \hline 51 \end{array}$$

Die Einerziffern werden addiert. Wenn die Summe  $\geq 10$  ist, dann wird die Einerziffer des Ergebnisses notiert, und die Zehnerziffer des Ergebnisses wird in die Zehnerpalte übertragen. Dies wird analog Spalte für Spalte von rechts nach links gemacht, und so erhält man das Gesamtergebnis.

Es stellt sich hier die Frage, wodurch sich der „Algorithmus“ auszeichnet, und wie man den Begriff so allgemein fassen kann, dass er auf alle betrachteten Vorgänge angewandt werden kann. Anschaulich gesehen findet die Eingabe von zwei Zahlen in eine Maschine statt, die sie sodann verarbeitet und wieder ausgibt:



Hier seien zunächst zwei einfache Definitionen betrachtet. Die erste lässt in der Jahrgangsstufe 11 im Informatikunterricht verwenden, die zweite stammt aus [Ba]:

**Definition 1.** Ein **Algorithmus** ist ein automatisiertes Verfahren, welches einen Input zu einem Output verarbeitet. Die Verarbeitung geschieht

1. in endlich vielen Schritten, von denen jeder in endlicher Zeit abgeschlossen ist, und
2. so, dass jeder Schritt aus einer eindeutig formulierten (unmissverständlichen) und ausführlichen Anweisung besteht.

Als Anweisungen kommen in Frage:

- einfache Anweisungen der Form **Tue dies:** \_\_\_\_\_ (*Sequenz*)
- Verzweigungen der Form **Falls *Bedingung* tue dies:** \_\_\_\_\_  
(*Verzweigung*)  
(sonst tue das: \_\_\_\_\_)
- Schleifen der Form **Für  $i = 1, \dots, n$  tue dies:** \_\_\_\_\_ (*Schleifen*)  
oder **Wiederhole dies:** \_\_\_\_\_ **bis *Bedingung* eintritt**  
oder **Solange *Bedingung* gilt, wiederhole dies:** \_\_\_\_\_

**Definition 2.** Ein **Algorithmus** ist eine Verarbeitungsvorschrift, die aus einer endlichen Folge von eindeutig ausführbaren Anweisungen besteht, mit der man eine Vielzahl gleichartiger Aufgaben lösen kann.

Ein Algorithmus gibt an, wie Eingabegrößen schrittweise in Ausgabegrößen umgewandelt werden.

Diese Definitionen sind nicht genau genug, da sie keine exakte Beschreibung des Vorgangs geben. Durch obige Skizze und die erste Definition des Algorithmus wird jedoch grob beschrieben, was der exakten Definition von Algorithmen zugrunde liegt, die auf dem Begriff der **Turing-Maschine** aufbaut.

## 1.2 Die Turing-Maschine

Bei der Turing-Maschine handelt es sich – wie der Name bereits sagt – um eine Maschine. Während man sich heute unter einer Maschine einen Computer vorstellt, handelt es sich bei einer Turing-Maschine hingegen um eine mathematische Maschine. Sie wurde vom englischen Mathematiker ALAN M. TURING (1912–1956) entwickelt. Er studierte in den dreißiger Jahren des zwanzigsten Jahrhunderts die mathematische Berechenbarkeit und beschrieb 1936 eine ganz einfache mathematische Maschine, mit der sich alle Algorithmen ausführen lassen. Hiermit lässt sich dann definieren:

**Definition.** Ein **Algorithmus** ist eine (anhaltende) Turing-Maschine (TM).

Es ist an dieser Stelle zu bemerken, dass es keinen „vorstellbaren“ Algorithmus gibt, der nicht durch eine TM darstellbar ist. Und es gibt auch andere Modelle für Maschinen gibt, z.B. die *Maschine mit wahlfreiem Zugriff*. Es sind jedoch bis heute keine Vorgänge in der Mathematik bekannt, die sich nicht durch eine TM, jedoch durch ein anderes Modell darstellen lassen. Dies führt zu

**These von A. Church.** Alles, was intuitiv berechenbar ist, kann durch eine TM berechnet werden.

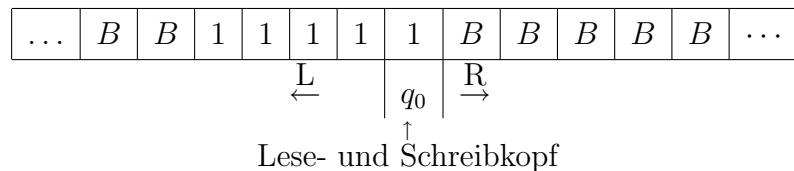
Nun widmen wir uns der Definition der TM, die für unsere Belange sinnvoll ist.

**Definition.** Eine **Turing-Maschine** besteht aus

- i) einem endlichen Alphabet  $A$  von Symbolen, die zur Ein- und Ausgabe von Informationen dienen,
- ii) einem endlichen Alphabet  $\bar{A} \supset A$  von Symbolen, die zur Berechnung der Zwischenschritte dienen,
- iii) einem Leerzeichen (blank)  $B \in \bar{A}$ ,
- iv) einem eindimensionalen (theoretisch unbegrenzten) Band mit Feldern, von denen jedes genau ein Symbol aus  $\bar{A}$  enthalten kann,
- v) einem beweglichen Lese/Schreibkopf, der ein Feld aufs Mal lesen und beschreiben kann und stets ein Feld nach links oder ein Feld nach rechts rücken oder stehen bleiben kann,
- vi) einer endlichen Menge  $Q$  sogenannter *Zustände* (deren Bedeutung später geklärt wird),
- vii) zwei speziellen Zuständen  $q_0, q_e \in Q$ , die *Anfangs-* und *Endzustand* heißen, und
- viii) einer Funktion  $\delta$ , die *Programm* heißt und einigen Paaren (Zustand, Symbol) ein Tripel aus einem Zustand, einem Symbol und einem der Zeichen  $L$  (für *Links*),  $R$  (für *Rechts*) und  $S$  (für *Stop*) zuordnet, mathematisch formuliert:

$$\delta: Q \times \bar{A} \rightarrow Q \times \bar{A} \times \{L, R, S\}, \quad (\text{Zustand, Symbol}) \mapsto (\text{Zustand, Symbol, Bewegung}).$$

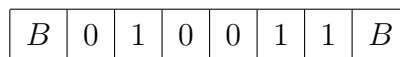
Ein zeichnerisches Modell zur Darstellung der TM kann man sich folgendermaßen vorstellen:



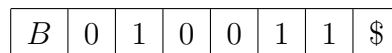
Hierbei handelt es sich um eine anhaltende TM, und es gibt den Zustand  $q_0$  sowie die Symbole  $A = \bar{A} = \{1, B\}$ .

Für die genauere Klärung folgt hier ein

**Beispiel 1.** Es seien hier  $\bar{A} = A = \{0, 1, \$, B\}$  und  $Q = \{q_0, q_1, q_e\}$ . Die Aufgabe besteht darin, aus dem Band



das Band



zu machen.

Das Programm besteht hierbei aus 6 Befehlen:

- $(q_0, 0) \mapsto (q_0, 0, R)$  Lies und schreibe 0 im Anfangszustand und gehe nach rechts.
- $(q_0, 1) \mapsto (q_0, 1, R)$  Lies und schreibe 1 im Anfangszustand und gehe nach rechts.
- $(q_0, B) \mapsto (q_0, \$, L)$  Lies  $B$  und schreibe  $\$$  im Anfangszustand und gehe nach links.
- $(q_1, 0) \mapsto (q_1, 0, L)$  Lies und schreibe 0 und gehe nach links.
- $(q_1, 1) \mapsto (q_1, 1, L)$  Lies und schreibe 1 und gehe nach links.
- $(q_1, B) \mapsto (q_e, \$, S)$  Lies  $B$  und schreibe  $\$$  und stopp das Programm.

Je nachdem, in welchem Feld die Maschine angesetzt wird, wird einer der ersten drei Schritte ausgeführt.

Wenn auf dem Feld 0 steht, bleibt 0 (Anfangszustand), und die Maschine rückt nach rechts.

Wenn dort 1 steht, bleibt 1 (Anfangszustand), und die Maschine rückt nach rechts.

Wenn die Maschine  $B$  im Anfangszustand vorfindet, überschreibt sie es mit  $\$$  und rückt nach rechts.

Nach diesem ersten Schritt werden die folgenden drei Schritte so lange wiederholt, bis einmal  $(q_1, B) \mapsto (q_e, \$, S)$  ausgeführt wird, der einen Endzustand herbeiführt und die Maschine stoppt.

**Bemerkung.** In diesem Algorithmus ist eine Schwäche: Was geschieht, wenn man auf der  $B$ -Stelle links startet?

**Aufgabe 1.** Verbessern Sie den Algorithmus aus Beispiel 1 so, dass die in der Bemerkung erwähnte Schwäche nicht mehr gilt.

*Lösung.* Die Schritte für  $q_0$  bleiben wie oben.

- $(q_1, 0) \mapsto (q_2, 0, L)$
- $(q_1, 1) \mapsto (q_2, 1, L)$
- $(q_1, B) \mapsto (q_2, B, R)$
- $(q_2, B) \mapsto (q_3, \$, R)$
- $(q_2, 0) \mapsto (q_3, 0, R)$
- $(q_2, 1) \mapsto (q_3, 1, R)$
- $(q_3, B) \mapsto (q_e, \$, S)$

Wir haben diesen Abschnitt mit dem Algorithmus der Addition begonnen. Dieser lässt sich mithilfe einer TM darstellen, wie in der nächsten Aufgabe zu zeigen ist.

**Aufgabe 2.** Es sollen zwei natürliche Zahlen addiert werden. Dazu sei  $\bar{A} = A = \{1, B\}$ , und eine Zahl  $n \in \mathbb{N}$  soll durch  $n + 1$  Einsen dargestellt werden. Stellen Sie eine TM dazu dar.

**Beispiel 2.** Hier wird die Erhöhung einer Dezimalzahl um 1 betrachtet werden. Nun werden die Zahlen nicht mehr nur durch Einsen dargestellt, sondern wir benutzen das Alphabet

$$A = \bar{A} = \{0, 1, 2, \dots, 9, B\}.$$

**Aufgabe 3.**  $\$010011\$BBBBBB \mapsto \$010011\$010011$

**Aufgabe 4.**  $\$x_1 \dots x_n\$y_1 \dots y_n \mapsto \$x_1 \dots x_n\$y_1 \dots y_n\$z_0 \dots z_n$

Ab jetzt werden nicht mehr alle Algorithmen mit der Turing-Maschine untersucht. Wir werden uns Algorithmen aus der Mathematik und der Anwendung im Unterricht widmen.

## 1.3 Erste Beispiele für Algorithmen

### 1.3.1 Induktion

Die Beweisform der Induktion ist eine grundlegende Strategie in der Mathematik. Sie fand auch in der Schulmathematik früher zahlreiche Anwendungen. Heute sind jedoch einige dieser Bereiche in den Richtlinien optional. Daher stellt sich die Frage, ob ihre Verwendung in der Schule notwendig ist.

**Definition. Induktion** Es sei  $n_0 \in \mathbb{Z}$  und  $I(n)$  eine für jedes  $n \geq n_0$  gültige Aussage. Um  $I(n)$  für alle  $n \neq n_0$  zu beweisen, ist folgendes zu zeigen:

- i) Es gilt  $I(n_0)$  (Induktions-Anfang).
- ii) Für alle  $n \neq n_0$ : falls  $I(n)$  gilt, so gilt auch  $I(n + 1)$  (Induktionsschritt, in Zeichen:  $n \rightarrow n + 1$ ).

In den folgenden Beispielen wird die Verwendung der Induktions-Voraussetzung durch  $\otimes$  gekennzeichnet.

**Beispiele. a)** Man zeige, dass  $\sum_{i=1}^n i = \frac{n(n+1)}{2}$  gilt.

Induktions-Anfang:  $n = 1 \Rightarrow \sum_{i=1}^1 i = \frac{1 \cdot 2}{2} = 1$

Induktions-Schritt  $n \rightarrow n + 1$ :

$$\sum_{i=1}^{n+1} i = \sum_{i=1}^n i + n + 1 \stackrel{\otimes}{=} \frac{n(n+1) + 2(n+1)}{2} = \frac{(n+2)(n+1)}{2}$$

**b)** Es gilt  $\sum_{i=1}^n (2i - 1) = n^2$

$n = 1 \Rightarrow \sum_{i=1}^1 (2i - 1) = 2 - 1 = 1 = 1^2$

$n \rightarrow n + 1$ :

$$\sum_{i=1}^{n+1} (2i - 1) = \sum_{i=1}^n (2i - 1) + 2(n + 1) - 1 \stackrel{\otimes}{=} n^2 + 2n + 1 = (n + 1)^2$$

c) Es gilt  $\sum_{i=0}^n x^i = \frac{1-x^{n+1}}{1-x}$ .

$$n=0 \Rightarrow \sum_{i=0}^0 x^i = x^0 = 1 = \frac{1-x^{0+1}}{1-x}$$

$n \rightarrow n+1$ :

$$\begin{aligned} \sum_{i=0}^{n+1} x^i &= \sum_{i=0}^n x^i + x^{n+1} \stackrel{\textcircled{*}}{=} \frac{1-x^{n+1}}{1-x} + \frac{(1-x)x^{n+1}}{1-x} \\ &= \frac{1-x^{n+1} + x^{n+1} - x^{n+2}}{1-x} = \frac{1-x^{n+2}}{1-x} \end{aligned}$$

### Anwendungsmöglichkeiten der Algorithmen in der Schulmathematik

a), b) Diese Beispiele finden z.B. in der Differential- und Integralrechnung Anwendung, wenn es um Intervallschachtelungen geht.

Beispiel c) findet unter Anderem in der Finanzmathematik Anwendung, die in der Schulmathematik verwendet werden kann. Dies bietet sich auch für fächerübergreifende Projekte der Mathematik mit Sozialwissenschaften oder Politik an.

**Aufgabe 1.** Zeigen Sie per Induktion:

a)  $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$

*Lösung.*  $n=1 \Rightarrow \sum_{i=1}^1 i^2 = 1^2 = \frac{1(1+1)(2+1)}{6}$

$n \rightarrow n+1$ :

$$\begin{aligned} \sum_{i=1}^{n+1} i^2 &= \sum_{i=1}^n i^2 + (n+1)^2 \stackrel{\textcircled{*}}{=} \frac{n(n+1)(2n+1)}{6} + \frac{6(n+1)^2}{6} \\ &= \frac{2n^3 + n^2 + 2n^2 + n + 6n^2 + 12n + 6}{6} = \frac{(n^2 + 3n + 2)(2n + 3)}{6} \\ &= \frac{(n+1)(n+2)(2(n+1) + 1)}{6} \end{aligned}$$

b)  $\sum_{i=1}^n i^3 = \frac{n^2(n+1)^2}{4}$

*Lösung.*  $n=1 \Rightarrow \sum_{i=1}^1 i^3 = 1 = \frac{1^2(1+1)^2}{4}$

$n \rightarrow n+1$ :

$$\begin{aligned} \sum_{i=1}^{n+1} i^3 &= \sum_{i=1}^n i^3 + (n+1)^3 \stackrel{\textcircled{*}}{=} \frac{n^2(n+1)^2}{4} + \frac{4(n+1)^3}{4} \\ &= \frac{(n^2 + 4(n+1))(n+1)^2}{4} = \frac{(n+2)^2(n+1)^2}{4} \end{aligned}$$

### 1.3.2 Multiplikation natürlicher Zahlen

Die Multiplikation wird bereits in der Primarstufe und zu Beginn der Sekundarstufe I verwendet. Dieser Algorithmus wird weiter unten noch benutzt und dort erläutert.

### 1.3.3 Euler-Algorithmus der Division

Ähnlich wie der Algorithmus der Multiplikation wird auch der Algorithmus der Division bereits in der Sekundarstufe I verwendet. Der Algorithmus in seiner eigentlichen Form und der allgemeinen Schreibweise wird jedoch in der Schule im Allgemeinen nicht eingeführt. Auch dieser Algorithmus wird später erläutert.

### 1.3.4 Fakultät

Anwendung findet sich in der Schulmathematik insbesondere in der Kombinatorik, dies unter Umständen bereits in der Jahrgangsstufe 10. Hierbei wird es auch mit den Binomialkoeffizienten verwendet. In der Oberstufe findet die Fakultät ebenfalls Anwendung in der Stochastik.

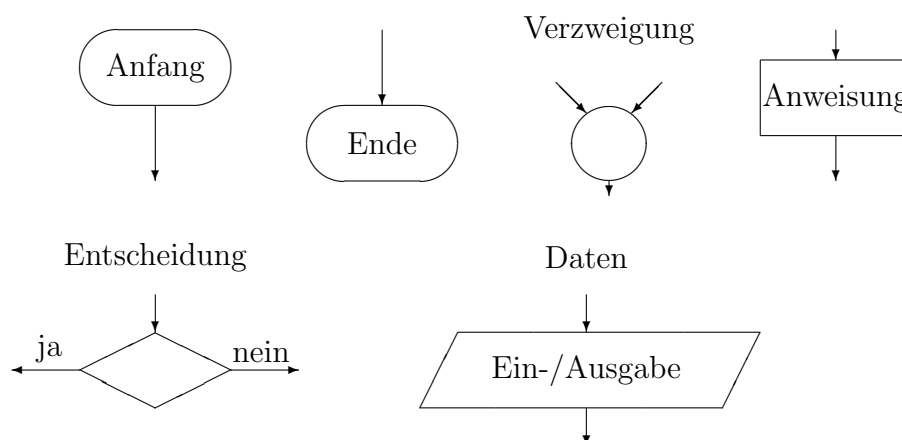
In der Informatik kann die Fakultät ebenfalls angewandt werden.

## 1.4 Flussdiagramme

### 1.4.1 Definition und erste Anwendung

**Flussdiagramme (Programmablaufpläne)** stellen Programme graphisch und vereinfacht dar. Sie geben *unabhängig von der Programmiersprache* die Wirkung und Vorgehensweise von Programmen wieder. Aufgrund der Unabhängigkeit von der Programmiersprache sind in den Flussdiagrammen nicht alle Schritte so durchgeführt wie in dem konkret umgesetzten Programm.

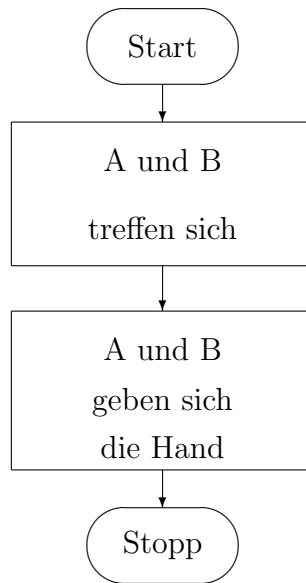
Die Flussdiagramme sind aus **Basisformen** zusammengesetzt:



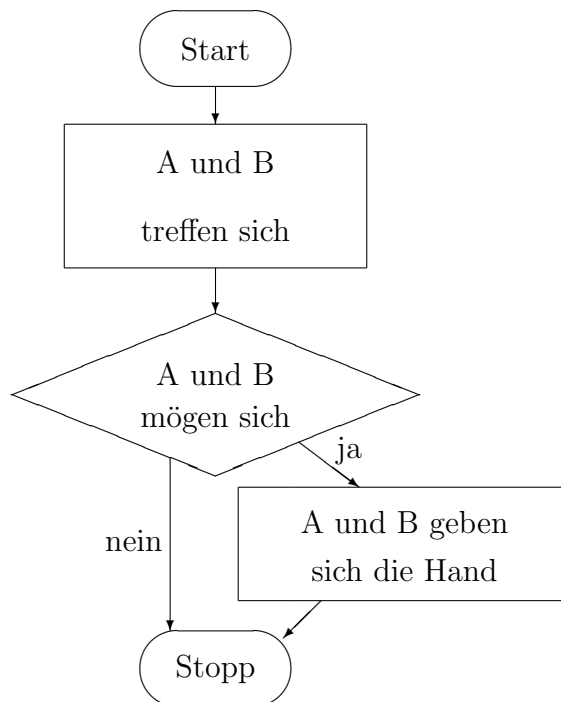
Die Flussdiagramme sollen zunächst an einigen einfachen Beispielen dargestellt werden. Es können bereits in der Jahrgangsstufe 8 Flussdiagramme im Mathematikunterricht eingeführt werden. Dies kann z.B. zur Erzeugung von Zufallszahlen geschehen (vgl. [VuW], S. 173ff oder Innenumschlag hinten). Eine weitere Möglichkeit besteht in der Untersuchung der Verwandtschaft von Vierecken (vgl. [MNW], S. 57ff). Zum Einstieg können daher einfache Beispiele betrachtet werden, mit denen man einen Einstieg zu *Wenn-dann-Beziehungen* gestalten kann, der in obigen Beispielen benötigt wird.



