Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals in den Bereichen Softwareentwicklung, Internettechnologie und IT-Management aktuell und kompetent relevantes Fachwissen über Technologien und Produkte zur Entwicklung und Anwendung moderner Informationstechnologien.

Gisela Engeln-Müllges · Klaus Niederdrenk
Reinhard Wodicka

Numerik-Algorithmen

Verfahren, Beispiele, Anwendungen

Neunte, vollständig überarbeitete und erweiterte Auflage
mit zahlreichen Abbildungen und Beispielen
sowie 2 CD-ROMs

Springer

Prof. Dr. rer. nat. Gisela Engeln-Müllges
Fachbereich Maschinenbau und Mechatronik
Fachhochschule Aachen
Goethestraße 1
52064 Aachen
engeln-muellges@fh-aachen.de

Prof. Dr. rer. nat. Klaus Niederdrenk
Fachbereich Chemieingenieurwesen
Fachhochschule Münster
Stegerwaldstraße 39
48565 Steinfurt
niederdrenk@fh-muenster.de

Stud. Prof. Dr. rer. nat. Reinhard Wodicka
Am Kupferofen 34
52066 Aachen

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Die Neuauflage fasst die 5. Auflage von „Numerische Mathematik für Ingenieure“ von Gisela Engeln-Müllges und Fritz Reutter(ursprünglich Bibliographisches Institut/Brockhaus AG, Mannheim) und die 8. Auflage von „Numerik-Algorithmen“ von Gisela Engeln-Müllges und Fritz Reutter (ursprünglich VDI-Verlag, Düsseldorf) zusammen.

ISSN 1439-5428
ISBN 3-540-62669-7 Springer Berlin Heidelberg New York

Springer ist ein Unternehmen von Springer Science+Business Media
springer.de

Satz: Druckfertige Daten der Autoren
Herstellung: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig
Umschlaggestaltung: KünkelLopka Werbeagentur, Heidelberg
Gedruckt auf säurefreiem Papier 33/3142/YL - 5 4 3 2 1 0

Im Gedenken an unseren akademischen Lehrer
Prof. Dr. rer. techn. Fritz Reutter (1911–1990)
Rheinisch-Westfälische Technische Hochschule Aachen

Vorwort zur 9. Auflage

Das vorliegende Werk steht in einer langen Tradition. Die erste Auflage erschien 1974 mit den Autoren Gisela Engeln-Müllges und Fritz Reutter unter dem Titel „Formelsammlung zur Numerischen Mathematik mit FORTRAN-Programmen“. Der Inhalt des Buches orientierte sich an Vorlesungen über Numerische Mathematik für Studierende der Ingenieurwissenschaften.
Bei den nachfolgenden Auflagen, an denen Fritz Reutter bis zu seiner schweren Erkrankung 1983 mitarbeitete, nahm der Umfang des Buches stark zu, und die Programmanhänge wurden um die Sprachen Turbo Pascal, C, Modula 2 und Quick-Basic erweitert. Dies hing zusammen mit der raschen Entwicklung der Computertechnologie und ihren Impulsen auf die Numerische Mathematik, die zu einer enormen qualitativen wie auch quantitativen Zunahme numerischer Verfahren führte.

Grundlage des vorliegenden Buches ist die achte Auflage des Buches von Gisela Engeln-Müllges und Fritz Reutter mit dem Titel „Numerik-Algorithmen“ (VDI-Verlag, 1996), die erstmals eine CD-ROM mit Fortran 77/90-, Turbo Pascal- und ANSI C-Programmen enthielt.

Wie bisher wird auch in dieser neunten Auflage von den unten genannten Verfassern besonderer Wert gelegt auf die Erläuterung der Prinzipien der behandelten Verfahren der Numerischen Mathematik, auf die exakte Beschreibung leistungsfähiger Algorithmen und die Entwicklung und Dokumentation zugehöriger Programme. Die ausführliche Darstellung der mathematischen Grundlagen, ergänzt durch viele Abbildungen und durchgerechnete Beispiele, soll den Zugang zur Numerischen Mathematik erleichtern und das Verständnis des algorithmischen Vorgehens fördern. Zahlreiche Beispiele aus ingenieurwissenschaftlichen Anwendungen belegen die Notwendigkeit der behandelten numerischen Verfahren.

Das Buch wendet sich in erster Linie an Ingenieure sowie an Naturwissenschaftler und Informatiker in Studium und Beruf, ferner an Lehrkräfte für mathematisch-naturwissenschaftlichen Unterricht. Deshalb ist besonders auf die Verwendbarkeit der behandelten Themen in der Praxis geachtet worden.

Beigefügt ist dem Buch eine CD-ROM, die umfassend getestete C-Programme zu nahezu allen angegebenen Algorithmen enthält, sowie weitere Software, zu der Informationen auf den folgenden Seiten zu finden sind.

Die inhaltliche Erweiterung des Buches erfordert es, die bisher in ihm enthaltenen numerischen Verfahren für Anfangs- und Randwertprobleme bei gewöhnlichen Differentialgleichungen in einen weiteren Band zu verlagern. Dieser wird, einem häufig geäußerten Wunsch folgend, zusätzlich Numerik partieller Differentialgleichungen sowie wichtige stochastische Methoden, insbesondere statistische Schätz- und Prüfverfahren enthalten und ebenfalls bei Springer erscheinen. Verfasser dieses Buches sind Gisela Engeln-Müllges, Klaus Niederdrenk und Wieland Richter.

Ganz besonders herzlich danken wir den Autoren der Programme und ebenso Doris und Uli Eggermann, die mit Hilfe des Satzprogramms LaTeX das reproduktionsreife Manuskript sowie eine Vielzahl der Abbildungen mit äußerstem Engagement und sehr großem Geschick fertig gestellt haben. Darüber hinaus hat uns Uli Eggermann tatkräftig beim Korrekturlesen und beim Rechnen der Beispiele zur Kubatur unterstützt.
Ein herzlicher Dank gilt nicht zuletzt Herrn Hermann Engesser vom Springer-Verlag für die hervorragende und effektive Zusammenarbeit.

Aachen, Münster, Juli 2004

Gisela Engeln-Müllges
Klaus Niederdrenk
Reinhard Wodicka

Informationen zur beigefügten Software (CD-1, CD-2)

Dem Buch sind zwei CDs beigefügt, welche unterschiedliche Demonstrationsprogramme enthalten sowie ANSI-C-Quellen zur Verwendung in selbstgeschriebenen Programmen.

