

Kapitel 5

Reelle Zahlen

Wir alle sind seit Schulzeiten an den Umgang mit reellen Zahlen gewöhnt. Sie gehören zum Alltag. Das war nicht immer so. Nach den Vorstellungen der im fünften vorchristlichen Jahrhundert ganz Unteritalien beherrschenden Pythagoreer offenbarte sich die Harmonie der erkennbaren Welt in alles umfassenden Verhältnissen ganzer Zahlen. Als Hippasus von Metapont den irrationalen Goldenen Schnitt $\phi = (1 + \sqrt{5})/2$ im Pentagramm entdeckte, also ausgerechnet im Geheimzeichen der Pythagoreer, kam es zu einem handfesten politischen Skandal, der sogar mit dem Aufstand von 445 in Zusammenhang gebracht wird [11, Kapitel 2, §1.1]. Noch für B. Bolzano (1781 - 1848), Professor für Philosophie und Religion in Prag und ein Wegbereiter der modernen Analysis, bedeutete die Begründung der reellen Zahlen letztlich eine Annäherung an Gott [8, S. 136] [3, §25]. Reelle Zahlen sind von unendlicher Natur. Und seit den präzisen Begriffsbildungen durch Cantor, Dedekind und Weierstraß im 19. Jahrhundert wissen wir auch, was damit gemeint ist. Die Weltanschauungskrisen früherer Jahrtausend sind vorbei, aber der „Nebel der Unendlichkeit“ bleibt: Es gibt keine Zifferndarstellung reeller Zahlen. Die Folge ist kein politischer Skandal sondern der Rest dieses Buches.

5.1 Konstruktion von \mathbb{R} durch Vervollständigung von \mathbb{Q} .

Wir skizzieren im folgenden beispielhaft den Zugang von G. Cantor (1845 - 1918) und C. Meray (1835 - 1911). Für eine ausführliche Darstellung und umfangreiche historische Bemerkungen verweisen wir wieder auf Ebbinghaus et al. [11, Kap. 2]. In Analogie zur Konstruktion ganzer und rationaler Zahlen in den Abschnitten 1.2 und 3.1 versuchen wir es wieder mit der Lösung passender Aufgaben. Nach Summen und Produkten wären nun Potenzen an der Reihe, also beispielsweise Aufgaben der Form

$$x^n - m = 0 \tag{5.1}$$

mit $n, m \in \mathbb{N}$ und $n, m \geq 2$. Bekanntlich hat (5.1) schon für $n = m = 2$ keine Lösung $x \in \mathbb{Q}$ (vgl. z.B. Courant und Robbins [7, Kap. 2, § 2]). Analog zu den Abschnitten 1.2 und 3.1 können wir nun die Lösung von (5.1) mit Paaren (n, m) identifizieren, Äquivalenzklassen bilden und so weiter. So erhalten wir immerhin die Wurzeln $\sqrt[n]{m}$. Aber das reicht nicht. Selbst wenn wir anstelle von der Nullstellen von $p(x) = x^n - m$ die Nullstellen beliebiger Polynome

$$p(x) = \sum_{i=1}^n a_i x^i = 0$$

mit ganzzahligen Koeffizienten a_i ins Spiel bringen, kommen auf diese Weise nur die sogenannten *algebraischen Zahlen* zustande, und auf der Zahlengeraden bleiben immer noch Löcher übrig.

Eines dieser Löcher ist π , die Fläche des Kreises mit Radius 1. An der Quadratur des Kreises, also an der Berechnung von π , haben sich seit den Babyloniern und Ägyptern viele Generationen von Mathematikern versucht. Eine kleine Auswahl der Erfolge und Misserfolge ist in Tabelle 5.1 dargestellt.

Wann	Wer	Approximation	Korrekte Stellen
ca. 2000 v. Chr.	Babylonier	$3\frac{1}{8}$	2
	Ägypter	$\frac{256}{81}$	2
ca. 1100 v. Chr.	Chinesen	3	1
ca. 550 v. Chr.	Altes Testament	3	1
3. Jhd v. Chr.	Archimedes	$3\frac{10}{71} < \pi < 3\frac{1}{7}$	3–4
2. Jhd.	Ptolemäus	$\frac{377}{120}$	4
3. Jhd.	Wang Fau	$\frac{142}{45}$	2
263	Liu Hui	$\frac{157}{50} = 3,14$	3
ca. 450	Tsu Ch'ung Chi	$\frac{355}{113}$	7
ca. 530	Aray-Bhata	$\frac{62832}{20000}$	4
ca. 650	Brahmagupta	$\sqrt{10}$	2

Tabelle 5.1 Approximation von π im Laufe der Zeiten

Erst seit der gefeierten Arbeit von Lindemann [37]¹ aus dem Jahre 1882 ist klar, daß alle Versuche, π durch rationale Zahlen und Wurzeln auszudrücken zum Scheitern verurteilt sind: π ist keine algebraische Zahl sondern *transzendent*.

Eine Möglichkeit, π dennoch auf die Spur zu kommen, besteht darin, die Fläche des Einheitskreises zu approximieren. Dazu überziehen wir ein die gesamte Ebene mit einem Gitternetz aus Quadraten mit Kantenlänge $h_n = 1/10^n$, $n \in \mathbb{N}$ beliebig aber fest gewählt, berechnen jeweils die Summe der Flächen von all jenen Gitterzellen, die ganz im Kreis liegen und erhalten als Ergebnis x_n (vgl. Abb 5.1). Das können

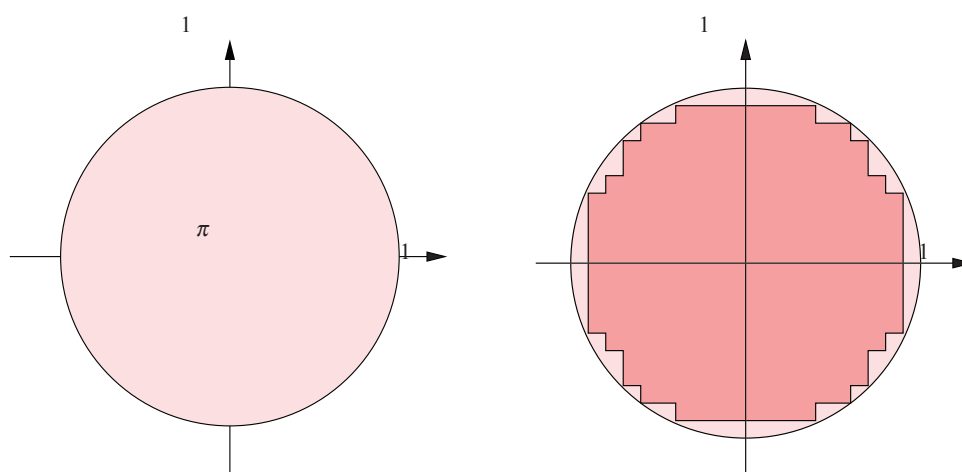


Abb. 5.1 Auf dem Weg zu π

wir für jedes $n \in \mathbb{N}$ machen und erhalten so die Folge (x_n) . Eine „kleine Nebenrechnung“ ergibt

$$x_0 = 0, x_1 = 2,68, x_2 = 3,1, x_3 = 3,137524, x_4 = 3,14119020, x_5 = 3,1415525416.$$