**Beispiele.** a) Es treffen sich zwei Personen, die sich bei ihrem Treffen die Hand geben. Das zugehörige Flussdiagramm kann folgendermaßen aussehen:



b) Das Beispiel a) soll dahingehend verbessert werden, dass die Personen sich nur dann die Hand geben, wenn sie sich auch mögen. Hier wird eine Entscheidung mit eingebaut:

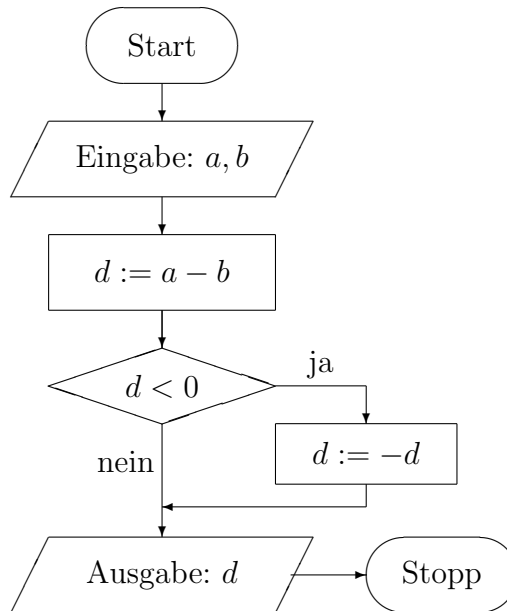


c) Es soll der Algorithmus zur Bestimmung des Betrags der Differenz zweier Zahlen

in einem Flussdiagramm dargestellt werden. Die Funktion ist definiert durch

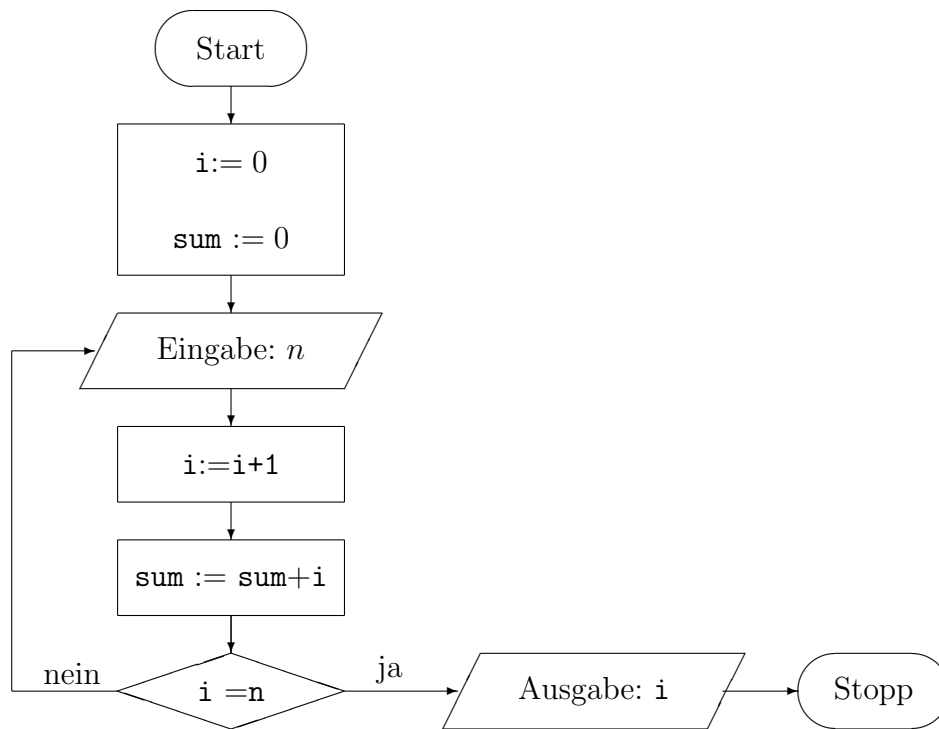
$$\text{abs}(a, b) = \begin{cases} a - b & \text{falls } a \geq b \\ b - a & \text{falls } a < b \end{cases}$$

Eine Möglichkeit für das Flussdiagramm ist gegeben durch



Selbstverständlich gibt es auch andere Möglichkeiten.

**d)** In dem Programm zum folgenden Flussdiagramm soll die Summe der Zahlen von 1 bis zur eingegebenen Zahl  $n \in \mathbb{N}$  berechnet werden. Hierzu muss die Zahl  $n$  eingegeben werden. Es lässt sich folgendermaßen darstellen:



**Aufgabe 1.** Notieren Sie Flussdiagramme zu den folgenden Algorithmen:

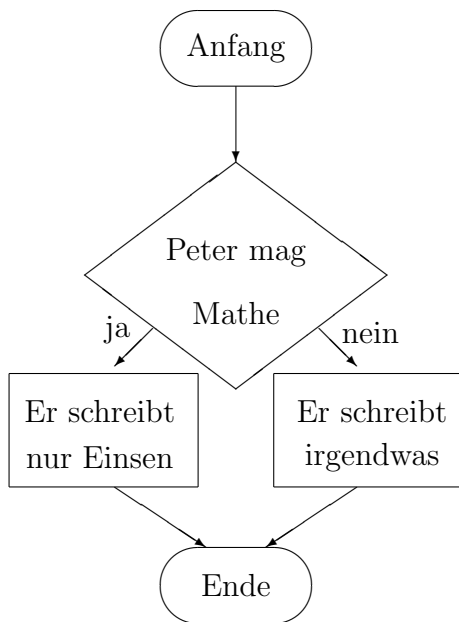
- „Wenn X der Täter ist, dann befand er sich zur Tatzeit am Tatort“ (aus [MNW8], S. 60, Aufgabe 11).
- Nach der Eingabe einer Zahl  $n \in \mathbb{N}$  soll die Summe von 1 bis  $n$  berechnet und geprüft werden, ob diese Summe gleich  $\frac{n(n+1)}{2}$  ist.
- Nach der Eingabe einer Zahl  $n \in \mathbb{N}$  soll  $n!$  berechnet werden.

**Aufgabe 2.** Entwickeln Sie in Gruppenarbeit mithilfe von geeigneten Schulbüchern ein Konzept zur Einführung und Anwendung von Flussdiagrammen in der Jahrgangsstufe 8.

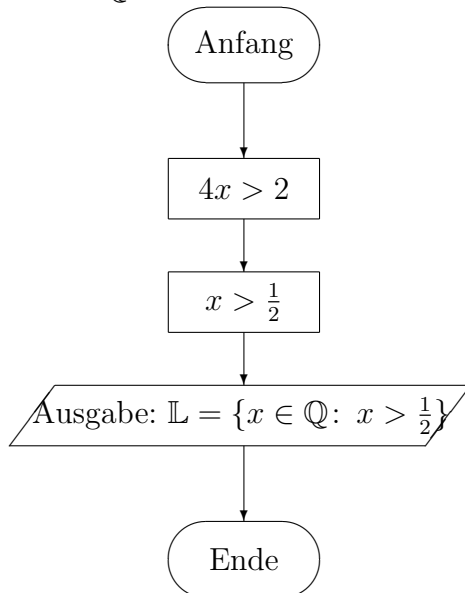
*Lösung:* Eine Möglichkeit zur Einführung der Flussdiagramme in der Jahrgangsstufe 8 besteht darin, zunächst *wenn-dann-Aussagen* mithilfe von Flussdiagrammen darzustellen, und danach mit *Zufallszahlen* zu arbeiten. Dieses Verfahren wird hier genauer dargestellt.

**Schritt 1.** Wenn-dann-Aussage

**Beispiele.** a) Wenn Peter Mathe mag, schreibt er nur Einsen.



b) Für  $x \in \mathbb{Q}$  ist  $4x > 2$  zu lösen.

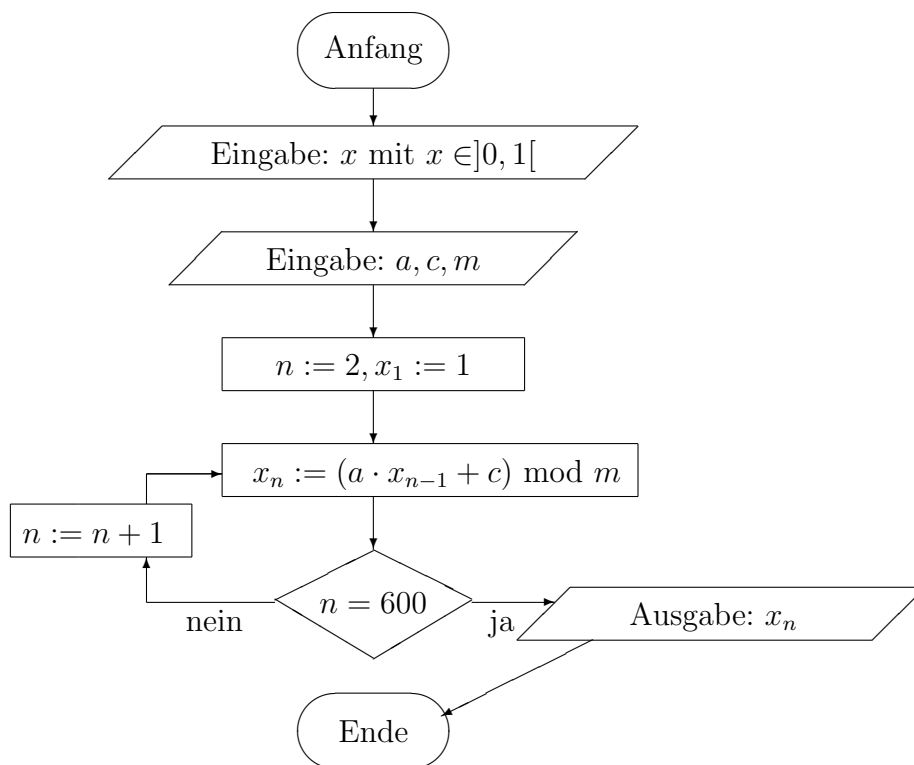


Weitere Beispiele können wie in den Beispielen a) und b) aus 1.4.1 gewählt werden.

## Schritt 2. Zufallszahlen

**Beispiel.** SchülerInnen sollen an Computern mit gleicher Zeiteinstellung Zufallszahlen bestimmen. Beobachtung: Alle haben dieselben Zufallszahlen. Dies legt die Vermutung nahe, dass ein festes Schema – d.h. ein Algorithmus – zur Bestimmung von Zufallszahlen vorliegt.

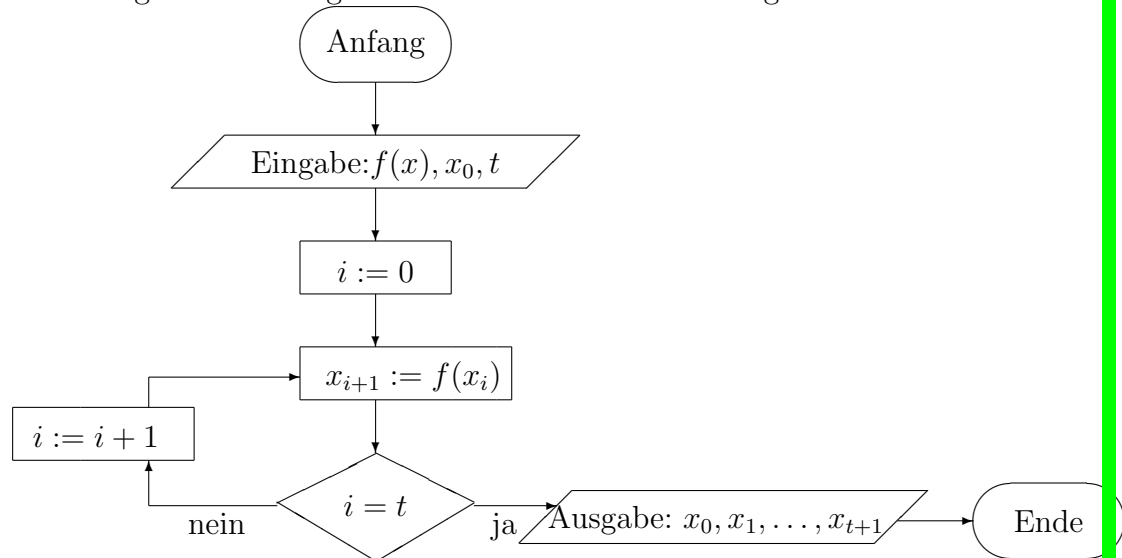
Ein einfaches Verfahren der folgenden allgemeinen *linearen Kongruenz* kann dann vorgestellt werden.



Die einzugebenden Zahlen  $a, b, c$  sind natürliche Zahlen ungleich Null. Damit kann das Flussdiagramm eingeführt werden. An diesem letzten Flussdiagramm sind alle Begriffe verwendet worden.

**Bemerkung.** Man kann die „Zufälligkeit“ von Zufallszahlen in einem Programm graphisch darstellen. Dies wird später durchgeführt.

**Bemerkung.** Für die Berechnung von Zufallszahlen lohnt es sich, die *Iteration* von Funktionen einzuführen. Wenn genug Zeit vorliegt, kann dies ausführlich gemacht werden. Hier können Anwendungen der *nichtlinearen Dynamik (Chaostheorie)* eingebaut werden, die SchülerInnen anregen, da sie zu einer graphischen Darstellung interessanter Objekte mithilfe des Computers führen. Auch hier ist fächerübergreifender Unterricht mit Biologie und Erdkunde möglich, da es Anwendungen gibt, die teilweise die Entwicklung der nichtlinearen Dynamik vorangetrieben haben. Das Flussdiagramm des Algorithmus der Iteration sieht folgendermaßen aus:



Hierbei gilt  $t \in \mathbb{N}$ ,  $x_0 \in \mathbb{R}$ , ferner ist  $f: \mathbb{R} \rightarrow \mathbb{R}$  eine lineare Funktion.

**Bemerkung.** Flussdiagramme werden auch im Pädagogik- und Sozialwissenschaftsunterricht verwendet, um Verfahren zu beschreiben. Hier ist fächerübergreifender Unterricht möglich.

#### 1.4.2 Multiplikation natürlicher Zahlen

Beispiele. a)

$$\begin{array}{r}
 13 \cdot 23 \\
 \hline
 26 \\
 39 \\
 \hline
 299
 \end{array}$$

Hier ist

$$13 \cdot 2 = 2 \cdot 10^1 + 6 \cdot 10^0 \quad \text{sowie}$$

$$13 \cdot 3 = 3 \cdot 10^1 + 9 \cdot 10^0.$$

Alle Faktoren vor den Zehnerpotenzen sind einstellig.

b)

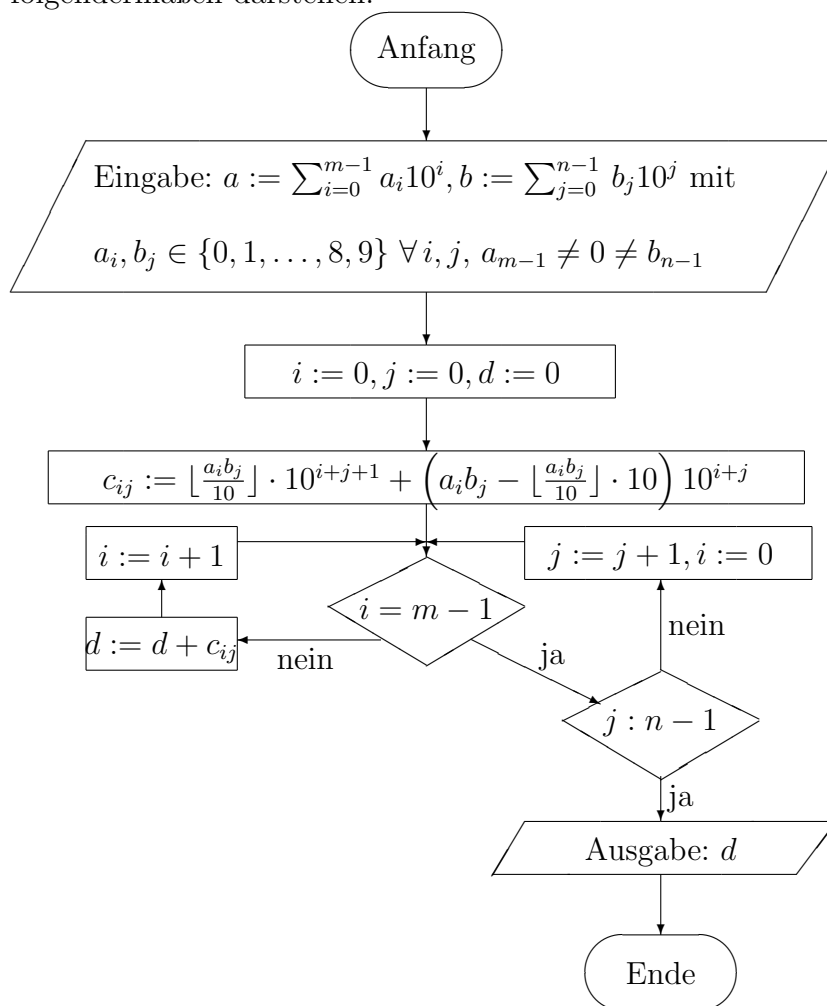
$$\begin{array}{r} 15 \cdot 33 \\ \hline 45 \\ 45 \\ \hline 495 \end{array}$$

Das Verfahren sieht etwas anders als in Beispiel a) aus, da  $5 \cdot 3 = 15$  zweistellig ist, und damit gilt

$$5 \cdot 3 = 1 \cdot 10^1 + 5 \cdot 10^0,$$

d.h. es wird eine 1 zum Faktor vor der um Exponent Eins größeren Zehnerpotenz addiert.

Um dies durchführen zu können, ist es in den Algorithmus einzubauen. Es lässt sich folgendermaßen darstellen:



Wendet man diesen Algorithmus auf die Beispiele a) und b) an, so ergibt sich:

**Beispiele. c)** Zunächst lauten die Darstellungen in Zehnerpotenzen  $a = 13 = 1 \cdot 10^1 + 3 \cdot 10^0$  und  $b = 23 = 2 \cdot 10^1 + 3 \cdot 10^0$ . Damit erhält man

$$c_{00} = \left\lfloor \frac{3 \cdot 3}{10} \right\rfloor \cdot 10^1 + \left( 3 \cdot 3 - \left\lfloor \frac{3 \cdot 3}{10} \right\rfloor \cdot 10 \right) \cdot 10^0 = 9,$$

$$c_{01} = \left\lfloor \frac{3 \cdot 2}{10} \right\rfloor \cdot 10^2 + \left( 3 \cdot 2 - \left\lfloor \frac{3 \cdot 2}{10} \right\rfloor \cdot 10 \right) \cdot 10^1 = 60,$$

$$c_{10} = \left\lfloor \frac{1 \cdot 3}{10} \right\rfloor \cdot 10^2 + \left( 1 \cdot 3 - \left\lfloor \frac{1 \cdot 3}{10} \right\rfloor \cdot 10 \right) \cdot 10^1 = 30,$$

und

$$c_{11} = \left\lfloor \frac{2 \cdot 1}{10} \right\rfloor \cdot 10^3 + \left( 2 \cdot 1 - \left\lfloor \frac{2 \cdot 1}{10} \right\rfloor \cdot 10 \right) \cdot 10^2 = 200.$$

Mit diesen Zwischenergebnissen ergibt sich

$$d = \sum_{i,j} c_{ij} = 200 + 30 + 60 + 9 = 299.$$

**d)** Für die Faktoren aus Beispiel b) gilt  $a = 15 = 1 \cdot 10^1 + 5 \cdot 10^0$  und  $b = 33 = 3 \cdot 10^1 + 3 \cdot 10^0$ , womit man

$$c_{00} = \underbrace{\left\lfloor \frac{5 \cdot 3}{10} \right\rfloor}_1 \cdot 10^1 + \left( 5 \cdot 3 - \left\lfloor \frac{5 \cdot 3}{10} \right\rfloor \cdot 10 \right) \cdot 10^0 = 15,$$

$$c_{01} = \left\lfloor \frac{5 \cdot 3}{10} \right\rfloor \cdot 10^2 + \left( 5 \cdot 3 - \left\lfloor \frac{5 \cdot 3}{10} \right\rfloor \cdot 10 \right) \cdot 10^1 = 150,$$

$$c_{10} = \left\lfloor \frac{1 \cdot 3}{10} \right\rfloor \cdot 10^2 + \left( 1 \cdot 3 - \left\lfloor \frac{1 \cdot 3}{10} \right\rfloor \cdot 10 \right) \cdot 10^1 = 30,$$

und

$$c_{11} = \left\lfloor \frac{3 \cdot 1}{10} \right\rfloor \cdot 10^3 + \left( 3 \cdot 1 - \left\lfloor \frac{3 \cdot 1}{10} \right\rfloor \cdot 10 \right) \cdot 10^2 = 300,$$

erhält, womit sich

$$d = \sum_{i,j} c_{ij} = 300 + 150 + 30 + 15 = 495$$

ergibt.

**Aufgabe 1.** Es sei  $m = n$ , dann sollen zwei Zahlen

$$a = \sum_{i=0}^{n-1} a_i 10^i \quad \text{und} \quad b = \sum_{j=0}^{n-1} b_j 10^j$$

multipliziert werden. Zeigen Sie:

Nach obigem Algorithmus werden  $n^2$  Multiplikationen einzelner Zahlen und mehr als  $n^2$  einzelne Additionen durchgeführt. (Damit werden mehr als  $2n^2$  Einzeloperationen durchgeführt.)