Informationen zur CD-ROM mit C-Programmen u.a. (CD-1)

Auf dieser CD befinden sich Ansi-C-Quellen von Unterprogrammen zu den meisten der im Buch angegebenen Algorithmen. Diese Quellen können compilerunabhängig in eigenen C-Programmen verwendet werden. Mittels beigefügter Makefile-Datei ist man in der Lage,

- gezielt einzelne Module zu übersetzen
- eine eigene Bibliothek zusammenzustellen, die zu selbstgeschriebenen Programmen hinzugelinkt werden kann
- spezielle Testprogramme zu den Unterprogrammen zu erstellen und mit geeigneten Testdatensätzen ablaufen zu lassen.

Weitergehende Informationen zur Verwendung der Ansi-C-Quellen und Antworten zu Compiler- und Makefile-Fragen sind auf der CD in der Datei *ReadMe.htm* und in der DMAKE-Datei *Makefile.mk* angegeben.

Systemvoraussetzungen

Die Unterprogramme wurden unter verschiedenen Betriebssystemen getestet, u.a. MS-DOS, Windows (95, 98, 2000, NT), OS/2, TOS 4.04, UNIX, Linux. Dabei wurden diverse C-Compiler verwendet.

C++-Programme und Campuslizenzen

Die C-Programme der CD-ROM sind bei der FH-Aachen auch in C++ erhältlich. Informationen über Lizenzen zu den C++-Programmen sowie über Campuslizenzen zu den C- und C++-Programmen können per e-Mail angefordert werden:

engeln-muellges@fh-aachen.de

Weitere Software auf der CD-1

Auf der CD-1 sind noch folgende Programme zu finden:

- CurveTrac
- CurveView
- Interaktive Lehrunterstützung

Zu allen drei Programmen finden Sie Informationen in den folgenden Abschnitten. In der HTML-Datei *ReadMe* auf der CD sind sie ebenfalls kurz beschrieben und können (unter Windows) direkt aus dem Browser (z.B. IE, Netscape, Mozilla, Opera) heraus gestartet werden.

Systemvoraussetzungen

Betriebssystem:	Windows (98, 2000 oder XP)
Arbeitsspeicher:	mindestens 256 MB RAM
Browser:	Netscape ab 6.2, Internet Explorer ab 5.5, Mozilla ab 1.4
Java:	aktivieren (ab Version 1.2); wenn von CD gestartet (aus ReadMe.htm), wird mitgeliefertes Java verwendet.

Informationen zum Expertensystem „CurveTrac“ (CD-1)

CurveTrac ist ein Baukastensystem von numerischen Anwendungen, die auf den C-Programmen aus diesem Buch basieren. Die Benutzer-Oberfläche ist in Java programmiert. Dadurch ist gewährleistet, dass die Anwendung auf den gängigsten Betriebsystemen genutzt werden kann. Getestet wurde CurveTrac unter Windows und Linux. In der hier vorliegenden Ausbaustufe ist das Modul „Höhenlinien“ eingebunden. Informationen über weitere Module können Sie bei engeln-muellges@fh-aachen.de anfordern. Dies sind im Einzelnen:

- Expertensystem zur Numerischen Mathematik
- Berechnung der Schnittkurve zweier Flächen im Raum
- Splinemodul (Flächensplines und Kurvensplines)
- u. a. m. ...

Beschreibung des Moduls „Höhenlinien“

Mit dem Modul „Höhenlinien“ können beliebig angeordnete Wertetripel $\big(x_i, y_i, z_i = f(x_i, y_i)\big)$, z. B. Messdaten, eingegeben werden. Die Funktion $f(x, y)$ wird näherungsweise mittels eines zweidimensionalen Oberflächenspline berechnet und als dreidimensionale Grafik dargestellt. Neben dieser Funktionalität ist es zusätzlich möglich, beliebige Schnitte parallel zur x, y-Ebene zu berechnen und das daraus erzeugte Höhenliniendiagramm in einer zweidimensionalen Grafik darzustellen. Bei Bedarf kann das Höhenliniendiagramm mit frei definierbaren Farben ausgefüllt werden.
CurveTrac mit dem Modul Höhenlinien ist auf der CD-ROM mit den C-Programmen zu finden.
Programmiert wurde CurveTrac von Dominikus Bartusch, Frank Hähling und Thomas Layh.

Informationen zu dem Modul „CurveView“ (CD-1)

CurveView ist wie CurveTrac ein in Java programmiertes Baukastensystem ohne Benutzeroberfläche. Mittels in XML-Dateien abgelegten numerischen Berechnungsvorschriften (basierend auf den C-Programmen aus diesem Buch) können die Ergebnisse einer Berechnung als zwei- bzw. dreidimensionale Grafik bereitgestellt werden. In der hier vorliegenden Ausbaustufe ist das Berechnungsmodul „Kurvensplines“ mit vielen Beispielen eingebunden. Wenn Interesse an weiteren Modulen besteht, z. B.

- Flächensplines
- Nullstellenverfahren
- Differentialgleichungen
- Matrizen
- Statistik
- u. a. m. ...

senden Sie eine Nachricht an engeln-muellges@fh-aachen.de.
Programmiert wurde CurveView von Stefan Kleemann.

Informationen zum Demo-Framework „Interaktive Lehrunterstützung“ auf der CD-ROM (CD-1)

Mit dem Framework können die Lernenden Themengebiete aus Lehrveranstaltungen selbstständig mit interaktiven Elementen vertiefen. Es gibt eine Auswahl von über 100 Interaktionen und Animationen, die gezielt zu speziellen Themengebieten zusammengestellt und dem Lernenden zur Verfügung gestellt werden können. In dieser Demo-Version sind zu jedem der folgenden Themengebiete beispielhaft Interaktionen und Animationen vorhanden:

- Quadratur
- Nullstellen
- Splines
- Differentialgleichungen

Wenn Interesse an weiteren Interaktionen und Animationen besteht, können Informationen unter der Mail-Adresse engeln-muellges@fh-aachen.de angefordert werden.
Programmiert wurde das Framework von Michael Neßlinger.

