

**Zeitschrift:** Schweizerische Bauzeitung  
**Herausgeber:** Verlags-AG der akademischen technischen Vereine  
**Band:** 91 (1973)  
**Heft:** 5: Datentechnik: Geräte und Anwendung

**Artikel:** Tischrechner mit problemorientierter Sprache. Zweiter Teil: Der Rechner Hewlett-Packard 9820  
**Autor:** Schilt, H.  
**DOI:** <https://doi.org/10.5169/seals-71787>

### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

### **Conditions d'utilisation**

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

### **Terms of use**

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

**Download PDF:** 17.04.2026

**ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>**

hat zwei Lämpchen, von denen immer nur eines leuchtet, nämlich dasjenige, das den jeweiligen Zustand der Taste angibt. Das Drücken der Taste DEG RAD wirkt nur auf die in der Anzeige befindliche Zahl, das heisst, die Anzeige wird von Radianten in Grad umgerechnet oder umgekehrt. Wenn die Taste vor dem Ausrechnen eines Programmes in der Stellung DEG ist, bewirkt das zusammen mit den Befehlen der Tasten  $\sin x$ ,  $\cos x$  oder  $\tan x$  (bzw.  $\arcsin x$ ) die Umrechnung der Argumente von Grad in Radianten (bzw. der Funktionswerte in Grad). CLEAR befiehlt ausser dem Löschen der Anzeige immer auch das Umschalten auf RAD.

Falls eine Adresse bei den bedingten Sprungbefehlen oder bei SEARCH Progr. oder RECORD nicht vollständig ist, leuchtet auch eine Lampe auf. Alle diese Lichter sind sehr angenehm, und ihre Beachtung erspart dem Benutzer manche Fehlmanipulation.

Folgendes kleine Beispiel möge die Eleganz der Programmierung mit dem Tektronix-Tischrechner demonstrieren. Es sei die quadratische Gleichung

$$x^2 + rx + r^2 \cos \alpha = 0$$

zu lösen. Der Gang der Rechnung ist auf Bild 3 angegeben.

Nach Eingabe der Konstanten in K01 und in K02 und Drücken der Taste START rechnet die Maschine unmittelbar das Resultat  $x_1$ , das in der Anzeige erscheint, CONTINUE befiehlt die nächste Rechnung mit  $x_2$  in der Anzeige und Rückkehr zum Anfang. Falls die Anzeige blinkt, bedeutet das erste Resultat absoluter Betrag von  $x_1$  und das zweite Resultat gibt den Polarwinkel der komplexen Zahl  $x_1$  an. CLEAR löscht das Blinken, und die Maschine steht für eine neue Eingabe der Konstanten bereit.

## Zweiter Teil: Der Rechner Hewlett-Packard 9820

### 4. Beschreibung des Gerätes

Von der Firma Hewlett-Packard wird nun auch mit dem Modell 9820 (kurz HP 20 genannt) ein Tischrechner angeboten, der nach einer problemorientierten Sprache organisiert ist. Eine Beschreibung dieses Modells ist daher hier angebracht. Die Grundausrüstung der HP 20 enthält eine Tastatur, eine Anzeige, ein Druckwerk, einen Magnetkartenleser, einen Speicher und zusätzlichen Platz für drei einschiebbare «Read Only Memories» (ROM genannt).

#### 4.1. Die Tastatur

Das Tastenfeld ist in fünf Blöcke aufgeteilt, Bild 4. Die linken drei enthalten Buchstaben und Sonderzeichen; sie dienen zugleich dem Aufrufen der ROM. Der vierte Block enthält Ziffern, Operationsbefehle und Satzzeichen; im fünften Block sind die Tasten für logische Operatoren, für Sprunganweisungen und zum Aufrufen der Register sowie drei Befehlstasten für die Arbeitsart des Rechners zusammengefasst. Oberhalb dieser Blöcke gibt es eine Reihe von Tasten, die sich auf die Anzeige, die Ein- und Ausgabe und den Magnetkartenleser beziehen und dafür eine klare Beschriftung tragen.

#### 4.2. Die Anzeige

In der Anzeige können auf einmal höchstens 16 Zeichen erscheinen. Ein Zeichen ist aus Lichtpunkten zusammengesetzt, die einer Matrix von 5 mal 7 Licht emittierenden Dioden (LED) entnommen sind. Die Schrift ist 7,5 mm hoch und sehr gut lesbar. Beim einmaligen Drücken der Taste BACK wird immer das letzte sichtbare Zeichen gelöscht; durch Bedienen der Taste FORWARD kommt das Zeichen wieder in die Anzeige. Wird eine Zeile mit mehr als 16 Zeichen eingegeben, so verschwinden die vordersten Zeichen aus der Anzeige, ohne dass deswegen im