*Lösung.* Die Multiplikation ist in folgende Schritte zu unterteilen:



- i) Es wird jede Ziffer der ersten Zahl mit jeder Ziffer der zweiten multipliziert, also fallen Multiplikationen einstelliger Zahlen an.
- ii) Nachdem i) durchgeführt wurde, wird folgendes Schema durchlaufen. Hierbei hilft die folgende Skizze:

$$\begin{array}{cccccccc}
 \bigcirc & \bigcirc & \bigcirc & \bigcirc & \cdot & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\
 \hline
 & & & & & (\circ) & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\
 & & & & & (\circ) & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\
 & & & & & (\circ) & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\
 & & & & & (\circ) & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\
 \hline
 \end{array}$$

- Die Addition der Ziffern wird spaltenweise von rechts nach links durchgeführt.
- Dabei ist in der ersten Spalte keine Addition durchzuführen.
- In der zweiten Spalte wird eine Addition durchgeführt.
- In der dritten Spalte werden zwei Additionen durchgeführt, evtl. eine weitere vom Übertrag innerhalb der zweiten Spalte.
- In der vierten Spalte werden drei Additionen durchgeführt, evtl. eine weitere vom Übertrag innerhalb der dritten Spalte.
- Der Schritt wird bis zur  $n$ -ten Spalte wiederholt.
- An der  $(n + 1)$ -ten Stelle nimmt zum Einen die Anzahl der mit Ziffern gefüllten Zeilen ab. Andererseits können ab hier zwei zusätzliche Additionen auftreten: Eine für den möglichen Übertrag innerhalb einer Zeile; ein weiterer kann jedoch für einen Übertrag der Addition der Ziffern der  $n$ -ten Spalte hinzukommen.
- Damit ergibt sich insgesamt für die maximale Anzahl der Rechenoperationen

$$\begin{aligned}
 & 0 + 1 + 2(+1) + 3(+1) + \dots + n(+1) + n(+1) + \dots + 3(+1) + 2(+1) + 1(+1) + 0 \\
 & \leq 2 \cdot \sum_{i=1}^n i + 2(n-1) + 1 = 2 \frac{n(n+1)}{2} + 2n - 1 \\
 & = n^2 + 3n - 1
 \end{aligned}$$

Es treten mindestens  $n^2$  dieser Schritte auf.

Damit kommen unter Umständen mehr als  $2n^2$  Einzeloperationen zustande.

**Aufgabe 2.** Multiplizieren Sie  $a = 999$  und  $b = 999$ . Fällt hierbei etwas auf? Wie kann dies gelöst werden?

*Lösung.* Wie auch in Aufgabe 1 im Algorithmus unter ii) berücksichtigt wurde, können bei der Addition der Zwischenergebnisse Überträge entstehen. Diese wurden in der Darstellung im Flussdiagramm nicht berücksichtigt, es könnten jedoch noch zwischen den Schritten  $d := d + c_{ij}$  und  $i := i + 1$  eingebaut werden, womit der Fehler behoben wäre.

### 1.4.3 Fakultät

Das Flussdiagramm sowie seine Beschreibung befinden sich in der Lösung zu Aufgabe 1 c) von Abschnitt 1.4.1.

Im Folgenden wird daher noch eine Möglichkeit der Einführung im Unterricht der Jahrgangsstufe 10 beschrieben. Diese Möglichkeit besteht aus mehreren Schritten:

- i) Zunächst wird das Urnenmodell zum *Ziehen mit Zurücklegen mit Beachtung der Reihenfolge* behandelt. Dass es  $n^k$  Möglichkeiten für  $n$  Kugeln und  $k$  Wiederholungen gibt leuchtet den SchülerInnen relativ schnell ein.
- ii) Es wird das Urnenmodell zum *Ziehen ohne Zurücklegen ohne Beachtung der Reihenfolge* behandelt. Die Anzahl der Möglichkeiten beträgt hierbei

$$\frac{n!}{(n-k)!} = n \cdot (n-1) \cdot \dots \cdot (n-k+1), \quad \textcircled{*}$$

wobei in dieser Schreibweise die Fakultät enthalten ist. Selbstverständlich kommt man an dieser Stelle auch ohne Fakultät aus; es ist jedoch von der Schreibweise und der Erkennung der Hintergründe sinnvoll, die Kurzschreibweise einzuführen. Zuerst wird der Fall der Vertauschung von  $n$  Kugeln auf  $k$  Plätzen behandelt. Hierbei ergeben sich  $n$  Möglichkeiten zur Belegung des ersten Platzes, dann noch  $n-1$  Möglichkeiten zur Belegung des zweiten Platzes,  $n-2$  Möglichkeiten zur Belegung des dritten Platzes usw. Für den  $k$ -ten Platz gibt es dann noch weitere  $n-k+1$  Möglichkeiten, womit sich für den Fall des Ziehens ohne Zurücklegen ohne Beachtung der Reihenfolgen genau  $n \cdot (n-1) \cdot \dots \cdot (n-k+1)$  Möglichkeiten ergeben.

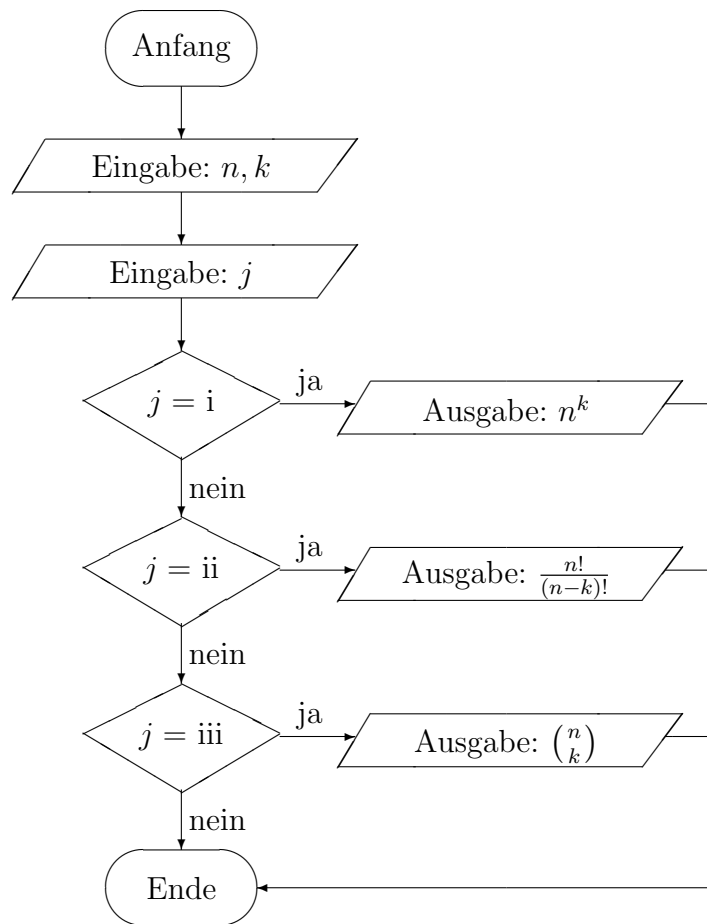
Den SchülerInnen gelingt es schnell, aus diesen Überlegungen die Gleichung  $\textcircled{*}$  herzuleiten.

- iii) Wenn die SchülerInnen bereits die Teile i) und ii) erarbeitet haben, fällt es ihnen nicht mehr schwer, den Fall *Ziehen ohne Zurücklegen ohne Beachtung der Reihenfolge* zu behandeln. Im Unterschied zu Teil ii) sind die Fälle zu identifizieren, die durch Permutation der  $k$  Kugeln auf  $k$  Plätzen auseinander entstehen. Hierfür existieren  $k!$  Möglichkeiten, woraus sich für Ziehen ohne Zurücklegen ohne Beachtung der Reihenfolge genau

$$\binom{n}{k} := \frac{n!}{(n-k)!k!} = \frac{n(n-1) \cdot \dots \cdot (n-k+1)}{k(k-1) \cdot \dots \cdot 2 \cdot 1}$$

ergibt. Auf diese Weise wird „ $n$  über  $k$ “ eingeführt.

**Aufgabe 1.** Schreiben Sie ein Flussdiagramm für die Darstellung eines der drei genannten Fälle i), ii) und iii) inklusive der Auswahl der Darstellung.



## 1.5 Effizienz von Algorithmen

Ein wichtiges Maß für die „Güte“ eines Algorithmus ist durch die für die Durchführung des Algorithmus benötigte Zeit gegeben. Zur Zeitmessung wird die benötigte Zeit als Funktion der Zahl der verarbeiteten Werte betrachtet. Eine Möglichkeit dies zu tun, besteht in der Zählung oder Abschätzung der Operationen in Abhängigkeit des Umfangs  $n$  der Eingabe.

Die Effizienz kann für unterschiedliche Vorgehensweisen derselben Berechnung verschieden ausfallen, wie das folgende Beispiel zeigt.

**Beispiel. a)** Es ist  $x^k$  für  $k = 2^n$  mit  $n \in \mathbb{N} \setminus \{0\}$  zu berechnen. Hier können unterschiedliche Algorithmen entwickelt werden, von denen zwei vorgestellt werden sollen.

1.  $x$  wird  $k$ -mal mit sich selbst multipliziert. Der Schritt sieht folgendermaßen aus:

$$x^k = \underbrace{x \cdot \dots \cdot x}_{2^n\text{-mal}}$$

Die Anzahl der zur Berechnung von  $x^{2^n}$  durchführbaren Schritte beträgt  $2^n$ .

2. Alternativ lässt sich  $x$  wählen und quadrieren. Das Ergebnis wird wieder quadriert u.s.w., d.h.

$$x^{2^n} = \underbrace{\left( \dots \left( (x^2)^2 \right) \dots \right)^2}_{n\text{-mal}}$$

Dieser Algorithmus wird genau  $n$ -mal durchgeführt.

Für große  $n$  liegen die Werte weit auseinander, wie die folgende Tabelle zeigt:

$n$	1	2	3	4	5	6
$2^n$	2	4	8	16	32	64

Nach diesen  $n$  bzw.  $2^n$  Rechenoperationen wird jeweils dasselbe Ergebnis erreicht. Der zweite Algorithmus ist damit effektiver.

**Aufgabe 1.** Sechs Algorithmen A, B, C, D, E und F umfassen  $n$ ,  $3n^2$ ,  $2n^2+4n$ ,  $n^3$ ,  $2^n$  und  $\ln n$  elementare Operationen. Schätzen Sie unter der Annahme, dass die Ausführung einer elementaren Operation 1 Millisekunde beansprucht, die Laufzeit für  $n = 1, 10, 100$  und  $1000$  ab.

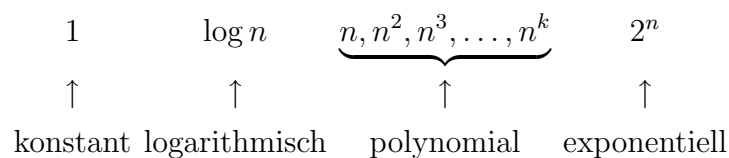
*Lösung.*

$n$	$3n^2$	$2n^2 + 4n$	$n^3$	$2^n$	$\ln n$
1ms	3ms	6ms	1ms	2ms	–
10ms	300ms	240ms	1s	1.024s	2.30ms
100ms	30s	20.4s	0.28s	$4 \cdot 10^{17}$ Jahrhunderte	4.61ms
1000ms	0.83h	0.56h	11.6 Tage	$10^{176}$ Jahrhunderte	6.91ms

Wie die Tabelle andeutet, lässt sich die Effizienz in Klassen einteilen. Eine Klasse ist gegeben durch *Polynom*-Funktionen, weitere durch *Exponential*-Funktionen, und durch *Logarithmus*-Funktionen. Es gibt jedoch zusätzlich die Möglichkeit der *konstanten* Funktion.

Die Polynom-Funktionen steigen mit steigenden Exponenten schneller an. Wie es sich insgesamt für die unterschiedlichen oben genannten Funktionen verhält, zeigt der folgende Graph. Hierbei ist zu beachten, dass die Werte erst ab einem gewissen Umfang von Interesse sind, da Algorithmen oft zahlreiche Male wiederholt werden, wie z.B. beim Zufallszahlen-Algorithmus aus der Lösung von Aufgabe 1 in Abschnitt 1.4.1.

Bei der *Effizienz* ergibt sich eine *Hierarchie*



Die bisherigen Überlegungen führen zu folgender

**Definition.** Es seien  $f(n)$  und  $g(n)$  Maße für die Effizienz von Algorithmen. Sie tragen den Namen **Zeitkomplexitätsfunktionen**.

$f(n)$  ist **höchstens von der Ordnung**  $g(n)$ , wenn ein  $C \in \mathbb{R}_{>0}$  existiert mit  $|f(n)| \leq C \cdot |g(n)|$ .

Die Menge aller Funktionen, die höchstens von der Ordnung  $g(n)$  sind, wird durch

$$O(g(n))$$

bezeichnet.

**Beispiel. b)** Es gilt  $2n^2 + 4n \in O(n^2)$ .

Da  $n \leq n^2$  für  $n \geq 1$ , gilt  $2n^2 + 4n \leq 2n^2 + 4n^2 = 6n^2$  für alle  $n \in \mathbb{N} \setminus \{0\}$ . Setzt man also  $C = 6$ , so gilt

$$|2n^2 + 4n| \leq 6 \cdot n^2 \quad \forall n \geq 1$$

$$\Rightarrow 2n^2 + 4n \in O(n^2).$$

Da auch  $2n^2 + 4n \geq n^2 \quad \forall n \geq 1$  gilt, folgt auch  $n^2 \in O(2n^2 + 4n)$ .  $n^2$  und  $2n^2 + 4n$  sind damit von *derselben Größenordnung*.

**Bemerkung.** In vielen praktischen Anwendungen zeigt es sich, dass es nicht schlimm ist, bestimmte Details wie Addition, Subtraktion, Multiplikation und Division von Termen oder Faktoren zu vernachlässigen. Dies gilt ebenfalls für Modulo-Operatoren oder Wertzuweisungen bei Funktionen.

Es liegt damit nahe, einen möglichst einfachen Term zur Festlegung der Ordnung zu verwenden, wozu obige Hierarchie dient. Dies wird unterstützt durch das Ergebnis des letzten Beispiels.

**Beispiel. c)** Es ist  $6n^4 + 3n - \ln(n) \in O(n^4)$ .

Die Koeffizienten 6 bzw. 3 spielen keine Rolle, daher gilt  $6n^4 \in O(n^4)$ ,  $3n \in O(n)$ ,  $-\ln(n) \in O(\ln(n))$ . Mithilfe der Hierarchie ergibt sich für ausreichend großes  $n \in \mathbb{N}$

$$\begin{aligned} O(\ln(n)) &\subset O(n) \subset O(n^4) \\ \Rightarrow 6n^4 + 3n - \ln(n) &\in O(n^4). \end{aligned}$$

**Aufgabe 2.** Ermitteln Sie unter Verwendung obiger Hierarchie eine möglichst kleine Größenordnung der folgenden Zeitkomplexitätsfunktionen.

a)  $n^4 + 2n^3 + 3$

b)  $6n^5 + 2^n$

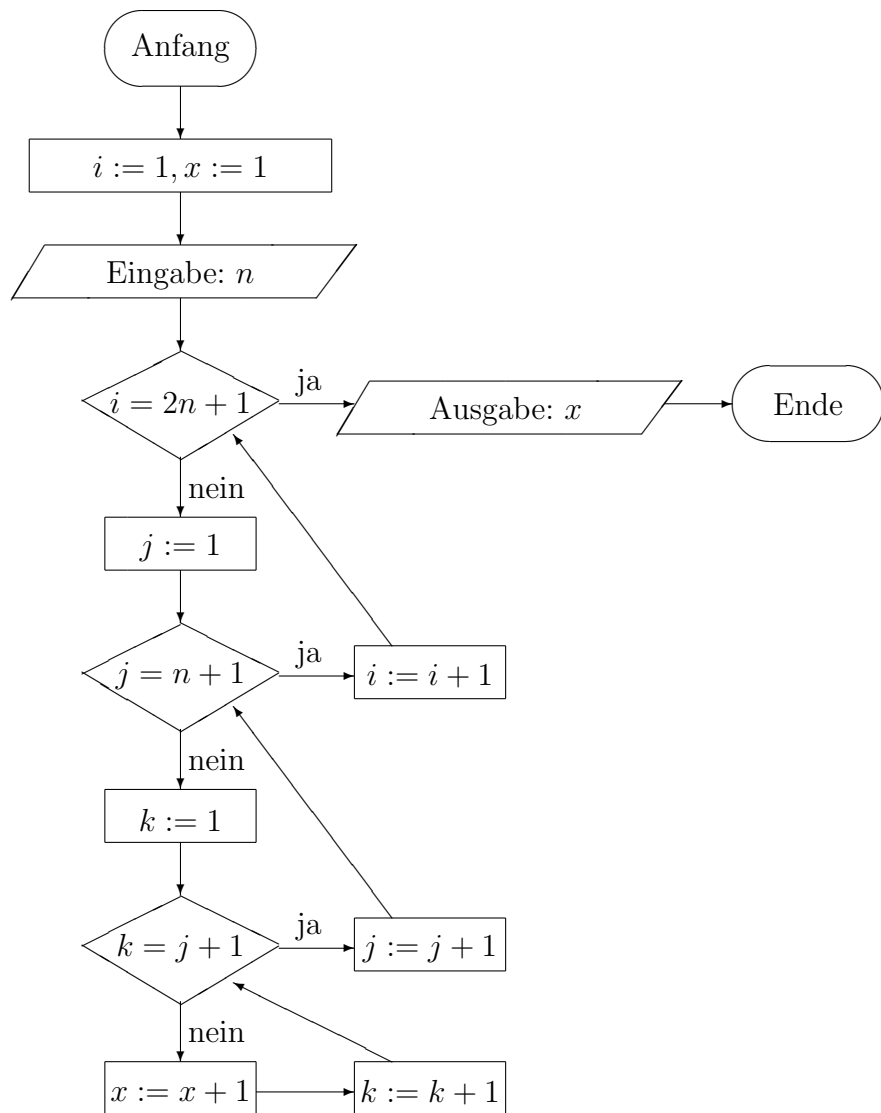
c)  $5n + n^2 \ln(n)$

*Lösung.* a)  $O(n^4)$     b)  $O(2^n)$

c) Der am schnellsten wachsende Term ist  $n^2 \ln(n)$ . Da er zwischen  $n^2$  und  $n^3$  liegt, liegt die Funktion z.B. in  $O(n^3)$ .

**Aufgabe 3.** Bestimmen Sie eine Zeitkomplexitätsfunktion für den Algorithmus zu folgendem Flussdiagramm.

*Hinweis.* Berechnen Sie, wie oft der Zuweisung  $x := x + 1$  ausgeführt wird.



*Lösung.* Die äußere Schleife (mit  $i$ ) wird genau  $2n$ -mal durchlaufen, die zweite (mit  $j$ ) genau  $n$ -mal, und die dritte (mit  $k$ ) wird für jedes  $j$  genau  $j$ -mal durchlaufen. Dies bedeutet, dass die Anweisung  $x := x + 1$  in der dritten Schleife genau  $j$ -mal stattfindet, und dies für jedes  $j$ . Also wird für jeden Wert von  $i$  die Zuweisung  $x := x + 1$  genau  $\sum_{\ell=1}^n \ell$ -mal, also  $\frac{n(n+1)}{2}$ -mal ausgeführt. Damit ergibt sich die Zeitkomplexitätsfunktion  $f(n)$  zu

$$f(n) = 2n \cdot \frac{n(n+1)}{2} = n^2(n+1) \Rightarrow f(n) \in O(n^3).$$

## 2 Algorithmen aus Sek I und Sek II

### 2.1 Lösung von Nullstellenproblemen

#### 2.1.1 Einleitung

**Aufgabe 1. a)** Entwerfen Sie (mithilfe von Schulbüchern) eine Unterrichtsreihe zur Einführung von Verfahren zur Lösung von Nullstellenproblemen von Polynomen in der Jahrgangsstufe 11.

Worauf ist hier zu achten?

**b)** Warum ist die Auswahl der Funktionen bei der Einführung von Nullstellenbestimmung auf Polynome niedrigen Grades beschränkt? Wie kann dies gelöst werden?

**Aufgabe 2.** Wie lässt sich das Verfahren in der Schule auf Polynome beliebigen Grades umsetzen?

Zur Bestimmung von Nullstellen existieren mehrere Verfahren wie z.B. das *Bisektionsverfahren*, das *Fixpunktverfahren* bzw. das *Newton-Verfahren*. Um zu testen, ob eine stetige Funktion überhaupt Nullstellen besitzt, in welchem Bereich bestimmte davon liegen und evtl. wieviele Nullstellen es gibt, dient der

**Zwischenwertsatz.** Es sei  $f: [a, b] \rightarrow \mathbb{R}$  stetig. Sei  $c \in \mathbb{R}$  so, dass entweder  $f(a) \leq c \leq f(b)$  oder  $f(b) \leq c \leq f(a)$  gilt. Dann existiert (mindestens) ein  $x \in [a, b]$  mit  $f(x) = c$ .