Informationen zur Demo-CD-ROM „NUMAS“ (CD-2)

Die beiliegende Demo-CD-ROM enthält das Lernfeld „Kurvensplines“ aus dem multimedialen Lehr- und Lernsystem NUMAS zur Numerischen Mathematik und Statistik. Der

Inhalt der Demo-CD-ROM entspricht den Kapiteln über Kurvensplines in diesem Buch und informiert auch über den Gesamtinhalt des Lernsystems.
NUMAS bietet didaktisch aufbereitetes Wissen zur Numerischen Mathematik und Statistik. Es werden Inhalte angeboten, die für die Hochschulausbildung vieler Fachrichtungen grundlegend sind und die sich gleichzeitig hervorragend dazu eignen, in Formen des angeleiteten Selbststudiums aufgenommen zu werden. Lernerinnen und Lerner werden zeitgemäß in ihren Selbstlern- und Selbstorganisationskompetenzen gefördert.
Für die Vollversion NUMAS wurden in großen Teilen die Inhalte dieses Buches und des im Vorwort angekündigten weiteren Bandes mit Differentialgleichungen und Statistik verwendet.
NUMAS ist aus einem vom BMBF und vom Land NRW geförderten Projekt innerhalb der Ausschreibungen „Neue Medien in der Bildung“ und „Neue Medien in der Hochschullehre“ hervorgegangen. An dem Projekt waren die Fachhochschule Aachen (Prof. Dr. Engeln-Müllges), die Freie Universität Berlin (Prof. Dr. Martus), die Fachhochschule Münster (Prof. Dr. Niederdrenk) und die Fachhochschule Südwestfalen (Prof. Dr. Richter) beteiligt.
NUMAS ist auch im Internet unter http://www.numas.de/ zu erreichen.

Der Umgang mit der CD-ROM selbst ist denkbar einfach: Nach dem Einlegen der CD-ROM in das Laufwerk erscheint selbstständig ein Auswahlbildschirm mit den im Gesamtsystem zur Verfügung stehenden Lernfeldern. Von hier aus kann man sich den Inhalt ansehen und bearbeiten. Bei der Demo-CD-ROM ist nur das Lernfeld Kurvensplines freigeschaltet.
Die CD-ROM stellt nur einen Teil der Funktionalität des Lernsystems zur Verfügung. Zum Beispiel sind die Kommunikationswerkzeuge, die an das Internet gebunden sind, nur im Online-System verfügbar.

Systemvoraussetzungen

Ihr Computersystem sollte folgende Bedingungen erfüllen, um einen reibungslosen Betrieb von NUMAS erreichen zu können:

Betriebssystem:	Windows (98, 2000 oder XP)
Browser:	Netscape ab 6.2, Internet Explorer ab 5.5, Mozilla ab 1.4
Javascript:	aktivieren (ab Version 1.2)
Cookies:	aktivieren

Um auch die interaktiven und multimedialen Elemente von NUMAS nutzen zu können, müssen zusätzlich folgende Plug-Ins installiert sein:
Macromedia Flash Player und SUN Java (JRE) ab Version 1.4 (mit Ausnahme der Version 1.4.1_03)
Weitere Informationen zu NUMAS können Sie unter der Mail-Adresse info@numas.de anfordern.

Bezeichnungen

$\Rightarrow$	wenn – dann bzw. hat zur Folge
$\Leftrightarrow$	dann und nur dann
$a := b$	a wird definiert durch b
$\stackrel{!}{=}$	geforderte Gleichheit
$<\quad \leq$	kleiner, kleiner oder gleich
$>\quad \geq$	größer, größer oder gleich
$a << b$	a ist wesentlich kleiner als b
$\approx$	ungefähr gleich
$\equiv$	identisch
$\sim$	proportional bzw. gleichmäßig zu
$\{a_1, a_2, \ldots\}$	Menge aus den Elementen a_1, a_2, ...
$\{x \mid \ldots\}$	Menge aller x für die gilt ...
$\in$	Element von
$\notin$	nicht Element von
$\subseteq$	enthalten in oder Teilmenge von
$\subset$	echt enthalten in oder echte Teilmenge von
$\not\subset$	nicht Teilmenge von
$\mathbb{N}$	Menge der natürlichen Zahlen
$\mathbb{N}_0$	Menge der natürlichen Zahlen mit Null
$\mathbb{Z}$	Menge der ganzen Zahlen
$\mathbb{Q}$	Menge der rationalen Zahlen
$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{C}$	Menge der komplexen Zahlen
$\mathbb{R}^+$, $\mathbb{R}^-$	Menge der positiven bzw. negativen reellen Zahlen
(a, b)	offenes Intervall von a bis b, $a < b$
$[a, b]$	abgeschlossenes Intervall von a bis b, $a < b$
$[a, b)$	halboffenes Intervall von a bis b (rechts offen), $a < b$
$(a, b]$	halboffenes Intervall von a bis b (links offen), $a < b$
$n!$	n Fakultät mit $n! = 1 \cdot 2 \cdot 3 \cdots n$, $n \in \mathbb{N}$, $0! := 1$
$\binom{n}{k}$	n über k mit $\binom{n}{k} := \frac{n!}{k!(n-k)!}$, $k \in \mathbb{N}_0$, $k \leq n$, $n \in \mathbb{N}$
$\prod_{i=1}^{n} a_i$	$a_1 \cdot a_2 \cdot a_3 \ldots a_n$
$\sum_{i=1}^{n} a_i$	$a_1 + a_2 + \ldots + a_n$
$\int_a^b$	Integral in den Grenzen a und b