Auflösung  
Mit den Abkürzungen  
 $\frac{r}{2} = A$  und  $\sqrt{A^2 - r^2 \cos \alpha} = D$

schreiben sich die Lösungen im reellen Fall:

$$x_1 = D - A, \quad x_2 = x_1 - 2D$$

im komplexen Fall:

$$|x_1| = \sqrt{A^2 + D^2}$$

$$\varphi = \arcsin \frac{D}{A}$$

Speicherzuordnung  
 $r = K01$        $\alpha = K02$

$$K11 = A \quad K12 = D$$

$$K03 = x_1 \quad K04 = x_2$$

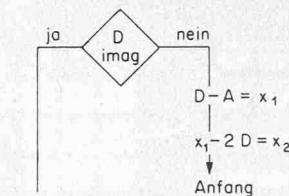
$$K05 = |x_1|$$

$$K06 = \varphi \text{ (Polarwinkel von } x_1 \text{)}$$

Flussdiagramm

$$\frac{r}{2} = A$$

$$\sqrt{A^2 - r^2 \cos \alpha} = D$$



Programm

LEARN

K01 : 2 = K11

(K11 x^2 - K01 x^2 · K02 cos) sqrt x = K12

IF RANGE 064

K12 - K11 = K03 STOP

K03 - 2 · K12 = K04 GOTO 000

K12 sqrt x^2 + y^2 K11 = K05 STOP

K11 IF DATA 086 CD pi +

(K12 : K11) arc tan = K06 END

LEARN

Bild 3. Rechnungsgang zum Beispiel im Text

Innern etwas von der Information verloren geht; im übrigen können die nicht mehr sichtbaren Zeichen mit Hilfe der Taste BACK wieder in die Anzeige geholt werden.

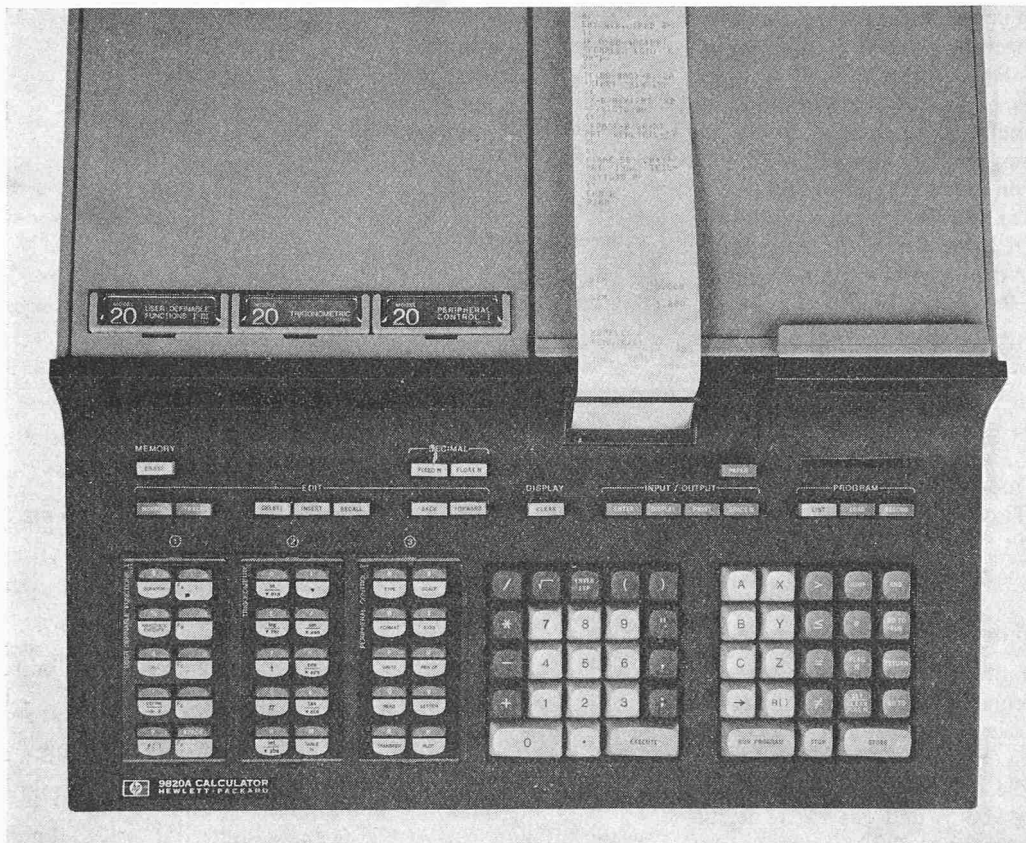
#### 4.3. Der Drucker

Die Maschine druckt auf einen 57 mm breiten Papierstreifen mit Buchstaben und Ziffern. Diese bestehen ebenfalls aus Punkten, die einer 5 mal 7 Matrix entnommen sind. Die Schrift ist 2,7 mm hoch und kann bequem und ohne Missverständnisse gelesen werden. Der Drucker druckt alles, was der Benutzer wünscht. Es seien z.B. die Register A, B, R7 mit gegebenen Werten gefüllt; mit den Befehlen PRINT A, B, R7; EXECUTE werden diese Werte auf je einer Zeile gedruckt. Da das ganze Alphabet zur Verfügung steht, kann auch ein kleiner Text gedruckt werden; z.B. wird durch PRINT «ERGEBNIS»; EXECUTE das Wort ERGEBNIS gedruckt. Wenn man das Resultat einer Rechnung im Register X gespeichert hat, ihm aber den Namen phi geben möchte, so wird durch PRINT «PHI =», X; EXECUTE der Name PHI = am Anfang einer Zeile und der Wert im Register X am Ende der nächsten Zeile gedruckt. Bild 5 zeigt den Streifen für das unter Abschnitt 6 angeführte Beispiel.