Für  $c = 0$  sagt damit der Zwischenwertsatz die Existenz einer Nullstelle im Intervall  $[a, b]$  voraus. Es können jedoch auch mehrere Nullstellen in  $[a, b]$  vorliegen. Um die Anzahl der Nullstellen und die Lage zu bestimmen, dienen andere Verfahren, bei denen z.B. die Intervalle verkleinert werden können, wie es das erste Verfahren zeigt.

#### 2.1.2 Bisektionsverfahren

Hier wird das im letzten Abschnitt angedeutete Verfahren der Intervallschachtelung auf eine bestimmte Art umgesetzt.

Man halbiert das Startintervall  $[a, b]$  in zwei Teilintervalle  $[a, \frac{a+b}{2}]$  und  $[\frac{a+b}{2}, b]$ . Liegt die Nullstelle  $a_0$  in  $[a, \frac{a+b}{2}]$ , so arbeitet man mit diesem Intervall weiter. Falls  $a_0 \notin [a, \frac{a+b}{2}]$ , so arbeitet man mit dem Intervall  $[\frac{a+b}{2}, b]$  weiter.<sup>1</sup> Wiederholt man dieses Verfahren, so kann dadurch die Nullstelle beliebig genau angenähert werden. Dieses Verfahren heißt *Bisektionsverfahren*.

**Beispiel. a)** Es sollen Intervalle bestimmt werden, in denen die Nullstellen von  $f(x) = x^3 - x + 0.3$  liegen.

*Lösung.*  $f$  besitzt maximal 3 Nullstellen, da es sich um ein Polynom vom Grad 3 handelt. Vernachlässigt man zunächst den Summanden 0.3, so liegt das Polynom

<sup>1</sup>Man kann an dieser Stelle jeweils auch testen, ob  $a_0 \in [a, \frac{a+b}{2}] \cap [\frac{a+b}{2}, b]$  gilt, jedoch macht dies den Algorithmus nicht schneller.



$g(x) = x^3 - x$  vor, das Nullstellen an den Stellen  $-1, 0$  und  $1$  hat. Es wird nun angenommen, dass die Nullstellen von  $f$  in der Nähe der Nullstellen von  $g$  liegen. Daher betrachtet man die Funktionswerte von  $f$  an den Nullstellen von  $g$  sowie Nachbarwerten, womit sich

$x$	$-2$	$-1$	$0$	$0.5$	$1$
$f(x)$	$-5.700$	$0.300$	$0.300$	$-0.075$	$0.300$

ergibt.  $f$  ist stetig, daher lässt sich der Zwischenwertsatz anwenden, und damit gibt es in  $[-2, -1]$ ,  $[0, 0.5]$  und  $[0.5, 1]$  jeweils mindestens eine Nullstelle. Da  $f$  den Grad 3 hat, liegt in jedem dieser Intervalle genau eine Nullstelle.

Notiert man das Verfahren aus Beispiel a) in sauberer Form, so ergibt folgender

**Satz zum Bisektionsverfahren.** Gegeben sei eine stetige Funktion  $f: [a, b] \rightarrow \mathbb{R}$  mit  $f(a) \cdot f(b) < 0$ . In jedem der für  $i = 0, 1, \dots$  erzeugten Intervalle

$$[a_0, b_0] := [a, b]$$

$$[a_{i+1}, b_{i+1}] := \begin{cases} [a_i, \frac{a_i+b_i}{2}], & \text{falls } f(\frac{a_i+b_i}{2}) \cdot f(a_i) \leq 0 \\ [\frac{a_i+b_i}{2}, b_i], & \text{sonst,} \end{cases}$$

befindet sich eine Nullstelle von  $f$ , und es gilt

$$b_i - a_i = \frac{b - a}{2^i}, \quad \text{insbesondere } \lim_{i \rightarrow \infty} (b_i - a_i) = 0.$$

**Aufgabe 1.** Bestimmen Sie die Nullstelle der Funktion  $f$  aus Beispiel a) im Intervall  $[0, 0.5]$  bis auf eine Stelle hinter dem Komma genau.

*Lösung.* Da  $f(0.25) > 0$  gilt, liegt die gesuchte Nullstelle in  $[a_0, b_0] = [0.25, 0.5]$ . Aus  $f(0.375) < 0$  folgt, dass sich die Nullstelle im Intervall  $[a_1, b_1] = [0.25, 0.375]$  befindet. Nach  $f(0.3125) > 0$  ergibt sich, dass die Nullstelle im Intervall  $[a_2, b_2] = [0.3125, 0.375]$  liegt. damit ist diese Aufgabe erledigt, da die Nullstelle bis auf eine Nachkommastelle genau gleich  $0.3$  lautet.

Wie in dem Beispiel schon zu sehen ist, ist das Verfahren aufwendig, so dass die Umsetzung in der Praxis sehr lang dauert. Zudem gibt es deutlich schnellere Algorithmen, weswegen das Bisektionsverfahren oft in der Praxis umgesetzt wird.

Das Bisektionsverfahren kann jedoch gut in der Jahrgangsstufe 9 umgesetzt werden, wenn die reellen Zahlen mithilfe der Intervallschachtelung eingeführt werden sollen oder bereits eingeführt wurden und Nullstellen bestimmt werden sollen.

**Aufgabe 2.** Gestalten Sie eine Unterrichtsreihe, mit der das Bisektionsverfahren in der Stufe 9 bzw. der Stufe 11 eingeführt und zur Bestimmung von Funktionsnullstellen umgesetzt werden kann.

**Aufgabe 3. a)** Schätzen Sie die Effizienz des Algorithmus ab.

**b)** überlegen sie sich ein Flussdiagramm des Algorithmus des Bisektionsverfahrens.

*Lösung.* a) Beim Bisektionsverfahren werden die Intervalle halbiert; damit nimmt der Abstand der Nullstellen zu den Intervallgrenzen linear ab bzw. die Genauigkeit linear zu. Daher ist die Effizienz  $O(n)$ .

### 2.1.3 Fixpunktverfahren

#### Einschub: Iteration

Für das Fixpunktverfahren benötigt man die Iteration von Funktionen. Auch in der Physik sind Iterationen von Bedeutung, wie das folgende Beispiel verdeutlicht: Wirkt auf einen Körper eine Kraft von Dauer, so kann die Wirkung der Kraft durch eine Funktion  $f$  dargestellt werden (wobei die Kraft hier anders als in der Physik üblich –  $F$  – bezeichnet wird). Ist diese Funktion integrierbar, so lässt sich die durch die Kraft  $f$  verrichtete Arbeit durch das Integral

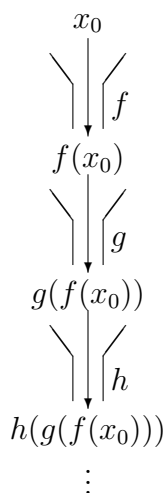
$$W = \int_{x_0}^{x_1} f(x) dx$$

berechnen. Es existieren jedoch auch Funktionen, die nicht oder nur schwer integrierbar sind. Dann kann man die Arbeit näherungsweise mithilfe einer Iteration der Funktion  $f$  berechnen.

Die Iteration kann in der Sekundarstufe 9 im Zusammenhang mit dem Heron-Verfahren eingeführt werden. Ferner gibt es eine Einführung in [Ti] – einem Buch für den Leistungskurs Mathematik. Dort werden einige der hier behandelten Lösungsverfahren besprochen.<sup>2</sup> Eine gute graphische Darstellung der Iteration lässt sich analog zu [P] einführen. Hier wird zunächst die allgemeine Verknüpfung von Funktionen betrachtet:

$$x_0 \xrightarrow{f} f(x_0) \xrightarrow{g} g(f(x_0)) \xrightarrow{h} h(g(f(x_0))) \cdots$$

Dies lässt sich anschaulich folgendermaßen darstellen:



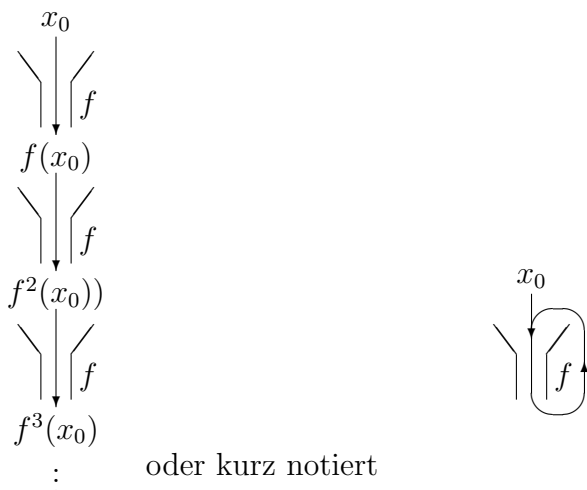
Für eine Iteration

$$x_0 \xrightarrow{f} f(x_0) \xrightarrow{f} f^2(x_0) \xrightarrow{f} f^3(x_0) \cdots ,$$

d.h. die Verknüpfung *ein und derselben* Funktion lässt sich das Diagramm einfacher darstellen:

---

<sup>2</sup>Es ist allerdings zu bemerken, dass dieses Buch nicht den heutigen Richtlinien entspricht.

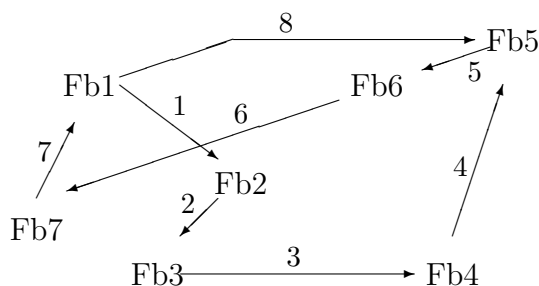


oder kurz notiert

Die Iteration hat eine besondere Bedeutung. Um diese Bedeutung übersichtlicher zu machen, führt man die *Orbits* der Funktionen an der Stelle  $x_0 \in \mathbb{D}_f$  ein.

Für die Hintereinanderausführung von Vorgängen lässt sich das folgende Beispiel betrachten.

**Beispiel. a)** Wir betrachten eine Folge von Pässen mit einem Fußball. Die FußballerInnen werden mit Fb bezeichnet. Dann werden Pässe ausgeführt. In der folgenden Abbildung ist eine Reihe von Pässen eingezeichnet und der Reihenfolge nach durchnummeriert. Der Ball kann hierbei selbstverständlich auch nach einiger Zeit zu Fb1 zurückkehren. Wenn dies geschieht, geht der Pass i.allg. zu einer anderen Person als der erste Pass; daher kann es sich nicht um den Orbit einer Funktion handeln.



**Bild 2.1**

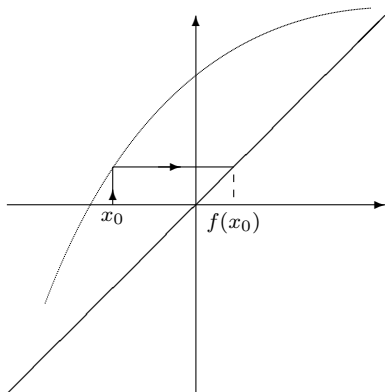
Nach dem Exkurs folgt nun die Definition des Orbits einer Funktion  $f$  zu einem Startpunkt  $x_0$ .

**Definition.** Es sei  $f: \mathbb{R} \rightarrow \mathbb{R}$  eine Funktion. Der *Orbit* von  $f$  zu einer reellen Zahl  $x_0 \in \mathbb{R}$  ist gegeben durch

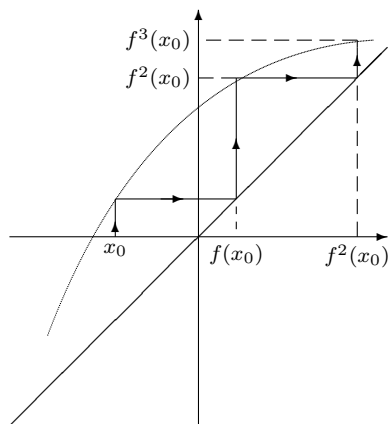
$$\mathcal{O}(x_0) := (x_0, f(x_0), f^2(x_0), \dots).$$

Bei reellen Funktionen ist ein Orbit recht gut an ihren Graphen zu untersuchen. Das geht so: Zunächst zeichnen wir zusätzlich zum Graphen der Funktion  $f$  die erste Winkelhalbierende, d.h. den Graphen der identischen Funktion  $g$  mit  $g(x) = x$  in das Koordinatensystem. Nun ziehen wir von dem Punkt  $(x_0, f(x_0))$  eine Linie

parallel zur  $x$ -Achse bis zur Diagonalen, siehe Bild 2.2. Wegen  $g(x_0) = x_0$  besitzt der Schnittpunkt der waagerechten Strecke mit der Diagonalen die Koordinaten  $(f(x_0), f(x_0))$ . Ziehen wir nun von diesem Punkt aus parallel zur  $y$ -Achse die Strecke bis zu einem Schnittpunkt mit dem Graphen von  $f$ , so besitzt dieser Schnittpunkt die Koordinaten  $(f(x_0), f^2(x_0))$ .



**Bild 2.2**



**Bild 2.3**

Jetzt ist das Verfahren klar. Ziehen wir vom Punkt  $(f(x_0), f^2(x_0))$  wiederum eine Linie parallel zur  $x$ -Achse bis zum Schnittpunkt mit der Diagonalen, so besitzt dieser Schnittpunkt die Koordinaten  $(f^2(x_0), f^2(x_0))$ . Nun ziehen wir wiederum die Verbindung zum Graphen von  $f$  parallel zur  $y$ -Achse, vgl. Bild 2.3, und erhalten so den Punkt  $(f^2(x_0), f^3(x_0))$ .

Dieses Verfahren kann beliebig lang fortgesetzt werden, und wir erhalten so eine Vorstellung davon, wie sich eine reelle Funktion bei der Iteration mit Startpunkt  $x_0$  verhält.

Wir behandeln nun einige

**Beispiele.** a)  $f: \mathbb{R} \rightarrow \mathbb{R}, x \mapsto 2x$ , an der Stelle  $x_0 = 1$ . Wir erhalten den Orbit

$$\mathcal{O}_f(1) = (1, 2, 4, 8, 16, \dots).$$

Der Graph von  $f$  und die anschauliche Darstellung dieses Orbits sind in Bild 2.4 zu finden.

**Bild 2.4**

Mit Maple kann der Orbit rechnerisch und zeichnerisch bestimmt werden. Definiert man die Funktion

```
> f:=s->2*s:
```

und den Startwert

```
> s:=1:
```

so können die ersten fünf Glieder des Orbits durch

```
> for t from 1 to 5 do
> f(%): s:=f(s):
> od;
```

bestimmt werden.

Bei der graphischen Darstellung von Pfeilen werden Vektoren verwendet. Ein Vektor im  $\mathbb{R}^2$  kann unter Maple durch  $\langle a|b \rangle$  definiert werden. Der zugehörige Pfeil, der am Endpunkt von  $\vec{b}$  angelegt wird und in Richtung von  $\vec{a}$  zeigt, kann durch

```
> arrow(b,a-b,shape=arrow);
```

gezeichnet werden. Für Genaueres siehe [Wa], 7.1. Der Graph in Bild 2.4 wurde durch

```
> with(plots):
> f:=s->2*s: s:=1: a[0] := <s|0>:
> for t from 1 to 5 do
> f(%): a[t]:=<s|f(s)>:
> b[t]:=<f(s)|f(s)>: s:=f(s):
> od:
> f1:=plot(f(x), x=0..20,
> y=0..20,axes=normal,scaling=constrained):
> f2:=plot(x, x=0..20,y=0..20,axes=normal,
> scaling=constrained, color=black):
> p0:=arrow(a[0],a[1]-a[0],shape=arrow):
> for i from 1 to 3 do
> p[i]:=arrow(b[i],a[i+1]-b[i],shape=arrow):
> q[i]:=arrow(a[i],b[i]-a[i],shape=arrow):
> od:
> display(f1,f2,p0,seq(p[i],i=1..3),seq(q[i],i=1..3));
```

erstellt.

b)  $f: \mathbb{R} \rightarrow \mathbb{R}, x \mapsto x^2 - 1$ . Hier gilt für den Orbit an der Stelle  $x_0 = 0$

$$\mathcal{O}_f(0) = (0, -1, 0, -1, 0, -1, \dots),$$

an der Stelle  $x_0 = \frac{1}{2}(1 - \sqrt{5})$

$$\mathcal{O}_f\left(\frac{1}{2}(1 - \sqrt{5})\right) = \left(\frac{1}{2}(1 - \sqrt{5}), \frac{1}{2}(1 - \sqrt{5}), \frac{1}{2}(1 - \sqrt{5}), \dots\right),$$

dieser Orbit besteht genau aus einem Punkt. Für  $x_0 = 2$  ergibt sich

$$\mathcal{O}_f(2) = (2, 3, 8, 63, 3968, 15745023, \dots).$$

Der Graph und die anschauliche Darstellung der ersten beiden Orbits befinden sich in Bild 2.5. Hierbei ist erkennbar, dass  $\mathcal{O}_f\left(\frac{1}{2}(1 - \sqrt{5})\right)$  aus einem Schnittpunkt

mit der Winkelhalbierenden des ersten Quadranten besteht. Es sei den LeserInnen überlassen sich zu überlegen, dass diese beiden Bedingungen gleichbedeutend sind. Dies werden wir im Laufe des Buches noch genauer untersuchen.

### Bild 2.5

c) Für  $f: \mathbb{R} \rightarrow \mathbb{R}$ ,  $x \mapsto 2 \cdot \sin(x)$ , soll  $\mathcal{O}_f(x_0)$  mit  $x_0 = \frac{4}{3}\pi$  bestimmt werden.

Unter Maple ist dies folgendermaßen möglich:

```
> f:=s->2*sin(s): a[0] := <4/3*Pi|0>:
> s:=4/3*Pi:
> for t from 1 to 7 do
> f(%): a[t]:=<s|f(s)>:
> b[t]:=<f(s)|f(s)>: s:=f(s):
> od:
> a[0] := <4/3*Pi|0>: s:=evalf(4/3*Pi,2):
```

Die Option 2 unter `evalf` dient zur Rundung der Werte von  $s$  auf zwei Nachkommastellen.

```
> for t from 1 to 7 do
> evalf(f(%),3):
> od;
```

-1.51

-2.00

-1.82

-1.94

-1.87

-1.91

-1.89

Damit ergibt sich

$$\mathcal{O}_f\left(\frac{4}{3}\pi\right) = \left(\frac{4}{3}\pi, -1.51, -2.00, -1.82, -1.94, -1.87, -1.91, -1.89, \dots\right),$$

### Bild 2.6

In obigen Beispielen verhalten sich die Funktionen unterschiedlich. In den Aufgaben 1 bis 3 werden die Beispiele a) bis c) genauer untersucht. Ein Ergebnis bei den Lösungen der Aufgaben ist, dass sich ein und dieselbe Funktion für verschiedene Anfangspunkte  $x_0$  recht unterschiedlich verhalten kann.

Wir wenden uns den Orbits komplexer Funktionen zu, die sich allerdings nicht mehr so anschaulich darstellen lassen wie die Orbits reeller Funktionen. Die reellen Orbits werden uns in Abschnitt 3.2.3 noch einmal begegnen.

Es sei  $f: \mathbb{C} \rightarrow \mathbb{C}$  eine Funktion. Der *Orbit* von  $f$  zum Punkt  $z_0 \in \mathbb{C}$  wird dann mit denselben Bezeichnungen wie in 2.1.2 definiert durch

$$\mathcal{O}_f(z_0) := (z_0, f(z_0), f^2(z_0), f^3(z_0), \dots).$$

Die Definition für komplexe Funktionen sieht genauso aus wie für reelle Funktionen, es ist jedoch zu bedenken, dass auch dann, wenn dieselbe Funktionsvorschrift betrachtet wird, ein Unterschied zwischen Definitions- und Wertebereich besteht, vgl. Beispiel 2.1.2 b) und Beispiel a) weiter unten. Wir werden nun auch eine Zeichnung zu den Orbits einführen, allerdings im voraus schon einmal auf den entscheidenden Unterschied zu den Zeichnungen zu reellen Orbits hinweisen. Die Punkte in den reellen Zeichnungen haben die Koordinaten  $(f^{n-1}(x), f^n(x))$ , in den komplexen Zeichnungen jedoch nur die Koordinaten  $(\operatorname{Re}(f^n(z)), \operatorname{Im}(f^n(z)))$ . Trotzdem kann Einiges an diesen Zeichnungen abgelesen werden, wie wir in Abschnitt 3.2 feststellen werden. Wenn alle „interessanten“ Punkte einer komplexen Funktion rein reelle Koordinaten haben, werden wir auch in Zukunft auf die reellen Zeichnungen zurückgreifen, da in ihnen mehr zu sehen ist als in komplexen Zeichnungen (vgl. Beispiel b)).