i	imaginäre Einheit $\mathrm{i} := \sqrt{-1}$
e	Eulersche Zahl $= 2.718\,281\,828\,459\ldots$
$\lvert a\rvert$	Betrag von a mit $\lvert a\rvert := \begin{cases} a & \text{für} \quad a \geq 0 \\ -a & \text{für} \quad a < 0 \end{cases}$
$\lVert\cdot\rVert$	Norm von $\cdot$
$\{a_k\}$	Folge von a_k
$\lim\limits_{k\to\infty} a_k$	Limes von a_k für $k \to \infty$
$\max\{f(x) \lvert x \in [a,b]\}$	Maximum aller Funktionswerte $f(x)$ für $x \in [a,b]$
$\min\{\lvert M_i\rvert \,\big\vert\, i = 1,2,\ldots,m\}$	Minimum aller $\lvert M_i\rvert$ für $i = 1,2,\ldots,m$
(x,y)	geordnetes Paar
$(x_1,x_2,\ldots,x_n)$	geordnetes n-Tupel
$f: I \to \mathbb{R}$	Abbildung f von I nach $\mathbb{R}$
$x \mapsto f(x), x \in D$	x wird $f(x)$ zugeordnet für $x \in D$
$f', f'', f''', f^{(4)}, \ldots, f^{(n)}$	erste, zweite, dritte, vierte, …, n-te Ableitung von f
$C[a,b]$	die Menge der auf $[a,b]$ stetigen Funktionen
$C^n[a,b]$	die Menge der auf $[a,b]$ n-mal stetig differenzierbaren Funktionen
$\mathbb{R}^n$	n-dimensionaler euklidischer Raum
$A = O(h^q)$	$\lvert A/h^q\rvert \leq C$ für $h \to 0$, $C =$ const.
$i = m(1)n$	$m,n \in \mathbb{Z}, \quad m \leq n, \quad i = m, m+1, \ldots, n$
sign (a), sgn (a)	$\begin{cases} 1 & \text{für} \quad a > 0 \\ 0 & \text{für} \quad a = 0 \\ -1 & \text{für} \quad a < 0 \end{cases}$
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \ldots$	Vektoren
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \ldots$	Matrizen
$\boldsymbol{0}$	Nullmatrix bzw. Nullvektor
$\boldsymbol{x} \times \boldsymbol{y}$	Vektorprodukt bzw. Kreuzprodukt
$\boldsymbol{E}$	Einheitsmatrix
$\boldsymbol{A}^{\mathsf{T}}$	transponierte Matrix von $\boldsymbol{A}$
$\boldsymbol{x}^{\mathsf{T}} = (x_1,x_2,\ldots,x_n)$	transponierter Vektor zu $\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$
$\boldsymbol{A}^{-1}$	inverse Matrix von $\boldsymbol{A}$
$\lvert\boldsymbol{A}\rvert$, $\det(\boldsymbol{A})$	Determinante von $\boldsymbol{A}$
[Verweis]	s. im Literaturverzeichnis unter [Verweis]
o. B. d. A.	ohne Beschränkung der Allgemeinheit
□	Ende eines Beispiels oder eines Beweises

Inhaltsverzeichnis

Kapitel 1

Darstellung von Zahlen und Fehleranalyse, Kondition und Stabilität

1.1 Definition von Fehlergrößen

Ein numerisches Verfahren liefert im Allgemeinen anstelle einer gesuchten Zahl a nur einen Näherungswert A für diese Zahl a. Zur Beschreibung dieser Abweichung werden Fehlergrößen eingeführt.

Definition 1.1. (*Wahrer* und *absoluter Fehler*)
Ist A ein Näherungswert für die Zahl a, so heißt die Differenz

$$\Delta_a = a - A$$

der *wahre Fehler* von A und deren Betrag

$$|\Delta_a| = |a - A|$$

der *absolute Fehler* von A.

Sehr oft wird in der mathematischen Literatur Δ_a bereits als absoluter Fehler und $|\Delta_a|$ als Absolutbetrag des Fehlers bezeichnet. In ingenieurwissenschaftlichen Anwendungen ist allerdings die Schreibweise in Definition 1.1 häufiger anzutreffen.

In den meisten Fällen ist die Zahl a nicht bekannt, so dass weder der wahre noch der absolute Fehler eines Näherungswertes A angegeben werden können. Daher versucht man, für den absoluten Fehler $|\Delta_a|$ von A eine möglichst kleine obere Schranke $\varepsilon_a > 0$ anzugeben, so dass $|\Delta_a| \leq \varepsilon_a$ gilt.

Definition 1.2. (*Fehlerschranke für den absoluten Fehler, absoluter Höchstfehler*)
Ist $|\Delta_a|$ der absolute Fehler eines Näherungswertes A und ist $\varepsilon_a > 0$ eine obere Schranke für $|\Delta_a|$, so dass

$$|\Delta_a| \leq \varepsilon_a$$

gilt, dann heißt ε_a eine *Fehlerschranke für den absoluten Fehler* von A oder *absoluter Höchstfehler* von A.

Bei bekanntem ε_a ist wegen $|\Delta_a| = |a - A| \leq \varepsilon_a$

$$A - \varepsilon_a \leq a \leq A + \varepsilon_a\,, \qquad \text{also} \qquad a \in [A - \varepsilon_a, A + \varepsilon_a]\,. \tag{1.1}$$

Um einen Näherungswert A unabhängig von der Größenordnung von a beurteilen zu können, wird der relative Fehler eingeführt.

Definition 1.3. (*Relativer Fehler*)
Ist $|\Delta_a|$ der absolute Fehler eines Näherungswertes A für die Zahl a, so heißt der Quotient

$$|\delta_a| = \frac{|\Delta_a|}{|a\,|} \quad \text{für} \quad a \neq 0$$

der *relative Fehler* von A.

Streng genommen müsste man $\delta_a = \Delta_a/a$ als relativen Fehler von A bezeichnen. Das Vorzeichen von δ_a gibt dann eine zusätzliche Information über die Richtung des Fehlers, d. h. für eine positive Zahl a hat $\delta_a = (a - A)/a < 0$ demnach $a < A$, also einen zu großen Näherungswert A für a zur Folge.

Da in der Regel a unbekannt ist, wird häufig auch

$$|\delta_a| = \frac{|\Delta_a|}{|A|} \qquad \text{für } A \neq 0$$

relativer Fehler von A genannt. Da dann auch Δ_a nicht exakt angebbar ist, wird man sich wieder mit der Angabe einer möglichst guten oberen Schranke für den relativen Fehler behelfen müssen.

Definition 1.4. (*Fehlerschranke für den relativen Fehler, relativer Höchstfehler*)
Ist $|\delta_a|$ der relative Fehler eines Näherungswertes A und gilt mit einem $\varrho_a > 0$

$$|\delta_a| \leq \varrho_a\,,$$

dann heißt ϱ_a eine *Fehlerschranke für den relativen Fehler* $|\delta_a|$ oder *relativer Höchstfehler* von $|\delta_a|$.

Ist ε_a ein absoluter Höchstfehler von A, so ist

$$\varrho_a = \varepsilon_a/|a| \quad \text{bzw.} \quad \varrho_a = \varepsilon_a/|A|$$

ein relativer Höchstfehler von A.

Definition 1.5. (*Prozentualer Fehler, Fehlerschranke für den prozentualen Fehler*)
Ist $|\delta_a|$ der relative Fehler des Näherungswertes A, so heißt

$$|\delta_a| \cdot 100$$

der *prozentuale Fehler* von A (relativer Fehler in Prozent), und σ_a mit

$$|\delta_a| \cdot 100 \;\le\; 100 \cdot \varrho_a \;=\; \sigma_a$$

heißt eine *Fehlerschranke für den prozentualen Fehler*.