Das Anzeigen oder Drucken von Zahlen kann in «Fixed Point n» (FXD n) oder in «Floating Point n» (FLT n) befohlen werden. Z.B. würde die Zahl  $100\pi$  in FXD 5 als 314.15927 und in FLT 4 als 3.1416 E 02 erscheinen. Die Zahlen werden in beiden Fällen gerundet, für die Rechnungen jedoch werden immer 12 Ziffern benutzt und so gespeichert.

#### 4.4 Die Magnetkarten

Es gibt zwei verschiedene Magnetkarten, die sich nur in der Länge und damit auch in der Kapazität unterscheiden; sie



```

0:
ENT A,B,C;FXD 0F
1:
IF 0>BB-4AC;PRT
"KOMPLEX";GTO "K
OM"
2:
I((BB-4AC)-B)/2A
+X;PRT "X1=";XF
3:
-X-B/A+Y;PRT "X2
=";Y;GTO 0F
4:
"KOM";-B/2A+X;
PRT "REALTEIL=";
XF
5:
I(4AC-BB)/2A+Y;
PRT "IMAG. TEIL=";
Y;GTO 0F
6:
END F
R300

```

```

X1=
2.000
X2=
1.000

KOMPLEX
REALTEIL=
.600
IMAG. TEIL=
.800

```

Bild 4 (links). Das Tastenfeld des Grundgerätes (HP 20) mit einem gedruckten Streifen, der zu dem weiter hinten beschriebenen Beispiel gehört. Bild 5 (rechts). Der Streifen zum Beispiel der quadratischen Gleichung in  $\frac{1}{2}$  natürlicher Grösse. Oben: Das gedruckte Programm. Unten: Die Ergebnisse der zwei im Text angeführten numerischen Beispiele

sind 5,1 mm breit und 15 cm bzw. 26,5 cm lang, tragen auf einer Seite eine magnetisierbare und auf der andern Seite eine beschreibbare Schicht. Sie können auf zwei Spuren je ein Programm oder Programmteil oder Register mit Daten aufnehmen. Durch die Befehle REC (record); EXECUTE werden sie mit dem gespeicherten Programm oder mit REC «DA», Rn; EXECUTE mit den Daten beschrieben, die in den Registern R0 bis Rn gespeichert sind. Mit LOD (load); EXECUTE kann der Inhalt einer Karte in die Maschine übertragen werden. Jede Spur der Karte kann gegen unerwünschtes Überschreiben durch Abreissen eines dafür vorgesehenen Stückes gesichert werden.

#### 4.5. Der Speicher

In der Grundausrüstung der HP 20 steht ein Speicher mit 179 Registern zur Verfügung; diesen kann man bis zu 429 Register ausbauen. Sechs Register tragen die Namen A, B, C, X, Y, Z; mit diesen ist eine vereinfachte und der algebraischen Sprache nahe Programmierung möglich. Die andern Register werden mit R(n) (n = Nummer des Registers) aufgerufen; mit ihnen ist eine indirekte Adressierung möglich. Es sei z.B. im Register R5 die Zahl 13 gespeichert; mit RR5 würde dann das Register R13 aufgerufen. Eine Adresse R(A/2) ist auch möglich; die Nummer dieses Registers wäre gleich dem ganzzahligen Teil von A/2. Negative Register-Nummern existieren nicht und R-3 würde als Fehler angezeigt (NOTE 01).

Der Speicher dient auch zur Aufnahme von Programm-Anweisungen; dabei werden, wie üblich, die höchsten Register-Nummern zuerst belegt. Beim Ausdrucken eines Programmes durch den Befehl LIST wird am Ende des ausgedruckten Programmes das nicht benutzte Register mit der höchsten Nummer angegeben; z.B. würde R 73 bedeuten, dass neben den mit

Buchstaben bezeichneten Registern noch die Register von R0 bis R 73 für Datenspeicherung zur Verfügung stehen.

Es gibt drei einschiebbare Zusatzspeicher (ROM), nämlich – einen für die *mathematischen Funktionen*  $\ln x$ ,  $e^x$ ,  $\lg x$ ,  $10^x$ ,  $\sin x$ ,  $\arcsin x$ ,  $\cos x$ ,  $\arccos x$ ,  $\tan x$ ,  $\arctan x$ ,  $\int x$ ,  $\text{abs } x$ ,  $x^y$ , die Zahl  $\pi$ , sowie die Taste TAB n (n = 1, 2, 3), mit welcher der Benutzer die Ein- und Ausgabe der Winkel für die trigonometrischen Funktionen und ihre Umkehrungen in Altgrad, Radianten oder Neugrad wählen kann.

