de Gruyter Lehrbuch
Spieß · Rheingans · Programmieren in FORTRAN

Wolfgang E. Spieß · Friedrich G. Rheingans

Einführung in das

Programmieren in FORTRAN

auf der Grundlage von FORTRAN 77

7., neubearbeitete und erweiterte Auflage

Walter de Gruyter · Berlin · New York 1985

Mit 20 Abbildungen und 15 Tabellen

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Spiess, Wolfgang E.: Einführung in das Programmieren in FORTRAN : auf d. Grundlage von FORTRAN 77 / Wolfgang E. Spiess; Friedrich G. Rheingans. – 7., neubearb. u. erw. Aufl. – Berlin ; New York ; de Gruyter, 1985.
(De-Gruyter-Lehrbuch)
ISBN 3-11-010433-4

NE: Rheingans, Friedrich G.:

Vorwort zur 7. Auflage

Als wir im Jahre 1969 die erste Auflage dieses Buches zum Druck vorbereiten ließen, hatten wir eigentlich damit gerechnet, daß es FORTRAN zehn Jahre später nicht mehr geben würde. Heute, nach über 15 Jahren, ist FORTRAN noch immer weltweit in Gebrauch, trotz der enormen Entwicklung, die auf dem Gebiet der elektronischen Datenverarbeitung stattgefunden hat. FORTRAN hat seinen Platz als **die** technisch-wissenschaftliche Programmiersprache behauptet und wird diesen Platz in der heute überschaubaren Zukunft wohl auch behalten.

Aber auch FORTRAN hat sich in dieser Zeit gewandelt; es ist leistungsfähiger und noch benutzerfreundlicher geworden. Diesem Wandel trug die 1978 unter dem Namen FORTRAN 77 herausgegebene amerikanische Norm Rechnung, die dann im Juni 1980 als DIN-Norm übernommen wurde. Sie ist identisch mit der amerikanischen Norm und der internationalen ISO-Norm, womit der Benutzer den Vorteil hat, daß Programme, die in FORTRAN abgefaßt sind, weltweit ausgetauscht werden können. Heute hat jede leistungsfähige Rechenanlage bis hin zu den kleinen Personal-Computern die Möglichkeit, FORTRAN-Programme zu verarbeiten.

Der Inhalt dieser 7. Auflage unseres Buches orientiert sich daher an der internationalen Norm und enthält gegenüber den vorigen Auflagen alle wesentlichen Erweiterungen, die FORTRAN 77 gebracht hat. Das bedingte eine komplette Überarbeitung des gesamten Buches und führte zu einer Zunahme von mehr als 30 Seiten. Wir hoffen, daß es uns dennoch gelungen ist, die Einführung in das Programmieren überschaubar und leicht verständlich wie die vorigen Auflagen zu halten.

Wir unterscheiden in diesem Buch zwischen dem FULL-FORTRAN (Gesamt-FORTRAN) und dem SUBSET-FORTRAN (Teil-FORTRAN), wie dies auch in der Norm der Fall ist und empfehlen dem Anfänger, sich zunächst die Sprachmöglichkeiten des Teil-FORTRAN einzuprägen, ehe er zu späterem Zeitpunkt, wenn er die Grundzüge beherrscht, sich nach und nach die erweiterten Möglichkeiten des Gesamt-FORTRAN zueigen macht. Um dem Anfänger den Weg ins Programmieren noch weiter zu vereinfachen, haben wir die für den Anfänger besonders wichtigen Passagen des Buches – wie schon in den früheren Auflagen – durch einen senkrechten dicken Strich am Rand gekennzeichnet. Wenn der Anfänger sich zunächst nur mit diesen Passagen beschäftigt, wird ihm der Einstieg in FORTRAN noch leichter fallen.

Wir hoffen, daß diese überarbeitete 7. Auflage unseres Buches eine ebenso breite Akzeptanz finden wird wie die vorigen Auflagen und bitten unsere Leser, weiter kritisch den Inhalt zu prüfen und durch Zuschriften uns die Teile aufzuzeigen, die einer Verbesserung bedürfen. Für diese Zuschriften möchten wir uns schon jetzt bedanken.

Berlin — *F. G. Rheingans*
Wesel, im Sommer 1985 — *W. E. Spieß*

Inhaltsverzeichnis

Dem Anfänger, der noch keinerlei Erfahrung mit Rechenautomaten hat, wird empfohlen, sich zunächst nur mit dem durch einen senkrechten Strich am Rand gekennzeichneten Text zu beschäftigen. Der nicht gekennzeichnete Text beschreibt Programmiermöglichkeiten, die der Anfänger sich erst anzueignen braucht, wenn er schon etwas Erfahrung mit FORTRAN-Programmen gesammelt hat.

1. Einführung in das Programmieren

Digitale Datenverarbeitungsanlagen finden heute Anwendung in nahezu allen Zweigen von Technik, Wissenschaft und Verwaltung. Sie dienen als Hilfsmittel zur Lösung sehr verschiedenartiger Probleme, und dem Unkundigen mag es oft scheinen, als hätten die Rechenautomaten dem Menschen das Denken abgenommen.

Das ist jedoch nicht der Fall. Probleme muß der Mensch selbst lösen, wobei er allerdings im Rechenautomaten ein Werkzeug zur Verfügung hat, das durch sehr hohe Rechengeschwindigkeit, große Memorierfähigkeit und hohe Rechengenauigkeit viele Probleme erst überschaubar und damit „entscheidbar" macht.

Wie jeder Automat muß auch der Rechenautomat vorbereitet werden, um die gewünschten Tätigkeiten in der richtigen Reihenfolge zu verrichten, er muß „programmiert" werden. In diesem Buch wollen wir uns mit einem Verfahren befassen, das uns die Programmierung digitaler Rechenautomaten gestattet.

Hierbei geht es darum, eine **Sprache** zu erlernen, die der Rechenautomat versteht und mit deren Hilfe dem Rechenautomaten eine Aufgabe gestellt werden kann, die dieser selbständig löst. Wir setzen dabei voraus, daß die Maschine alles tut, was wir ihr befehlen, sofern wir es in der richtigen Form gesagt haben, schließen also Funktionsfehler des Rechenautomaten aus.

Obwohl FORTRAN (FORmula TRANslation) auch zur Programmierung verwaltungstechnischer und kommerzieller Probleme eingesetzt wird, liegt die Hauptanwendung dieser Programmiersprache im technisch-wissenschaftlichen Bereich. Die in diesem Buch enthaltenen Beispiele und Aufgaben tragen diesem Umstand Rechnung.