Die Differenz $|x_m - x_n|$ ist gerade der durch Verkleinerung der Kantenlänge bewirkte Zuwachs der einbeschriebenen Fläche. Damit wird $|x_m - x_n|$ beliebig klein, falls n und m genügend groß sind. Folgen (x_n) mit dieser Eigenschaft nennt man *Cauchy-Folgen* (siehe Abschnitt A.2 im Anhang). Offenbar ist $(x_n) \subset \mathbb{Q}$ und mit wachsendem n kommt x_n dem Flächeninhalt namens π beliebig nahe. π selbst ist aber keine ratio-

¹ C.L.F. von Lindemann (1852 - 1939) war der Lehrer des Königsbergers David Hilbert (1862 - 1943), eines der bedeutendsten Mathematiker aller Zeiten.

nale Zahl. Die Cauchy-Folge $(x_n) \subset \mathbb{Q}$ hat also keinen Grenzwert in \mathbb{Q} . Ziel ist es nun, \mathbb{Q} so zu erweitern, daß alle Cauchy-Folgen einen Grenzwert haben. Diese Erweiterung nennt man *Vervollständigung* von \mathbb{Q} .

Wir beschränken uns auf π . Am einfachsten wäre es, π direkt mit der Folge (x_n) zu identifizieren. Allerdings gibt es noch beliebig viele andere Folgen rationaler Zahlen, welche π auch beliebig genau approximieren, beispielsweise die Folge

$$d_0 = 3, d_1 = 3, 1, d_2 = 3, 14, d_3 = 3, 141, d_4 = 3, 1415, d_5 = 3, 14159 \dots,$$

und die Zahl π wäre so nicht eindeutig bestimmt. Ein ähnliches Problem tauchte auch schon bei der Konstruktion von \mathbb{Z} und \mathbb{Q} auf, und wir lösen es in bewährter Weise. Weil sowohl (x_n) als auch (d_n) dem Flächeninhalt π beliebig nahe kommen, wird der Abstand $|x_n - d_n|$ für wachsende n beliebig klein. Wenn zwei Cauchy-Folgen $(x_n), (d_n)$ diese Eigenschaft haben, schreiben wir $(x_n) \cong (d_n)$. Dadurch wird eine Äquivalenzrelation definiert. Die Menge \mathbb{R} der reellen Zahlen besteht aus allen Äquivalenzklassen $x = [(x_n)]$ von Cauchy-Folgen $(x_n) \subset \mathbb{Q}$ bezüglich dieser Relation. Eine reelle Zahl ist also durch *unendlich viele* Folgenglieder charakterisiert, denen die unendlich vielen Stellen hinter dem Komma entsprechen. Im Gegensatz dazu kommt eine rationale Zahl mit endlich vielen Stellen Zähler und Nenner aus. Etwas poetisch ausgedrückt bedeutet das:

Die reellen Zahlen sind von unendlicher Natur.

Addition, Multiplikation, Anordnung und die entsprechenden Rechenregeln vererben sich von den Folgengliedern auf Cauchy-Folgen und damit auf \mathbb{R} . Als Vertreter jeder Äquivalenzklasse $x \geq 0$ wählen wir die Folge endlicher Dezimalbrüche

$$d_0 = z_n \cdots z_0, d_1 = z_n \cdots z_0, z_{-1}, d_2 = z_n \cdots z_0, z_{-1} z_{-2}, d_3 = z_n \cdots z_0, z_{-1} z_{-2} z_{-3}, \dots$$

mit $z_i \in \mathcal{Z} = \{0, 1, \dots, 9\}$ und schreiben x als *unendlichen Dezimalbruch*

$$x = z_n \cdots z_0, z_{-1} z_{-2} z_{-3} z_{-4} \dots$$

In diesem Sinne ist

$$\pi = 3, 1415 \dots \in \mathbb{R}.$$

Dabei wird die unendliche Natur von π schlicht durch drei Punkte repräsentiert.

Die *Darstellung* der einzelnen Folgenglieder x_n oder d_n spielte bei unseren Überlegungen keine Rolle. Wir hätten anstelle von Dezimalbrüchen genauso gut auch eine andere q -adische Darstellung wählen können. So erhält man beispielsweise im Dualsystem

$$\pi = 11,0010\,0100\,0011\,1111\dots_2. \quad (5.2)$$

5.2 Keine Zifferndarstellung reeller Zahlen

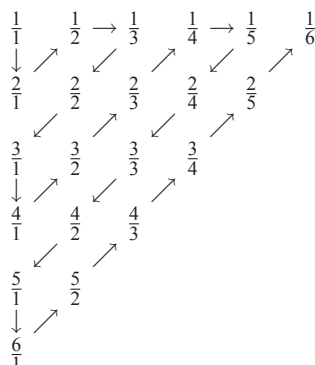
Die Voraussetzung für numerisches Rechnen mit reellen Zahlen ist deren Darstellung durch ein Ziffernsystem (vgl. die Abschnitte 1.1.1 und 1.1.2). Betrachtet man reelle Zahlen als unendliche Dezimalbrüche, so kommt man zwar mit endlich vielen Ziffern aus, die entsprechenden Ziffernketten sind aber unendlich lang. Die unendlichen Dezimalbrüche sind also noch kein Ziffernsystem im Sinne von Absatz 1.1.2. Wir wollen untersuchen, ob es solch eine Zifferndarstellung der reellen Zahlen überhaupt geben kann.

Nach Definition 1.2 heisst eine Menge M *abzählbar*, wenn sich die natürlichen Zahlen bijektiv auf M abbilden lassen. Anschaulich bedeutet das: Eine abzählbare Menge M lässt sich durchnummerieren, also in der Form

$$M = \{m_0, m_1, m_2, \dots\}$$

aufschreiben. Nach Satz 1.1 ist die Menge $\mathcal{D}(\mathcal{Z})$ aller Ziffernketten aus einer endlichen Ziffernmeng \mathcal{Z} abzählbar. Da unendliche Teilmengen abzählbarer Mengen wieder abzählbar sind, ist eine Menge M von Zahlen genau dann abzählbar, wenn es ein Ziffernsystem zur Darstellung von M gibt (vergleiche Definition 1.3). Aus Satz 1.5 und Satz 3.1 folgt also die Abzählbarkeit von \mathbb{Z} und \mathbb{Q} .