- einen für fünf beliebig *definierbare Funktionen*<sup>2)</sup>
- einen für die *Kontrolle* der anschliessbaren Geräte

#### 4.6. Anschliessbare Geräte

An das Grundgerät können als zusätzliche Geräte angeschlossen werden (Bild 6):

- eine Schreibmaschine
- ein Zeichengerät (Plotter)
- ein Kartenleser für bleistiftmarkierte Karten im Oktalcode
- ein Digital-Umsetzer für graphisch gegebene Daten (fehlt in Bild 6).
- ein Magnetbandleser
- ein Lochstreifenleser
- Messinstrumente mit digitaler Anzeige usw.

### 5. Die Sprache der HP 20

#### 5.1. Zeilen. (Lines)

Ein Programm der HP20 besteht aus einzelnen Zeilen, die nur als Ganzes gespeichert werden können. Ihre Länge kann der Benutzer bestimmen; sie ist naturgemäss nach oben be-

<sup>2)</sup> Falls keine andern Zusatzspeicher benutzt werden, können damit bis zu 25 verschiedene Funktionen definiert werden.

schränkt; je nach Zeichen und Operationen kann eine Zeile höchstens 35 bis 68 Zeichen umfassen. Falls man zu viele Zeichen in eine Zeile eingeben will, erscheint in der Anzeige NOTE 09; man muss dann die Zeile unterteilen und neu eingeben. Wenn man eine Zeile geschrieben hat, kann ihr Inhalt gespeichert werden; die Taste STORE gibt dem Gerät die entsprechende Anweisung. Die Zeile wird dabei vorn mit einer Zeilennummer versehen; ihr Ende erscheint wieder in der Anzeige, unter Umständen vereinfacht geschrieben, und wird mit dem Zeichen | abgeschlossen. Dieses Zeichen dient dem Benutzer als Quittung für die erfolgte Speicherung der Zeile. Unmittelbar nachher kann mit dem Eintippen einer neuen Zeile begonnen werden.

### 5.2. Sätze (Statements)

Jede Zeile besteht aus einzelnen Sätzen, welche Anweisungen für bestimmte abgeschlossene Tätigkeiten des Gerätes enthalten. Jeder Satz ist gegenüber dem nächsten mit einem Strichpunkt abzutrennen; vergisst man dieses Zeichen, so erscheint in der Anzeige NOTE 01; mit dem Drücken von BACK und Strichpunkt ist der Fehler korrigiert. – Will man nicht programmieren, sondern nur einzelne Rechnungen direkt ausführen, so beendet man den Satz durch EXECUTE. Die befohlene Anweisung wird dann ausgeführt und das Ergebnis angezeigt. Angenommen  $6 + (3 + 2) * 4 \rightarrow A$  sei ein Satz, da er weniger als 16 Zeichen enthält ist er noch ganz in der Anzeige sichtbar. Nach dem Drücken der Taste EXECUTE erscheint in der Anzeige die Zahl 26 und diese wird dem Register A zugewiesen. Will man diese Zahl für eine neue Rechnung benutzen, so muss man sie aus dem Speicher A aufrufen und so in die neue Rechnung einbauen. Nach dem Befehl  $A + 5/2$  EXECUTE erscheint in der Anzeige 28.5 und da keine Zuweisung erfolgte, wird die Zahl automatisch dem Register Z zugewiesen; damit steht das Resultat in diesem Register für eine weitere Verwendung bereit.

### 5.3. Zweistellige Operationen

Die Zeichen für zweistellige Operationen werden immer zwischen die Operanden geschrieben, so wie es in der algebraischen Sprache üblich ist. Als Zeichen für die arithmetischen Operationen sind in der HP20 folgende vorgesehen: +, -, \*, /, ↑. Das Malzeichen \* ist nur zwischen Zahlen nötig, z.B.  $6 * 3$ . Wenn die Inhalte der Register A und B miteinander zu multiplizieren sind, kann man kürzer AB schreiben.  $A \uparrow B$  bedeutet, dass die Zahl  $a$  im Register A mit der Zahl  $b$  in Register B potenziert werden soll, also  $A \uparrow B$  ist gleichwertig wie  $a^b$ . Die Zeichen + und - können, wie in der Algebra, Operatoren darstellen aber auch Vorzeichen bedeuten. Die Stufenregel (Hierarchie) der Operationen wird in der Sprache der HP20 streng beachtet. Klammern können beliebig verschachtelt werden; jedoch muss jeder öffnenden Klammer rechts von ihr eine schliessende Klammer entsprechen, sonst erscheint eine Fehleranzeige: NOTE 03 für zu viele öffnende oder NOTE 04 für zu viele schliessende Klammern. Mit diesen Eigenschaften der Operationen und der Klammern ist eine unmittelbare Übertragung der in algebraischen Zeichen vorliegenden Rechnungen in den Tischrechner möglich.

### 5.4. Logische Operatoren

Die Tatsache, dass die HP20 auch logische Operatoren enthält, ist für den Benutzer besonders wertvoll. Als logische Operatoren sind vorhanden: =, ≠, >, ≤. Im Rechenwerk wird für  $A = B$  entweder 0 gesetzt, wenn der Inhalt  $a$  von Register A verschieden vom Inhalt  $b$  des Registers B oder 1, wenn  $a = b$  ist. Entsprechendes gilt für die andern logischen Operatoren. Das Gleichheitszeichen hat in der HP20 also ausschliesslich eine logische Bedeutung.