Beispiel 1.6.
Für die Zahl $x = \pi$ sind $X = 3.14$ ein Näherungswert und $\varepsilon_x = 0.0016$ eine Schranke für den absoluten Fehler $|\Delta_x|$. Also gilt nach (1.1)

$$3.1384 = 3.14 - 0.0016 \le \pi \le 3.14 + 0.0016 = 3.1416\,,$$

und der relative Höchstfehler ergibt sich zu

$$|\delta_x| \le \varrho_x = \frac{0.0016}{3.14} \approx 0.00051\,.$$

Für den prozentualen Fehler folgt nach Definition 1.5

$$100 \cdot |\delta_x| = 0.051\%\,.$$

□

1.2 Zahlensysteme

1.2.1 Darstellung ganzer Zahlen

Für jede ganze Zahl a gibt es genau eine Potenzentwicklung zur Basis 10 (*Zehnerpotenzen*) der Gestalt

$$a \;= v \cdot (a_n 10^n + a_{n-1}10^{n-1} + \ldots + a_1 10^1 + a_0 10^0) = v \cdot \sum_{k=0}^{n} a_k 10^k$$

mit dem Vorzeichen $v \in \{-1, 1\}$, den Koeffizienten $a_k \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ und einer nicht negativen ganzen Zahl n ($n \in \mathbb{N}_0$). Diese *Dezimaldarstellung* von a erhält man, indem man die Ziffern, die zur Bezeichnung der Zahlen a_k dienen, in absteigender Reihenfolge aufschreibt:

$$a \;= v \cdot a_n a_{n-1} \ldots a_1 a_0 \tag{1.2}$$

Die Ziffern a_k in der Dezimaldarstellung (1.2) werden auch *Stellen* von a genannt; die Stellung einer Ziffer in der Zahl gibt ihren Wert (Einer – Zehner – Hunderter usw.) wieder.

Es gibt keinen triftigen Grund, nur das uns gewohnte und vertraute *Dezimalsystem* zu benutzen. Wegen der einfachen technischen Realisierbarkeit benutzen digitale Rechenanlagen durchweg das *Dualsystem*, das auf den zwei verschiedenen Ziffern 0 und 1 beruht. Jede ganze Zahl a kann damit in eindeutiger Weise als Potenzentwicklung zur Basis 2 (Zweierpotenzen)

$$a = v \cdot \sum_{k=0}^{n} a_k \cdot 2^k \qquad \text{mit } a_k \in \{0, 1\}$$

und einem Vorzeichen $v \in \{-1, 1\}$ geschrieben werden. Und wiederum gibt nur die Stellung einer Ziffer a_k Auskunft über ihren Stellenwert, so dass die Angabe der dualen Ziffern in der richtigen Reihenfolge zur Charakterisierung ausreicht:

$$a = v \cdot (a_n a_{n-1} \ldots a_1 a_0)_2$$

Der Index 2 soll hierbei auf die Darstellung als Dualzahl verweisen; weggelassen wird nur der Index 10 für die üblichen Dezimalzahlen. So hat man beispielsweise

$$\begin{aligned} 2004 &= (+1) \cdot (2004)_{10} \\ &= (+1) \cdot (1 \cdot 2^{10} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^2) \\ &= (+1) \cdot (11111010100)_2 \,. \end{aligned}$$

Definition 1.7. (*Stellenwertsystem zur Basis β*)
Sei $\beta \in \mathbb{N}$, $\beta \geq 2$. Das System der β verschiedenen Ziffern 0, 1, …, $\beta-1$ bildet ein *Stellenwertsystem zur Basis β*, und jede ganze Zahl a lässt sich darin in der Form

$$a = v \cdot \sum_{k=0}^{n} a_k \cdot \beta^k = v \cdot (a_n a_{n-1} \ldots a_1 a_0)_\beta$$

mit eindeutig bestimmten Ziffern $a_k \in \{0, 1, \ldots, \beta-1\}$ und einem Vorzeichen $v \in \{-1, 1\}$ darstellen.

Die Wahl $\beta = 10$ führt auf das geläufige Dezimalsystem und $\beta = 2$ auf das Dualsystem. Anwendungen finden immer wieder auch die Basiswahlen $\beta = 8$ (*Oktalsystem*) und $\beta = 16$ (*Hexadezimalsystem* mit den Ziffern 0, 1, …, 9, A, B, C, D, E, F)

Die gegenseitige Konvertierung von Zahlen in unterschiedlichen Stellenwertsystemen lässt sich einfach bewerkstelligen, wobei es reicht, als ein Bezugssystem das Dezimalsystem anzunehmen. Die Umwandlung einer β-Zahl in die zugehörige Dezimalzahl erfolgt ökonomisch mit dem *Horner-Schema* (s. Abschnitt 3.2) für Polynome:

$$\begin{aligned} a &= v \cdot (a_n \beta^n + a_{n-1}\beta^{n-1} + a_{n-2}\beta^{n-2} + \ldots + a_1\beta + a_0) \\ &= v \cdot \Big(\Big\{ \ldots \big[\underbrace{(a_n\beta + a_{n-1})}_{s_{n-1}} \beta + a_{n-2} \big] \beta + \ldots + a_1 \Big\} \beta + a_0 \Big) \\ &\qquad \underbrace{\qquad\qquad\qquad\qquad}_{s_{n-2}} \\ &\qquad\quad \ddots \\ &\qquad \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{s_1} \\ &\quad \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{s_0} \end{aligned}$$

Algorithmus 1.8. (*Umwandlung einer β-Zahl in eine Dezimalzahl*)
Berechnet man für eine im Stellenwertsystem zur Basis β gegebene ganze Zahl $a = v \cdot (a_n a_{n-1} \dots a_1 a_0)_\beta$, $v \in \{+1, -1\}$, ausgehend von $s_n = a_n$, nacheinander die Größen

$$s_k = \beta \cdot s_{k+1} + a_k\,, \qquad k = n{-}1,\, n{-}2,\, \dots,\, 1,\, 0\,,$$

dann ist $a = v \cdot s_0$ im Dezimalsystem.

Umgekehrt folgt aus dem letzten Zwischenergebnis

$$a \;= v \cdot s_0 = v \cdot (s_1 \cdot \beta + a_0)\,,$$

dass a_0 als der Rest der Division der ganzen Zahl s_0 durch β angesehen werden kann:

$$\frac{s_0}{\beta} = s_1 + \frac{a_0}{\beta} \qquad \text{oder} \qquad \frac{s_0}{\beta} = s_1 \quad \text{Rest } a_0$$

Geht man schrittweise weiter zurück, so folgen die nächsten β-Ziffern a_1, a_2 und so weiter.