1.1. Ein Programmierbeispiel

Zunächst wollen wir an Hand eines kompletten Beispiels einen allgemeinen Überblick über die beim Programmieren notwendigen Tätigkeiten und Überlegungen vermitteln. Der Anfänger braucht sich dabei noch nicht alle Einzelheiten einzuprägen, da die verschiedenen Anweisungen und Vorschriften von FORTRAN eingehend in den Kapiteln 2 bis 4 besprochen werden.

Die Aufgabe besteht darin, eine Tabelle für Z_1 und Z_2 nach der Gleichung

$$Z_{1,2} = \frac{A}{5 \pm \sqrt{B - A}}$$

für beliebig viele Wertepaare A und B zu erstellen.

Löst man die Aufgabe mittels eines einfachen Taschenrechners, so kann dabei wie folgt vorgegangen werden:

In die erste Spalte einer Tabelle (Abb. 1.1) tragen wir den Rechenablauf ein. Die auszuführenden Tätigkeiten bezeichnen wir durch Symbole, wie es in der zweiten Spalte angedeutet ist.

Spalte / Zeile	1	2	3	4	
1	A	①	33,80	22,50	
2	B	②	45,70	18,30	
3	B − A	③ ⇐ ② − ①	11,90	−4,20	
4	$\sqrt{B - A}$	④ ⇐ $\sqrt{③}$	3,45	Wurzel imaginär	
5	$5{,}0 + \sqrt{B - A}$	⑤ ⇐ 5,0 + ④	8,45	—	
6	$5{,}0 - \sqrt{B - A}$	⑥ ⇐ 5,0 − ④	1,55	—	
7	$Z_1 = \frac{A}{5{,}0 + \sqrt{B - A}}$	⑦ ⇐ ① / ⑤	4,00	—	
8	$Z_2 = \frac{A}{5{,}0 - \sqrt{B - A}}$	⑧ ⇐ ① / ⑥	21,80	—	

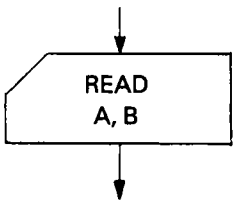
Abb.1.1: Tabellarische Auswertung einer gegebenen Funktion

Der Kreis mit einer Ziffer bezeichnet den Inhalt (den Wert) einer jeden Zeile, der Pfeil ist zu lesen als „ergibt", und die Operationen sind durch die übliche mathematische Symbolik dargestellt. Beispielsweise ist die Zeile 3 wie folgt zu lesen: Der Wert von ③ ergibt sich aus dem Wert von ②, vermindert um den Wert von ①. Die Zeile enthält also nicht eine mathematische Formel, sondern eine **Anweisung** (Befehl), was zu tun ist.

Bevor wir mit der Berechnung beginnen können, müssen wir in Spalte 3 Werte für A und B in die Zeilen 1 und 2 eintragen. Den Inhalt der beiden „Speicherzellen" ① und ② tippen wir dann in den Taschenrechner, lösen dort die Operation Subtrahieren aus und notieren das Ergebnis in Zeile ③ (Speicherzelle ③). Daraufhin bestimmen wir die Quadratwurzel von ③ mittels einer entsprechenden Hilfstabelle oder der entsprechenden Funktion des Taschenrechners und notieren das Ergebnis in Zeile 4. In dieser Art fahren wir bis zum Ende der Befehlsfolge fort. Nach dem ersten Durchlauf wiederholt sich der Vorgang, beginnend mit der Bereitstellung neuer Werte für A und B.

Die Spalte 2 in Abb. 1.1 ist das Programm für die Auswertung der Aufgabe mit dem Taschenrechner. Die Aufstellung dieses Programms ist die Planung für den anschließenden Rechenvorgang. Die Planung für die Berechnung auf einer digitalen Rechenanlage benötigt gegenüber der „Handrechnung" weitere Punkte, die wir an Hand des Flußdiagramms in Abb. 1.3 im einzelnen besprechen wollen.

(Start) kennzeichnet den Anfang des Programms. Die erste Operation, die vom Rechenautomaten ausgeführt werden soll, ist das Einlesen (lies = engl. READ) von Zahlenwerten für die Größen A und B. Im Flußdiagramm ist dieser Vorgang durch den Umriß einer Lochkarte gekennzeichnet.



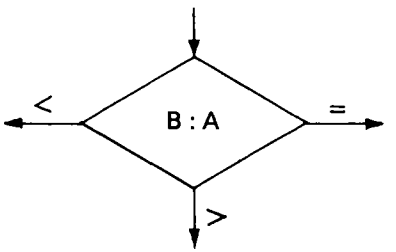

```
1   READ (1,10) A, B
10  FØRMAT (2F10.0)
```

Neben dem Symbol für das Flußdiagramm sind die zugehörigen Anweisungen an den Rechenautomaten in FORTRAN notiert. Die erste Anweisung veranlaßt den Lesevorgang, wobei von einer Lochkarte, die im Lesefach der Rechenanlage liegen muß, eine Zahl für die Speicherzelle A und eine weitere für die Speicherzelle B gelesen wird. Die Unterscheidung verschiedener Speicherzellen wird im Rechenautomaten durch eine fortlaufende Numerierung erreicht, ähnlich wie wir es in der Tabelle in Abb. 1.1 getan haben. Um eine bessere Verbindung der mathematischen Formulierung einer Aufgabe mit dem zugehörigen Programm herzustellen, werden in FORTRAN die Speicherzellen jedoch durch Namen symbolisiert. Die Zuordnung der Namen zu den Nummern der Speicherzellen macht der Rechenautomat während eines besonderen Übersetzungsvorganges selbständig. Der zweiten Anweisung entnimmt der Rechenautomat die Information über das Format der beiden Zahlen, das ist z. B., wieviel Stellen jede Zahl maximal haben darf.

Da im vorhinein nicht bekannt ist, welche Zahlenwerte die Variablen A und B annehmen werden, muß der Programmierer für den Fall, daß der Zahlenwert von A größer als der von B ist, verhindern, daß die Wurzel aus B-A gezogen wird. In

diesem Fall würde die Wurzel imaginär. Der Rechenautomat muß daher prüfen, ob der Zahlenwert von B kleiner, gleich oder größer als der von A ist. Diese Prüfung wird bei der Handrechnung nicht extra geplant sondern beim Ausfüllen jeder Spalte unbewußt durchgeführt.