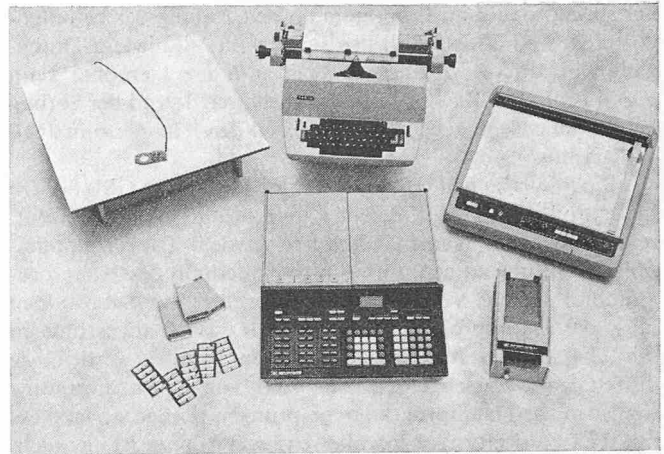


Bild 6. Ansicht des Grundgerätes mit einigen Zusatzgeräten. In der Mitte unten das Grundgerät, oben die Schreibmaschine, von dort im Uhrzeigersinn: das Zeichengerät, der Kartenleser, die einschiebbaren ROM, der Digital-Umsetzer

Verknüpfungen von logischen Beziehungen sind bei der HP20 auch möglich. Dem logischen «und» (= sowohl als auch) entspricht bekanntlich die Multiplikation. N sei das Symbol für einen logischen Ausdruck (z.B. für  $A = B$ ) und M stehe für einen andern logischen Ausdruck (z.B. für  $X > Y$ ), MN ist nur dann = 1, wenn sowohl N als auch M wahr sind.

z. B. ist:

$$(A > B) (B > C) = \begin{cases} 1, & \text{wenn } A > B > C \\ 0 & \text{in allen andern Fällen, insbesondere} \\ & \text{auch für } A = B \text{ oder } B = C \end{cases}$$

Dem logischen «oder» (= entweder oder, oder beides) entspricht die Addition. So ist  $M + N \neq 0$ , wenn entweder M wahr oder N wahr oder beide wahr sind. Man beachte, dass in der Booleschen Algebra  $M + N$  nur die Werte 0 oder 1 annehmen, während in der HP-Sprache  $N + M$  auch 2 geben kann.

z. B. ist der Ausdruck

$$(A > B) + (B > C) = \begin{cases} 0, & \text{wenn } A < B \text{ und } B < C \\ 1, & \text{wenn } d < B \text{ und } B > C \\ & \text{oder } A > B \text{ und } B < C \\ 2, & \text{wenn } A > B \text{ und } B > C \end{cases}$$

### 5.5. Sprungbefehle.

In der Sprache der HP20 gibt es verschiedene Möglichkeiten für unbedingte Sprünge.

1. Die absoluten GO TO Befehle: GO TO 7 befiehlt dem Rechner, das Programm bei der Zeile Nr. 7 fortzusetzen. GO TO «TAG» bedeutet, dass der Rechner mit jener Zeile weiterfahren soll, an deren Anfang das Wort «TAG» steht. Sprungbefehle dieser letzten Art haben den Vorteil, dass sie nicht von der Zeilennummer abhängen und somit nicht beeinflusst werden, wenn im Programm Zeilen eingefügt oder gestrichen werden.

2. Die relativen GO TO Befehle: GO TO +7 bedeutet: das Programm sei erst nach 7 Zeilen weiterzuführen. GO TO -2 befiehlt einen Sprung von 2 Zeilen nach rückwärts.

Der Befehl JMP (Jump) hat ähnliche Eigenschaften wie das relative GO TO; zusätzlich kann bei diesem Befehl die Anzahl der zu überspringenden Zeilen berechnet werden: JMP (A + 3) heisst, dass das Programm nach (a + 3) Zeilen fortgesetzt werden soll, wenn a die in A gespeicherte Zahl bedeutet. (Die Dezimalstellen von a werden dabei unterdrückt). Es sei  $JMP -((A + 1 \rightarrow A) = 5)$  am Schlusse einer Zeile. Solange  $a \neq 4$  ist, beträgt der logische Ausdruck null, und der Rech-

ner springt daher im Programm an den Anfang der betreffenden Zeile; die Zeile wird also wiederholt. Bei jedem Durchgang wird nun die Zahl  $a$  im Register A um 1 erhöht; wenn  $a = 4$  ist, hat  $A + 1 = 5$  den Wahrheitswert 1, und der Sprung beträgt eine Zeile nach rückwärts, weil der Rechner nun  $JMP -1$  «sieht».