Wie bereits erwähnt, besteht das Koordinatensystem der Skizze eines Orbits aus Real- und Imaginärteil. Dabei tragen wir wie in Abschnitt 1.1 den Realteil auf der waagerechten Achse und den Imaginärteil auf der orthogonalen Achse ab. Alle

Ergebnisse  $f^n(z_0)$  für  $n \in \mathbb{N}$  sowie beliebiges  $z_0$  werden in dieses Koordinatensystem eingetragen. Damit wir Einiges an diesem Graphen erkennen können und nicht nur eine wild zusammengewürfelte Menge von Punkten sehen, wird nun  $z_0$  durch einen Pfeil mit  $f(z_0)$  verbunden,  $f(z_0)$  durch einen Pfeil mit  $f^2(z_0)$  u.s.w., also wird für alle  $n$  der Punkt  $f^n(z_0)$  mit dem Punkt  $f^{n+1}(z_0)$  verbunden. Ein Beispiel war in Abschnitt 1.1 durch den Graph der Exponential-Funktion gegeben, wobei wir uns den Kreis wegdenken müssen.

Um solche Orbits kennenzulernen, betrachten wir einige

**Beispiele. d)** Es sei  $f: \mathbb{C} \rightarrow \mathbb{C}$ ,  $z \mapsto z^2 - 1$ . Hier gilt für den Orbit an der Stelle  $z_0 = i$

$$\mathcal{O}_f(i) = (i, -2, 3, 8, 63, \dots) .$$

In diesem Fall ist der Orbit schnell aus dem Bild verschwunden.

### Bild 2.7

Um den Orbit mit Maple zu bestimmen, wird die Isomorphie  $\mathbb{C} \cong \mathbb{R}^2$  als Vektorräume ausgenutzt, vgl. [Fis], 1.3.4. Ist  $c \in \mathbb{C}$ , so wird der Isomorphismus durch  $c \mapsto \langle \text{Re}(c), \text{Im}(c) \rangle$  gegeben, wobei  $\text{Re}(c)$  für den Realteil und  $\text{Im}(c)$  für den Imaginärteil von  $c$  stehen. Der Orbit der ersten sechs Glieder kann hiermit unter Maple über

```
> f:=s->s^2-1: c:=I: a[0] :=
> <Re(c)|Im(c)>: s:=c:
> for t from 1 to 5 do
> f(%): a[t]:=<Re(f(s))|Im(f(s))>:
> s:=f(s):
> od:
> a[0] := <Re(c)|Im(c)>: s:=evalf(c):
> for t from 1 to 5 do
> f(%):
> od;
```

berechnet werden. Der Graph in Bild 2.7 wurde durch

```
> f1:=plot(0,
> x=-2..10,y=-2..2,axes=normal,color=black):
> p0:=arrow(a[0],a[1]-a[0],shape=arrow):
```



```

> for i from 1 to 3 do
> p[i]:=arrow(a[i],a[i+1]-a[i],shape=arrow):
> od:
> display(f1,p0,seq(p[i],i=1..3));

```

erstellt.

e)  $f: \mathbb{C} \rightarrow \mathbb{C}, z \mapsto 2z$ , ist die komplexe Fortsetzung der Funktion aus Beispiel 2.1.3 a). Der Graph zu  $z_0 = 1$  in Bild 2.8 sieht hierbei ganz anders aus als der Graph zu Beispiel 2.1.3 a) und enthält tatsächlich weniger Information. Wir können an diesem Graphen nur erkennen, *wie* er verläuft, jedoch nicht, *warum* er so verläuft. Aus diesem Grund werden wir in solchen Fällen demnächst den reellen Graph auch bei komplexen Funktionen vorziehen, wenn es möglich ist.

### Bild 2.8

f) Nun betrachten wir  $f: \mathbb{C} \rightarrow \mathbb{C}, z \mapsto z^3 + iz^2$ . An der Stelle  $z_0 = 0$  erhalten wir

$$\mathcal{O}_f(0) = (0, 0, 0, \dots),$$

und für  $z_0 = i$  ergibt sich der Orbit mit Maple zu

$$\mathcal{O}_f(i) = (i, -2i, 4i, -80i, 505600i, -0.1292474712 \cdot 10^{18}i, \dots) .$$

Für zwei verschiedene Startpunkte sehen die Orbits hierbei recht unterschiedlich aus, wie in Bild 2.9 erkennbar ist.

### Bild 2.9

g) Ein weiteres Beispiel sei  $f: \mathbb{C} \rightarrow \mathbb{C}$ ,  $z \mapsto z^2 - 0.5 + 0.56i$ . Für den Startpunkt  $z_0 = -0.15 + 0.65i$  ergibt sich mit Maple

```
> f:=s->s^2-.5+.56*I:
> v:=-.15+.65*I: a[0] := <Re(v)|Im(v)>: s:=v:
> for t from 1 to 10 do
> f(%): a[t]:=<Re(f(s))|Im(f(s))>:
> s:=f(s):
> od:
> s:=evalf(v):
> for t from 1 to 10 do
> f(%):
> od;
```

```
-.9000 + .3650 I
.17677500 - .09700000 I
-.4781595994 + .5257056500 I
-.5477298279 + .0572575940 I
-.2032704677 + .4972766158 I
-.7059651496 + .3578366995 I
-.1296603111 + .0547595218 I
-.4861868090 + .5457997267 I
-.5615197284 + .0292787450 I
-.1855528395 + .5271188141 I
```

Der Orbit ist in Bild 2.10 sichtbar und sieht auf den ersten Blick sehr „wild“ aus. Trotz der Tatsache, dass einige Funktionsvorschriften recht einfach aussehen (wie in Beispiel g) mit  $z \mapsto z^2 + c$  mit  $c \in \mathbb{C}$ ), existieren viele komplizierte Orbits.

### Bild 2.10

Wie an den obigen Beispielen erkennbar ist, verhalten sich verschiedene Funktionen unterschiedlich, sogar ein und dieselbe Funktion verhält sich an verschiedenen Stellen nicht immer gleich, vergleiche hierzu Beispiel 2.1.3 b) und Beispiel f). Einige Orbits können im Unendlichen verschwinden, andere nähern sich an einen oder mehrere Punkte in der komplexen Ebene an. Iterieren wir mit diesen Punkten, an die sich die Orbits annähern, so stellen wir fest, dass die entsprechenden Orbits genau auf diesen Punkten liegen, denen sich die Orbits zuvor angenähert haben. Nach einer gewissen Anzahl von Iterationen gelangen wir wieder an den Anfangspunkt des Orbits. Dies motiviert die folgende

**Definition.** Es sei  $f: \mathbb{C} \rightarrow \mathbb{C}$  eine beliebige Funktion. Ein Punkt  $z_0 \in \mathbb{C}$  heißt *Fixpunkt* von  $f$ , falls  $f(z_0) = z_0$  gilt. Falls ein  $n \in \mathbb{N}$  existiert mit  $f^n(z_0) = z_0$ , so heißt  $z_0 \in \mathbb{C}$  *periodisch*. Das minimale  $n$  mit dieser Eigenschaft heißt *Periode* von  $z_0$  und  $\mathcal{O}_f(z_0)$  heißt *Periode- $n$ -Orbit*.

Wie gut erkennbar ist, handelt es sich bei Fixpunkten um Orbits der Periode  $n = 1$ <sup>3</sup>. Orbits, welche die Orbits zu Punkten aus einer Umgebung jedes seiner Punkte „anziehen“, heißen *attraktiv*. Solche Orbits werden wir in Kapitel 4 genauer untersuchen. Die Funktion in Beispiel b) ist linear. Wie man allerdings feststellen kann, ist die Dynamik linearer Funktionen aufgrund ihrer konstanten ersten Ableitung recht langweilig, vergleiche hierzu Beispiel 2.1.3 a), Beispiel e). Wir werden diese Funktionen im Folgenden nicht betrachten.

In den Beispielen d) und g) wurden jeweils quadratische Funktionen betrachtet. In gewisser Hinsicht handelt es sich bei quadratischen Funktionen um die „nächstschwierigen“ Funktionen im Vergleich zu den linearen Funktionen, die einige elementare Eigenschaften haben.

Die in Beispiel f) betrachtete Funktion verhält sich etwas komplizierter als lineare und quadratische Funktionen, auch wenn es in obigen Beispielen nicht erkennbar ist. Die Hintergrundtheorie, in der dies zu erkennen ist, befindet sich in [?], Part 3.

Ab jetzt erforschen wir quadratische Funktionen. Ihre allgemeine Form ist  $f(z) = az^2 + bz + c$  mit  $a, b, c \in \mathbb{C}$  und  $a \neq 0$ . Wir werden im folgenden die speziellen quadratischen Funktionen

$$Q_c: \mathbb{C} \rightarrow \mathbb{C}, \quad z \mapsto z^2 + c,$$

mit  $c \in \mathbb{C}$  betrachten. Der Grund hierfür ist, dass der *kritische* Punkt mit verschwindender Ableitung ein Punkt mit interessantem Orbit ist. Dieser kritische Punkt liegt für die Funktionen  $Q_c$  im Ursprung und ist damit leicht zu behandeln.

Wie in Beispiel b) erkennbar ist, kann sich eine Funktion  $Q_c$  für verschiedene  $z_0 \in \mathbb{C}$  unterschiedlich verhalten. Einige der Orbits scheinen dabei in Richtung unendlich

---

<sup>3</sup>Benutzen wir die Homöomorphie von  $\mathbb{C}$  zur Riemannschen Einheitskugel, so kann  $f(\infty) = \infty$  notiert werden, und so ergibt sich ein weiterer Fixpunkt. Aus diesem Grund muss nicht notwendig zwischen der Annäherung an einen Fixpunkt und der Annäherung an das Unendliche unterschieden werden.

zu laufen. Der folgende Satz hilft bei der Erkenntnis, ob ein Orbit gegen unendlich läuft.<sup>4</sup>

**Satz.** Es sei  $Q_c$  wie oben definiert, ferner  $z_0 \in \mathbb{C}$  mit  $|z_0| > 2$  und  $|z_0| \geq |c|$ . Dann gilt

$$|Q_c^n(z_0)| \rightarrow \infty.$$

Wir führen den *Beweis* über  $n$ . Wählen wir die Bezeichnungen des Satzes sowie  $z_n := Q_c^n(z_0)$ , so werden wir einige Male  $|z_n^2 + c| \stackrel{\circledast}{\geq} |z_n|^2 - |c|$  für alle  $n \geq 0$  benutzen, was weiter unten bewiesen wird. Damit ergibt sich

$$\begin{aligned} |z_1| &= |Q_c(z_0)| = |z_0^2 + c| \\ &\geq |z_0|^2 - |c| \geq |z_0|^2 - |z_0| \\ &= |z_0| \cdot (|z_0| - 1). \end{aligned} \tag{1}$$

Definieren wir nun  $\lambda := |z_0| - 1$ , so gilt  $\lambda > 1$  wegen  $|z_0| > 2$ . Führen wir den obigen Schritt noch einmal aus, so erhalten wir zunächst mit (1) und  $\circledast$

$$\begin{aligned} |z_2| &= |Q_c(z_1)| = |z_1^2 + c| \\ &\geq |z_1|^2 - |c| \geq \lambda^2 \cdot |z_0|^2 - |c|. \end{aligned}$$

Hieraus folgt sofort

$$\begin{aligned} |z_2| &\geq \lambda^2 \cdot |z_0|^2 - |c| \stackrel{i)}{>} \lambda \cdot |z_0|^2 - |c| \\ &\stackrel{ii)}{\geq} \lambda \cdot |z_0|^2 - \lambda \cdot |z_0| = \lambda \cdot |z_0| \cdot (|z_0| - 1) \\ &= \lambda^2 \cdot |z_0|, \end{aligned}$$

wobei an den Stellen i) und ii) jeweils  $\lambda > 1$  und  $|z_0| \geq |c|$  und bei i) zusätzlich  $|z_0| > 2$  eingeht.

Wir erhalten durch  $n$ -malige Hintereinanderausführung von (1)

$$\begin{aligned} |z_{n+1}| &= |Q_c(z_n)| = |z_n^2 + c| \\ &\geq |z_n|^2 - |c| \geq \lambda^n |z_0|^2 - |c| \\ &\geq \lambda^n |z_0| (|z_0| - 1) = \lambda^{n+1} |z_0|. \end{aligned}$$

Für den Grenzwert erhalten wir daraus unter Berücksichtigung von  $\lambda > 1$  sowie  $|z_0| > 2$

$$\lim_{n \rightarrow \infty} |Q_c^n(z_0)| \geq \lim_{n \rightarrow \infty} \lambda^n |z_0| = \infty,$$

<sup>4</sup>Die Schreibweise  $|Q_c^n(z_0)| \rightarrow \infty$  bedeutet, dass  $|Q_c^n(z_0)|$  für  $n \rightarrow \infty$  gegen Unendlich geht, also beliebig groß wird. Wenn dies anders geschrieben werden soll, so kann es durch  $\lim_{n \rightarrow \infty} |Q_c^n(z_0)| = \infty$  notiert werden, wobei zu beachten ist, dass  $\infty$  keine komplexe Zahl ist. Diese Darstellung ist insbesondere mit Hilfe der Homöomorphie zur Riemannschen Einheitskugel möglich.

also die Behauptung.

Es bleibt der Beweis von  $\circledast$ . Für zwei komplexe Zahlen  $c_1$  und  $c_2$  gilt mit der Dreiecksungleichung aus Abschnitt 1.1

$$|c_1| + |c_2| \geq |c_1 + c_2|,$$

woraus

$$|c_1| \geq |c_1 + c_2| - |c_2|$$

folgt. Setzen wir nun  $c_1 = z_0^2 + c$  und  $c_2 = -c$  ein, so erhalten wir

$$|z_0^2 + c| \geq |z_0^2 + c - c| - |-c| = |z_0^2| - |c|.$$

Damit haben wir die Lücke im Beweis gefüllt.  $\square$

Auf den ersten Blick kann man erstaunt darüber sein, dass im Satz die Voraussetzung  $|z_0| > 2$  und nicht  $|z_0| \geq 2$  steht. Um dies einzusehen, betrachten wir nun  $Q_{-2}$  an der Stelle  $z_0 = -2$  mit  $|z_0| = 2$ . Für den Orbit erhalten wir

$$\mathcal{O}_{Q_{-2}}(-2) = (-2, 2, 2, \dots),$$

also  $|Q_{-2}^n(-2)| \not\rightarrow \infty$ .

Mit Hilfe des soeben bewiesenen Satzes können wir nun entscheiden, ob ein bestimmter Orbit gegen Unendlich läuft oder nicht. Wir kennen jedoch noch nicht alle Orbits, die gegen Unendlich laufen und wissen auch nicht, welche Eigenschaften die Orbits der Punkte besitzen, die nicht gegen das Unendliche laufen. Im folgenden werden wir daher die Globale Dynamik der Funktionen  $Q_c$  untersuchen.

Bei Funktionen mit  $\mathbb{D} \subset \mathbb{R}$  lässt sich das Verhalten in der Umgebung von Fixpunkten  $x^F$  leichter untersuchen als bei Funktionen mit einem komplexen Definitionsbereich. Es gilt:

**Satz.**  $f: [a, b] \rightarrow \mathbb{R}$  sei stetig differenzierbar und  $x^F \in [a, b]$  ein Fixpunkt von  $f$ . Dann gilt für eine Iteration  $x_{n+1} := F(x_n)$  in der Umgebung von  $x^F$ :

- Ist  $|f'(x^F)| < 1$ , so konvergiert  $x_n \rightarrow x^F$ , wenn  $x_0$  nahe genug an  $x^F$  liegt.  $x^F$  heißt dann *attraktiver Fixpunkt*.
- Ist  $|f'(x^F)| > 1$ , so divergiert  $x_n$  vom Fixpunkt  $x^F$ .  $x^F$  heißt dann *abstoßender Fixpunkt*.

**Bemerkung.** Ein Fixpunkt  $x^F$  mit  $|f'(x^F)| = 1$  heißt *neutraler Fixpunkt*.

**Beispiel. h)** Welcher der drei Punkte  $x_1^F = -1.125 \dots$ ,  $x_2^F = 0.3389 \dots$  und  $x_3^F = 0.7864 \dots$  von  $f(x) = x^3 + 0.3$  ist attraktiv, welcher abstoßend?

Lösung.  $f(x) = x^3 + 0.3 \Rightarrow f'(x) = 3x^2$

$$\Rightarrow |f'(x_1^F)| > 1, \quad |f'(x_2^F)| < 1 \quad \text{und} \quad |f'(x_3^F)| > 1.$$

Dann sind  $x_1^F$  und  $x_3^F$  abstoßend,  $x_2^F$  ist attraktiv.

**Aufgabe 1.** Prüfen Sie, ob  $x_3^F$  für  $f(x) = \sqrt[3]{x - 0.3}$  attraktiv oder abstoßend ist.

Im letzten Satz stand eine Aussage über einen Fixpunkt, wenn man bereits weiß, dass einer existiert. Es muss jedoch einerseits entschieden werden, ob ein solcher Fixpunkt existiert, andererseits können noch zusätzliche Aussagen zu Fixpunkten gemacht werden. Dies sieht folgendermaßen aus:

**Banachscher Fixpunktsatz.** Sei  $f: [a, b] \rightarrow [a, b]$  stetig, es existiere ein  $\alpha < 1$  mit

$$|f(x) - f(y)| \leq \alpha |x - y| \quad \text{für alle } x, y \in [a, b].$$

Dann gilt:

- i) Es existiert ein eindeutiger Fixpunkt  $x^F \in [a, b]$ .
- ii)  $f^n(x_0) \rightarrow x^F$  für alle  $x_0 \in [a, b]$ .
- iii) Es gelten die Fehlerabschätzungen

$$\begin{aligned} |x_n - x^F| &\leq \frac{\alpha^n}{1-\alpha} \cdot |x_1 - x_0| && \text{a-priori-Abschätzung,} \\ |x_n - x^F| &\leq \frac{\alpha}{1-\alpha} \cdot |x_n - x_{n-1}| && \text{a-posteriori-Abschätzung.} \end{aligned}$$

**Beispiel. i)** Finden Sie ein Intervall  $[a, b]$  und ein  $\alpha$ , so dass die Voraussetzungen des Banachschen Fixpunktsatzes für die Funktion aus dem letzten Beispiel erfüllt sind.

Führen Sie eine a-priori-Abschätzung für  $x_0 = 0$  und einen Fehler von maximal  $10^{-4}$  durch.

Wenden Sie die a-posteriori-Abschätzung für den Fehler von  $x_9$  durch.

*Lösung.* Man versuche es mit  $[a, b] = [0, 0.5]$  für  $x_2^F = 0.3389 \dots$ . Für alle  $x \in [0, 0.5]$  gilt  $f(x) = x^3 + 0.3 \geq 0.3$  und  $f(x) \leq 0.5^3 + 0.3 \leq 0.5$ . Damit erhält man als Funktion  $f: [0, 0.5] \rightarrow [0, 0.5]$ .

Mit  $\alpha = 0.75$  ist die a-priori-Abschätzung erfüllt für

$$\begin{aligned} |x_n - x^F| &\leq \frac{\alpha^n}{1-\alpha} |x_1 - x_0| = \frac{0.75^n}{0.25} \cdot 0.3 \\ &= 1.2 \cdot 0.75^n \leq 10^{-4} \Leftrightarrow n \geq 32.7. \end{aligned}$$

Daher sind mindestens 33 Iterationen für die gesuchte Genauigkeit durchzuführen. Für die a-posteriori-Abschätzung erhält man damit für  $x_9$

$$|x_9 - x^F| \leq \frac{0.75}{0.25} \cdot |x_9 - x_8| = 3 \cdot |x_9 - x_8| = 3.8 \cdot 10^{-5}.$$

**Bemerkung.** Bei der Lösung des letzten Beispiels wird deutlich, dass die gewünschte Genauigkeit nicht erst bei der Anzahl der durch die a-priori-Abschätzung geforderten Schritte, sondern bereits zuvor erreicht wird.

Die a-priori-Abschätzung ist pessimistischer als die a-posteriori-Abschätzung.

**Aufgabe 2.** Entwerfen Sie eine Unterrichtsreihe zur Einführung der Nullstellenbestimmung von Funktionen in der Jahrgangsstufe 11 mithilfe des Fixpunkt- oder Bisektionsverfahrens.

### 2.1.4 Newton-Verfahren

Nachdem bisher Verfahren besprochen wurden, die auch in der Mittelstufe und zu Beginn der Oberstufe verwendet werden können, beschäftigen wir uns jetzt mit einem Verfahren zur Nullstellenbestimmung, das erst in der Sekundarstufe II nach der Einführung der Ableitung verwendet werden kann.

$f$  sei eine differenzierbare Funktion, und  $x_0 \in \mathbb{D}_f$  ein Ausgangspunkt. Der Ausgangspunkt ist – soweit Vorkenntnisse es erlauben – möglichst nah an der Nullstelle zu wählen, damit die Anzahl der Iterationsschritte möglichst gering ist.