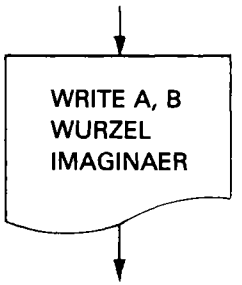
Diese Entscheidung während der Rechnung wird im Flußdiagramm durch eine Entscheidungsraute dargestellt.




```
IF (B–A) 5, 3, 2
```

Die Entscheidungsraute hat einen Eingang und drei Ausgänge. Die Rechnung wird an dem Zweig fortgesetzt, dessen Bedingung durch die Zahlenwerte von B und A erfüllt wird. Ist beispielsweise B kleiner als A, wird die Rechnung entlang dem linken Pfeil fortgesetzt. Neben der Entscheidungsraute ist die entsprechende FORTRAN-Anweisung angegeben. Sie besagt: Springe zur Programmstelle 5, 3 oder 2, wenn (engl. IF) das Ergebnis von B–A kleiner als null, gleich null oder größer als null ist.

Für den Fall, daß B kleiner als A ist, soll der Rechenautomat für das augenblickliche Wertepaar A und B die Meldung ausschreiben (schreiben = engl. WRITE), daß die Wurzel imaginär ist. Im Flußdiagramm wird das Ausdrucken durch einen abgerissenen Papierbogen symbolisiert.

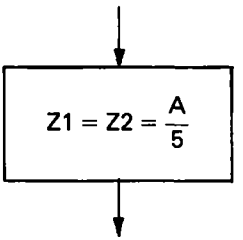



```
 5  WRITE (2,21) A, B
21  FØRMAT('_', 2F15.5,'    WURZEL IMAGINAER')
```

Von den entsprechenden FORTRAN-Anweisungen veranlaßt die erste den eigentlichen Schreibvorgang, während die zweite dem Rechenautomaten angibt, in welchem Format diese Zahlen ausgegeben werden sollen. Die FØRMAT-Anweisung enthält ebenfalls den Orientierungstext. FØRMAT-Anweisungen gehören zu den Lese- und Schreibanweisungen. Zahlen vor FORTRAN-Anweisungen ermöglichen es, an anderen Stellen des Programms auf diese – beispielsweise als Sprung-

ziel – Bezug zu nehmen. Sie heißen **Anweisungsnummern.** So trägt die WRITE-Anweisung die Nummer 5, da sie eines der Sprungziele der Anweisung IF (B–A) 5, 3, 2 ist.

Ist der Zahlenwert von B gleich dem von A, so folgt der Rechenablauf dem rechten Ausgang aus der Entscheidungsraute. Die Berechnungsformel vereinfacht sich zu:




```
3  Z1 = A/5.0
   Z2 = Z1
```

Das Rechteck wird im Flußdiagramm zur Kennzeichnung von Rechenoperationen verwendet. Die Rechenoperation Z1 = A/5.0 hat zur Folge, daß nach Ablauf der Operation die Variable Z1 den Zahlenwert hat, der sich aus der Division des Zahlenwertes von A durch 5.0 ergibt. Das Zeichen = ist beim Programmieren als „ergibt" zu interpretieren. Der Wert der links vom Zeichen = stehenden Variablen ergibt sich aus der Auswertung des rechts vom Zeichen = stehenden Ausdrucks. Nach der Bestimmung von Z1 und Z2 können die Ergebnisse ausgedruckt werden. Im Flußdiagramm in Abb. 1.3 geht der Pfeil daher direkt zu dem Feld, das das Ausschreiben der Ergebnisse symbolisiert.

Die bisher besprochenen Fälle waren Sonderfälle. Ist das momentane Wertepaar so, daß die Bedingungen für die Sonderfälle nicht auftreten, so kann Z1 berechnet werden. Hierzu muß die Quadratwurzel aus B–A gezogen werden. Bei der Handrechnung geschieht das gewöhnlich mit Hilfe einer speziellen Funktionstaste. Ganz ähnlich verhält es sich bei einer digitalen Rechenanlage.

Da Wurzelberechnungen in wissenschaftlichen Programmen sehr häufig vorkommen, faßt man die notwendigen Anweisungen in einem Teilprogramm (Unterprogramm) zusammen und gibt diesem einen Namen, über den man es im Bedarfsfall aufrufen kann.

Der Aufruf eines solchen **Unterprogramms** bewirkt einen Sprung zum Anfang der entsprechenden Anweisungsfolge, die ausgeführt wird und von deren letzter Anweisung ein Rücksprung in das eigentliche Programm erfolgt.

In unserem Beispiel muß dreimal die Quadratwurzel berechnet werden, es wird daher dreimal das Unterprogramm Quadratwurzel (engl. square root und in

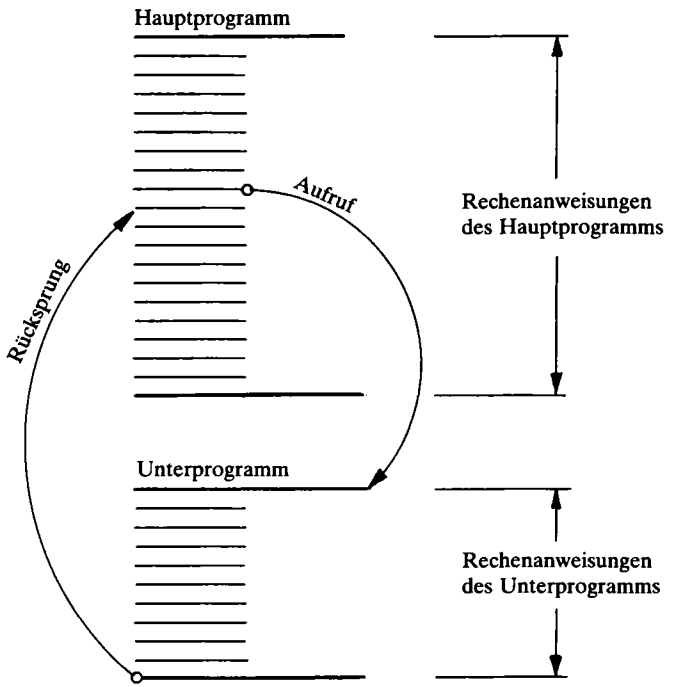
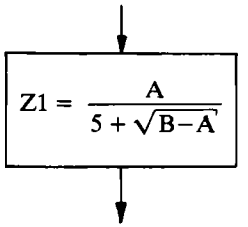