Ebenfalls zu den GO TO Befehlen gehört das GSB (= Go to Subroutine). Dieser Befehl kann wie die andern absolut, relativ oder mit Namen adressiert werden. Die Programmzeilen der Subroutinen dürfen nicht innerhalb des Hauptprogrammes stehen. Vor den Programmzeilen der Subroutinen muss das Hauptprogramm z.B. mit GO TO 0 abgeschlossen sein. Es können mehrere Subroutinen benutzt werden, sie dürfen sogar verschachtelt sein. Soll von einer Subroutine wieder in das Hauptprogramm gesprungen werden, so ist diese mit RET (= Return) abzuschliessen; eine Adresse ist hier nicht nötig. Erst nach allen Zeilen der Subroutinen kommt END; als Zeichen für den Rechner, dass nun keine weiteren Programmschritte folgen.

*Bedingte Befehle* werden durch die Taste IF eingeleitet. Folgt auf IF eine logische Operation, so werden die nächsten Sätze der betreffenden Zeile nur dann ausgeführt, wenn die logische Operation wahr ist, d.h. wenn ihr Wert grösser als null ist; andernfalls springt der Rechner auf die nächste Zeile. Damit steht dem Benutzer eine grosse Auswahl von Verzweigungsmöglichkeiten zur Verfügung, die ihm ein vielseitiges Programmieren erlauben.

#### 5.6. «Flags»

Der Benutzer der HP20 hat die Möglichkeit mit 16 verschiedenen Flags zu arbeiten. Dazu sind die Tasten FLG n, SET/CLEAR FLG n zu benützen, wobei die Zahl n von 0 bis 15 gehen kann. Ihre Verwendung ist durch folgende logische Beziehung geklärt: Falls FLG 3 gesetzt ist, gilt  $FLG 3 = 1$  andernfalls ist  $FLG 3 = 0$ ; für die bedingten Verzweigungen kann man entweder  $IF FLG 3 = 1$  oder  $IF FLG 3 = 0$  verwenden, je nach den Umständen, unter denen man verzweigen will. FLG 13 und FLG 15 werden von der Maschine gesetzt, wenn besondere Operationen vorkommen. So setzt das Gerät  $FLG 13 = 1$ , wenn während der Eingabe-Stellung (Enter Mode) der Maschine keine Zahlen eingegeben werden, sondern nur mit RUN PROGRAMM der Befehl zum Fortsetzen des Programmes gegeben wird. FLG 13 erlaubt daher, das Programm zu verzweigen, wenn nach mehreren Durchläufen des normalen Programmes keine Zahlen mehr eingetippt werden.

Falls eine unerlaubte Operation (z.B. Division durch null,  $\arcsin > 1$  usw.) befohlen wird, setzt das Gerät  $FLG 15 = 1$ , stoppt das Programm und gibt eine Anmerkung in der Anzeige. (Z.B. würde NOTE 10 IN 5 angeben, dass in der Zeile 5 eine unerlaubte Operation befohlen wird.) Um in einem solchen Falle das Anhalten des Programmes zu vermeiden, kann der Benutzer am Anfang  $FLG 14 = 1$  setzen.

#### 5.7. Fehleranzeige

Falls der Benutzer fehlerhafte Sätze oder Zeilen in den HP Rechner gibt, so erscheint in vielen Fällen eine Bemerkung in der Anzeige. Es sind 16 verschiedene Anmerkungen, sogenannte NOTES eingebaut, deren Bedeutungen in einer Tabelle zusammengestellt sind und von denen wir einige beschrieben haben. Diese Fehleranzeigen sind für den Benutzer, namentlich am Anfang, eine grosse Hilfe.

#### 5.8. Weitere Programmierhilfen

Beim Programmieren kann man die HP20 zum Aus-schreiben jeder gespeicherten Zeile durch TRACE EXECUTE veranlassen. In dieser Arbeitsart wird bei RUN PROGRAMM jede Zeilennummer und der numerische Wert jeder gerechneten Grösse ausgedruckt, so dass man die einzelnen Schritte

nachkontrollieren kann. NORMAL EXECUTE schaltet das Gerät wieder in die gewöhnliche Arbeitsweise. Falls man in einem fertigen Programm z.B. nach der Zeile Nr n eine neue Zeile einfügen will, springt man durch GO TO n zur n.ten Zeile; durch RECALL erscheint das Ende dieser Zeile in der Anzeige. Man kann nun die einzufügende Zeile programmieren, und durch INSERT STORE wird diese Zeile im Programm eingebaut; dabei werden alle folgenden Zeilennummern um eins erhöht. Durch DELETE kann man eine angezeigte Zeile im Programm löschen, wobei alle folgenden Zeilennummern um eine Einheit herabgesetzt werden. Die relativen und absoluten GO TO Befehle mit numerischer Adresse werden dabei allerdings nicht angepasst.

#### 6. Beispiel

Wir geben noch ein Beispiel mit Programmierläuterungen für das Bestimmen der Lösungen von quadratischen Gleichungen von der Form

$$Ax^2 + Bx + C = 0$$

Zuerst beschreiben wir den Inhalt der Zeile und geben darunter das Programm für HP20 an.

0. Zeile: Eingabe der Werte für A, B, C und Befehl für dreistelliges Ausdrucken der Resultate.

0: ENT A, B, C; FXD 3 |—

1. Zeile: Wenn  $0 > B^2 - 4AC$ , dann soll das Wort «KOMPLEX» gedruckt und das Programm bei der Zeile mit der Adresse «KOM» fortgesetzt werden.