Der grundsätzliche Hintergedanke liegt in der Taylor-Entwicklung und in Verbindung der *Linearität* der Funktion  $f$ . Wählt man  $x_0$ , so gilt für  $x \in U(x_0)$  (wobei  $U(x_0)$  eine hinreichend klein gewählte Umgebung von  $x_0$  ist)

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0).$$

Bei dem Graphen handelt es sich um eine Tangente an den Funktionsgraphen an der Stelle  $x_0$ . Die Tangente hat die Gleichung

$$y = f(x_0) + f'(x_0)(x - x_0) = f'(x_0) \cdot x + f(x_0) - f'(x_0) \cdot x_0.$$

Die Hoffnung besteht darin, dass die Nullstellen dieser Tangente in der Nähe der gesuchten Nullstelle von  $f$  liegt. Da sie vermutlich nicht exakt auf der Nullstelle liegt, wird man dieses Verfahren wiederholen. An einer Nullstelle gilt  $y = 0$

$$\Rightarrow 0 = f'(x_0) \cdot x_1 + f(x_0) - f'(x_0) \cdot x_0 \Rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)},$$

falls  $f'(x_0) \neq 0$  ist. Damit ergibt sich die Iteration nach dem *Newton-Verfahren*:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad \text{für } n = 0, 1, \dots \quad (1)$$

**Bemerkung.** Das Newton-Verfahren funktioniert nur dann, wenn  $f'(x_n) \neq 0$  für alle  $n$  gilt.

**Beispiel. a)** Bestimmen Sie die Nullstelle für  $f(x) = x^2 - 2$  näherungsweise mit dem Newton-Verfahren.

*Lösung.* Für die Iteration gilt

$$x_{n+1} = x_n - \frac{x_n^2 - 2}{2x_n} = \frac{1}{2}x_n + \frac{1}{x_n}$$

Startet man mit  $x_0 = 2$ , so ergeben sich

$n$	$x_n$
1	1.5
2	1.4167
3	1.41421569
4	1.414213562

Im vierten Schritt wird daher die Nullstelle auf zehn Stellen genau bestimmt.

**Bemerkung.** Eine Alternative zur Bestimmung der Nullstellen von quadratischen Funktionen nach dem Newton-Verfahren besteht in der Durchführung mithilfe des *Heron-Verfahrens*.

Umständlich ist es teilweise, stets die Ableitung  $f'(x_n)$  zu bestimmen. Möchte man dies umgehen, so verwendet man die Gleichung

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)} \quad \text{für } n \in \mathbb{N}. \quad (2)$$

Eine weitere Variante besteht in der Verwendung der Steigung der Sekante durch  $(x_{n-1}, f(x_{n-1}))$  und  $(x_n, f(x_n))$  statt der Steigung der Graphen von  $f$ . Damit erhält man

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \cdot f(x_n) \quad \text{für } n \in \mathbb{N}. \quad (3)$$

**Beispiel. b)** Ein natürlicher Spline soll die Funktion  $f$  mit  $f(x) = \frac{1}{1+x^2}$  an den Stellen  $x = -3, -2, -1, 0, 1, 2, 3$  interpolieren. Die Spline-Funktion  $p$  ist achsensymmetrische zur  $y$ -Achse, da es auch für  $f$  gilt. Damit gilt also

$$p(x) = ax^6 + bx^4 + cx^2 + d.$$

Dies führt zu einem LGS mit der Lösung

$$p(x) = -\frac{1}{100}x^6 + \frac{3}{20}x^4 - \frac{16}{25}x^2 + 1.$$

Die Graphen von  $f$  und  $p$  sehen folgendermaßen aus:



**Bemerkung.** Bei diesem Verfahren sind zwei Startwerte  $x_0$  und  $x_1$  zu wählen.

**Aufgabe 1.** Bestimmen Sie die Nullstellen der Funktion  $f$  mit  $f(x) = x^2 - 2$  mit Verfahren (3).

**Aufgabe 2.** Berechnen Sie eine Nullstelle von  $f$  mit  $f(x) = x^3 + x - 1$  im Intervall  $[0, 1]$  mithilfe des Newton-Verfahrens und der Verfahren aus (2) und (3). Schätzen Sie hiermit die Genauigkeit der Verfahren gegeneinander ab. Verwenden Sie z.B. den Startwert  $x_0 = 1$ .

*Lösung.* (1) Nach dem Newton-Verfahren ergibt sich

$n$	$x_n$	$f(x_n)$
0	1.0	1.0
1	0.75	0.171875
2	0.686046512	0.008941037
3	0.682339583	0.000028231
4	0.682327804	0.0000000009

Welche Bedeutung kann man hierbei den Werten  $f(x_n)$  zuordnen?

(2) Berechnet man den Ableitungswert nur einmal, so ergibt sich mit  $f'(x_0) =$

$$3x_0^2 + 1 = 4$$

$n$	$x_n$
0	1.0
1	0.75
2	0.6919134110
3	0.6861226806
4	0.6838414890
5	0.6829333481
6	0.6825703313

(3) Nach Gleichung (2) ist das Verfahren von der Anzahl der Rechenschritte geringer, auch wenn es auf den ersten Blick komplizierter wirkt. Außerdem sind zwei Punkte vorzugeben, durch die die erste Sekante verlaufen soll. Hierzu wählt man sinnvollerweise  $[x_0, f(x_0)] = [0, -1]$  und  $[x_1, f(x_1)] = [1, 1]$ . Dann ergibt sich

$n$	$x_n$
0	0
1	1
2	0.5
3	$0.\overline{63}$
4	0.6900523560
5	0.6820204196
6	0.6823257814

Die Verfahren (2) und (3) erreichen nach dem sechsten Schritt dieselbe Genauigkeit wie das Newton-Verfahren bereits nach dem vierten Schritt.

## 2.2 Lösung von Linearen Gleichungssystemen

### 2.2.1 Einleitung

Auf dem mathematischen Hintergrund der linearen Algebra wird in diesem Abschnitt nicht eingegangen. Es wird als Grundwissen vorausgesetzt, da es sich hierbei um Inhalt der ersten beiden Semester handelt. Um Grundlagen über lineare Algebra nachzulesen und auf Übungsaufgaben zurückzugreifen vgl. [Fis] und [StGr].

Ein lineares Gleichungssystem mit  $n$  Gleichungen und  $n$  Unbekannten lässt sich in der Form  $A \cdot x = b$  mit  $A = (a_{ij})_{1 \leq i, j \leq n} \in M(n, \mathbb{R})$ ,  $x \in \mathbb{R}^n$  und  $b \in \mathbb{R}^n$  notieren. Ausgeschrieben bedeutet dies

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}. \quad (*)$$

Bei numerischen Lösungen von linearen Gleichungssystemen unterscheidet man zwischen zwei Verfahren:

- i) *Direktes Verfahren.* Dies sind solche Verfahren, die in endlich vielen Rechenschritten eine exakte Lösung von  $\textcircled{*}$  liefern. Es ist nur möglich, wenn jeder Schritt numerisch exakt lösbar ist.
- ii) *Iteratives Verfahren.* Dies sind solche Verfahren, die eine gegen eine exakte Lösung von  $\textcircled{*}$  konvergente Folge von Lösungen erzeugen.

Hier beschäftigen wir uns zunächst mit einem „klassischen“ direkten Verfahren, dem Gauß-Algorithmus.

### 2.2.2 Gauß-Algorithmus

**Beispiel 1.** Es soll

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{pmatrix} \cdot x = \begin{pmatrix} 8 \\ 7 \\ 6 \end{pmatrix}$$

gelöst werden. Aus der letzten Zeile kann man  $x_3 = 1$  ablesen. Damit lautet die zweite Zeile

$$4x_2 + 5 = 7 \Rightarrow x_2 = \frac{1}{2},$$

und damit folgt

$$\begin{aligned} x_1 + 1 + 3 &= 8 \Rightarrow x_1 = 4 \\ \Rightarrow x &= {}^t(4, \frac{1}{2}, 1). \end{aligned}$$

Das Verfahren aus Beispiel 3.1 lässt sich stets auf lineare Gleichungssysteme in oberer Dreiecksform anwenden, wenn

$$\begin{pmatrix} a_{11} & \cdots & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix} \cdots \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad \textcircled{*}$$

gilt. Man erhält sodann

**Bemerkung.** Das LGS  $\circledast$  besitzt genau dann ein Lösung  $x \in \mathbb{R}^n \setminus \{0\}$ , wenn  $a_{ii} \neq 0$  für alle  $i = 1, \dots, n$  gilt. Eine Lösung ist gegeben durch

$$x_i := \frac{1}{a_{ii}} \left( b_i - \sum_{j=i+1}^n a_{ij} x_j \right).$$

Der Aufwand dieser Methode beträgt

$$\frac{n(n+1)}{2} \text{ Punktoperationen.}$$

**Aufgabe 1.** Zeigen Sie, dass die Aufwandangabe stimmt.

Nach der Theorie der linearen Algebra ist die Multiplikation einer Zeile mit  $\lambda \neq 0$  und die Addition des  $\lambda$ -fachen einer Zeile der Matrix zu einer anderen Zeile zulässig. Für mathematische Hintergründe vgl. [Fis], Kapitel 0. Fasst man es zusammen, so ergibt sich:

**Bemerkung.** Bei der Bearbeitung eines LGS sind die folgenden Verfahren zugelassen:

Sind  $z_i$  und  $z_j$  die  $i$ -te und  $j$ -te Zeile des LGS mit  $i < j$  und  $\lambda \in \mathbb{R}$ , so kann  $z_j := z_j - \lambda z_i$  gesetzt und  $z_i \leftrightarrow z_j$  vertauscht werden, ohne die Lösungsmenge zu verändern.

Mithilfe dieses Verfahrens kann ein LGS auf obere Dreiecksform gebracht werden. OBdA sei  $a_{11} \neq 0$ , denn ansonsten kann dies durch Vertauschung von Zeilen erreicht werden falls ein  $a_{i1} \neq 0$  existiert. Falls  $a_{i1} = 0$  für alle  $i = 1, \dots, n$ , so ist der Rang von  $A$ ,  $\text{rg} A < n$  (d.h.  $\det A = 0$ ). Daraus folgt, dass keine eindeutige Lösung existiert. Dieses Argument lässt sich auch auf die folgenden Schritte übertragen. Führt man die soeben geschilderten Schritte wiederholt durch, so erhält man damit

### Gauß-Algorithmus zur Überführung von $A \in M(n, K)$ in obere Dreiecksform.

Für  $i = 1, \dots, n - 1$  führe die folgenden Schritte durch:  
Erzeuge Nulle unterhalb des Diagonalelements  $a_{ii} \neq 0$ .

1. falls

- $a_{ii} \neq 0$ : fertig
- $a_{ii} = 0$ :
  - falls  $a_{ji} = 0 \forall j = i + 1, \dots, n$ :  
 $A$  ist nicht regulär (d.h. es exist. keine eindeutige Lösung)  $\leadsto$  stopp
  - falls  $a_{ji} \neq 0$  für ein  $j = i + 1, \dots, n$  geeignet:  
sei  $j \geq i + 1$  minimal mit  $a_{ji} \neq 0 \leadsto z_i \leftrightarrow z_j$

2. Eliminationsverfahren:

- für  $j = i + 1, \dots, n$ : eliminiere  $a_{ji}$  durch:

$$z_j := z_j - \frac{a_{ji}}{a_{ii}} \cdot z_i$$

**Aufgabe 2.** Notieren Sie das Flussdiagramm zum Gauß-Algorithmus.

**Bemerkung.** Der Aufwand des Gauß-Algorithmus beträgt  $\frac{n^3}{3} - \frac{n}{3}$  Operationen.

**Aufgabe 3.** Zeigen Sie die Bemerkung.

*Lösung.* Beweis per Induktion.

$n = 1 \Rightarrow \frac{1}{3} - \frac{1}{3} = 0$ , also kein Aufwand.

$n \rightarrow n + 1$ :

$$\begin{aligned} \frac{(n+1)^3}{3} - \frac{n+1}{3} &= \frac{1}{3} (n^3 + 3n^2 + 3n + 1 - n - 1) \\ &= \frac{1}{3} (n^3 - n + (3n^2 + 3n)) \\ &= \frac{1}{3} (n^3 - n) + \underbrace{\frac{n^2 + n}{n(n+1)}} \end{aligned}$$

Für die Vertauschung innerhalb der Zeile bzw. der Spalte sind jeweils  $\frac{n(n+1)}{2}$  Schritte nötig (nochmal Induktion), und damit insgesamt  $2 \cdot \frac{n(n+1)}{2}$  Schritte.

**Beispiel 2.** Es soll  $Ax = b$  mit

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 4 & -2 & 6 \\ 3 & 1 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 9 \\ -4 \\ 9 \end{pmatrix}$$

auf rechte obere Dreiecksform gebracht werden.

Es gilt

$$(A|b) = \left( \begin{array}{ccc|c} 1 & 2 & -1 & 9 \\ 4 & -2 & 6 & -4 \\ 3 & 1 & 0 & 9 \end{array} \right) \rightsquigarrow \left( \begin{array}{ccc|c} 1 & 2 & -1 & 9 \\ 0 & -10 & 10 & -40 \\ 0 & -5 & 3 & -18 \end{array} \right) \rightsquigarrow \left( \begin{array}{ccc|c} 1 & 2 & -1 & 9 \\ 0 & -10 & 10 & -40 \\ 0 & 0 & -2 & 2 \end{array} \right)$$

**Bemerkung.** An obigem Beispiel wird deutlich, dass für die rechte obere Dreiecksmatrix  $B = (b_{ij})$ , die aus  $A = (a_{ij})$  erzeugt wird, folgendes gilt:

$$\det A = (-1)^\ell \det B = (-1)^\ell \prod_{i=1}^n b_{ii},$$

wobei  $\ell = \#$  (nötiger Zeilenvertauschungen). Dies gilt allgemein und lässt sich aus den Eigenschaften der Determinante herleiten, vgl. [Fis], Satz 3.1.3.

**Beispiel 3.** Für die Matrix  $A$  aus Beispiel 2 ergibt sich

$$\det A = 1 \cdot (-10) \cdot (-2) = 20,$$

da keine Zeilenvertauschungen vorgenommen wurden.

**Aufgabe 3.** Bestimmen Sie die Lösungsmenge von

$$\left( \begin{array}{ccc|c} 3 & -2 & 5 & 3 \\ 5 & 2 & 1 & -1 \\ 6 & -4 & 10 & 5 \end{array} \right).$$

*Lösung.* Es existiert keine Lösung, denn die Matrix kann zu

$$\left( \begin{array}{ccc|c} 3 & -2 & 5 & 3 \\ 0 & \frac{16}{3} & -\frac{22}{3} & -6 \\ 0 & 0 & 0 & -1 \end{array} \right)$$

umgeformt werden.

**Bemerkung.** Bei LGS existieren drei verschiedene Arten von Lösungsmengen:

- i) Lösungsmengen mit einem Element,
- ii) leere Lösungsmenge,
- iii) Lösungsmenge mit unendlich vielen Elementen.

**Aufgabe 4.** a) In welchen Jahrgangsstufen werden LGS benötigt, und wo macht es Sinn, Matrizen einzuführen?

b) Wann kann der Gauß-Algorithmus eingeführt werden?

## 3 Interpolation

### 3.1 Einleitung

Häufig ist man der Verallgemeinerung der Ergebnisse eines Versuchs, einer Messreihe interessiert. Hierzu ist es notwendig, eine Formel / Funktion zu definieren, die die Messergebnisse beschreibt, und mit der näherungsweise andere Werte vorhergesagt werden können. Diese Definition einer Funktion ist nur dann möglich, wenn ein hinreichender Umfang an Ergebnissen vorliegt, wie das folgende Beispiel zeigt.

**Beispiel 1.** Es wurde im Abstand von 1s die Temperatur eines stark gekühlten Stoffs gemessen. Die Ergebnisse lauten

$$T_1 = 1\text{K}, \quad T_2 = 2\text{K}.$$

Es soll eine Funktion bestimmt werden, mit der die Ergebnisse beschrieben werden können.

*Lösung.* Zwei Möglichkeiten von unendlich vielen bei den Messreihen  $(0, 1)$  und  $(1, 2)$  sind

$$f_1(x) = x + 1 \quad \text{und} \quad f_2(x) = x^2 + 1.$$

Wie das Beispiel zeigt sind Einschränkungen für die Funktionen zu machen. Dies wird in den nächsten Abschnitten geschehen. Zunächst wird jedoch die grundlegende Definition gegeben.

**Definition (Interpolationsproblem).** Gegeben sind  $n + 1$  Wertepaare  $(x_i, f_i)$ ,  $i = 0, \dots, n$  mit  $x_i \leq x_j$  für alle  $i, j$  mit  $i \neq j$ . Gesucht ist eine stetige Funktion  $f$  mit der Eigenschaft  $f(x_i) = f_i$  für alle  $i$ . Man nennt die  $x$ -Werte *Stützstellen*, die  $y$ -Werte *Stützwerte*, die Wertepaare *Stützpunkte*. Eine Funktion  $f$  mit Eigenschaften  $f(x_i) = f_i$  für alle  $i$  heißt *Interpolierende* der Wertepaare  $(x_i, f_i)$ .

### 3.2 Polynominterpolation

**Beispiel 2.** Es seien die Werte

$$\begin{array}{c|c|c|c|c} x_i & -1 & 0 & 1 & 2 \\ \hline f_i & 5 & -2 & 9 & -4 \end{array}$$

gegeben. Bestimmen Sie ein Polynom maximalen Grades, dessen Graph die Punkte enthält.

*Lösung.* Es handelt sich um ein Polynom  $f$  vom Grad  $n$ , d.h.

$$f(x) = \sum_{i=0}^n a_i x^i.$$

Hier existieren vier Werte, d.h. es können vier Gleichungen aufgestellt werden.

Da ein Polynom  $n + 1$  unbekannte Koeffizienten hat, kann der Grad des Polynoms höchstens  $n - 1 = 3$  sein. Also

$$f(x) = a_3x^3 + a_2x^2 + a_1x + a_0.$$

Damit ergibt man das LGS

$$f(-1) = 5 \Rightarrow -a_3 + a_2 - a_1 + a_0 = 5$$

$$f(0) = -2 \Rightarrow a_0 = -2$$

$$f(1) = 9 \Rightarrow a_3 + a_2 + a_1 + a_0 = 9$$

$$f(2) = -4 \Rightarrow 8a_3 + 4a_2 + 2a_1 + a_0 = -4$$

$$\begin{aligned} \rightsquigarrow \left( \begin{array}{ccc|c} -1 & 1 & -1 & 7 \\ 1 & 1 & 1 & 11 \\ 8 & 4 & 2 & -2 \end{array} \right) &\rightsquigarrow \left( \begin{array}{ccc|c} 1 & 1 & 1 & 11 \\ 0 & 1 & 0 & 9 \\ 0 & 2 & -1 & 9 \end{array} \right) &\rightsquigarrow \left( \begin{array}{ccc|c} 1 & 1 & 1 & 11 \\ 0 & 1 & 0 & 9 \\ 0 & 0 & 1 & 9 \end{array} \right) \\ &\rightsquigarrow \left( \begin{array}{ccc|c} 1 & 0 & 0 & -7 \\ 0 & 1 & 0 & 9 \\ 0 & 0 & 1 & 9 \end{array} \right) \end{aligned}$$

$$f(x) = -7x^3 + 9x^2 + 9x - 2$$

Das Verfahren lässt sich verallgemeinern, es gilt:

**Satz.** *Existenz und Eindeutigkeit des Interpolationspolynoms*

Gegeben seien  $n + 1$  Wertepaare  $(x_i, f_i)$ ,  $i = 0, \dots, n$ . Dann existiert ein eindeutiges Polynom  $f$  mit  $\deg f \leq n$  mit  $f(x_i) = f_i$  für alle  $i$ .

**Bemerkung.** Für  $n = 1$  spricht man von *linearer Interpolation*, für  $n = 2$  von *quadratischer Interpolation*.

**Aufgabe 1.** Bestimmen Sie analog zu Beispiel 2 die Polynome zu den folgenden Wertetabellen.

$$\text{a) } \begin{array}{c|c|c|c|c} x_i & -1 & 0 & 1 & 2 \\ \hline f_i & -11 & -2 & 3 & 22 \end{array} \quad \text{b) } \begin{array}{c|c|c|c|c} x_i & -1 & 0 & 2 & 3 \\ \hline f_i & 1 & -3 & 1 & -27 \end{array}$$

*Lösung.*

a)  $f(x) = -11 + 9(x + 1) - 2(x + 1)x + 3(x + 1)x(x - 1)$

b)  $f(x) = 1 - 4(x + 1) + 2(x + 1)x - 3(x + 1)x(x - 2)$

**Aufgabe 2.** a) In welcher Jahrgangsstufe lässt sich das Verfahren aus Beispiel 2 einsetzen?

b) Entwickeln Sie einen kurzen Entwurf für eine Einführungsstunde dieses Interpolationsverfahrens.