Abb. 1.2: Aufruf eines Unterprogramms

FORTRAN mit SQRT abgekürzt) aufgerufen. Die entsprechenden Notierungen im Flußdiagramm und in FORTRAN sind:



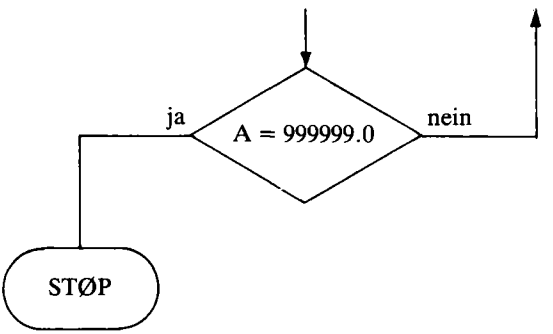

```
2  Z1 = A/ (5.0 + SQRT (B–A))
```

Diese FORTRAN-Anweisung zeigt deutlich, wie sehr sich die Schreibweise in FORTRAN und die der Arithmetik ähneln. Sogar der Aufruf des Unterprogramms SQRT ist in der Anweisung enthalten. Eine einzige FORTRAN-Anweisung kann eine Vielzahl von Rechenoperationen enthalten. Der Zahlenwert, aus dem die Quadratwurzel berechnet werden soll, ist in Klammern hinter dem Namen des Unterprogramms angegeben.

Die Bildung von Unterprogrammen ist ein wichtiges Hilfsmittel, umfangreiche Programme übersichtlich zu gestalten. Unterprogramme und das Hauptprogramm bezeichnet man als **Programmsegmente.**

Zur Berechnung von Z2 muß ein weiterer Sonderfall beachtet werden: Der Nenner in der Formel für Z2 darf nicht null werden, da der Zahlenwert für Z2 dann unendlich groß würde (singulärer Punkt). Jeder Rechenautomat hat einen begrenzten Zahlenbereich. Die größte darstellbare Zahl liegt bei den Rechenautomaten zwischen 10^{40} und 10^{80}. Bei Überschreitungen der größten Zahl meldet der Rechenautomat einen Fehler und unterbricht den Rechenablauf. Um dies auszuschließen, ist im Flußdiagramm ein Zweig vorgesehen, auf dem im Falle $\sqrt{B-A} = 5$ die Meldung „Funktion singulär" mit dem zugehörigen Wertepaar A und B ausgedruckt wird. Ist der Nenner ungleich null, so kann Z2 berechnet werden, und danach können die Zahlenwerte von A, B, Z1, und Z2 ausgedruckt werden.

In der Aufgabenstellung wurde gefordert, Z1 und Z2 für beliebig viele Wertepaare A und B berechnen zu lassen. Bei der Handrechnung wird die Berechnung abgebrochen, wenn das letzte Wertepaar verarbeitet worden ist. Der Rechenautomat weiß jedoch nicht, wann er das letzte Wertepaar gelesen hat. Der Programmierer wählt daher für eine der Eingabegrößen einen bestimmten Zahlenwert, an dem der Rechenautomat erkennen kann, daß die Berechnung abgebrochen werden soll. Im vorliegenden Beispiel geschieht dies, wenn A=999999.0 ist. Die Entscheidungsraute dafür im Flußdiagramm ist



und die FORTRAN-Anweisung lautet

```
7  IF (A.NE.999999.0) GØTØ 1
```

Die FORTRAN-Anweisung besagt: Wenn A ungleich 999999.0 ist, springe (engl. GØTØ) zur Anweisung mit der Nummer 1 (zum Einlesen neuer Werte für A und B), andernfalls führe die im Programm auf die IF-Anweisung folgende Anweisung aus, das heißt, die Anweisung STØP, die den Rechenautomaten veranlaßt, die Berechnungen zu beenden. Der Rechenautomat wird also die Berechnung ununterbrochen fortsetzen bis er für A den Wert 999999.0 erhält. Das kann beim ersten, fünfzigsten oder zehntausendsten Durchlauf des Programms sein. Günstiger wäre es gewesen, diese Abfrage hinter der Leseanweisung am Anfang des Programms

vorzusehen, da dann die überflüssige Durchrechnung des Wertepaars mit A = 999999.0 und die letzte Zeile im Ergebnisausdruck (s. S. 20) wegfallen würden.

An diesem einfachen Beispiel haben wir die wichtigsten Typen von Anweisungen für digitale Rechenanlagen kennengelernt. Es sind:

- Anweisungen zur Eingabe von Daten
- Entscheidungsanweisungen (Abfragen)
- Anweisungen für Rechenoperationen
- Sprunganweisungen
- Unterprogrammanweisungen
- Anweisungen zur Ausgabe von Daten.

Das Flußdiagramm in Abb. 1.3 ist ein flächenhaftes Gebilde. Rechenprogramme sind aber linear aufgebaut, da der Rechenautomat nur eine Anweisung nach der anderen ausführt und nicht zwei oder mehrere Anweisungen gleichzeitig. Die Sprungbefehle ermöglichen es, daß der lineare Programmablauf dennoch Programmverzweigungen, wie sie das Flußdiagramm enthält, ermöglicht. In unserem Beispiel stehen daher an all den Stellen Sprungbefehle, wo der Programmablauf nicht linear fortgesetzt werden soll.

Im folgenden sind das komplette Flußdiagramm und das dazugehörige FORTRAN-Programm angegeben. Der Leser braucht sich noch nicht die FORTRAN-Anweisungen im einzelnen einzuprägen. Mit diesem Beispiel sollen lediglich die Flußdiagrammtechnik und das Verständnis für die Notwendigkeit der verschiedenen Typen von FORTRAN-Anweisungen vermittelt werden.

Man beachte, daß in diesem Buch in FORTRAN-Anweisungen zur Unterscheidung der Buchstabe O als Ø und die Ziffer 0 (null) als 0 geschrieben wird. Diese Art der Unterscheidung erleichtert die Arbeit des Datentypisten, die Rechenanlagen drucken jedoch O (Buchstabe) und 0 (Ziffer), unterscheiden also nicht so deutlich.