1: IF  $0 > BB - 4AC$ ; PRT «KOMPLEX»; GTO «KOM» |—

2. Zeile: Rechne  $(\sqrt{B^2 - 4AC} - B)/2A = x_1$ ; drucke « $x_1 =$ » und dessen Wert.

2:  $(\sqrt{BB - 4AC} - B)/2A \rightarrow X$ ; PRT « $X1 =$ », X |—

3. Zeile: Rechne  $-x_1 - B/A = x_2$ ; drucke « $X2 =$ » und dessen Wert; gehe an den Anfang.

3:  $-X - B/A \rightarrow Y$ ; GTO 0 |—

4. Zeile: Zeilenname «KOM»;  $-B/2A = \text{Realteil}$ ; drucke «REALTEIL =» und dessen Wert.

4: «KOM»;  $-B/2A \rightarrow X$ ; PRT «REALTEIL =», X |—

5. Zeile: Rechne  $\sqrt{4AC - B^2}/2A = \text{Imaginärteil}$ ; drucke «IMAG.TEIL =» und dessen Wert; gehe an den Anfang.

5:  $\sqrt{4AC - BB}/2A \rightarrow Y$ ; PRT «IMAG.TEIL =», Y; GTO 0 |—

6. Zeile: Ende Programm

6: END |—

Um die Ausführung des Programmes zu befehlen, drückt man END, die Taste RUN PROGRAMM; es erscheint in der Anzeige A. Der Benutzer wird damit aufgefordert, den Zahlenwert für A einzugeben. Er kann dies direkt tun oder aus andern Werten eine Rechnung vorbereiten (z.B.  $7 - 5/3$ ), durch den Befehl RUN PROGRAMM wird in beiden Fällen das Resultat dem Speicher A zugewiesen, und in der Anzeige erscheint B. Nun ist der Wert für B und nachher derjenige für C in gleicher Weise einzutasten, und nach RUN PROGRAMM folgt gleich das Ergebnis.

In Bild 5 ist dieses Beispiel mit dem Programm zusammen auf dem Streifen sichtbar, zuerst für die Werte  $A = 1$ ,  $B = -3$  und  $C = 2$ , man erkennt das Ergebnis  $X1 = 2.000$  und  $X2 = 1.000$ ; dann noch für die Werte  $A = 5$ ,  $B = -6$  und  $C = 5$  mit dem Resultat: «KOMPLEX, REALTEIL = .600, IMAG.TEIL = .800.

In der Eingabe-Stellung (ENTER MODE) ist es sogar möglich, den einzugebenden Grössen diejenigen Namen zu geben, die man ihnen in der mathematischen Ableitung gegeben hat. Nehmen wir an, in unserem Beispiel hätte der Koeffizient A den Namen  $\phi$  und B den Namen R, dann könnte man die 0. Zeile folgendermassen beginnen: ENT «PHI», A, «R», B, C; usw. Auf RUN PROGRAMM erscheint PHI, man gibt den Wert dafür ein und nach einem RUN PROGRAMM erscheint R. A wird übersprungen, aber der eingegebene Wert wird trotzdem dem Register A zugewiesen. Diese Möglichkeit erspart in vielen Fällen weitere Kommentare zum Benutzen der Programme.

## 7. Schlussbemerkung

Mit dieser kurzen Darstellung dürfte die Anpassungsfähigkeit und leichte Programmierbarkeit der HP20 deutlich geworden sein. Diese Eigenschaften verdankt man unter anderem dem Umstand, dass jede Zeile einzeln bei den Befehlen EXECUTE oder STORE mit Hilfe eines Übersetzers (Com-

piler genannt) in die Maschinensprache übersetzt und für die Anzeige der gespeicherten Zeile wieder mit einem Uncompiler zurückübersetzt wird. Deswegen sind die quitierten Zeilen oft einfacher als die eingetasteten. Ein aufmerksamer Benutzer kann daraus lernen, welche Abkürzungen erlaubt sind; z. B. kommt JMP (-5) als JMP -5 zurück, die Klammern sind hier also nicht nötig. Ferner erlaubt die Übersetzung jeder Zeile, dass einstellige Operationen (Funktionen:  $f(x)$ ,  $\sqrt{x}$ ,  $\sin x$  usw.) nicht nach der logischen Folge, zuerst  $x$  dann  $f$ , sondern nach der in der Mathematik üblichen Schreibweise  $f(x)$  (also z. B. SIN 30) eingegeben werden können.

Die Entwicklung der Sprache der HP20 bedeutet eine grosse Leistung der Konstrukteure. Dank dieser Sprache kann der Benutzer seine Aufmerksamkeit dem Problem zuwenden und erspart so viel Zeit bei der Ausarbeitung von anspruchsvollen Programmen. Das Aneignen dieser Sprache ist einfach und bildet eine gute Vorbereitung zum Erlernen höherer Programmiersprachen wie Fortran, Algol usw.