Allgemein lässt sich das Polynom besser als durch LGS bestimmen. Eine ist die *Lagrange-Form*. Es gilt

$$f(x) = \sum_{i=0}^n f_i l_i(x) \quad \text{mit } l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}.$$

Auch hierbei ist der Aufwand hoch. Das Verfahren lässt sich allerdings durch einen Algorithmus durchführen.

**Definition.** Gegeben seien die Stützpunkte  $(x_i, f_i)$ ,  $i = 1, \dots, n$ . Man definiert *dividierte Differenzen* rekursiv durch

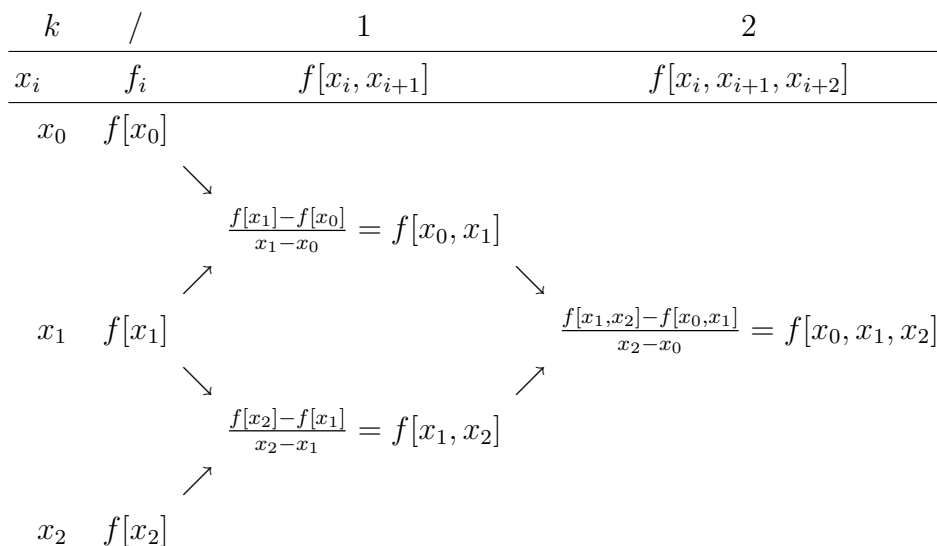
1.  $f[x_i] := f_i$  für alle  $i = 0, \dots, n$ .

2. Für  $k = 1, \dots, n$ :

- Für  $i = 0, \dots, n - k$ :

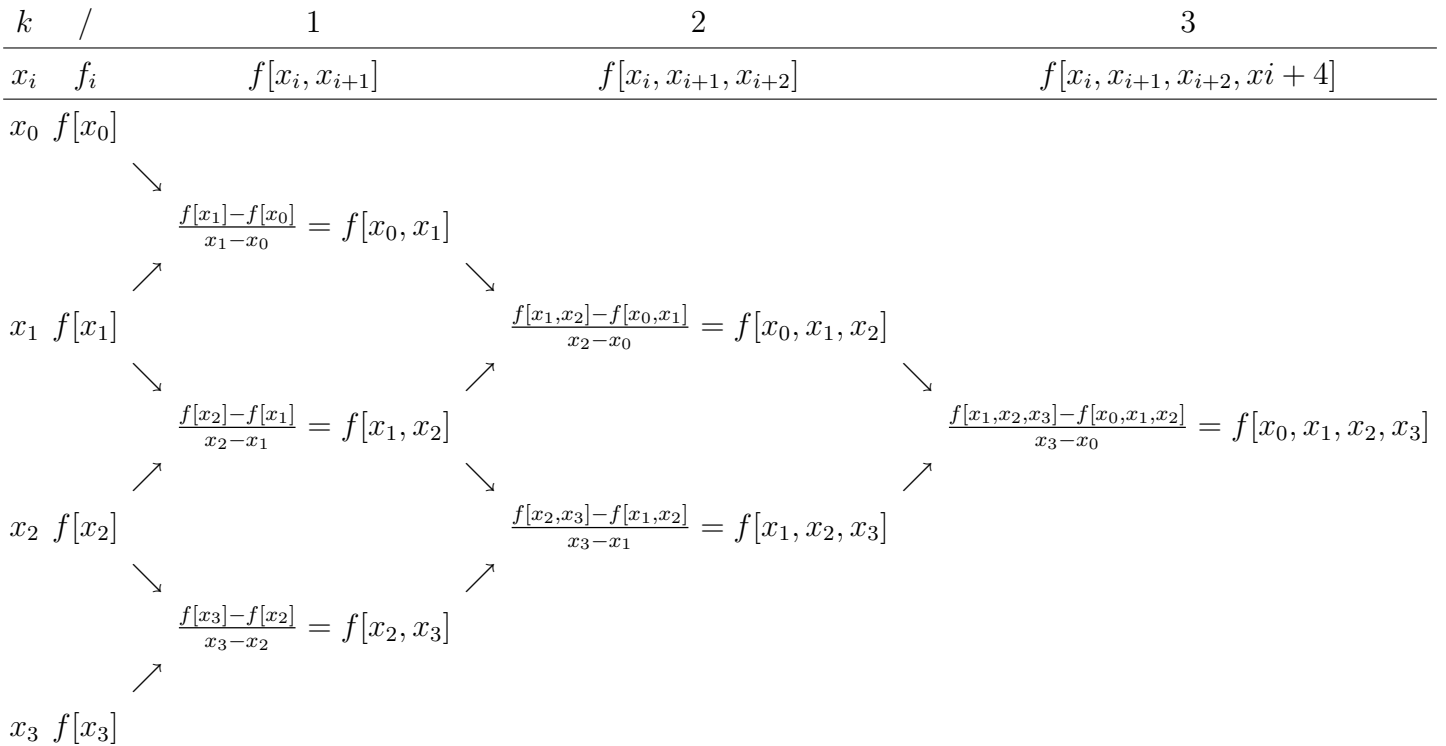
$$f[x_i, x_{i+1}, \dots, x_{i+k}] := \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

**Bemerkung.** Die Umsetzung obiger rekursiver Definition ist ein Algorithmus. Es lässt sich übersichtlich darstellen:



**Aufgabe 3.** Ergänzen Sie das Diagramm aus der Bemerkung für  $k = 3$ .

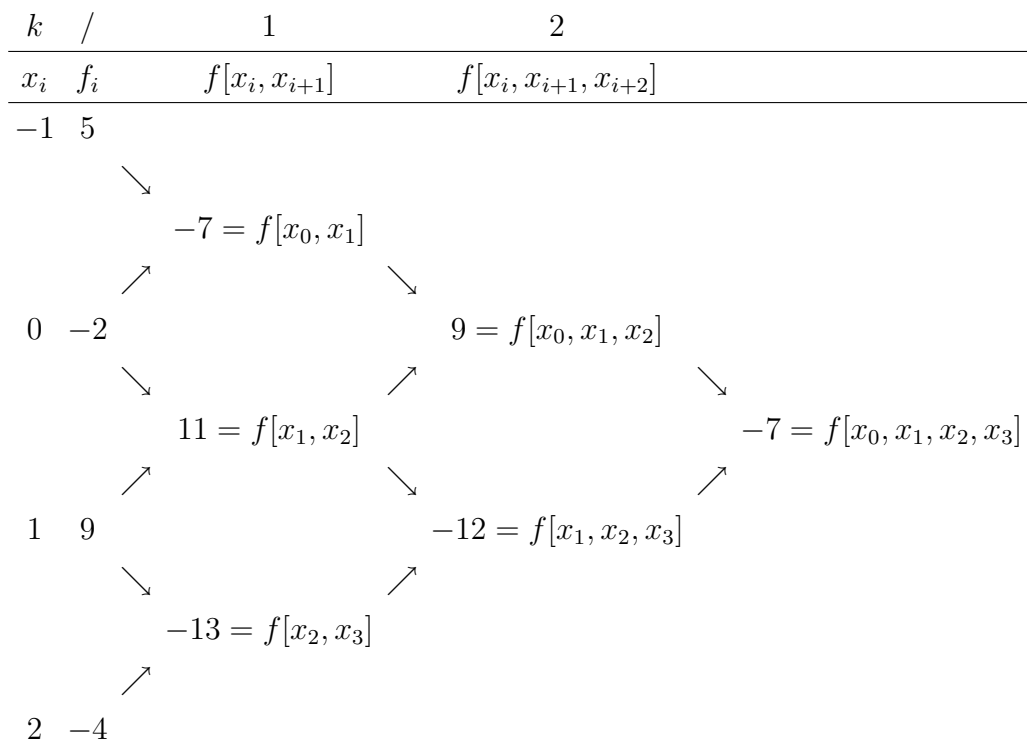
Lösung.



**Aufgabe 4.** Zeichnen Sie ein Flussdiagramm zu dem Algorithmus.

**Beispiel 3.** Berechnen Sie die dividierten Differenzen zu Beispiel 2.

Lösung.



**Bemerkung.**

- Die  $x_i$  müssen nicht der Größe nach geordnet werden.
- Weitere Wertepaare sind bedenkenlos ergänzbar.

Mit den dividierten Differenzen können Interpolationspolynome leicht berechnet werden.

**Newtonsche Interpolationsformel.** Gegeben seien die Stützpunkte  $(x_i, f_i)$ ,  $i = 1, \dots, n$ . Das Interpolationspolynom lautet dann

$$f(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots + a_n \prod_{j=0}^{n-1} (x - x_j)$$

mit  $a_i = f[x_0, \dots, x_i]$ .

Kurz notiert

$$f(x) = \sum_{i=0}^n f[x_0, \dots, x_i] \prod_{j=0}^{i-1} (x - x_j).$$

**Bemerkung.** Man beachte, dass das Interpolationspolynom eindeutig bestimmt ist, und damit die Polynome nach der Newtonschen Interpolationsformel und der Lagrange-Form übereinstimmen.

**Beispiel 4.** Auf den Datensatz aus Beispiel 2 wird die Newtonsche Formel angewandt. Dann ergibt sich mit  $n = 3$

$$\begin{aligned} f(x) &= \sum_{i=0}^3 f[x_0, \dots, x_i] \prod_{j=0}^{i-1} (x - x_j) \\ &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &\quad + f[x_0, \dots, x_3](x - x_0)(x - x_1)(x - x_2) \\ &= 5 - 7(x + 1) + 9(x + 1)x - 7(x + 1)x(x - 1) \end{aligned}$$

**Aufgabe 5.** Wenden Sie die Newtonsche Interpolationsformel auf die Werte aus Aufgabe 1 an, und bestätigen Sie damit die alten Ergebnisse.

**Aufgabe 6.** a) Setzen Sie den Algorithmus der Newtonschen Interpolationsformel unter MAPLE um.

b) In welcher Stufe der Schule lässt sich dieses Verfahren inklusive der Verwendung von MAPLE einbauen?

### 3.3 Splineinterpolation

Bei der Polynominterpolation treten Probleme auf:

- Pro Messwert wird der Grad des Polynoms um eins erhöht.
- Damit entstehen starke Schwankungen am Rande des Stützstellenintervalls.

Diese Probleme können mit folgender Idee umgangen werden: Das Intervall ist in Teilintervalle aufzuteilen. In diesen Teilintervallen wird dann das Interpolationspolynom niedrigsten Grades bestimmt und verwendet. Zum Beispiel können Zweierpaare an Punkten linear interpoliert werden.

**Beispiel 5.** Die Daten aus Beispiel 2 in Abschnitt 3.2 sollen abschnittsweise linear interpoliert werden. Hierdurch ergeben sich Paare von benachbarten Punkten. Der Graph sieht dann folgendermaßen aus:

**Aufgabe 7.** Bestimmen Sie stückweise lineare Interpolierende zu Beispiel 5 rechnerisch.

*Lösung.* Die Funktion lautet

$$s(x) = \begin{cases} -7x - 2 & \text{für } x \in [-1, 0] \\ 11x - 2 & \text{für } x \in [0, 1] \\ -13x + 22 & \text{für } x \in [1, 2] \end{cases}$$

**Achtung!** Die lineare Interpolation weist Knickstellen auf, was unerwünscht ist. Dadurch ist die zugehörige Funktion nicht an allen Stellen differenzierbar.

Man fordert im allgemeinen eine zweifache Differenzierbarkeit der Interpolierenden. Diese Funktionen heißen dann *Splines*.<sup>5</sup> Wir schränken uns hier auf Funktionen vom Grad drei ein.

---

<sup>5</sup>Der Begriff *Spline* stammt aus dem Schiffbau.

### 3.3.1 Interpolation mit kubischen Splines

**Definition.**

- a) Eine Funktion  $s: [a, b] \rightarrow \mathbb{R}$  heißt *kubischer Spline* zu Stützstellen  $a = x_0 < x_1 < \dots < x_n = b$ , falls gilt:
- $s \in \mathcal{C}^2([a, b])$ ,
  - $s$  ist auf  $[x_i, x_{i+1}]$  jeweils ein Polynom vom Grad 3 für alle  $i = 0, \dots, n-1$ .
- b)  $s$  heißt *interpolierender Spline* zu den Werten  $y_0, \dots, y_n$ , wenn zusätzlich zu a) gilt:

$$s(x_i) = y_i \quad \text{für alle } i.$$

- c)  $s$  heißt *periodischer interpolierender Spline* mit Periode  $b - a$ , wenn zusätzlich zu a) und b) gilt:

$$s(a) = s(b), \quad s'(a) = s'(b), \quad s''(a) = s''(b).$$

- d)  $s$  heißt *natürlicher interpolierender Spline*, wenn zusätzlich zu a) und b) gilt:

$$s''(a) = s''(b) = 0.$$

Ein Polynom dritten Grades besitzt vier Koeffizienten. Teilt man das zu untersuchende Intervall in  $n$  Teilintervalle, so ergeben sich damit  $4n$  Freiheitsgrade für die Funktion, die stückweise ein kubisches Polynom ist.

Pro Intervall sind zwei Bedingungen zu erfüllen, womit sich  $2n$  Bedingungen ergeben. Da die erste und die zweite Ableitung an den Stellen  $x_1, \dots, x_{n-1}$  stetig sein sollen, ergeben sich  $2(n-1)$  zusätzliche Bedingungen.

Insgesamt müssen daher  $4n - 2$  Bedingungen erfüllt sein.

**Bemerkung.** Es dürfen zwei weitere Bedingungen gestellt werden.

- a) *Periodischer Spline:* Es gilt  $y_0 = y_n$  und

$$s'(a) = s'(b), \quad s''(a) = s''(b).$$

- b) *Natürlicher Spline:* Die zusätzliche Bedingung lautet hier

$$s''(a) = 0 = s''(b).$$

Leichter ist die Berechnung über die *Momente*  $M_i = s''(x_i)$ .

**Satz 1 (Berechnung der Momente).** Für die Momente  $M_i = s''(x_i)$ ,  $i = 0, \dots, n$  gilt:

$$\mu_i M_{i-1} + 2M_i + (1 - \mu_i)M_{i+1} = 6f[x_{i-1}, x_i, x_{i+1}] \quad \text{für alle } i = 1, \dots, n-1.$$

Hierbei bedeutet

$$\mu_i = \frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}}.$$

Für die Varianten c) und d) von Splines kommen die folgenden Bedingungen hinzu:

i) periodischer Spline:  $M_0 = M_n$  und

$$\mu_n M_{n-1} + 2M_n + (1 - \mu_n)M_1 = 6f[x_{n-1}, x_n, x_{n+1}]$$

mit

$$x_{n+1} := x_n + x_1 - x_0, \quad f_n = f_{n+1},$$

$$\mu_n = \frac{x_n - x_{n-1}}{x_n + x_1 - x_0 - x_{n-1}}.$$

ii) natürlicher Spline:  $M_0 = M_n = 0$ .

**Bemerkung.** Falls  $x_i - x_{i-1} = \text{const}$  für alle  $i = 1, \dots, n$ , so gilt  $\mu_j = 0.5$  für alle  $j = 0, \dots, n$ .

Nun kommt die Umsetzung von Splines.

**Satz 2 (Berechnung von Splines aus Momenten).** Gegeben seien die Wertepaare  $(x_i, y_i)$  für  $i = 0, \dots, n$  und dazugehörige Momente  $M_i := s''(x_i)$  eines interpolierenden Splines. Dann gilt auf den Intervallen  $[x_{i-1}, x_i]$ ,  $i = 1, \dots, n$ :

$$s(x) = M_{i-1} \frac{(x_i - x)^3}{6h_i} + M_i \frac{(x - x_{i-1})^3}{6h_i} + C_i \left( x - \frac{x_{i-1} + x_i}{2} \right) + D_i$$

mit

$$h_i := x_i - x_{i-1}$$

und

$$C_i := \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}), \quad D_i := \frac{y_i - y_{i-1}}{2} - \frac{h_i^2}{12}(M_i + M_{i-1}).$$

**Beispiel 6.** Die Daten aus Beispiel 2 sollen mit natürlichem Spline interpoliert werden. Aus Beispiel 3 haben wir

$$f[x_0, x_1, x_2] = 9 \quad \text{und} \quad f[x_1, x_2, x_3] = -12.$$

Nach Satz 1 lauten die Gleichungen der Momente

$$0.5M_0 + 2M_1 + 0.5M_2 = 6 \cdot 9 = 54,$$

$$0.5M_1 + 2M_2 + 0.5M_3 = 6 \cdot (-12) = -72.$$

Da ein natürlicher Spline vorliegen soll, gilt

$$M_0 = M_3 = 0.$$

Damit ergeben sich  $M_1$  und  $M_2$  aus den Gleichungen zu

$$M_1 = 38.4 \quad \text{und} \quad M_2 = -45.6.$$

Aus Satz 2 folgt dann

$$s(x) = \begin{cases} 38.4 \frac{(x+1)^3}{3} + C_1(x+0.5) + D_1 & \text{für } x \in [-1, 0] \\ 38.4 \frac{(1-x)^3}{6} + 45.6 \frac{x^3}{6} + C_2(x-0.5) + D_2 & \text{für } x \in [0, 1] \\ -45.6 \frac{(2-x)^3}{6} + C_3(x-1.5) + D_3 & \text{für } x \in [1, 2] \end{cases}$$

Aus Satz 2 ergibt sich damit

$$C_1 = -7 - \frac{1}{6} \cdot 38.4 = -13.4, \quad D_1 = 1.5 - \frac{1}{12} \cdot 38.4 = -1.7,$$

$$C_2 = 11 - \frac{1}{6} \cdot (-82) = 25, \quad D_2 = 3.5 - \frac{1}{12} \cdot (-7.2) = 4.1,$$

$$C_3 = -13 - \frac{1}{6} \cdot 45.6 = 20.6, \quad D_3 = 2.5 - \frac{1}{12} \cdot (-45.6) = 6.3.$$

Oben eingesetzt ergibt sich damit

$$s(x) = \begin{cases} 6.4x^3 + 19.2x^2 + 5.8x - 2 & \text{für } x \in [-1, 0] \\ -14x^3 + 19.2x^2 + 5.8x - 2 & \text{für } x \in [0, 1] \\ 7.6x^3 - 45.6x^2 + 70.6x - 23.6 & \text{für } x \in [1, 2] \end{cases}$$

**Aufgabe 8.** Fertigen Sie unter MAPLE einen Graphen von  $s$  an.

*Lösung.*

```
> f := x -> piecewise(x<=0, 6.4*x^3+19.2*x^2+5.8*x-2, 0<x,  
> -14*x^3+19.2*x^2+5.8*x-2, 1<x, 7.6*x^3-45.6*x^2+70.6*x-23.6):  
> plot(f(x), x=-1..2, y=-12..12);
```

**Bemerkung.** Es ist erkennbar, dass an den Rändern der Messung das Polynom deutlich stärker ausbricht als bei Spline.

Noch deutlicher wird es im folgenden Beispiel.

**Beispiel 7.** Es sei  $f(x) = \frac{1}{1+x^2}$ , und  $x = -3, -2, -1, 0, 1, 2, 3$ .

Ein natürlicher Spline soll die Funktion  $f$  mit  $f(x) = \frac{1}{1+x^2}$  an den Stellen  $x = -3, -2, -1, 0, 1, 2, 3$  interpolieren. Die Spline-Funktion  $p$  ist achsensymmetrisch zur  $y$ -Achse, da es auch für  $f$  gilt. Damit gilt also

$$p(x) = ax^6 + bx^4 + cx^2 + d.$$

Dies führt zu einem LGS mit der Lösung

$$p(x) = -\frac{1}{100}x^6 + \frac{3}{10}x^4 - \frac{16}{25}x^2 + 1.$$

Die Graphen von  $f$  und  $p$  sehen folgendermaßen aus:



**Aufgabe 9.** Berechnen Sie die natürlichen Splines zu Aufgabe 1.

*Lösung.* a)

$$s(x) = \begin{cases} -2x^3 - 6x^2 + 5x - 2 & \text{für } x \in [-1, 0] \\ 6x^3 - 6x^2 + 5x - 2 & \text{für } x \in [0, 1] \\ -4x^3 + 24x^2 - 25x + 8 & \text{für } x \in [1, 2] \end{cases}$$

b)

$$s(x) = \begin{cases} 3x^3 + 9x^2 + 2x - 3 & \text{für } x \in [-1, 0] \\ -4.5x^3 + 9x^2 + 2x - 3 & \text{für } x \in [0, 2] \\ 6x^3 - 54x^2 + 128x - 87 & \text{für } x \in [2, 3] \end{cases}$$

## 4 Einführung und erste Anwendungen von Maple im Mathematikunterricht

Hier folgt eine Einführung in MAPLE. Diese Einführung wurde in wesentlichen Teilen bereits in Kursen angewendet – insbesondere in einem Kurs zur Förderung begabter SchülerInnen der Klassen 9 und 10 in Gelsenkirchen und im Schülerlabot der Universität Bochum. Aus diesem Grund werden hier die LeserInnen auch mit „Du“ angesprochen. Es handelt sich um fertige Arbeitsblätter mit Lösungen. Die zugehörigen MPLE-Dateien werden gern mitgeliefert.