Bei einigen Autoren und Rechenanlagenherstellern wird die Unterscheidung zwischen dem Buchstaben O und der Ziffer 0 in umgekehrter Weise vorgenommen.

Zur Bearbeitung des Programms werden die Anweisungen entweder von einem Bildschirm aus über die zugehörige Tastatur oder mittels entsprechend gelochter

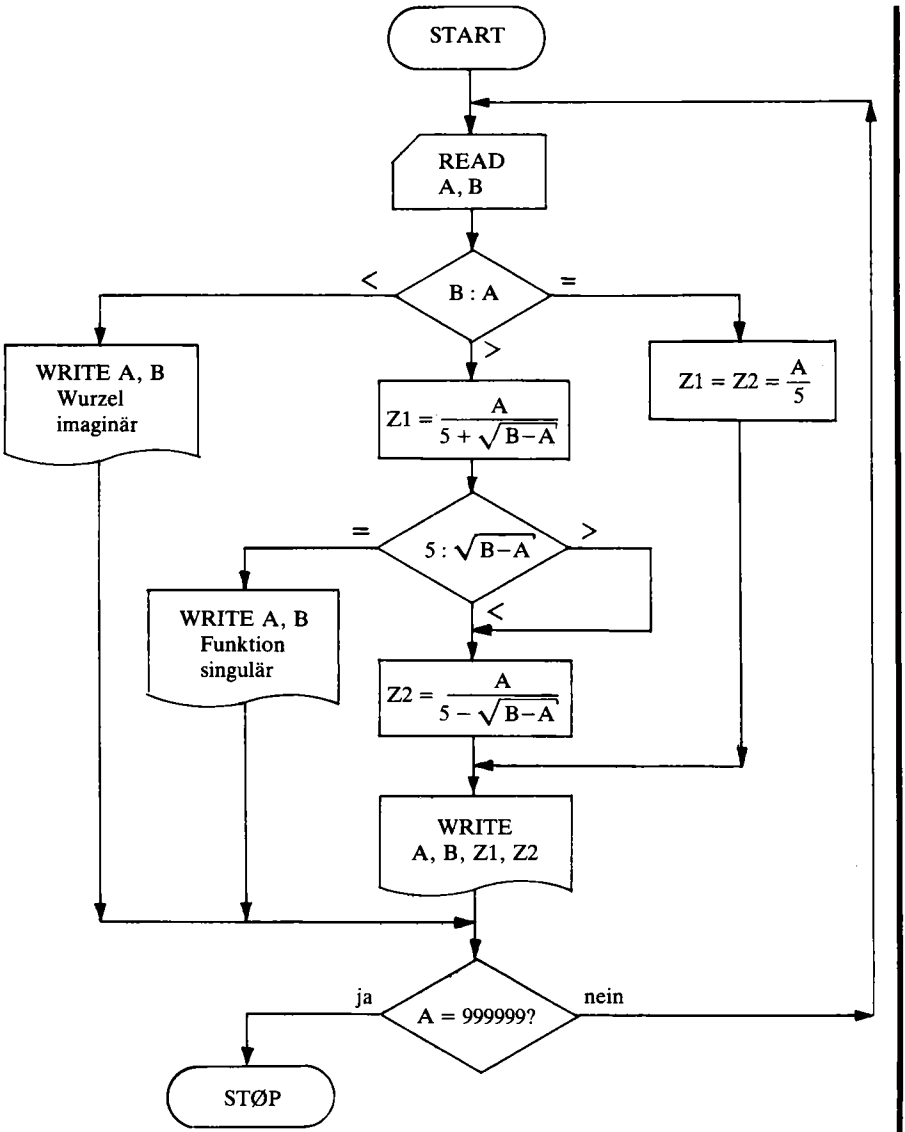


Abb. 1.3: Flußdiagramm zur Auswertung von $Z_{1,2} = \frac{A}{5 \pm \sqrt{B-A}}$

```
      PROGRAM BEISPL
    1 READ(1,10)A,B
   10 FORMAT(2F10.3)
      IF(B-A)5,3,2
    2 Z1=A/(5.0 + SQRT(B-A))
      IF(5.0 - SQRT(B-A))4,6,4
    4 Z2=A/(5.0 - SQRT(B-A))
      GOTO 8
    3 Z1=A/5.0
      Z2=Z1
    8 WRITE(2,20)A,B,Z1,Z2
   20 FORMAT(' ',4F15.5)
      GOTO 7
    5 WRITE(2,21)A,B
   21 FORMAT(' ',2F15.5,'   WURZEL IMAGINAER')
      GOTO 7
    6 WRITE(2,22)A,B
   22 FORMAT(' ',2F15.5,'   FUNKTION SINGULAER')
    7 IF(A.NE.999999.0)GOTO 1
      STOP
```

Lochkarten in den Computer eingegeben, der die Anweisungen in obenstehender Form protokolliert und sie in Maschinenbefehle übersetzt. Nach der Übersetzung muß der Benutzer die Daten eingeben, mit denen das Programm gerechnet werden soll. Im folgenden Ausdruck erscheinen die Zahlenwerte für A in der ersten Spalte, für B in der zweiten Spalte und die Ergebnisse in der dritten und vierten Spalte.

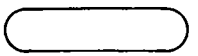
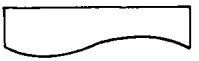
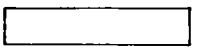
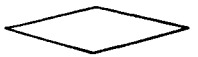
```
    33.80000      45.70000       4.00017      21.80136
    52.30000      65.22000       6.08533      37.20937
   105.30000     130.30000   FUNKTION SINGULAER
    27.60000      27.60000       5.52000       5.52000
    22.50000      18.30000   WURZEL IMAGINAER
     0.00000      15.30000       0.00000       0.00000
999999.00000       0.00000   WURZEL IMAGINAER
```

Bei umfangreichen Programmen werden keine Flußdiagramme wie in Abb. 1.3 erstellt, sondern es werden zwei getrennte Pläne aufgestellt. Der eine ist der Datenflußplan, er gibt die Daten an, die verarbeitet werden sollen, und zeigt den Transport der Daten zwischen verschiedenen Datenträgern[1]. Der andere ist der Programmablaufplan, der die Verarbeitung der Daten, das sind die eigentlichen Berechnungen, angibt. Bei technisch-wissenschaftlichen Problemen ist der Datenanfall meist gering, so daß Datenflußplan und Programmablaufplan meist in einem Diagramm, ähnlich dem in Abb. 1.3, zusammengefaßt werden.