Man kann übrigens \mathbb{Q} auch direkt durchnummerieren. Dazu listet man zunächst alle positiven rationalen Zahlen entsprechend dem folgenden Dreiecksschema auf.



Mehrfach auftauchende Zahlen, wie etwa $1 = \frac{2}{2} = \frac{3}{3}, \dots$, lässt man einfach weg. Dann schreibt man noch 0 an den Anfang und $-\frac{m}{n}$ jeweils direkt hinter $\frac{m}{n}$. So erhält man schließlich die Aufzählung

$$\mathbb{Q} = \left\{ 0, 1, -1, 2, -2, \frac{1}{2}, -\frac{1}{2}, \frac{1}{3}, -\frac{1}{3}, 3, -3, 4, -4, \frac{3}{2}, -\frac{3}{2}, \dots \right\}.$$

Diese Methode geht auf G. Cantor (1845 - 1918) zurück. Eine andere Aufzählung von Calkin und Wilf ist in [2, Kap. 16] beschrieben. Und nun der Hammer:

Satz 5.1 (Überabzählbarkeit von \mathbb{R}). *Die Menge \mathbb{R} der reellen Zahlen ist nicht abzählbar.*

Beweis. Wäre \mathbb{R} abzählbar, so hätte auch jede unendliche Teilmenge diese Eigenschaft. Es reicht daher, eine unendliche Teilmenge $M \subset \mathbb{R}$ zu finden, die nicht abzählbar ist. Wir wählen $M = \{x \in \mathbb{R} \mid 0 < x \leq 1\}$. Im Widerspruch zur Behauptung sei M abzählbar, also $M = \{x_0, x_1, x_2, \dots\}$. Wir schreiben nun jedes Element $x_n \in M$ als unendlichen Dezimalbruch, also beispielsweise 0,9 statt 1. Auf diese Weise erhalten wir das unendliche Schema

$$\begin{array}{l} x_0 = 0, z_{00} z_{01} z_{02} \dots \\ x_1 = 0, z_{10} z_{11} z_{12} \dots \\ \vdots \quad \quad \quad \vdots \\ x_n = 0, z_{n0} z_{n1} z_{n2} \dots \\ \vdots \quad \quad \quad \vdots \end{array}$$

Zu jedem $n \in \mathbb{N}$ wählen wir ein $a_n \in \{1, \dots, 8\}$ mit $a_n \neq z_{nn}$ und setzen $a = 0, a_1 a_2 \dots$. Offenbar ist $a \in M$, also $a = x_n$ für ein geeignetes $n \in \mathbb{N}$. Das kann aber nicht sein, denn nach Konstruktion ist $a_n \neq z_{nn}$ und daher unterscheidet sich a von jedem x_n . Widerspruch. \square

Satz 5.1 hat weitreichende Konsequenzen. Eine davon ist:

Satz 5.2. *Es gibt kein Ziffernsystem zur Darstellung von \mathbb{R} .*

Denn die Menge $\mathcal{D}(\mathcal{Z})$ aller Ziffernketten aus einer endlichen Ziffernmengemenge \mathcal{Z} ist abzählbar, im Gegensatz zu \mathbb{R} .

Es ist also nicht möglich, allen reellen Zahlen einen eigenen Namen zu geben. Symbolisch können daher mit reellen Zahlen rechnen², numerisch ausrechnen können wir reelle Zahlen aber nicht, nicht einmal $\pi + 1$.

² Symbolisches Rechnen ist nichts anderes als das altbekannte „Buchstabenrechnen“ aus der Schule. Jeder der endlich vielen in einer speziellen Rechnung auftretenden Zahlen werden Namen π, a, b, \dots oder $c = a + b, \dots$ zugeordnet, die nur für diese spezielle Rechnung gültig sind. Das Ergebnis ist eine Formel $y = f(\pi, a, b, \dots)$, mit deren Hilfe man die Lösung y für spezielle Werte der Eingaben π, a, b, \dots berechnen kann. Es gibt mittlerweile hochentwickelte Symbolik-Programme wie MAPLE oder MATHEMATICA, die auf Grundlage faszinierender Verbindungen zwischen Analysis und Algebra weit mehr als nur Bruchrechnung beherrschen. Symbolisches Rechnen stößt allerdings an prinzipielle Grenzen, wenn es nachweislich keine Formel gibt, mit der man die gesuchte Lösung aus den Eingabewerten berechnen könnte. Das ist leider bei den meisten praktisch relevanten Problemen der Fall, angefangen mit der Berechnung der Nullstellen von Polynomen mindestens fünften Grades über die Berechnung von Integralen bis hin zur Lösung gewöhnlicher oder partieller Differentialgleichungen.

Numerisches Rechnen mit reellen Zahlen ist nicht möglich!

Was tun? Beim numerischen Rechnen mit reellen Zahlen müssen wir uns als endliche Wesen mit Approximationen aus einer geeigneten Menge $\mathbb{G} \subset \mathbb{R}$ von darstellbaren Zahlen begnügen. Dabei treten unvermeidlicherweise Fehler auf. Das liegt an der unendlichen Natur der reellen Zahlen. Auch zukünftige Computer-Generationen können daran nichts ändern. Mit den Auswirkungen werden wir uns im Rest dieses Buches beschäftigen.

5.3 Gleitkommazahlen und Rundungsfehler

5.3.1 Absoluter und relativer Fehler

Bevor wir uns um die Approximation reeller Zahlen kümmern, wollen wir klären wie der dabei auftretende Fehler gemessen werden soll. Sei also \tilde{x} eine Approximation von $x \in \mathbb{R}$. Ein natürliches Maß für den Fehler ist der Abstand von x und \tilde{x} auf der Zahlengeraden. Dieser Abstand heißt

$$\text{absoluter Fehler:} \quad |x - \tilde{x}|.$$

Zum Beispiel approximiert $\tilde{x} = 3$ die Zahl $x = \pi$ bis auf den absoluten Fehler $|\pi - 3| < 2 \cdot 10^{-1}$.

Der absolute Fehler ist nicht die einzige Möglichkeit, die Genauigkeit einer Approximation zu messen. So führt beispielsweise die Approximation a) von $x = 4$ durch $\tilde{x} = 0$ und b) von $x = 12344$ durch $\tilde{x} = 12340$ auf denselben absoluten Fehler $|x - \tilde{x}| = 4$, obwohl im Fall b) die ersten vier Stellen von x und \tilde{x} übereinstimmen und im Fall a) keine einzige. Das Verhältnis von absolutem Fehler $|x - \tilde{x}|$ und approximierter Zahl $|x|$ ist in beiden Fällen sehr unterschiedlich. Dieses Verhältnis heißt

$$\text{relativer Fehler:} \quad \frac{|x - \tilde{x}|}{|x|}, \quad x \neq 0.$$

Im Falle $x = 0$ ist der relative Fehler nicht definiert.