Adresse des Verfassers: Prof. Dr. H. Schilt, 2502 Biel, Höweg 5.

# Computer im Bauingenieurwesen

Von Dr. E. Anderheggen<sup>1)</sup>, Zürich

DK 681.3:624

## Einführung

Seit der Einführung von Computern gegen Mitte der fünfziger Jahre wurde in vielen Gebieten des Bauingenieurwesens eine grosse Zahl neuer numerischer Verfahren entwickelt. Besonders spektakulär waren die Erfolge auf dem Gebiet der Computerstatik, da es innerhalb weniger Jahre gelang, die meisten Probleme der Statik und der Kontinuumsmechanik in einer erstaunlich einheitlichen und theoretisch befriedigenden Art zu lösen. Dabei erweckte vor allem die Methode der Finiten Elemente weltweit Begeisterung.

Seit einiger Zeit jedoch, wenn man die technische Fachliteratur durchschaut oder wenn man sich mit Leuten trifft, die seit Jahren auf dem Gebiet der Computerstatik arbeiten, bekommt man immer deutlicher den Eindruck, dass eine solche optimistische Einstellung nicht ganz am Platz ist. Die Stimmung hat sich irgendwie geändert. Nachdem so viele Computerprogramme geschrieben worden sind, merkt man heute, wenn man zurückblickt, wie wenig rationell man vorgegangen ist und wie oft die Ziele übersehen wurden, für die man eigentlich arbeitete. Viel zu oft betrachtete man nämlich den Computer nur als Instrument zur Lösung von selten vorkommenden, besonders schwierigen Aufgaben, so dass die Probleme übersehen wurden, welche sich aus dem nicht immer leichten Zusammenleben von Computer, Programmierer und Programmbenutzer ergeben, und zwar vor allem dann, wenn der Computer als Werkzeug des projektierenden Ingenieurs für alltägliche Aufgaben eingesetzt werden soll.

In der Zukunft wird der Computer einen noch viel breiteren Kreis von Anwendungen finden, nicht zuletzt als Folge dieser drei wichtigen technologischen Entwicklungen:

Erstens das sogenannte «Time Sharing», womit es möglich ist, aus entfernten und einfachen Ein- und Ausgabegeräten (Konsolen) mit dem Computer eine Art Dialog zu führen, zweitens die erweiterten Möglichkeiten der graphischen Datenverarbeitung und drittens die sogenannten Computernetzwerke, bei denen eine Koppelung zwischen verschiedenen Grosscomputern verwirklicht wird.

Heute stehen wir folglich Rationalisierungsproblemen gegenüber, die im wesentlichen auch Wachstumsprobleme sind:

<sup>1)</sup> Antrittsvorlesung an der ETH Zürich vom 4. Dezember 1972.

Zuerst das Problem der Programmqualität und der Programmdokumentation. Soll ein Programm einen breiten Kreis von Benutzern finden, treten die Mann-Maschine-Kommunikationsprobleme ganz in den Vordergrund. Dafür ist aber oft wenig Verständnis vorhanden, nicht zuletzt, weil bei Handrechnungen sich solche Probleme gar nicht stellen: nur der eigentliche Algorithmus ist dabei wichtig. Unzählige Programme liegen deswegen in irgendwelchen Schubladen begraben, ohne die geringste Hoffnung, je wieder aufgegriffen zu werden.

Dann das Problem der ungenügenden praktischen Anwendbarkeit vieler Programme: Gegenüber den Mann-Maschine-Kommunikationsproblemen wurde den Algorithmen oft zu viel Aufmerksamkeit geschenkt, mit dem bedauerlichen Ergebnis, dass als Folge der Einführung von Computern Theorie und Praxis in manchen Fällen noch weiter auseinander geraten sind.

Ein weiteres Problem ist das der Duplikationen, und zwar sind hier in erster Linie nicht die Duplikationen ganzer Programme wichtig. Viel mehr ins Gewicht fällt, dass beim Programmieren oft solche Probleme am meisten Mühe bereiten, die unabhängig von der spezifischen Anwendung immer wieder vorkommen. Viel Arbeit könnte eingespart werden, wenn zwischen Programmierern aus verschiedenen Anwendungsgebieten eine bessere Koordination vorhanden wäre. Wir werden sehen, dass dies einer der Hauptgründe für die Entwicklung von «integrierten» Systemen gewesen ist.

Ein verwandtes Problem ist das der Maschinenabhängigkeit vieler Programme. Da die Programmentwickler die Erreichung einer maximalen Effizienz oft als eines ihrer Hauptziele betrachten, wird gern übersehen, dass die Programme einmal auf einem anderen Computer laufen sollen und dies, was heute zu einer oft unwesentlichen Verkürzung der Rechenzeit führt, morgen einen grossen und undankbaren Mehraufwand verursachen kann.

Im folgenden sei zuerst über zwei Projekte berichtet, die in diesem Zusammenhang wesentlich erscheinen.

## Das ICES-System

Anfangs der sechziger Jahre wurden am Massachusetts Institute of Technology zwei historisch wichtige Programme geschrieben. Das Programm COGO für Vermessungsaufgaben und das Programm STRESS zur Berechnung von elastischen