In Tab. 1.1 sind ausgewählte Symbole aus der Norm DIN 66001 für Datenflußpläne und Programmablaufpläne zusammengestellt. Es empfiehlt sich, diese Zeichen in Flußdiagrammen zwecks allgemeiner Verständlichkeit zu verwenden.

1 Datenträger sind z. B. Lochkarten, Lochstreifen, Magnetband, Druckpapier usw.

Tab. 1.1: Genormte Symbole für Flußdiagramme

	Symbol für Beginn und Ende des Programms
	Umriß einer Lochkarte, symbolisiert die Dateneingabe
	Symbol für Ausgabeoperationen
	Rechteck zur Kennzeichnung von Operationen, die nicht durch Sonderzeichen gekennzeichnet sind, meist Rechenoperationen
	Entscheidungsraute für Abfragen
	Kennzeichnung von Unterprogrammen
	Übergangsstelle

1.2. Kurze Beschreibung einer digitalen Rechenanlage

In diesem Abschnitt geben wir einen kurzen Abriß über den Aufbau und die internen Zusammenhänge in einem Rechenautomaten. Die Kenntnis dieses Abschnitts ist zum Programmieren in FORTRAN zwar nicht unbedingt notwendig, sie trägt jedoch zum Verständnis gewisser FORTRAN-Vorschriften bei.

1.2.1. Aufbau

Zwischen der im vorangegangenen Abschnitt dargestellten „Handrechnung" (vgl. Abb. 1.1) und der Arbeitsweise einer digitalen Rechenanlage ergeben sich eine Reihe von Parallelen.

Die Aufgabe, die dem Menschen bei der Handrechnung zufällt, ist das Steuern des Rechenablaufs und das Übertragen der Ergebnisse der einzelnen Operationen zwischen dem Taschenrechner und der Tabelle. Dem Taschenrechner entspricht in der Datenverarbeitungsanlage das Rechenwerk, dem Menschen das Steuerwerk, der Tabelle der Arbeitsspeicher und der Befehlsfolge das Programm. Die Aufgabe der Hilfstabelle beziehungsweise der Funktionstaste des Taschenrechners übernehmen in den meisten Rechenanlagen fertige Unterprogramme, die vom Hersteller mitgeliefert werden. Der Schritt von Spalte 1 nach Spalte 2 ist vergleichbar mit der Übersetzung des Programms in Befehle, die die Maschine direkt verarbeiten kann. Vor dem Beginn der Berechnungen mußten wir in unsere Tabelle die Befehlsfolge und die Werte A und B eintragen. Entsprechend müssen dem Rechenautomaten das **Programm** und die **Daten** eingegeben werden.

Während bei der Handrechnung die Endergebnisse direkt aus Zeile 7 und 8 ablesbar sind, ist bei der digitalen Rechenanlage eine zusätzliche Tätigkeit erforderlich. Dies ist das Übertragen der im Arbeitsspeicher stehenden Endergebnisse, wo sie der Benutzer nicht direkt ablesen kann, beispielsweise auf ein Blatt Papier oder einen Bildschirm (Ausgabe). Ein vereinfachtes Schema einer digitalen Rechenanlage mit den Analogien zur Handrechnung zeigt Abb. 1.4.

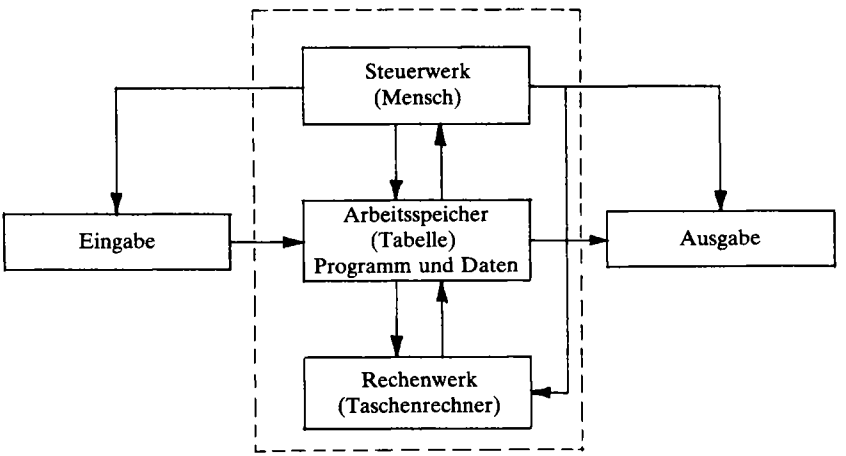


Abb. 1.4: Schematische Darstellung einer digitalen Rechenmaschine mit Analogien zur Handrechnung

Maschinentechnisch läßt sich eine elektronische Datenverarbeitungsanlage in die Zentraleinheit und die peripheren Geräte einteilen. Einen Überblick über eine mittelgroße Rechenanlage zeigt die Abbildung 1.5. Die **Zentraleinheit** beherbergt das zentrale Steuerwerk, das zentrale Rechenwerk und den zentralen Arbeitsspeicher. Gewisse Informationen benötigt die Zentraleinheit von außen, um die danach automatisch ablaufenden Operationen steuern zu können. Eine derartige Steuerinformation ist z. B. die Angabe, mit der Bearbeitung eines bestimmten Programms zu beginnen oder die Bearbeitung eines Programms zu unterbrechen. Diese Anweisungen an die Zentraleinheit werden über ein Bedienungspult, das auch **Konsole** genannt wird, oder periphere Standardgeräte gegeben. Die Konsole erfüllt noch eine zweite Funktion: Über sie informiert die Zentraleinheit den Benutzer über die Vorgänge in der Rechenanlage, sie teilt ihm z. B. mit, welches Programm im Augenblick bearbeitet wird, welche peripheren Geräte von ihr benutzt werden oder veranlaßt den Benutzer, gewisse Handgriffe an der Anlage vorzunehmen. Die Ein- und Ausgabe von Informationen an der Konsole geschieht mittels einer Steuerschreibmaschine oder eines Bildschirms mit Tastatur. Bei kleinen Rechenanlagen wie den Personalcomputern ist die Zentraleinheit und die Ein- und Ausgabe in einem Gerät vereint. Die Eingabe geschieht dabei über die Ta-

statur am Bildschirm und die Ausgabe über den Bildschirm, oder – wenn vorhanden – über einen Drucker.