Der relative Fehler ist ein Maß für die Anzahl der gültigen Stellen.

In unserem Beispiel ist

$$\frac{4 - 0}{4} = 10^0, \quad \frac{12344 - 12340}{12344} \approx 3 \cdot 10^{-4}.$$

Bisweilen wird der *relative Fehler in Prozent* als $100 \cdot |x - \tilde{x}|/|x|\%$ angegeben. In Beispiel a) haben wir offenbar einen Fehler von 100 Prozent, während im Fall b) ein Fehler von weniger als 1 Promille vorliegt.

5.3.2 Fest- und Gleitkommazahlen

Wir wollen die reellen Zahlen durch *endliche* q -adische Brüche approximieren. Die Anzahl der verfügbaren Stellen bezeichnen wir mit ℓ .

Zunächst versuchen wir es mit einer *festen* Anzahl von n Vor- und m Nachkommastellen aus. Die resultierenden *Festkommazahlen* haben die Gestalt

$$z_{n-1} z_{n-2} \cdots z_0, z_{-1} \cdots z_{-m} = \sum_{i=-m}^{n-1} z_i q^i, \quad z_i \in \{0, \dots, q-1\}.$$

Als Beispiel betrachten wir das Dezimalsystem, also $q = 10$, und $\ell = 4$. Bei Wahl von $n = 3$, $m = 1$ erhalten wir als Approximation von $x = \pi$ durch Runden auf eine Nachkommastelle

$$\tilde{x} = 003,1, \quad |\pi - \tilde{x}| > 4 \cdot 10^{-2}.$$

Die ersten beiden Stellen sind verschenkt. Hätten wir $n = 1$ und $m = 3$ gewählt, wäre die Approximation durch bessere Ausnutzung der vorhandenen $\ell = 4$ Stellen mit

$$\tilde{x} = 3,142, \quad |\pi - \tilde{x}| < 5 \cdot 10^{-4}$$

um zwei Größenordnungen besser ausgefallen. Bei Wahl von $x = 123,4$ hätte allerdings alles wieder ganz anders ausgesehen.

Im Sinne einer optimalen Ausnutzung der vorgegebenen ℓ Stellen wollen wir die *Anzahl der Vor- und Nachkommastellen variabel* halten. Dazu führen wir nun die *Gleitkommazahlen* ein.

Definition 5.3 (Gleitkommazahlen). Jede in der Form

$$\tilde{x} = (-1)^s a \cdot q^e \tag{5.3}$$

mit Vorzeichenbit $s \in \{0, 1\}$, Exponent $e \in \mathbb{Z}$ und *Mantisse*

$$a = 0, a_1 \cdots a_\ell = \sum_{i=1}^{\ell} a_i q^{-i}, \quad a_i \in \{0, \dots, q-1\}, \quad a_1 \neq 0,$$

oder $a = 0$ darstellbare Zahl \tilde{x} heißt *Gleitkommazahl* mit *Mantissenlänge* $\ell \in \mathbb{N}$, $\ell \geq 1$. Die Menge all dieser Gleitkommazahlen heißt $\mathbb{G}(q, \ell)$. Die Darstellung (5.3) heißt *normalisierte Gleitkommadarstellung*.

Die Menge $\mathbb{G}(10, 4)$ besteht aus allen Dezimalbrüchen, bei denen höchstens die $\ell = 4$ führenden Ziffern von Null verschieden sind. Beispielsweise ist

$$0,000001234, \quad 3,142, \quad 123,4, \quad 12340000000 \in \mathbb{G}(10, 4).$$

Im Extremfall erhält man

$$\mathbb{G}(2, 1) = \{0, \dots, \pm 0,001, \pm 0,01, \pm 0,1, \pm 1, \pm 10, \pm 100, \pm 1000, \dots\}.$$

Die Gleitkommazahlen häufen sich in 0 und dünnen mit wachsendem Betrag aus, denn offenbar gilt

$$|\tilde{x} - \tilde{y}| \geq |\tilde{x}| q^{-\ell} \quad \forall \tilde{x}, \tilde{y} \in \mathbb{G}(q, \ell). \tag{5.4}$$

Aus (5.4) folgt, daß der relative Abstand zweier Gleitkommazahlen \tilde{x} und \tilde{y} mindestens $q^{-\ell}$ beträgt.

5.3.3 Bestapproximation durch Runden

Wir sind an einer möglichst guten Approximation reeller Zahlen interessiert. Deshalb wollen wir nun zu jedem $x \in \mathbb{R}$ ein $\text{rd}(x) \in \mathbb{G}(q, \ell)$ mit minimalem Abstand finden, also eine Lösung der Aufgabe

$$|x - \text{rd}(x)| \leq |x - \tilde{x}| \quad \forall \tilde{x} \in \mathbb{G}(q, \ell). \tag{5.5}$$

angeben. Dazu beschreiben wir das *Runden reeller Zahlen*. Sei $x \in \mathbb{R}$ und $x > 0$. Wir schreiben x in der Form

$$x = a^* q^e$$

und wählen dabei $e \in \mathbb{Z}$ so, daß $q^{-1} \leq a^* < 1$ gilt. In der q -adischen Darstellung

$$a^* = \sum_{i=1}^{\infty} a_i q^{-i} = 0, a_1 a_2 \cdots a_\ell a_{\ell+1} \dots, \quad a_i \in \{0, \dots, q-1\}, \tag{5.6}$$

ist also $a_1 \neq 0$. Um Eindeutigkeit zu erzielen, schließen wir aus, daß von einem gewissen Index an für alle i durchweg $a_i = q - 1$ gilt. Wir wählen also beispielsweise $a^* = 0,124$ anstelle von $0,123\bar{9}$. Runden von a^* auf ℓ Stellen liefert dann

$$a = 0, a_1 a_2 \cdots a_\ell + \begin{cases} 0 & \text{falls } a_{\ell+1} < \frac{1}{2}q \\ q^{-\ell} & \text{falls } a_{\ell+1} \geq \frac{1}{2}q \end{cases} . \quad (5.7)$$

Nun definieren wir

$$\text{rd}(x) = aq^\ell . \quad (5.8)$$

Nach Konstruktion ist a ein höchstens ℓ -stelliger q -adischer Bruch, also ist $\text{rd}(x) \in \mathbb{G}(q, \ell)$. Man überzeugt sich auch leicht, daß $\text{rd}(x)$ eine Lösung von (5.5) ist, übrigens im allgemeinen nicht die einzige. Im Falle $x < 0$ definieren wir $\text{rd}(x) = -\text{rd}(-x)$ und schließlich wird $\text{rd}(0) = 0$ vereinbart.