### 4.1 Einführung

**Aufgabe 1.** Gib unter Maple die folgenden Zeilen ein:

> 1 + 1:

> 1 + 1;

Wo liegt der Unterschied?

Gib ferner die Terme

> 8/4/2;

und

> 8/(4/2);

ein. Was wird hier genau berechnet?

**Aufgabe 2.** Berechne unter Maple

a)  $8/4/2$    b)  $8/(4/2)$

Notiere die Terme in richtiger mathematischer Form.

**Aufgabe 3.** Berechne unter Maple

a)  $2 \wedge (2 \wedge 3)$    b)  $(2 \wedge 2) \wedge 3$

Notiere die Terme in richtige mathematischer Schreibweise.

**Aufgabe 4.** Gib die folgenden Terme ein:

$a+b$ ;    $a*c$ ;    $b/c$ ;    $b:c$ ;   sowie

`> a^b; a:=%`

Beobachte die Ausgaben. Was fällt Dir auf?

**Aufgabe 5.**

a) Ein Graph einer Funktion  $f$  mit  $f(x) = x^2$  lässt sich unter MAPLE mit

`> plot(x^2, x=-2..2, y=-2..2);`

zeichnen. Verändere die Parameter für  $x$  und  $y$  sowie die Funktionsvorschriften zu  $x^3, 1/x, \dots$

b) Verändere den MAPLE-Befehl zu

`> plot(x^2, x=-2..2, y=-2..2, style=point);`

Was wird hier mit dem Graphen gemacht? Welche Werte werden nur noch für  $x$  eingesetzt?

**Aufgabe 6.**

a) Gib den folgenden Term ein:

`> evalf(sqrt(2),10);`

und deute das Ergebnis. Verändere auch die hintere Zahl. Was wird hier angegeben?

b) Berechne mithilfe von Maple  $\sqrt{5}, \sqrt{5134}, \sqrt{16}$ .

### Aufgabe 7.

a) Gib unter MAPLE die folgende Zeile ein:

```
> Sum(i, i=1..10) = sum(i, i=1..10);
```

b) Berechne unter MAPLE die Summen

$$1) \sum_{i=3}^{20} i, \quad 2) \sum_{i=1}^{15} i^2, \quad 3) \sum_{i=1}^{15} i^3, \quad \sum_{i=1}^{15} (i+1)^3.$$

**Aufgabe 8.** Berechne die Werte der folgenden Funktionen an unterschiedlichen Stellen:

a)  $f(x) = x^3 - 10$ ,   b)  $f(x) = \sqrt{5x}$ ,   c)  $f(x) = -x^4 + 1$ .

**Aufgabe 9.** Fertige eine Zeichnung der Funktionen aus Aufgabe 8 in unterschiedlichen Intervallen an. Achte hierbei darauf, dass auf  $x$ - und  $y$ -Achse dieselben Intervalle für die Abbildung ausgewählt werden.

**Aufgabe 10.** Öffne die Datei `einfafg10.mws` und führe sie durch. Was lässt sich hier beobachten? Ersetze die Funktionen aus dieser Datei durch Funktionen aus Aufgabe 8.

## Grundlagen des Programmierens unter Maple

**Aufgabe 11.** Gib die folgenden Schleifen ein und vergleiche ihre Wirkung:

a)

```
> for a from 0 to 5 do
>   a
> od;
```

b)

```
> for a from 0 to 10 by 2 do
>   a
> od;
```

**Aufgabe 12.** Schleifen können auch mit Abbildungen durchgeführt werden. Gib die beiden Programme ein und untersuche ihre Arbeitsweise.

```
> f := x -> x^2+1;
> for x from 0 to 4 do
>   f(x)
> od;
```

**Aufgabe 13.** Gib die folgende Schleife ein.

```
> a := 0;
> while a <= 5 do
>   a := a+1
> od;
```

**Aufgabe 14.** `for` und `while` werden oft kombiniert:

```
> y := 1;
> for a from 1 to 220 while y < 100 do
>   y := a^2+1;
> od;
```

**Aufgabe 15.** Gib das folgende Programm ein und mach Dich seiner Funktionsweise klar, indem Du `a` und `b` veränderst.

```
> a := 2; b := 3;
> if a < b then
>   print('Hallo!')
> fi;
```

**Aufgabe 16.** Gib dieses Programm ein und probiere es mit unterschiedlichen Werten für `a` aus. Erkläre die Wirkung des Programms.

```
> a := 1;
> if a > 0 then
>   print('positiv')
> elif a = 0 then
>   print('null')
> else
>   print('negativ')
> fi;
```

**Aufgabe 17.** `for` und `if` können gut kombiniert werden.

```
> for a from -2 to 2 do
>   if a > 0 then
>     print(a^2)
>   else
>     print('Hallo')
>   fi;
> od;
```

## 4.2 (Pseudo)-Zufallszahlen

### 4.2.1 Iteration

**Aufgabe 1.** Öffne die MAPLE-Datei `iterat1.mws` und lass den Algorithmus für die Funktionen  $f$  mit

$$f_1(x) = 2x, \quad f_2(x) = x^2 + 1, \quad f_3(x) = x^2 + \frac{3}{8}, \quad f_4(x) = x^2 - 1,$$

und den Startwerten

$$x = 0 \quad \text{und} \quad x = 1$$

für etwa

$$t = 15$$

durchlaufen. Hierbei sollten die Graphen der Funktionen in den Intervallen

$$x = [0; 20], y = [0; 20] \quad \text{bei } f_1,$$

$$x = [0; 20], y = [0; 20] \quad \text{bei } f_2,$$

$$x = [0; 5], y = [0; 5] \quad \text{bei } f_3,$$

$$x = [0; 5], y = [0; 5] \quad \text{bei } f_4,$$

betrachtet werden.

**Aufgabe 2.** Öffne die MAPLE-Datei `iterat2.mws` und lass den Algorithmus für dieselben Funktionen, Startpunkte und Anzahl der Schritte wie in Aufgabe 1 und den Startwerten

$$s = 0 \quad \text{und} \quad s = 1$$

für etwa

$$t = 15$$

durchlaufen. Erkennst Du die Orbits? Was sagen sie aus?

## 4.2.2 Lineare Kongruenz-Algorithmus

**Aufgabe 1.** Öffne die Datei `linkong1.mws` und führe den Algorithmus aus. Verändere dann die Parameter (es sollte  $n \leq 2000$  sein).

**Aufgabe 2.**

a) Öffne die Datei `linkong2.mws` und versuche, den Algorithmus unter a) zu verstehen. Was wird in dem Graphen genau abgebildet?

**Aufgabe 2.**

b) Lasse den Teil b) aus `linkong2.mws` durchlaufen; er variiert die Parameter  $a$  und  $c$ . Was fällt bei der Betrachtung der Graphen auf? Verändere die Startwerte und lass die Programme nochmal durchlaufen.

## 4.3 Monte-Carlo-Simulation

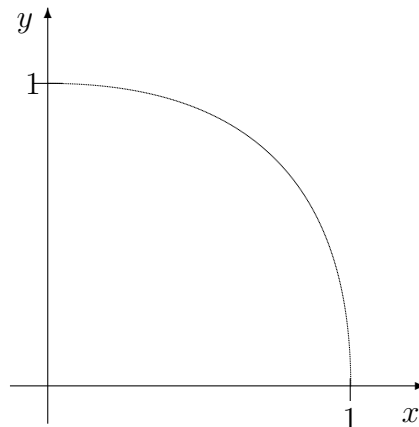
Unter **Monte-Carlo-Methoden** werden Verfahren zusammengefasst, bei denen näherungsweise Berechnungen durchgeführt werden, indem aus einer großen Grundmenge mithilfe von Zufallszahlen Stichproben genommen werden.

Unter **Simulationen** versteht man die Analyse und Bewertung des Verhaltens von Systemen mithilfe eines Rechners.

Unter einer **Monte-Carlo-Simulation** versteht man eine Simulation, bei der die Variablen mithilfe von *Pseudo-Zufallszahlen* berechnet werden.

### 4.3.1 Bestimmung von $\pi$

Wir betrachten das folgende Bild:



der Flächeninhalt des Quadrats ist

$$F_Q = 1.$$

Der Flächeninhalt eines Kreises vom Radius 1 ist gegeben durch

$$F_K = \pi,$$

wobei  $\pi$  eine Zahl ist, die zu bestimmen ist. Sie kann nicht durch das Lösen von Gleichungen mit rationalen Koeffizienten bestimmt werden, wie es bisher bei Euch der Fall war.

Wir können es hier näherungsweise bestimmen:

Für die Fläche des Viertelkreises vom Radius 1 gilt

$$F_{K/4} = \frac{\pi}{4}.$$

Das Verhältnis der Flächen zueinander ergibt sich zu

$$\frac{F_{K/4}}{F_Q} = \frac{\pi/4}{1} = \frac{\pi}{4}.$$

Hieraus folgt

$$\pi = 4 \cdot \frac{F_{K/4}}{F_Q}. \quad (1)$$

Kennt man also die Flächeninhalte des Viertelkreises und des Quadrates, so kann  $\pi$  berechnet werden.

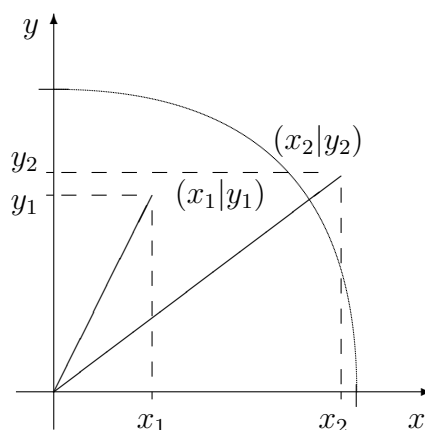
$F_Q = 1$  ist bekannt, aber wie kommen wir zu  $F_{K/4}$ ?

Wir bestimmen das Verhältnis  $\frac{F_{K/4}}{F_Q}$  näherungsweise, indem wir rein zufällig Punkte aus dem Quadrat auswählen und prüfen, wie viele davon im Viertelkreis liegen.

#### Aufgabe 1.

- Überlege Dir ein Verfahren unter MAPLE, mit dem sich Punkte aus einem Quadrat mit der Kantenlänge 1 auswählen lassen.

b) Zeichne diese Punkte in den Graphen eines Quadrates ein.



**Bemerkung.** Ein Punkt  $(x|y)$  liegt innerhalb des Einheitskreises genau dann, wenn

$$x^2 + y^2 \leq 1$$

gilt.

**Aufgabe 2.** Setze das Verfahren zur Bestimmung der Anzahl der Punkte, die im Viertelkreis liegen, in MAPLE um.

Das Verhältnis

$$\frac{\# \text{ Punkte im Viertelkreis}}{\# \text{ Punkte insgesamt}}$$

ist für eine ausreichend große Zahl an ausgewählten Punkten näherungsweise gleich dem Verhältnis der Flächen  $F_{K/4}/F_Q$ , d.h.

$$\frac{F_{K/4}}{F_Q} \approx \frac{\# \text{ Punkte im Viertelkreis}}{\# \text{ Punkte insgesamt}}.$$

**Bemerkung.** Für die Zahl  $\pi$  gilt bei einer ausreichend großen Stichprobe

$$\pi \approx 4 \cdot \frac{\# \text{ Punkte im Viertelkreis}}{\# \text{ Punkte insgesamt}}.$$

**Algorithmus.**

*Eingabe:*  $n \in \mathbb{N}$  (Anzahl der Schritte)

1. Setze  $ct = 0$  (Anzahl der Treffer).
2.  $n$ -mal werden die folgenden Schritte wiederholt:
  - (a) Wähle zufällig einen Punkt  $(x|y)$  mit  $0 \leq x, y \leq 1$ .
  - (b) Teste, ob  $x^2 + y^2 \leq 1$  gilt (d.h. ob der Punkt im Viertelkreis liegt).
  - (c) Falls ja, erhöhe  $ct$  um 1.
3.  $4 \cdot \frac{ct}{n}$  ist ein Näherungswert für  $\pi$ .

*Ausgabe:*

$$4 \cdot \frac{ct}{n}.$$

**Aufgabe 3.**

- a) Programmiere den Monte-Carlo-Algorithmus für  $\pi$  unter MAPLE.
- b) Untersuche den Algorithmus anhand unterschiedlicher Schrittzahlen  $n$  (bis ca. 3000).
- c) Stelle die zufällig bestimmten Punkte und den Viertelkreis unter MAPLE graphisch dar.

**4.3.2 Monte-Carlo-Integration und Flächeninhalt**

Wir beginnen mit einem Beispiel:

$$f: \mathbb{R} \rightarrow \mathbb{R}, \quad x \mapsto x^2.$$

Nun sehen wir uns einen Graphen zur Funktion im Intervall  $[0; 2]$  an:

Es soll die Fläche zwischen dem Graphen und der  $x$ -Achse berechnet werden.

**Algorithmus 1.**

*Eingabe:* Funktion  $f$ , die integriert werden soll.

1. Wähle  $k$  Zufallszahlen  $x_1, \dots, x_k \in [0; 2]$  ( $k = 5000$  reicht halbwegs).
2. Berechne  $f(x_i)$  für  $i = 1, \dots, k$ .
3. Berechne den Mittelwert der Funktionswerte  $f(x_i)$  für  $i = 1, \dots, k$ .
4. Multipliziere den Mittelwert aus Schritt 3 mit der Breite 2 des Intervalls.

*Ausgabe:* Ergebnis der Berechnung in den Schritten 1 bis 4.

**Aufgabe 1.**



- a) Schreib unter MAPLE ein Programm, das den oben stehenden Algorithmus enthält.
- b) Berechne die Fläche für unterschiedliche  $k \leq 100000$ .  
Vergleiche die Ergebnisse mit dem exakten Wert 4.

**Aufgabe 2.** Schreibe das Programm aus Aufgabe 1 so um, dass es für  $[0; r]$  mit  $r > 0$  verwendet werden kann.

**Aufgabe 3.** Wende Dein Programm aus Aufgabe 2 für  $r = 2$  auf die Funktionen  $f$  mit

- a)  $f(x) = -x^2$  und  
b)  $f(x) = x - 1$

an, und betrachte Graphen im Intervall  $[0; 2]$  dazu. Was fällt Dir hierbei auf?

### Algorithmus 2.

*Eingabe:*

- Funktion  $f$ , für die die Fläche zwischen  $x$ -Achse und dem Funktionsgraphen berechnet werden soll.
- Zahl  $r$ , so dass die Fläche im Intervall  $[0; r]$  berechnet werden soll.
  1. Wähle  $k$  Zufallszahlen  $x_1, \dots, x_k \in [0; r]$  ( $k = 5000$  reicht halbwegs).
  2. Berechne  $|f(x_i)|$  für  $i = 1, \dots, k$ .
  3. Berechne den Mittelwert der Funktionswerte  $|f(x_i)|$  für  $i = 1, \dots, k$ .
  4. Multipliziere den Mittelwert aus Schritt 3 mit der Breite  $r$  des Intervalls.

*Ausgabe:* Ergebnis der Berechnung in den Schritten 1 bis 4.

**Aufgabe 4.** Schreibe das Programm aus Aufgabe 2 so um, dass der Algorithmus 2 umgesetzt wird.

### 4.3.3 Monte-Carlo-Primzahltest

## Modulo

**Aufgabe 1.** Gib die folgende Zeile ein:

$> 10 \bmod 3;$

Ist das Ergebnis verwunderlich?

**Aufgabe 2.**

- a) Definiere unter MAPLE eine **for**-Schleife, mit der für  $a = 1, \dots, 15$  die Terme  $a \bmod 3$  ausgegeben werden. Kannst Du die Wirkung von `mod` erklären?
- b) Schreibe die Schleife aus Teil a) so um, dass  $a \bmod 4, a \bmod 7$  für  $a = 1, \dots, 20$  ausgegeben werden. Unterstützt dies Deine Vermutung?

**Definition.** Die Zahl

$$m := a \bmod n$$

ist die **Restklasse** von  $a$  bzgl. der **Basis**  $n$ , d.h. es existiert eine Zahl  $z \in \mathbb{Z}$  mit

$$a = z \cdot n + m.$$

**Bemerkung.**

- a) Es gilt  $a \bmod n = 0$  genau dann, wenn  $m = 0$  gilt, d.h.  $n$  ist ein Teiler von  $a$ , in Zeichen  $n \mid a$ .
- b) Ist  $n$  ein Teiler von  $a - b$ , so gilt

$$a \bmod n = b \bmod n.$$

**Aufgabe 3.** Berechne unter MAPLE

- a)  $a \bmod 25$  für  $a = 1, \dots, 30$ ,  
 b)  $a \bmod 19$  für  $a = 1, \dots, 20$ .

**Aufgabe 4.** Berechne unter MAPLE die folgenden Schleifen:

- a)  $(k \cdot 2) \bmod 12$   
 b)  $(k \cdot 2) \bmod 17$   
 c)  $(k \cdot 5) \bmod 19$   
 d)  $(k \cdot 5) \bmod 20$

für  $k = 1, \dots, 6$ . Was fällt hierbei auf? Kannst Du eine Vermutung äußern?

**Bemerkung.** Eine Zahl  $p \in \mathbb{N}$  ist genau dann prim, wenn für alle  $a, b \in \mathbb{Z}$  mit  $a \bmod p \neq 0$  und  $b \bmod p \neq 0$  gilt:

$$(a \cdot b) \bmod p \neq 0.$$

## Primzahltest

**Aufgabe 1.**

- a) Wähle eine beliebige Dir bekannte Primzahl  $n$  und lasse unter MAPLE berechnen:

$$a^{n-1} \bmod n \quad \text{für alle } a = 1, 2, \dots, n-1.$$

Was fällt Dir hierbei auf?

- b) Führe dasselbe Verfahren für eine nicht prime Zahl durch. Vergleiche das Ergebnis mit dem Ergebnis aus a).

**Satz.** (kleiner Fermat'scher Satz)

$$n \in \mathbb{N} \text{ prim} \Rightarrow a^{n-1} = 1 \bmod n \quad \text{für alle } a \in \mathbb{Z}_n \setminus \{0\}$$

**Bemerkung.** Die Umkehrung dieses Satzes gilt nicht, wie obige Aufgabe 1 b) zeigt.

Wählt man einige (z.B.  $k = 50$ ) Zahlen  $a_k \in \mathbb{Z}_n \setminus \{0\}$ , so kann man, wenn  $a_i^{n-1} \neq 1 \bmod n$  für irgendein  $i$  gilt, sicher sein, dass  $n$  nicht prim ist.

Wenn  $a_i^{n-1} = 1 \bmod n$  für alle  $i$  gilt, dann ist  $n$  „wahrscheinlich“ eine Primzahl.

**Algorithmus.**

*Eingabe:*  $n \in \mathbb{N}$  (die zu prüfende Zahl)

1. Wähle  $k$  Zufallszahlen  $a_1, \dots, a_k \in \mathbb{Z}_n \setminus \{0\}$  ( $k = 50$  reicht meistens).
2. Wiederhole die folgenden Anwendungen für  $i = 1, \dots, k$ :
  - (a) Berechne  $a_i^{n-1} \bmod n$ .
  - (b) Falls  $a_i^{n-1} \bmod n \neq 1$  gilt, sage: „ $n$  ist nicht prim“ und breche den Algorithmusschritt 2 ab.
3. Falls Schritt 2  $k$ -mal durchgeführt wurde (d.h.  $i = k$ ), so gib aus: „ $n$  ist wahrscheinlich prim“.

**Aufgabe 2.**

- a) Schreib unter MAPLE ein Programm, das den oben stehenden Algorithmus enthält.
- b) Teste das Programm aus a) für Zahlen, von denen Dir bekannt ist, dass sie prim sind.
- c) Prüfe bei sehr großen Zahlen aus, ob sie prim sind.

**Aufgabe 3.** Führe mit den folgenden großen Zahlen den Test durch, ob es sich hierbei um Primzahlen handelt.

- a)  $n = 318949$ ,
- b)  $n = 318949$ ,

- c)  $n = 318950$ ,
- d)  $n = 600009$ ,
- e)  $n = 600010$ ,
- f)  $n = 600011$ ,
- g)  $n = 95789$ ,
- h)  $n = 95790$ ,
- i)  $n = 2^{24036583} - 1$ .