Unter den **peripheren Geräten** versteht man die Ein- und Ausgabegeräte und die externen Speicher. Über die Eingabegeräte werden die Programme und die dazugehörigen Daten eingelesen. Übliche Eingabegeräte sind die Bildschirmstation und der Lochkartenleser. Darüber hinaus gibt es auch Eingabegeräte für bestimmte Sonderzwecke, wie z. B. Belegleser für Klarschrift, die das Übertragen von Klarschrift direkt in den Computer ermöglichen. Die Ausgabegeräte dienen zur Dokumentation der Rechenergebnisse. Sie lassen sich in zwei Gruppen einteilen:

a) Ausgabegeräte für den Benutzer
b) Ausgabegeräte für den Rechenautomaten

Die Ausgabegeräte für den Benutzer sind in Abb. 1.5 der Zeilendrucker, der digitale Kurvenschreiber und das Bildschirmgerät. Sie dokumentieren die Ergebnisse in Form von Texten, Zahlen oder graphischen Darstellungen, die ein Mensch in angemessener Zeit interpretieren kann. Die Diskette, das Magnetband und der Lochkartenstanzer werden vornehmlich eingesetzt, um Ergebnisse auszugeben, die einem Rechenautomaten zu späterer Zeit wieder als Eingabedaten bereitgestellt werden sollen.

In den externen Speichern werden diejenigen Programme und Daten aufbewahrt, die der Rechner im Augenblick nicht benötigt, die aber im Bedarfsfall in den zentralen Arbeitsspeicher eingelesen und dort verarbeitet werden können. Drei Parameter sind neben den Kosten bei der Auswahl geeigneter externer Speicher wichtig: die Speicherkapazität, die Zugriffszeit und die Übertragungsgeschwindigkeit. Die Speicherkapazität gibt an, wieviele Daten ein Speicher aufnehmen kann. Die Zugriffszeit gibt Auskunft darüber, wie lange der Rechenautomat braucht, um einen bestimmten Speicherplatz zu finden und seinen Inhalt zu lesen; die Übertragungsgeschwindigkeit schließlich besagt, wie schnell Daten vom externen in den Arbeitsspeicher übertragen werden können. Durch die beiden letzten Parameter wird die Arbeitsgeschwindigkeit der gesamten Anlage entscheidend beeinflußt.

Zu den peripheren Geräten wird auch die Datenfernübertragung gerechnet. Sie ermöglicht dem Benutzer den Verkehr mit einer digitalen Rechenanlage über große Entfernung. Beispiele für die Anwendung der Datenfernübertragung sind Rechenanlagen der Universitäten, die von den Benutzern mit Hilfe einer Bildschirmstation und des Telefons benutzt werden können, die zentrale Platzbuchung für Fluggesellschaften und die Informationssysteme der Polizei.

Die Steuerung des Betriebs einer Rechenanlage, das ist das Zusammenwirken der Zentraleinheit mit der oft sehr umfangreichen Periphere, wird heutzutage durch Steuerprogramme erledigt, die ebenfalls im zentralen Arbeitsspeicher stehen.

Diese Steuerprogramme, wie auch die Übersetzungsprogramme und die Programmbibliotheken, die zu den Rechenautomaten mitgeliefert werden und eine