Algorithmus 1.9. (*Umwandlung einer Dezimalzahl in eine β-Zahl*)
Berechnet man für eine im Dezimalsystem gegebene ganze Zahl a, ausgehend von $s_0 = |a|$, aus der Gleichung

$$\frac{s_k}{\beta} = s_{k+1} \quad \text{Rest } a_k$$

nacheinander für $k = 0,\, 1,\, 2,\, \dots$ die Größen s_1 und a_0, s_2 und a_1 usw., bis für ein $k = n$ der Wert $s_{n+1} = 0$ ist, dann ist mit den auf diese Weise gewonnenen Ziffern $a_0, a_1, \dots, a_n \in \{0,\, 1, \dots,\, \beta{-}1\}$ die β-Darstellung von a gegeben durch

$$a \;= v \cdot (a_n a_{n-1} \dots a_1 a_0)_\beta\,,$$

wobei $v \in \{-1,\, 1\}$ das Vorzeichen von a bezeichnet.

Die Zahl 2004 besitzt demnach wegen

$$\begin{array}{rcrll}
\frac{2004}{2} & = & 1002 & \text{Rest } 0 & (= a_0) \\
\frac{1002}{2} & = & 501 & \text{Rest } 0 & (= a_1) \\
\frac{501}{2} & = & 250 & \text{Rest } 1 & (= a_2) \\
\frac{250}{2} & = & 125 & \text{Rest } 0 & (= a_3) \\
\frac{125}{2} & = & 62 & \text{Rest } 1 & (= a_4) \\
\frac{62}{2} & = & 31 & \text{Rest } 0 & (= a_5) \\
\frac{31}{2} & = & 15 & \text{Rest } 1 & (= a_6) \\
\frac{15}{2} & = & 7 & \text{Rest } 1 & (= a_7) \\
\frac{7}{2} & = & 3 & \text{Rest } 1 & (= a_8) \\
\frac{3}{2} & = & 1 & \text{Rest } 1 & (= a_9) \\
\frac{1}{2} & = & 0 & \text{Rest } 1 & (= a_{10})
\end{array} \quad \uparrow$$

die Dualdarstellung

$$2004 = (+1) \cdot (11111010100)_2\,.$$

Rechnerintern wird eine ganze Zahl $a = v \cdot |a|$ als Dualzahl durch eine Reihe von Bits ("binary digits") abgespeichert, gewöhnlich in zwei oder vier Bytes (1 Byte = 8 Bits). Bei einer 2-Byte-Hinterlegung bedeutet das:

1. Byte								2. Byte							
$\mu \vdots \alpha_{14}$	α_{13}	α_{12}	α_{11}	α_{10}	α_9	α_8		α_7	α_6	α_5	α_4	α_3	α_2	α_1	α_0

Das Vorzeichen wird dann über

$$\mu = \begin{cases} 0\,, & \text{falls } v = +1 \\ 1\,, & \text{falls } v = -1 \end{cases}$$

gewählt, und im Fall einer positiven Zahl entsprechen α_{14} bis α_0 den Dualziffern von a:

$$a \ = (+1) \cdot (\alpha_{14}\alpha_{13} \ldots \alpha_1\alpha_0)_2$$

Als größte darstellbare positive Zahl ergibt sich dann

$$a \ = (+1) \cdot (111111111111111)_2 = \sum_{k=0}^{14} 1 \cdot 2^k = \frac{2^{15}-1}{2-1} = 32\,767\,.$$

Damit eine Subtraktion auf eine Addition über $a - b = a + (-b)$ zurückgeführt werden kann, werden negative Zahlen rechnerintern durch ein Komplement dargestellt, d. h. im Fall $\mu = 1$ bzw. $v = -1$ stimmen die Hinterlegungen α_k nicht mehr mit den Dualziffern a_k der Zahl $a = (-1) \cdot (a_{14}a_{13} \ldots a_1a_0)_2$ überein.

Vorsicht ist geboten, wenn bei einer Addition oder Subtraktion das Ergebnis nicht mehr im darstellbaren Bereich liegt: Addiert man beispielsweise bei einer 1-Byte-Zahl [größte darstellbare positive Zahl: $2^7-1 = 127$] $97 = (+1)\cdot(1100001)_2$ und $43 = (+1)\cdot(0101011)_2$, so folgt

$$\begin{array}{rr} & 0 \vdots 1\ 1\ 0\ 0\ 0\ 0\ 1 \\ + & 0 \vdots 0\ 1\ 0\ 1\ 0\ 1\ 1 \\ \hline & 1 \vdots 0\ 0\ 0\ 1\ 1\ 0\ 0 \end{array}$$

d. h. ein Übertrag beeinflusst letztendlich das Vorzeichenbit und führt auf ein völlig inplausibles negatives Resultat! Solche Effekte treten leider häufiger auf, ohne dass vom Rechnersystem eine Fehlermeldung oder zumindest eine Warnung ausgesprochen wird. Deshalb ist **bei Rechnerergebnissen stets eine Prüfung auf Plausibilität** erforderlich!

1.2.2 Darstellung reeller Zahlen

Jede nicht ganze, reelle Zahl a besitzt eine Entwicklung der Form

$$a \ = v \cdot \left(\sum_{k=0}^{n} a_k 10^k + \sum_{k=1}^{\infty} b_k 10^{-k} \right)$$

mit einem Vorzeichen $v \in \{-1, 1\}$ und Ziffern a_k, $b_k \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, wobei mindestens ein Term

$$b_k 10^{-k} \neq 0$$

auftritt. Die Dezimaldarstellung einer nicht ganzen Zahl heißt *Dezimalbruch*. Im Dezimalbruch einer Zahl a wird zwischen a_0 und b_1 ein Punkt, der sog. *Dezimalpunkt*, gesetzt:

$$a = v \cdot a_n a_{n-1} a_{n-2} \ldots a_1 a_0 \,.\, b_1 b_2 \ldots b_t \ldots \tag{1.3}$$

Die rechts vom Punkt notierten Stellen heißen *Dezimalstellen* oder *Dezimalen*. Gibt es in einem Dezimalbruch eine Dezimale $b_j \neq 0$, so dass alle folgenden Dezimalen $b_{j+1} = b_{j+2} = ... = 0$ sind, dann heißt der Dezimalbruch *endlich*, andernfalls *unendlich*. Eine Darstellung (1.3) mit endlich vielen Dezimalstellen heißt *Festpunktdarstellung*. Demnach vergegenwärtigt $2\frac{1}{8} = 2.125$ einen endlichen Dezimalbruch und $\frac{1}{3} = 0.3333\ldots$ nicht. Dezimalbruchdarstellungen sind, abgesehen vom Sonderfall der Neunerperiode (z. B. ist $-1.29999\ldots = -1.2\overline{9} = -1.3$), eindeutig.

In einem Stellenwertsystem zur Basis β gilt analog die allgemeine Darstellung

$$\begin{aligned} a &= v \cdot \left(\sum_{k=0}^{n} a_k \, \beta^k + \sum_{k=1}^{\infty} b_k \, \beta^{-k} \right) \\ &= v \cdot \left(a_n a_{n-1} \ldots a_1 a_0 \,.\, b_1 b_2 \ldots \right)_\beta \end{aligned}$$

mit $v \in \{+1, -1\}$ und a_k, $b_k \in \{0, 1, \ldots, \beta-1\}$; der Index β verweist wieder auf das zugrunde liegende Stellenwertsystem und entfällt nur im vertrauten Fall $\beta = 10$. Der zwischen den Ziffern a_0 und b_1 gesetzte Punkt heißt nun *β-Punkt*, und man nennt $(a_n a_{n-1} \ldots a_1 a_0)_\beta$ – manchmal auch unter Berücksichtigung des Vorzeichens v – den *ganzzahligen* Anteil und $(\,.\, b_1 b_2 b_3 \ldots)_\beta$ den *gebrochenen* oder *fraktionierten* Anteil von a. Der Sonderfall $\beta = 2$ führt nun zur *Dualbruchdarstellung* einer reellen Zahl. Gibt es nur endliche viele „Nachkomma“-Stellen, so spricht man von einem *endlichen β-Bruch*, andernfalls von einem *unendlichen β-Bruch*.

Es gilt der

Hilfssatz 1.10.
Jede rationale Zahl p/q mit teilerfremden $p \in \mathbb{Z}$ und $q \in \mathbb{N}$ wird durch einen endlichen oder durch einen unendlichen periodischen Dezimal- oder Dualbruch dargestellt, jede irrationale Zahl durch einen unendlichen nicht periodischen Dezimal- oder Dualbruch.

Zur Umwandlung reeller Zahlen in ein β-System reicht es, den ganzzahligen Anteil – siehe Abschnitt 1.2.1 – und den gebrochenen Anteil unabhängig voneinander zu konvertieren. Beispielsweise folgt mit $\beta = 2$ aus

$$\begin{aligned} 0.625 &= (+1) \cdot (\,.\, b_1 b_2 b_3 \ldots)_2 \qquad\qquad \text{mit } b_k \in \{0, 1\} \\ &= b_1 \, 2^{-1} + b_2 \, 2^{-2} + b_3 \, 2^{-3} + \ldots \end{aligned}$$

durch Multiplikation mit $\beta = 2$

$$\begin{aligned} 1.25 &= b_1 + b_2 \, 2^{-1} + b_3 \, 2^{-2} + \ldots \\ &= (b_1 \,.\, b_2 b_3 \ldots)_2 \,. \end{aligned}$$

Somit entspricht b_1 dem ganzzahligen Anteil 1 von 1.25 und weiter

$$0.25 \;=\; (.\, b_2 b_3 \ldots)_2 \;=\; b_2\, 2^{-1} + b_3\, 2^{-2} + \ldots$$

Fährt man nun mit einer erneuten Multiplikation mit $\beta = 2$ fort

$$0.5 \;=\; (b_2\,.\, b_3 \ldots)_2 \;=\; b_2 + b_3\, 2^{-1} + \ldots\ ,$$

so führt der Vergleich der ganzzahligen und der gebrochenen Anteile in dieser Gleichung auf $b_2 = 0$ und $0.5 = (.\, b_3 b_4 \ldots)_2$. Der nächste gleichartige Schritt ergibt über

$$1.0 \;=\; (b_3\,.\, b_4 \ldots)_2 \;=\; b_3 + b_4\, 2^{-1} + \ldots$$

schließlich $b_3 = 1$ und $b_k = 0$ für $k \geq 4$. Mithin gilt

$$0.625 = (+1) \cdot (.\, 101)_2\ ,$$

was auch plausibel ist, denn 0.625 ist die Summe von $\frac{1}{2} = 2^{-1}$ und $\frac{1}{8} = 2^{-3}$.

Allgemein formuliert heißt das:

Algorithmus 1.11. (*Umwandlung einer echt gebrochenen Zahl in einen β-Bruch*)
Gegeben sei eine echt gebrochene Zahl a, $-1 < a\ < 1$.
Berechnet man, ausgehend von $c_0 = |a|$, für $k = 1, 2, 3, \ldots$ nacheinander die Größen

$$\begin{array}{ll} b_k = \text{int}\ (\beta \cdot c_{k-1}) & \text{„ganzzahliger Anteil von } \beta \cdot c_{k-1}\text{“}, \\ c_k = \beta \cdot c_{k-1} - b_k & \text{„echt gebrochener Anteil von } \beta \cdot c_{k-1}\text{“}, \end{array}$$

dann ist

$$a \;= v \cdot (.\, b_1 b_2 b_3 \ldots)_\beta\ ,$$

wobei $v \in \{-1, 1\}$ das Vorzeichen von a wiedergibt.

Die Berechnung wird abgebrochen, wenn hinreichend viele „Nachkomma“-Stellen ermittelt wurden.

Beispiel 1.12.
Algorithmus 1.11 führt mit $a = 0.1$ auf die folgenden Größen für das Dualsystem:

$$\begin{array}{ll} b_1 = \text{int}\ (0.2) = 0\,, & c_1 = 0.2 \\ b_2 = \text{int}\ (0.4) = 0\,, & c_2 = 0.4 \\ b_3 = \text{int}\ (0.8) = 0\,, & c_3 = 0.8 \\ b_4 = \text{int}\ (1.6) = 1\,, & c_4 = 0.6 \\ b_5 = \text{int}\ (1.2) = 1\,, & c_5 = 0.2 \\ b_6 = \text{int}\ (0.4) = 0\,, & c_6 = 0.4 \\ \ldots & \ldots \end{array}$$

Man erkennt, dass sich die Dualziffernfolge periodisch wiederholt:

$$\begin{array}{rcl} 0.1 & = & (+1) \cdot (.\, 00011001100110011 \ldots)_2 \\ & = & (+1) \cdot (.\, 0\overline{0011})_2\ . \end{array}$$

Mithin entspricht 0.1 einem unendlichen periodischen Dualbruch. □

Definition 1.13. (*Tragende Ziffern*)
Alle Ziffern in der β-Darstellung einer Zahl $a = v \cdot (a_n a_{n-1} \dots a_0 \,.\, b_1 b_2 \dots)_\beta$ mit $v \in \{+1, -1\}$ und $a_k,\ b_k \in \{0, 1, \dots, \beta-1\}$, beginnend mit der ersten von Null verschiedenen Ziffer ($a_n \neq 0$), heißen *tragende Ziffern*.

Beispiel 1.14.

$a = 0.0024060$	besitzt 7 Dezimalen	und	5 tragende Ziffern,
$a = 1573800$	besitzt keine Dezimalen	und	7 tragende Ziffern.
$a = 47.110$	besitzt 3 Dezimalen	und	5 tragende Ziffern.
$a = 0.1$	besitzt 1 Dezimale	und	1 tragende Ziffer.

Für die letzte Zahl ergibt sich im Dualsystem $a = (0.0\overline{0011})_2$, eine Zahl mit unendlich vielen Dualstellen und somit unendlich vielen tragenden Ziffern. □

Definition 1.15. (*Normalisierte Gleitpunktdarstellung*)
Jede reelle Zahl $a \neq 0$ kann als β-Zahl in der Form

$$a = v \cdot (\,.\, d_1 d_2 d_3 \dots d_s d_{s+1} \dots)_\beta \cdot \beta^k \tag{1.4}$$

für ein $k \in \mathbb{Z}$ dargestellt werden, wobei $v \in \{+1, -1\}$ das Vorzeichen von a angibt und $d_1 \neq 0$ gilt. (1.4) heißt *normalisierte β-Gleitpunktdarstellung* von a,
$m = (\,.\, d_1 d_2 d_3 \dots)_\beta$ ihre *β-Mantisse*, und k ihr *β-Exponent*. Besitzt die Mantisse s tragende Ziffern, $s \in \mathbb{N}$, so heißt sie *s-stellig*.

So besitzen $a = 346.5201$ und $b = -0.005386$ im Dezimalsystem die normalisierten Gleitpunktdarstellungen

$$\begin{aligned} a &= 0.3465201 \cdot 10^3 \quad \text{(7-stellige Mantisse)}, \\ b &= -0.5386 \cdot 10^{-2} \quad \text{(4-stellige Mantisse)}. \end{aligned}$$

Einem Computer stehen für Berechnungen nur endlich viele in ihm darstellbare Zahlen, die *Maschinenzahlen*, zur Verfügung. Die Mantissen m dieser Maschinenzahlen haben gewöhnlich eine feste Anzahl von Ziffern. Ferner ist der Exponent $k \in \mathbb{Z}$ durch $-k_1 \leq k \leq k_2$ mit $k_1,\ k_2 \in \mathbb{N}$ begrenzt. Eine weltweit gültige Norm nach ANSI (American National Standards Institute) und IEEE (Institute of Electrical and Electronics Engineers) schreibt für reelle Zahlen, als Dualbruch mit 4 Bytes (= 32 Bits) im Rechner hinterlegt, das Format

$$\begin{array}{|c:cccc:cccccccc|} \hline \mu & e_1 & \dots & e_8 & d_2 & d_3 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & d_{24} \\ \hline \end{array}$$

vor, wobei

- $\rightarrow \mu$ das Vorzeichen der Zahl wiedergibt,
- $\rightarrow\ e_1 \dots e_8$ zur Darstellung des Exponenten k in (1.4) gehört; als nicht negative ganze Zahl, auch *Charakteristik* genannt, wird hiervon stets die Konstante 127 abgezogen, d. h. $k = (e_1 \dots e_8)_2 - 127$,
- $\rightarrow\ d_2 \dots d_{24}$ die Mantisse (bei negativen Zahlen wieder komplementär genommen) der Zahl darstellt, wobei $d_1 = 1$ zu ergänzen ist (als normalisierte Gleitpunktdarstellung bleibt für $d_1 \neq 0$ im Dualsystem nur $d_1 = 1$ übrig!).

Hinzu kommen zusätzliche Kennungen für Fehlerfälle wie "Overflow" (etwa bei Division durch 0) und anderes mehr.

Für die Darstellung von $0.1 = (+1) \cdot (.\,0\overline{0011})_2 = (+1) \cdot (.\,\overline{1100})_2 \cdot 2^{-3}$ bedeutet das

$$\boxed{0 \,\vdots\, \underbrace{01111100}_{=124} \,\vdots\, \underbrace{100\,1100\,1100\,1100\,1100\,1100}_{d_2 \dots d_{23}}} \,1100\,11\ldots$$

Als 2-Exponent ergibt sich $124 - 127 = -3$, und $d_1 = 1$ ist unterdrückt worden. Mithin ist klar, dass 0.1 im Rechner nicht exakt wiedergegeben werden kann! Rundet die Rechnerarithmetik auf, so wird d_{24} zu 1 gewählt, und die interne Maschinenzahl ist größer als 0.1 (genauer: 0.1000000015...); wird abgerundet, also $d_{24} = 0$ gesetzt, dann hat die interne Maschinenzahl einen Wert unterhalb von 0.1 (genauer: 0.09999999404...). Deshalb wird ein Rechner – übrigens unabhängig vom Speicherformat – für das Produkt $10 \cdot 0.1$ nie genau den Wert 1 ermitteln.

Dass mit Gleitpunktzahlen auf Rechenanlagen überhaupt gearbeitet wird, liegt daran, dass man damit bei gleicher Speichergröße einen enorm viel größeren Zahlenbereich überstreicht.

Beispiel 1.16.
Nimmt man – abgesehen vom Vorzeichen – an, dass 8 dezimale Speichereinheiten zur Verfügung stehen, so ließen sich damit in den einzelnen Systemen die folgenden positiven Zahlen darstellen:

1) Ganzzahlsystem
   - kleinste darstellbare positive Zahl 0000 0001
   - größte darstellbare positive Zahl 9999 9999
2) Festpunktsystem mit 4 Dezimalen
   - kleinste darstellbare positive Zahl 0000.0001
   - größte darstellbare positive Zahl 9999.9999
3) Normalisiertes Gleitpunktsystem mit 6 Mantissen- und 2 Exponentenziffern (Exponent = Charakteristik−50)
   - kleinste darstellbare positive Zahl $.100\,000 \cdot 10^{-50}$
   - größte darstellbare positive Zahl $.999\,999 \cdot 10^{49}$