Die so erklärte Abbildung $\text{rd} : \mathbb{R} \rightarrow \mathbb{G}(q, \ell)$ verallgemeinert das bekannte Runden von Dezimalzahlen. Für $q = 2$ und $\ell = 4$ erhalten wir beispielsweise

$$\text{rd}(101110_2) = 110000_2, \quad \text{rd}(11,010110_2) = 11,01_2 .$$

5.3.4 Rundungsfehler und Maschinengenauigkeit

Nun wollen wir untersuchen, wie groß der Fehler beim Runden werden kann. Insbesondere wollen wir wissen, ob wir eine obere Fehlerschranke finden können, die nicht von $x \in \mathbb{R}$ abhängt. Die Antwort hängt davon ab, welches Fehlermaß wir verwenden. Für den *absoluten Rundungsfehler* $|x - \text{rd}(x)|$ gibt es eine solche Schranke nicht:

Satz 5.4. *Zu jedem $N \in \mathbb{N}$ gibt es ein $x \in \mathbb{R}$, so daß*

$$|x - \text{rd}(x)| \geq q^N .$$

Beweis. Die Behauptung folgt durch Wahl von $x = 0, z_1 \cdots z_\ell z_{\ell+1} \cdot q^{\ell+1+N}$ mit $z_1, z_{\ell+1} \neq 0$ und beliebigem $N \in \mathbb{N}$. \square

Der absolute Rundungsfehler kann beliebig groß werden.

In $\mathbb{G}(10, 4)$ erhält man beispielsweise für $x = 0,10001 \cdot 10^{15}$ den absoluten Rundungsfehler $|x - \text{rd}(x)| = 10^{10}$.

Anders sieht es für den *relativen Rundungsfehler* $|x - \text{rd}(x)|/|x|$ aus:

Satz 5.5. *Es sei q eine gerade Zahl. Dann gilt*

$$\frac{|x - \text{rd}(x)|}{|x|} \leq \frac{q}{2} q^{-\ell} = \text{eps}(q, \ell) \quad \forall x \in \mathbb{R}, x \neq 0 . \quad (5.9)$$

Die Zahl $\text{eps}(q, \ell)$ heißt Maschinengenauigkeit.

Beweis. Es sei $x = a^* q^\ell$ und $\text{rd}(x) = aq^\ell$ mit $q^{-1} \leq a^* < 1$ wie in (5.6) und gerundeter Mantisse a aus (5.7). Wir zeigen

$$|a - a^*| \leq \frac{1}{2} q^{-\ell} \quad (5.10)$$

und betrachten dazu zwei Fälle.

Im ersten Fall ist $a_{\ell+1} < \frac{1}{2}q$. Es wird also abgerundet. Aus $\frac{1}{2}q \in \mathbb{N}$ folgt zunächst $a_{\ell+1} + 1 \leq \frac{1}{2}q$. Weiter gilt

$$|a^* - a| = \sum_{i=\ell+1}^{\infty} a_i q^{-i} = \left(a_{\ell+1} + \sum_{j=1}^{\infty} a_{j+\ell+1} q^{-j} \right) q^{-(\ell+1)} < (a_{\ell+1} + 1) q^{-(\ell+1)} \leq \frac{1}{2} q^{-\ell} .$$

Das Kleinerzeichen ist richtig, weil nach Voraussetzung nicht durchweg $a_{j+\ell+1} = q - 1$ sein darf.

Im zweiten Fall ist $a_{\ell+1} \geq \frac{1}{2}q$. Es wird also aufgerundet. Nun erhält man

$$\begin{aligned} |a^* - a| &= a - a^* = q^{-\ell} - \sum_{i=\ell+1}^{\infty} a_i q^{-i} = \left(1 - \sum_{j=1}^{\infty} a_{j+\ell} q^{-j}\right) q^{-\ell} \\ &\leq (1 - a_{\ell+1} q^{-1}) q^{-\ell} \leq \left(1 - \frac{1}{2}\right) q^{-\ell} = \frac{1}{2} q^{-\ell}. \end{aligned}$$

Damit ist (5.10) bewiesen. Daraus folgt unmittelbar

$$|x - \text{rd}(x)| = |a^* - a| q^e \leq \frac{1}{2} q^{e-\ell},$$

und wegen $a_1 \geq 1$ gilt

$$|x| = |a^*| q^e \geq a_1 q^{-1} q^e \geq q^{e-1}.$$

Insgesamt erhält man

$$\frac{|x - \text{rd}(x)|}{|x|} \leq \frac{1}{2} \frac{q^{e-\ell}}{q^{e-1}} = \frac{1}{2} q^{-\ell} = \text{eps}(q, \ell)$$

und damit die Behauptung. □

Wir gehen im folgenden davon aus, daß q geradzahlig ist, wie etwa $q = 2$ oder $q = 10$.

Der relative Rundungsfehler ist für alle $x \in \mathbb{R}$ durch die Maschinengenauigkeit $\text{eps}(q, \ell)$ beschränkt.

Wir notieren noch eine einfache aber nützliche Folgerung aus Satz 5.5.

Satz 5.6. Für $\tilde{x} = \text{rd}(x)$ gilt

$$\tilde{x} = x(1 + \varepsilon), \quad |\varepsilon| \leq \text{eps}(q, \ell). \quad (5.11)$$

Beweis. Auflösen von (5.11) nach ε liefert

$$|\varepsilon| = \frac{|x - \tilde{x}|}{|x|} \leq \text{eps}(q, \ell).$$

□

Falls (5.11) vorliegt, wird x durch \tilde{x} bis auf ℓ gültige Stellen approximiert. Nur beim Abrunden stimmen die führenden ℓ Ziffern allerdings auch tatsächlich überein. So wird im Falle $q = 10$ (Dezimalsystem), $\ell = 4$ und

$$x = 12344, \tilde{x} = 12340, \quad x = 12,995, \tilde{x} = 13,00$$

jeweils x durch \tilde{x} bis auf $\ell = 4$ gültige Stellen approximiert.

Im folgenden schreiben wir meist kurz eps statt $\text{eps}(q, \ell)$.

Mit Blick auf Satz 5.6 kann man eine untere Schranke eps für die aktuelle Maschinengenauigkeit eps mit folgendem Programm berechnen.

Algorithmus 5.7 (MachinePrec).

```
function epsilon=MachinePrec
    epsilon=1;
    while 1 + epsilon > 1
        epsilon = epsilon/2;
    end
```

Man beachte, daß Algorithmus 5.7 nur deshalb terminiert, weil es Rundungsfehler gibt.

5.3.5 Gleitkommazahlen als Äquivalenzklassen reeller Zahlen

Wir wollen die reellen Zahlen in einzelne Pakete verpacken, ähnlich den Restklassen in Abschnitt 4.2. Dazu definieren wir die Äquivalenzrelation

$$x \cong y \iff \text{rd}(x) = \text{rd}(y), \quad x, y \in \mathbb{R}, \quad (5.12)$$

und betrachten die entsprechenden Äquivalenzklassen (siehe Anhang A.1)

$$[x] = \{y \in \mathbb{R} \mid \text{rd}(x) = \text{rd}(y)\}, \quad x \in \mathbb{R}.$$

Beispielsweise ist $[1] = [1 - \text{eps}, 1 + \text{eps})$. In jeder Äquivalenzklasse $[x]$ liegt genau eine Gleitkommazahl, nämlich $\tilde{x} = \text{rd}(x)$. Diese können wir jeweils als Repräsentanten wählen und so die Gleitkommazahlen mit den Äquivalenzklassen identifizieren:

$$\tilde{x} \longleftrightarrow [\tilde{x}], \quad \tilde{x} \in \mathbb{G}(q, \ell). \quad (5.13)$$

Eine Gleitkommazahl $\tilde{x} \in \mathbb{G}(q, \ell)$ repräsentiert alle reellen Zahlen, die auf \tilde{x} gerundet werden.

Das Rechnen mit Gleitkommazahlen $\tilde{x}, \tilde{y}, \dots$ entspricht also dem Rechnen mit *Paketen* $[\tilde{x}], [\tilde{y}], \dots$ reeller Zahlen. Verschiedene reelle Zahlen $x, x' \in [\tilde{x}]$ können wir nicht unterscheiden.

Wir wollen wissen, wie die Äquivalenzklassen $[\tilde{x}], \tilde{x} \in \mathbb{G}(q, \ell)$, aussehen.

Satz 5.8. *Es sei*

$$\tilde{x} = aq^e \in \mathbb{G}(q, \ell), \quad q^{-1} \leq a < 1.$$

Dann ist $[\tilde{x}]$ das halboffene Intervall

$$[\tilde{x}] = [\tilde{x} - q^{e-1}\text{eps}, \tilde{x} + q^{e-1}\text{eps}) \subset \mathbb{R}. \quad (5.14)$$

Beweis. Wegen $a < 1$ ist $a = 0, a_1 a_2 \dots a_\ell$, und wegen $a \geq q^{-1}$ ist $a_1 \neq 0$. Nach (5.7) werden daher alle Zahlen $a^* \in \mathbb{R}$ mit den Eigenschaften

$$0 \leq a^* - a < \frac{q}{2} q^{-(\ell+1)}$$

auf a abgerundet. Daraus folgt

$$x = a^* q^e < aq^e + \frac{q}{2} q^{-(\ell+1)} q^e = \tilde{x} + q^{e-1}\text{eps}.$$

Umgekehrt werden nach (5.7) alle $a^* \in \mathbb{R}$ mit den Eigenschaften

$$a^* - (a - q^{-\ell}) \geq \frac{q}{2} q^{-(\ell+1)}$$

auf a aufgerundet. Daraus folgt

$$x = a^* q^e \geq aq^e - \frac{1}{2} q^{-\ell} q^e = \tilde{x} - q^{e-1}\text{eps}.$$

□

Wie erwartet, wird die Länge $2q^{e-1}\text{eps} = q^{e-\ell}$ des Intervalls $[\tilde{x}]$ mit wachsendem $|\tilde{x}|$ immer größer.

Zum Schluß wollen wir feststellen, wie die Rundungs-Äquivalenz (5.12) zweier reeller Zahlen x, y mit ihrem Abstand zusammenhängt.

Satz 5.9. *Es gilt*

$$\text{rd}(x) = \text{rd}(y) \implies |x - y| \leq (|x| + |y|)\text{eps}, \quad x, y \in \mathbb{R}.$$

Umgekehrt gibt es zu jedem $\varepsilon > 0$ reelle Zahlen x, y mit den Eigenschaften $|x - y| \leq \varepsilon$ und $\text{rd}(x) \neq \text{rd}(y)$.

Beweis. Der Fall $x = y = 0$ ist trivial. Es sei also $x, y \neq 0$ und $\text{rd}(x) = \text{rd}(y)$. Nach Satz 5.6 gilt dann

$$x(1 + \varepsilon_x) = \text{rd}(x) = \text{rd}(y) = y(1 + \varepsilon_y)$$

mit $|\varepsilon_x|, |\varepsilon_y| \leq \text{eps}$. Auflösen nach $x - y$ liefert

$$|x - y| = |y\varepsilon_y - x\varepsilon_x| \leq (|x| + |y|)\text{eps}.$$

Den zweiten Teil der Behauptung zeigen wir durch ein Beispiel. Die Zahl

$$x = 0, a_1 \cdots a_\ell a_{\ell+1}, \quad a_1 \neq 0, a_{\ell+1} = \frac{1}{2}q,$$

wird aufgerundet, aber die Zahl $y = x - \varepsilon$ wird für alle ε mit $0 < \varepsilon \leq a_{\ell+1}q^{-(\ell+1)}$ abgerundet. Damit gilt offenbar $|x - y| \leq \varepsilon$ und $\text{rd}(x) \neq \text{rd}(y)$. \square

Auch ein beliebig kleiner *relativer Abstand*

$$\frac{|x - y|}{|x|} = \varepsilon \ll \text{eps}$$

von $x, y \in \mathbb{R}$ kann wegen Satz 5.9 durch Runden auf zwei unterschiedliche Gleitkommazahlen zu einem relativen Abstand von

$$\frac{|\text{rd}(x) - \text{rd}(y)|}{|\text{rd}(x)|} \geq 2q^{-1}\text{eps} \gg \varepsilon$$

vergrößert werden (siehe (5.4)).

5.3.6 Gleichheitsabfragen

Da wir es wegen (5.13) im Rechner nicht mit reellen Zahlen x, y sondern nur mit entsprechenden Paketen $[\tilde{x}], [\tilde{y}] \subset \mathbb{R}$ zu tun haben, wirken sich Gleichheitsabfragen in Programmen nicht immer so aus wie erwartet. Das wollen wir anhand eines einfachen Beispiels illustrieren.

Die stetige Funktion

$$f(x) = \begin{cases} \frac{x - \pi}{\sin(x)} & \text{falls } \sin(x) \neq 0 \\ -1 & \text{falls } \sin(x) = 0 \end{cases}, \quad x \in [\frac{1}{2}\pi, \frac{3}{2}\pi], \quad (5.15)$$

soll graphisch dargestellt werden. Dazu schreiben wir ein Programm, das $y_i = f(x_i)$ an den $n + 1$ Gitterpunkten $x_i = \frac{1}{2}\pi + ih$, $i = 0, \dots, n$, im Abstand von $h = \pi/n$ auswertet und dann die stückweise lineare Interpolation der Punkte (x_i, y_i) mit einem vorhandenen Unterprogramm `plot(x, y)` ausgibt. Bei direkter Umsetzung der Formel (5.15) erhält man:

Algorithmus 5.10 (TumbPlot).

```
function tumbplot(n)
    h=pi/n;
    for i=1:n+1
        x(i) = pi/2 + (i-1)*h;
        if (sin(x(i))==0) y(i) = -1;
        else y(i) = (x(i)-pi)/sin(x(i)); end;
    end;
    plot(x,y);
```

Im linken Bild von Abbildung 5.2 sieht man das Ergebnis für $n = 5$. Um eine bessere Darstellung zu erhalten wählen wir mehr Gitterpunkte, also $n = 10$, und erleben eine böse Überraschung, die im rechten Bild zu sehen ist.

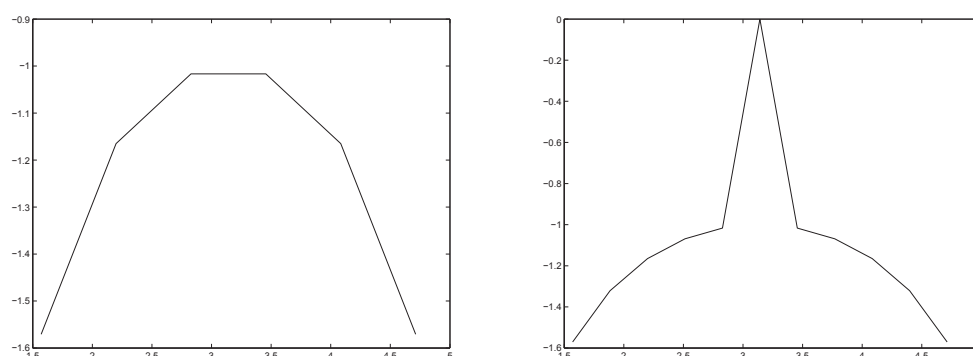


Abb. 5.2 Das Ergebnis von TumbPlot für $n = 5$ und $n = 10$. Was ist passiert?

Wir suchen einen Programmierfehler, finden keinen, probieren eine Reihe anderer Werte von n und stellen nach einer Weile fest, daß ungerade n immer zu richtigen und gerade n immer zu völlig falschen Ergebnissen führt. Woran liegt das?

Ist n ungerade, so kommt der in der Mitte des Intervalls $[\frac{1}{2}\pi, \frac{3}{2}\pi]$ liegende Punkt $x = \pi$, also der Fall $\sin(x) = 0$, nicht vor, für gerade n aber schon. Im Falle $n = 10$ erhalten wir

$$x(5) - \pi = 0 \quad \sin(x(5)) = 1.2246e-16$$

und daher $y(5) = 0 \neq -1$. Runden von x_5 und π führt also auf dasselbe Ergebnis (siehe (5.13)). Die Auswertung der Sinusfunktion am gerundeten Wert $x(5) = \text{rd}(x_5) \neq \pi$ ist aber richtigerweise von Null verschieden. Pech gehabt.

Im allgemeinen besteht zwischen der Gleichheit von Rechenergebnissen x, y bei exakter und \tilde{x}, \tilde{y} bei gerundeter Rechnung kein Zusammenhang: Aus $x = y$ folgt im allgemeinen nicht $\tilde{x} = \tilde{y}$ und umgekehrt. Beim Programmieren gilt deshalb die Regel:

Gleichheitsabfragen von Gleitkommazahlen sind verboten!

An die Stelle von Gleichheitsabfragen treten *Ungleichungsbedingungen*, welche sich aus den Konditions- und Stabilitätsüberlegungen ergeben, die wir in den folgenden Kapiteln anstellen wollen (vgl. zum Beispiel Satz 7.5 in Abschnitt 7.2). Im vorliegenden Fall sollte man die Gleichheitsabfrage bei der Auswertung von $f(x_i)$ in TumbPlot durch

```
if (abs(x(i)-pi)<h/2) y(i) = -1; end;
else y(i) = (x(i)-pi)/sin(x(i)); end;
```

ersetzen. Dann kann nichts mehr schief gehen.

5.4 Praktische Realisierung

Gleitkommazahlen werden im Computer entsprechend Definition 5.3 in normalisierter Form $\tilde{x} = (-1)^s a q^e$ mit $q = 2$ dargestellt. Die Speicherlänge von N Bits wird also auf das Vorzeichen s , die Mantisse $a = 0, a_1 a_2 \dots a_\ell$ und den ganzzahligen Exponenten $e = e_{M-1} e_{M-2} \dots e_0$ verteilt. Da wegen der Normalisierung $a \geq q^{-1} = 0,1_2$ immer $a_1 = 1$ sein muss, kommt man für die Mantisse mit $\ell - 1$ Bits für a_2, \dots, a_ℓ aus (hidden bit).

Im Gegensatz zu natürlichen, ganzen und rationalen Zahlen ist das Abspeichern von Gleitkommazahlen in der Form

$$\boxed{s \mid e_{M-1} e_{M-2} \dots e_0 \mid a_2 \dots a_\ell}$$

durch eine Industrienorm, die Norm IEEE 754 standardisiert³. Die beiden wichtigsten dort festgelegten Datentypen sind `float` mit einfacher Genauigkeit (single precision) und `double` mit doppelter

³ Bereits Konrad Zuse hat in seinen ersten Rechenmaschinen Z1 und Z3 eine ähnliche Darstellung verwendet.

Genauigkeit (double precision). Das Vorzeichen wird jeweils durch $s = 0$ (positiv) oder $s = 1$ (negativ) repräsentiert. Die Speicherlänge von `float` ist $N = 32$ Bits, von denen 1 Bit auf das Vorzeichen s , $M = 8$ Bits auf den Exponenten $e = e_7 \cdots e_0$ und $\ell - 1 = 23$ Bits auf die Mantisse $a = 0,1a_2 \cdots a_\ell$ entfallen. Die entsprechende Maschinengenauigkeit ist

$$\text{eps}(2, 24) = 2^{-24} \approx 6.0 \cdot 10^{-8}.$$

Für den Datentyp `double` sind $N = 64$ Bits vorgesehen, die in 1 Bit für das Vorzeichen s , $M = 11$ Bits für den Exponenten e und $\ell - 1 = 52$ Bits für die Mantisse a aufgeteilt werden. Das führt auf die Maschinengenauigkeit

$$\text{eps}(2, 52) = 2^{-52} \approx 1.1 \cdot 10^{-16}.$$

Der Wertebereich des Exponenten $e = e_{M-1}e_{M-2} \cdots e_0$ reduziert sich für ein festes M offenbar ein beschränktes Intervall

$$e \in [e_{\min}, e_{\max}] \cap \mathbb{Z}.$$

Für normalisierte Gleitkommazahlen $\tilde{x} = (-1)^s a 2^e$ erhält man daraus die Abschätzung

$$x_{\min} = 2^{e_{\min}-1} \leq |\tilde{x}| \leq (1 - 2^{-\ell}) 2^{e_{\max}} = x_{\max}. \quad (5.16)$$

Die entsprechenden Werte von e_{\min} , e_{\max} und x_{\min} , x_{\max} für `float` und `double` sind neben den anderen Kenngrößen in Tabelle 5.2 dargestellt.

	float	double
Bitlänge	32	64
Vorzeichen s	1	1
Exponent e	8	11
Mantisse a	23	52
Maschinengenauigkeit eps	$6,0 \cdot 10^{-8}$	$1,1 \cdot 10^{-16}$
min. Exponent e_{\min}	-126	-1022
max. Exponent e_{\max}	128	1024
min. Betrag x_{\min}	$1,2 \cdot 10^{-38}$	$2,2 \cdot 10^{-308}$
max. Betrag x_{\max}	$3,4 \cdot 10^{+38}$	$1,8 \cdot 10^{+308}$

Tabelle 5.2 Eigenschaften normalisierter Gleitkommazahlen der Typen `float` und `double`.

Offenbar ist $\tilde{x} = 0$ nicht als normalisierte Gleitpunktzahl darstellbar, sondern ein Sonderfall, der durch $e = e_{M-1} = \cdots = e_0 = a_2 = \cdots = a_0 = 0$ repräsentiert wird. Alle Zahlen x mit $|x| < x_{\min}$ werden auf 0 gerundet. Weitere Sonderfälle sind $+\infty$, $-\infty$ und NaN (Not a Number), für die ebenfalls spezielle Bitfolgen reserviert sind. Schließlich kann man die untere Schranke in (5.16) noch um den Faktor $2^{-(\ell-1)}$ drücken, indem man unter gewissen Umständen von der Normalisierungsbedingung $a \geq 2^{-1}$ abgeht. Beim Runden auf die resultierenden *denormalisierten Gleitkommazahlen* ist Satz 5.9 allerdings nicht mehr richtig. Moderne Prozessoren sind mit einer Gleitkomma-Einheit (floating point unit) ausgestattet, die zumindest die beiden Standarddatentypen `float` und `double` unterstützen.

Die Datentypen `float` und `double` stehen in allen gängigen Programmiersprachen wie C oder JAVA zur Verfügung. Jede in MATLAB erzeugte Variable ist standardmäßig vom Typ `double`. Will man einfache Genauigkeit verwenden, so muß dies explizit angegeben werden. Zur Konvertierung verwendet man dann die Funktion `single()`.

5.5 Aufgaben

Aufgabe 5.1 Wir betrachten Gleitkommazahlen mit Basis $q = 2$, Mantissenlänge $l = 5$ und Exponentenlänge $M = 3$.

Rechnen Sie die Zahlen $x = \frac{27}{4}$, $y_1 = 1$ und $y_2 = \frac{1}{16}$ in die normalisierte Gleitkommadarstellung um und notieren Sie das zugehörige Bitmuster. Bilden Sie anschliessend in Bit-Darstellung die Summen $x + y_1$ und $x + y_2$.

Was beobachten Sie? Bestimmen Sie die kleinste Dezimalzahl, die man in dieser Weise zu x addieren kann, um noch ein korrektes Ergebnis zu bekommen.

Aufgabe 5.2 Geben Sie die größte und kleinste positive Gleitkommazahl an, wenn man von einer Mantissenlänge von 52 Bit und einem Exponenten von 11 Bit ausgeht. Die Mantisse sei $1 \leq a < 2$ normalisiert und der Exponent mit einem Bias von 1023 als nicht negativer Integer dargestellt. Desweiteren ist für den Exponenten zu beachten, dass die Darstellungen $000 \dots 0$ und $111 \dots 1$ reserviert sind und nicht zur Zahlendarstellung zur Verfügung stehen.

Aufgabe 5.3 1. Rechnen Sie folgende Zahlen aus $\mathbb{G}(5,6)$, also Gleitkommazahlen zur Basis $q = 5$ mit Mantissenlänge 6, in Gleitkommazahlen aus $\mathbb{G}(10,5)$ um:

$$x = 13,2441_5$$

$$x = 1,32144_5$$

$$x = 0,01332_5$$

2. Geben Sie für die obigen Werte jeweils die nächstgrößere und nächstkleinere Zahl in $\mathbb{G}(5,6)$ und $\mathbb{G}(10,5)$ an. Warum geht das?

Aufgabe 5.4 Die quadratische Gleichung

$$x^2 - 2px - q = 0$$

soll für $p = 1$, $q = 0.0002$ gelöst werden. Dazu betrachten wir die beiden Verfahren

1. $d = p^2 + q$, $x_1 = p + \sqrt{d}$, $x_2 = p - \sqrt{d}$
2. $d = p^2 + q$, $x_1 = p + \sqrt{d}$, $x_2 = -q/x_1$

Was passiert, wenn die beiden Verfahren in 4- bzw. 5stelliger Gleitkommaarithmetik (Dezimalsystem, 4 bzw. 5 Dezimalstellen) ausgeführt werden? Wie erklären Sie die unterschiedlichen Resultate?

(Dies ist keine Programmieraufgabe! Schreiben Sie auf, was ein Rechner in den einzelnen Schritten tut und beachten Sie, dass der Rechner in diesem Beispiel nur 4 bzw. 5 Dezimalstellen speichern kann.)

Aufgabe 5.5 Gegeben ist das Polynom $p(x) = x^4 + 2$, mit $x > 0$

Es steht Ihnen ein selbstgebauter Rechner zur Verfügung, der Ihnen den Wert von $p(x_0)$ ausrechnet. Dummerweise hat der Programmierer dieses Rechners nicht bedacht, dass eine Eingabe x_0 auch aus dem Bereich der reellen Zahlen stammen kann. Das Programm ignoriert in diesem Fall einfach alle Zahlen nach dem Komma, d.h. aus $x_0 = 3,24$ wird ein $x_0 + \Delta x = 3$. Das daraus resultierende falsche Ergebnis $p(x_0 + \Delta x)$ wird sich also von $p(x_0)$ um den Betrag $\Delta p(\Delta x)$ unterscheiden.

Erstellen Sie per Hand eine Grafik, die das Problem beschreibt. Zeichnen Sie dazu die Funktion $p(x)$ skizzenhaft auf und markieren Sie anhand dieser alle relevanten Bereiche, vor allem $\Delta p(\Delta x)$.

Beurteilen Sie anschließend folgende Aussage:

Ist der Eingabefehler immer gleich groß (Geben Sie also z.B. in obengenannten Rechner $x_0 = 3.7, x_1 = 6.7, x_2 = 152.7$ ein, so ist Δx in allen drei Fällen 0.7), so wird der Fehler im Ergebnis auch immer gleich gross bleiben. Zeigen oder widerlegen sie die Aussage anhand Ihrer Skizze.