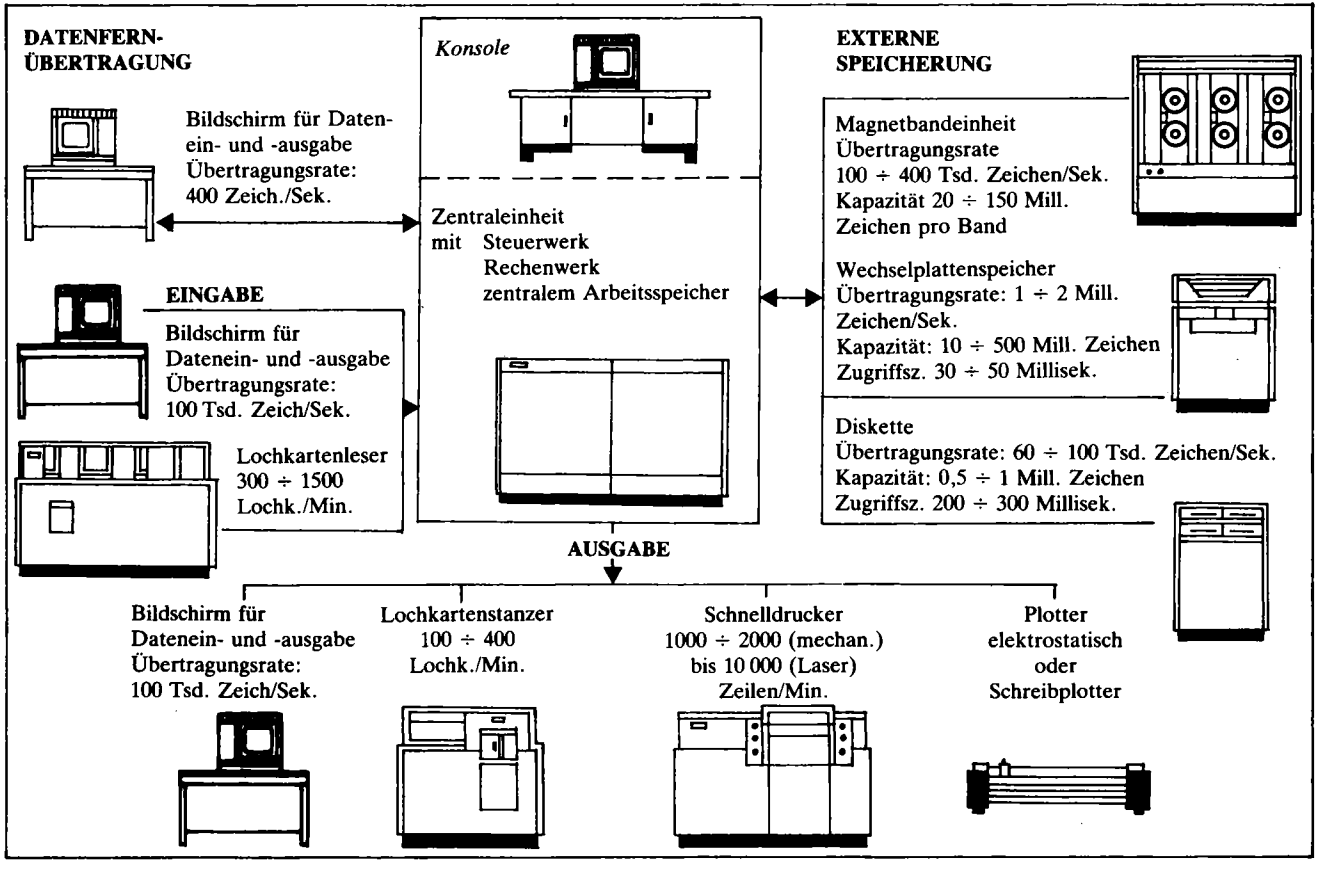


Abb. 1.5 Übersicht über eine mittelgroße Rechenanlage

Fülle von Programmen zur Lösung mathematischer, technisch-wissenschaftlicher und kommerzieller Standardprobleme enthalten, machen heute zu einem wesentlichen Teil die Qualität einer digitalen Rechenanlage aus. Da sie zur Rechenanlage gehören, aber keine Geräte sind, hat es sich eingebürgert, sie als „software" zu bezeichnen zur Unterscheidung von den Geräten, die man unter dem Begriff „hardware" vereinigt.

1.2.2. Informationsdarstellung

Bisher haben wir erläutert, aus welchen Elementen eine digitale Datenverarbeitungsanlage zusammengesetzt ist und wie diese zusammenwirken. Im folgenden wollen wir auf die Darstellung von Daten und Befehlen in einem Rechenautomaten eingehen.

Im täglichen Leben verständigen wir uns mit Hilfe von Wörtern und Zahlen, deren Grundelemente Buchstaben bzw. Ziffern sind, die wir durch Symbole (z. B. C oder 6) darstellen. Wörter und Zahlen sind digitale Daten (von digit = Zeichen, Ziffer), weil sie aus Datenelementen, nämlich 26 Buchstaben und 10 Ziffern, aufgebaut sind.

Dem Rechenautomaten stehen **nur zwei** Elemente zur Darstellung von Daten zur Verfügung, nämlich **etwas** oder **nichts** oder auch **wahr** oder **falsch.** Sie werden durch L (wahr) und O (falsch) symbolisiert. Das Datenelement wird **Bit** (von BInary digiT = Binärzeichen) genannt. Da ein Bit nur die zwei genannten Zustände einnehmen kann, benötigt man für die Darstellung der 26 Buchstaben und 10 Ziffern eine Gruppe von Bits.

Bei den heutzutage üblichen Rechenanlagen werden 6–8 Bits zu einer Gruppe zusammengefaßt. Hiermit lassen sich 64 (6-Bit-Gruppe) bzw. 256 (8-Bit-Gruppe) verschiedene **Zeichen** (engl. Character) binär verschlüsseln, so daß es möglich ist, in einer Rechenanlage außer den Buchstaben und Ziffern auch andere Zeichen wie z. B. Komma, Klammer, etc. zu verarbeiten. Eine 8-Bit-Gruppe wird als **Byte** bezeichnet.

Beispiel: Binäre Verschlüsselung von Buchstaben und Dezimal-Ziffern (Byte)

B ≙ | L | L | O | O | O | O | L | O |

3 ≙ | L | L | O | L | O | O | L | L |

Technisch darstellen und speichern lassen sich Bits beispielweise durch eine bestimmte Stelle auf einer Lochkarte, die gelocht (L) oder nicht gelocht (O) ist, oder durch einen Flip-Flop, der zwei verschiedene Schaltzustände annehmen kann. Einer

Gruppe von Bits entspricht im Arbeitsspeicher der digitalen Rechenanlage eine Gruppe von Flip-Flops, die man **Speicherstelle** nennt. Mehrere Speicherstellen zusammen bilden eine **Speicherzelle.** Eine Speicherstelle enthält immer ein Zeichen z. B. ein A, eine Speicherzelle enthält ein **Datenwort,** das aus mehreren Zeichen besteht, z. B. den Namen ANTON.

In der folgenden Tabelle ist der Zusammenhang zwischen Informations- und Speichereinheiten noch einmal übersichtlich zusammengefaßt.

Tab. 1.2: Übersicht über Informationseinheiten und Speichereinheiten.

Informationseinheit	Speichereinheit
bit	Bitspeicher
Zeichen, Byte (6 ÷ 8 bit)	Speicherstelle
Datenwort (16 ÷ 60 bit bzw. 2 ÷ 10 Zeichen)	Speicherzelle

Die besprochene Art der Zahlendarstellung, die binäre Verschlüsselung von Dezimalziffern, ist für die numerische Bearbeitung in einem Rechenautomaten zu aufwendig, denn beispielsweise lassen sich in einem 24-Bit-Wort nur 3 bis 4 Ziffern unterbringen. Sollen Zahlenwerte rechnerisch verarbeitet und nicht nur wie Worte gespeichert werden, so werden sie zwar im Dezimalsystem eingegeben, zur eigentlichen Rechnung aber in das interne Zahlensystem des Rechenautomaten, das Dualsystem, umgesetzt.

Im Dualsystem werden Zahlenwerte als Summen von Potenzen von 2 ausgedrückt und können daher direkt durch Bitmuster dargestellt werden.

Beispiel: Die Schreibweise der Zahl 26,125 besagt im Dezimalsystem:

$$26{,}125 = 2\cdot10^{1} + 6\cdot10^{0} + 1\cdot10^{-1} + 2\cdot10^{-2} + 5\cdot10^{-3}$$

Im Dualsystem wird sie zerlegt in (im Dualsystem wird zur Vermeidung von Verwechslungen mit dem Dezimalsystem die 1 als L geschrieben):

$$1\cdot2^{4}+1\cdot2^{3}+0\cdot2^{2}+1\cdot2^{1}+0\cdot2^{0}+0\cdot2^{-1}+0\cdot2^{-2}+1\cdot2^{-3} = 26{,}125$$

L L O L O O O L = LLOLO,OOL

Die Zeichenfolge 26,125 im Zehnersystem und die Zeichenfolge LLOLO,OOL im Dualsystem stehen für denselben Wert.

Der Rechenautomat unterscheidet mehrere Typen von Zahlen mit unterschiedlicher Darstellungs- und Verarbeitungsweise. Die beiden wichtigsten sind: