Angewandte Datenverarbeitung und Computertechnik als Bestandteile der modernen Ingenieurausbildung

Autor(en): **Vogel, J.**

Objekttyp: **Article**

Zeitschrift: **Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association Suisse des Electriciens, de l'Association des Entreprises électriques suisses**

Band (Jahr): **66 (1975)**

Heft 10

PDF erstellt am: **12.07.2024**

Persistenter Link: https://doi.org/10.5169/seals-915287

BULLETIN
des Schweizerischen Elektrotechnischen Vereins
de l'Association Suisse des Electriciens
des Verbandes Schweizerischer Elektrizitätswerke
de l'Union des Centrales Suisses d'Electricité

Elektrotechnik – Electrotechnique

Angewandte Datenverarbeitung und Computertechnik als Bestandteile der modernen Ingenieurausbildung [1])

Von J. Vogel

372.868.13.01

1. Einleitung

Absolventen der Abteilung für Elektrotechnik, die vor 15...20 Jahren ihr Studium an der ETH Zürich beendeten, verliessen die Hochschule in der Überzeugung, eine weitgehende Ausbildung in reiner und zumindest eine Basisschulung in angewandter Mathematik erhalten zu haben. Fast ausnahmslos waren diese angehenden Ingenieure nach acht Semestern der festen Ansicht, dass für ein mathematisch formulierbares Problem in erster Linie eine geschlossene, analytische Lösung gefunden werden müsse. Eine Näherungslösung unter Benutzung des Rechenschiebers durfte nur dann in Betracht gezogen werden, wenn man auf andere Weise nicht zum Ziel kam. Nur ein einziges Mal im Verlauf des damaligen Studienganges wurde jeweils am Institut für angewandte Mathematik eine kleine Demonstration an elektromechanischen Rechengeräten veranstaltet, und die Teilnehmer bekamen Gelegenheit, darüber zu staunen, dass man mit diesen Apparaten in der Lage war, Operationen wie z. B. das Wurzelziehen, auszuführen.

Dass an der ETH auch ein regelrechter Computer vorhanden war, wussten die Studierenden höchstens vom Hörensagen. Wie man sich dieser Wundermaschine näherte oder ihr gar irgendwelche Wünsche unterbreitete, davon erzählte niemand etwas.

So ergab es sich zwangsläufig, dass die Absolventen der Abteilung IIIB frühestens während einer Assistenten- oder Doktorandenzeit in Versuchung kamen, den Computer für elektrotechnische Probleme zu missbrauchen und dies auch nur dann, wenn es sich als unmöglich erwies, eine analytische Lösung zu finden. Nur Avantgardisten verlegten sich auf die in dieser Abteilung damals noch ziemlich abwegige Idee, die Verifikation von Ergebnissen, die durch Näherungsverfahren gefunden worden waren, auf dem Computer vorzunehmen. Da nur spärliche Kenntnisse über numerische Verfahren und überhaupt keine über die Programmierung existierten, waren diese Computerbenützer auf umfassende Hilfe der Mitarbeiter aus dem Institut für angewandte Mathematik angewiesen.

Die vom Rechner erzielten Resultate dienten vielfach nur zum Vergleich mit Näherungswerten, die auf manuelle Art erhalten worden waren.

In der Folge bemühten sich die Dozenten der Abteilung für Mathematik, auch den angehenden Ingenieuren Programmierkenntnisse beizubringen. Heute finden immer mehr Studenten und Mitarbeiter bei rechenintensiven Problemen den Zugang zum Digitalrechner, wenn auch die verwendeten Methoden und Programme nicht immer dem letzten Stand der Datenverarbeitungstechnik entsprechen.

2. Wann soll ein Ingenieur den Computer einsetzen?

Die Gelegenheiten für den Ingenieur, aus der Verwendung des Computers Vorteile für seine Arbeit zu ziehen, sind mannigfaltig.

2.1 Tischcomputer

Der problemloseste Einsatz des Computers ist derjenige der Anwendung als Tischrechner oder «Desk Calculator». Aufgaben, die man früher mühsam und oft in mehreren Schritten mittels Rechenschieber und elektromechanischer Rechengeräte behandelte, kann man heute von einem Terminal aus unter Verwendung einer geeigneten Sprache oder auf einem Kompaktrechner in Sekundenschnelle lösen.

2.2 Problemlösung

Im Bereich der Problemlösung wird der Ingenieur Abläufe, die meistens durch Gleichungen und Differentialgleichungen beschrieben sind, in irgendeiner höheren Computersprache programmieren. Der Computer verhilft dem Ingenieur in diesem Fall zumindest zu einer grossen Zeitersparnis, weil die Lösungen nicht auf eine andere, zeitaufwendigere Art, z. B. durch das Experiment, gefunden werden müssen. In den meisten Fällen stellt der Computer dank seiner Fähigkeit, vor-

[1]) Antrittsvorlesung, gehalten am 4. Juli 1974 an der ETHZ.

gegebene Schritte beliebig oft in extrem kurzer Zeit zu wiederholen, überhaupt das einzige Werkzeug dar, mit dem gewisse Probleme angepackt werden können.

2.3 Simulation

In manchen Fällen wird man sich des Rechners sowie eines passenden Modells bedienen, obwohl die gestellte Aufgabe prinzipiell auch im Laboratorium behandelt werden könnte. Die Gründe dafür können vielfältiger Natur sein:

a) Manchmal ist das physikalische System nicht verfügbar,
- weil der Zeitaufwand für dessen Bau zu gross wäre,
- weil es sich um eine Projektstudie handelt,
- oder wenn das Experimentieren am Originalsystem Gefahren für den Experimentator in sich birgt.

b) Vielfach ist eine analytische Behandlung des Problems nicht möglich, weil keine passende Rechenverfahren vorhanden sind, oder wenn komplexe stochastische Prozesse mitspielen. Die Simulation unter Verwendung von Zufallszahlengeneratoren bietet sich oft als einziges Hilfsmittel an.

c) Die Simulation auf dem Computer erlaubt es, im Zeitraffertempo zu arbeiten, wodurch vielfach die Kosten für eine Studie erheblich gesenkt werden können.

d) Die Untersuchung am Modell gestattet es, den Experimentierbereich zu vergrössern. Kritische Fälle können untersucht werden, ohne dass man ein Risiko für teure Elemente und Instrumente eingeht.

e) Die im Laboratorium auftretenden Messfehler infolge von Instrumenteneinflüssen können eliminiert und Störgrössen nach Belieben eingeführt oder unterdrückt werden.

f) Struktur- und Parameteränderungen sind an Simulationsmodellen viel einfacher vorzunehmen als am System selber, ausserdem kann mit Funktionen und idealen Elementen gearbeitet werden, unabhängig von deren späterer technischer Realisierung.

2.4 Datenspeicher

Der Einsatz des Rechners für Zwecke der Datenspeicherung darf nicht getrennt von dessen Benützung zur Problemlösung gesehen werden. Das Hauptproblem bei einer Reihe von Ingenieurwissenschaften, z. B. bei der Orts- und Regionalplanung oder Forst- und Landwirtschaft, liegt nicht unbedingt in der Lösung von komplexen mathematischen Aufgaben, sondern in der Erfassung, Klassifizierung, Organisation und Abspeicherung von grossen Datenmengen. Diese Daten müssen später auf einfache Weise wieder ergänzt, korrigiert und für Auswertungen herangezogen werden können. Der moderne Computer mit seinen grossen Speichermöglichkeiten offeriert sich hier als idealer Datenträger.

2.5 Real-time Datenerfassung und Prozeßsteuerung

Bei vielen Experimenten will der Ingenieur nicht über längere Zeit bei seinen Apparaturen stehen und Instrumente ablesen. Der Prozessrechner kann ihm die Erfassung der Messwerte abnehmen und auch gleich noch eine Auswertung dieser Resultate vornehmen. In einem weiteren Schritt können die Ergebnisse dann dazu gebraucht werden, um die Abläufe zu beeinflussen, d. h. den Prozess zu steuern.

2.6 Lernhilfsmittel

Der Computer kann heute auf effiziente Weise als Hilfsmittel im Bereich der Ausbildung und Weiterbildung gebraucht werden. Auf diesen sich im Aufschwung befindenden Zweig des Rechnereinsatzes für den computerunterstützten Unterricht wird in einem späteren Abschnitt näher eingegangen.

3. Was soll der Ingenieur lernen?

Damit der Ingenieur im Berufsleben den Computer in diesen vielfältigen Bereichen zweckmässig einsetzen kann, muss er bereits an der Hochschule eine entsprechende Ausbildung erhalten (Tabelle I).

a) Für den Einsatz des Computers als Tischrechner benötigt er die Kenntnis einer einfachen Sprache, die wenn möglich auch für die generellere Problemlösung eingesetzt werden kann, sowie Instruktionen darüber, wie ein Terminal bedient wird. Dies setzt voraus, dass Terminals in genügender Anzahl an den Instituten vorhanden sind.

Ein Beispiel für eine Sprache, welche sich für solche Labor-Rechenaufgaben eignet, ist APL, die sog. Iverson-Sprache. Wünscht ein Student 16 Zahlen in einer 4×4-Matrix anzuordnen und diese Matrix zu invertieren, so tippt er die aus Fig. 1 ersichtliche Instruktions-Zeile auf seinem Terminal ein und erhält in Sekundenschnelle die Antwort. Er braucht zu diesem Zweck also kein eigentliches Programm zu schreiben.

Für die entsprechende Ausbildung müsste man etwa einen halben Tag veranschlagen, während die Bedienung eines Kompaktrechners in 1...2 Stunden erlernbar ist.

b) Um den Computer für die Problemlösung einsetzen zu können, benötigt der angehende Ingenieur vor allem einmal eine gründliche Einführung in eine der gängigen *höheren Programmiersprachen* wie ALGOL, PASCAL, FORTRAN oder PL/1. Dabei darf sich diese Ausbildung nicht auf die reine Codiertechnik beschränken, sondern muss dem Studenten auch zeigen, wie er eine optimale Synthese zwischen der Formulierung seines Problems und den Möglichkeiten, die ihm die Sprache bietet, findet. Er muss lernen, sein Programm derart zu strukturieren, dass es übersichtlich ist, dass es leicht ergänzt oder verändert werden kann und dass bei Bedarf auch andere es benützen können. Er muss wissen, wie er die Richtigkeit seines Programms verifizieren kann, und schliesslich muss er lernen, sein Programm so zu dokumentieren, dass er ein Jahr später noch weiss, was sein Code bedeutet und unter Umständen auch die Nachwelt noch verstehen kann, was er mit dem Programm bezweckt hat.

Da in der Industrie diverse Programmiersprachen in Gebrauch sind, sollte sich die Ausbildung nicht nur auf die Basissprache beschränken, in der der Student seine ersten Gehversuche macht, sondern im Laufe des Studiums sollten sich die Kenntnisse auf 2...3 Sprachen erweitern, wobei eine Terminal-Steuersprache, wie z. B. Venus an der ETHZ oder TSO an der Universität Zürich, unbedingt mit dazu gehört. Dies besonders, weil in der Industrie diese Arbeitsweise von einer Station aus, die weit weg vom Rechenzentrum steht, immer stärkere Verbreitung findet.

Alle Hochschulrechenzentren besitzen heute grössere Sammlungen von erprobten *Subroutinen und Prozeduren*. Es ist reine Zeitverschwendung, und leider kommt dies heute noch häufig vor, wenn der

Ausbildung Tabelle I

Einsatzbereich	Erforderliche Kenntnisse
Tischrechner	Einfache Benützersprache Terminalbedienung
Problemlösung	Höhere Programmiersprache (ALGOL, PASCAL, FORTRAN, PL/1) Verwendung von Bibliotheks-Rechenroutinen Einsatz von anwendungsorientierter Software (Modularprogramme)
Simulation	Technik der Simulation Einsatz von Simulationsprogrammen
Datenspeicherung	Technik der Manipulation von Datenmengen
On-line-Datenerfassung und Prozess-Steuerung	Hardware, Software, Interfaces Arbeit am Prozessrechner
Lernhilfsmittel	Terminal- bzw. Konsolenbedienung

```
      A P L \ 3 6 0

      ⌹ 4 4  ⍴ 35 1 7 9 1 5 9 3 5 7 2 4 8 2 26 1

 0.004178272981  ¯0.1908077994   0.1197771588   0.05571030641
¯0.06115472271   ¯0.2981767536   0.3378070399   0.09369460623
¯0.000506457331   0.05343124842 ¯0.04482147379 ¯0.02355026589
  0.1020511522    0.7336034439  ¯0.4684730311  ¯0.2453785769
```

Fig. 1 Tischrechnermodus von APL (Iverson Sprache): Befehl zur Anordnung von 16 Zahlen in einer 4 × 4-Matrix und deren Inversion

Benützer solche bereits codierte Abläufe, z. B. für Fourieranalyse, Inversion, Regression usw. neu zu programmieren versucht. Es ist deshalb notwendig, in passender Weise die potentiellen Benützer der Rechenzentrumsbibliothek über die vorhandenen Routinen zu informieren, ihnen zu sagen, welche Methoden verwendet werden, welche Genauigkeit man erwarten kann, welche Vorsichtsmassnahmen am Platze sind und wie man diese Routinen optimal in Rahmenprogramme einsetzen kann.

Für den ganzen technischen Bereich existiert heute eine sehr umfangreiche, *anwendungsorientierte Software*. Es handelt sich um grössere, meist modular geschriebene Programmpakete, die einen möglichst breiten Applikationsbereich der betreffenden Ingenieurrichtung abdecken. Charakteristisch ist hier z. B. das Bauingenieurwesen mit Programmen wie STRESS, STAUB, METRO, SAUD, FIPS und ICES. Im Betriebsingenieurbereich interessiert man sich für Programmsysteme, mit welchen Probleme der Netzplantechnik, der Lagerhaltung, der Produktionsplanung und der Fertigungssteuerung behandelt werden können. In der Elektrotechnik ist eine Unmenge von Modularprogrammen für die Analyse und Synthese von analogen und digitalen Systemen vorhanden.

Die Zusammenstellung in Tabelle II zeigt, was allein im Bereich der Analyse von analogen elektrischen Schaltungen heute zur Verfügung steht. Es handelt sich dabei nur um eine Auswahl von allgemein verfügbaren Programmen, die von Industriefirmen, Computerherstellern oder Softwareunternehmen entwickelt worden sind, für die diversesten Computersysteme gebraucht werden können und ausnahmslos gut dokumentiert sind.

Wenn ein Ingenieur ein solches Programmpaket einsetzt, so verwendet er Makrobefehle wie ANALYZE, INVERT, SOLVE und INTEGRAL und erspart sich vor allem viel Detailprogrammierarbeit. Er muss sich nicht mit numerischen und programmiertechnischen Fragen herumschlagen und kann sich weitgehend auf seine Hauptaufgabe, die Lösung des vorliegenden Problems, konzentrie-

General information

Tabelle II

Programm Name	Type of analysis		Computer on which operating [1]	Time in use (Years)	Topological Limits	Documentation	Programm availability
	Linear	Nonlinear					
ANP 3	×		1, 2, 3, 4, 5, 7, 8	2	100 Branches 40 Nodes	Good	No Restr.
BELAC	×		4	5	None	Good	No Restr.
COMPACT	×		2, 6, 7	2	1 + 2 Port Topologies	Good	No Restr.
CORNAP	×		2, 4, 6, 7	6	30 Elements to 70 Elements	Good	No Restr.
LISA	×		7	5	125 Elements 50 Nodes	Good	No Restr.
MARTHA	×		7	3	2 Port Topologies	Good	No Restr.
PTNA	×		7	2	30 Elements	Good	No Restr.
SCAP	×		7	0,5	200 Nodes 500 Branches 100 Coupled Elements	Good	No Restr.
SNAP	×		2	3	25 Nodes 100 Branches	Good	No Restr.
ADA-74	×	×	4	8	None	Good	No Restr.
ASTAP	×	×	7	4	None	Good	No Restr.
DACSEP	×	×	2	0,5	300 Nodes	None	Restr.
CURE 2000	×	×	2	0,5	200 Elements	Good	Restr.
IMAG 2	×	×	4, 5, 7, 10	3	None	Good	No Restr.
R-CAP	×	×	9	2,5	–	Good	Restr.
CIRCUS-2	×	×	4, 7, 2	3	None	Good	No Restr.
NAP 2		×	7	1	200 Branches 50 Nodes or 1500 Branches 500 Nodes	Preliminary Users Manual	No Restr.
SCEPTRE	×	×	7, 4, 2	5	300 Nodes	Good	No Restr.
SYSCAP	×	×	2	4	255 Nodes	Good	No Restr.

[1]) *1* = Burroughs, *2* = CDC, *3* = ICL, *4* = GE/Honeywell, *5* = UNIVAC, *6* = PDP, *7* = IBM, *8* = RC4000, *9* = SPEKTRA, *10* = C.I.I.

ren. Aber die Benützung dieser Software will auch gelernt sein, und die Hochschule hat die Aufgabe, diese Kenntnisse in passender Weise zu vermitteln. In welcher Form dies geschehen kann, soll in einem nachfolgenden Abschnitt umrissen werden.

c) Was die Simulation anbelangt, so gilt für sie dasselbe wie für die anwendungsorientierten Programme. Es existiert eine grosse Anzahl von Simulationssystemen für die Behandlung von stochastischen und kontinuierlichen Prozessen.

In der Schweiz sind Programme wie SIMULA, SIMSCRIPT, GPSS, DYNAMO, CSMP und MIMIC im Gebrauch, deren zweckmässiger Einsatz nicht auf der Hand liegt, sondern gelernt werden muss.

d) Was die Organisation und Manipulation von grossen Datenmengen anbelangt, so haben sich an der ETH schon seit einigen Jahren Dozenten der Informatik dieser Aufgabe angenommen und vermitteln den Interessenten im Rahmen einer Vorlesung die notwendigen Kenntnisse.

e) Um mit der Datenerfassung am Experiment und der Regelung und Steuerung von Prozessen durch Computer vertraut zu werden, ist es unumgänglich, dass der Student in Tuchfühlung mit dem Rechner tritt. So trägt beispielsweise das Institut für Automatik der ETHZ diesem Bedürfnis Rechnung. Der Andrang der Studenten zu den entsprechenden Veranstaltungen zeigt, dass mit der Bereitstellung dieses Unterrichtsmittels der richtige Weg beschritten worden ist.

f) Was den computerunterstützten Unterricht anbelangt, so ist vom Benützer aus gesehen keine besondere Schulung notwendig. Die Ausbildung hat auf der Seite der Lehrenden, welche Kurse entwickeln, zu erfolgen.

4. Wer soll die Ausbildung geben?

Es wäre verlockend einfach, zu verlangen, dass die gesamte Ausbildung in allen auf den Computer ausgerichteten Gebieten vom Institut für Informatik gegeben werden sollte. Eine solche Forderung ist aber nicht realistisch, weil beispielsweise fünf Informatikdozenten der ETHZ schon personell gar nicht in der Lage sind, allen Ansprüchen einer Schule dieser Grösse zu genügen. Ausserdem würde dies auch bedingen, dass sich die Informatiker in die spezifischen Ingenieurfachbereiche einarbeiteten, um den Unterricht anwendungsorientiert zu geben. Wegen der Vielfalt der Anwendungsbereiche ist dies aber eine unzumutbare Forderung. Was die Ingenieurabteilungen von den Informatikern erhalten können, ist die Grundausbildung in

- Numerik und Programmierung,
- Behandlung grosser Datenmengen,
- Verwendung vorhandener Programme,
- Benützung von Terminals und zugehöriger Sprachen.

Eine Reihe von Gebieten kann je nach Verfügbarkeit und Interessengebiet der Dozenten entweder vom Institut für Informatik oder von der entsprechenden Ingenieur-Abteilung behandelt werden. Dies wären z. B. Kurse über

- den Aufbau und die Funktionsweise von Digitalrechnern,
- die Systemprogrammierung,
- zusätzliche Programmiersprachen ausser der Basissprache,
- Logik, Automatentheorie und
- Simulation.

Für den Unterricht in bezug auf den spezifischen Computereinsatz im Fachbereich muss aber die betreffende Abteilung selber verantwortlich sein. Dabei geht es vorwiegend um die Entwicklung und den Einsatz von anwendungsorientierter Software, den Einsatz von Computern am Experiment sowie um Vorlesungen über den Aufbau und die Funktionsweise von Computer-Hardware-Komponenten, letzteres in erster Linie für angehende Elektroingenieure. Ein Dozent aus der Fachabteilung ist hier am besten in der Lage, den Unterricht durch realistische, praxisnahe Beispiele, Übungen und Experimente zu untermauern.

Percent of Schools Teaching Courses Recommended for Computer Engineering Tabelle III

Course	Percent of Schools		
	Not taught at school	Taught at school outside EE Dept.	Taught in EE Dept.
Basic Subjects			
Introductory Computer Programming	2	63	35
Switching Theory and Logical Design I	2	2	96
Switching Theory and Logical Design II	20	2	78
Machine Structure and Machine Language Programming	10	35	55
Computer Organization	10	21	69
Systems Programming and Operating Systems	20	51	29
Recommended Electives			
Programming Languages and Translation I	20	58	22
Programming Languages and Translation II	34	51	15
Numerical Analysis	4	74	22
Logic and Automata Theory	27	23	50
Operations Research	18	71	11
Simulation and Modeling	19	35	46
Field Analysis	69	16	15
Other Courses			
Introduction to Discrete Structures	49	30	21
Computer Graphics	57	28	15

In den Vereinigten Staaten und Kanada wurde eine Untersuchung über den Computerunterricht an 150 Hochschulen, welche eine Abteilung für Elektrotechnik besitzen, durchgeführt und das Ergebnis kürzlich veröffentlicht. Aus Tabelle III ist die Aufteilung des Unterrichts zwischen dem Computer-Science-Department und der Abteilung für Elektrotechnik ersichtlich.

Die sechs wichtigsten Kurse für das sog. Computer Engineering, welches nachfolgend noch zur Sprache kommt, werden an 80 % aller Schulen gegeben. Dabei werden nur die Einführung in die Programmierung und die Vorlesungen über Systemprogrammierung vorwiegend von den Informatikern gelesen, während die mehr schaltungsorientierten Fächer an der Abteilung für Elektrotechnik gelehrt werden. Bei den empfohlenen Fächern ist die Fachabteilung vorwiegend für Automatentheorie und Simulation verantwortlich, während Numerik, zusätzliche Programmiersprachen und Operations Research meist zentral für die ganze Schule gelesen werden.

Interessant ist die Zusammenstellung über die Anzahl der computerorientierten Kurse, die an diesen Schulen gegeben werden (Tabelle IV). 48 % der Schulen offerieren mindestens fünf derartige Veranstaltungen, und 15 % bieten zehn oder mehr Kurse an. Die elektrotechnischen Abteilungen offerieren

Number of Computer courses Tabelle IV

Number	Percent (%) of Schools	
0	1	
1	10	
2	19	
3	12	
4	10	
5	11	
6	11	
7...8	8	
9...10	5	48
11...15	8	
16...20	2	
21...25	2	
25	1	

einen grossen Teil dieser Kurse auf der Graduate-Stufe, was unseren höchsten Semestern entspricht. Dabei besitzen nur 60 % der Schulen ein eigentliches Computer Science Department, während bei den übrigen 40 % die Fachabteilungen selber für den gesamten Computer-Unterricht verantwortlich sind.

5. Organisation der Ausbildung

Die nächste Frage, die sich stellt, ist: In welcher Form soll der Unterricht in anwendungsorientierter Datenverarbeitung an den Ingenieurabteilungen gegeben werden? Man kann hier fünf verschiedene Ansichten antreffen, mit mehr oder weniger gut fundierten Argumenten (Tabelle V).

a) Vorwiegend versierte Programmierer vertreten die Meinung, dass die Kenntnis einer Basissprache wie FORTRAN, ALGOL, PASCAL oder PL/1 genüge, um alle auftauchenden Probleme zu lösen, und dass man von Fall zu Fall die gerade interessierenden Abläufe selber programmieren solle. Diese auf das Problem zugeschnittenen Programme seien bestimmt effizienter als jedes Allzweck-Programm, und ausserdem müsse der Benützer nicht vorerst neue, höhere Programmiersprachen lernen. Es genügt nach dieser Ansicht vollkommen, wenn der angehende Ingenieur eine gründliche Ausbildung in angewandter Mathematik und Programmiertechnik erhält.

Der Nachteil dieser Methode dürfte ebenso klar sein. Der Ingenieur muss einen grossen Teil seiner Zeit programmiertechnischen Fragen widmen, er wiederholt Arbeiten, die schon vielfach vorher durchgeführt worden sind, und ausserdem wird sein Programm im allgemeinen später zu nichts anderem mehr nützlich sein.

b) Bei der zweiten Art des vorgeschlagenen Unterrichts wird versucht, die erwähnten Nachteile zu reduzieren. Im Zusammenhang mit der Behandlung von diversen Problemkreisen im Rahmen einer Vorlesung sollen jeweils Spezialprogramme entwickelt werden, welche der Behandlung eines bestimmten Problemtypus dienen. Auch hier liegt zwar wieder eine gewisse Duplikation von bereits bestehenden Programmen vor. Die Entwicklung von solchen Spezialprogrammen, die etwa 50 bis höchstens 300 Instruktionen umfassen sollten, lehrt den Studenten aber das anwendungsorientierte Programmieren und erlaubt es dem Dozenten, eine Reihe von Übungen, die eng auf den Vorlesungsstoff zugeschnitten sind, effizient zu lösen. Die Entwicklung solcher Programme kann auch im Rahmen von Studienarbeiten oder Dissertationen erfolgen, wobei diese Software im allgemeinen auch für andere Arbeiten des betreffenden Instituts anwendbar sein wird.

c) Nach der dritten Ansicht sollen im Rahmen des Unterrichts grössere Allzweck-Softwarepakete entwickelt werden. Diese Unterrichtsmethode gestattet es dem Dozenten, numerische, programmiertechnische, syntaktische und organisatorische Probleme anhand eines handfesten Programms zu lehren. Der Student lernt dabei auch, bereits bestehende Programmteile zu verwenden und optimal in das neue Programm zu integrieren.

Nachteilig bei dieser Methode ist ihre Zeitaufwendigkeit. Es ist schwierig, mit unerfahrenen Studenten in 1...2 Semestern ein anspruchsvolles Anwendungsprogramm zu entwickeln und auszutesten. Ausserdem kommt der Computereinsatz für die eigentliche Problemlösung auf diese Art zu kurz. Es dürfte deshalb eher angezeigt sein, solche Entwicklungen im Rahmen grösserer Forschungsprojekte auf Institutsebene auszuführen, wobei die Verantwortlichen bereits mit der EDV vertraute Mitarbeiter sein müssen. Noch besser können derartige Software-Projekte vorangetrieben werden, wenn mehrere Institute mit ähnlichen Interessen oder gar mehrere Hochschulen zusammenspannen.

Ein sehr erfolgreiches Beispiel haben in dieser Hinsicht etwa 20 Universitäten und 10 industrielle Laboratorien in den USA gegeben, welche gemeinsam das Netzwerkanalysenprogramm NASAP entwickelt haben. Die beteiligten Hochschulen benützen das Programm, das nach Wahl für diverse Computersysteme eingesetzt werden kann, für ganz verschiedenartige Applikationen.

d) Die vierte Variante geht davon aus, dass der Ingenieur sowenig als möglich selber programmieren sollte. Da es für sehr viele Problemkreise bereits fixfertige Softwarepakete gibt, genügt es nach dieser Meinung, zu wissen, welche verfügbaren Programme man für welche Fälle anwenden soll, welche Möglichkeiten die Programme dem Benützer offerieren und wie man die Programme praktisch verwendet. Im Unterricht sollten deshalb möglichst viele Softwarepakete behandelt und anhand von Übungen einexerziert werden. Dass bei dieser Unterrichtsart nicht viel Zeit für theoretische und methodologische Aspekte übrig bleibt, liegt auf der Hand. Ausserdem ist der auf diese Weise ausgebildete Ingenieur hilflos, wenn er einmal eine Aufgabe zu lösen hat, für die es eben doch keine fixfertige Software gibt.

e) Nach der letzten Ansicht soll der Unterricht von der eigentlichen Programmierung und vom Einsatz der verfügbaren Software völlig losgelöst werden. Es genüge, wird argumentiert, dass man den Studenten erzähle, welche numerischen Methoden bei der Lösung der fachspezifischen Probleme zu verwenden seien und ihnen Angaben darüber mache, wie diese Methoden in einem Anwendungsprogramm zu implementieren seien. Zusammen mit generellen Erläuterungen darüber, wie solche Programme aufgebaut und verwendet werden können, sollte diese Ausbildung den angehenden Ingenieur genügend auf alle zukünftigen Begegnungen mit dem Computer vorbereiten.

Die Erfahrung zeigt jedoch, dass man sich Illusionen hingibt, wenn man auf diese Art und Weise Unterricht betreiben will. Solange der Student die Angstschwelle, und eine solche Schwelle ist bei den meisten vorhanden, nicht überwunden hat, wird er jede Ausrede

Unterricht in anwendungsorientierter Datenverarbeitung Tabelle V

Nur Basissprache (ALGOL, FORTRAN, PASCAL, PL/I usw.) für Problemlösung verwenden	Aufgrund einer Basissprache Spezialprogramme zur Lösung gewisser Problemtypen entwickeln ↓ Spezialvorlesungen Studienarbeiten Dissertationen	Entwicklung grösserer «Allzweck»-Softwarepakete ↓ Forschungsprojekte auf Institutsebene, mehrere Institute oder Hochschulen	Verwendung bestehender Softwarepakete	Vermittlung der theoretischen und methodologischen Grundlagen, welche für die Entwicklung von Applikationssoftware nötig sind
			Synthese für Vorlesungen, Übungen, Praktika + Kurse in Codier- und Anwendungstechnik	

finden, um nicht an den Computer zu gehen. Deshalb sind Übungen am Rechner unter Verwendung von einfach zu handhabender Software unerlässlich.

Die optimale Lösung für den Unterricht dürfte eine Synthese der letzten beiden Varianten sein. Anhand eines allgemein zugänglichen, flexiblen Anwendungsprogramms, welches als Aufhänger dient, soll der Student im Rahmen einer Vorlesung die notwendigen theoretischen Grundlagen vermittelt erhalten. Ausserdem lernt er auf diese Weise zumindest eine höhere, anwendungsorientierte Computersprache kennen und im Rahmen von Übungen und Praktika einsetzen. Damit er sich auch Kenntnisse über die anderen verfügbaren Softwarepakete aneignet, müssen Kurse organisiert werden, in denen nur reine Codier- und Anwendungstechnik vermittelt wird. Diese Kurse können konzentriert jeweils an einigen Tagen zu Beginn oder am Ende der Semesterferien durchgeführt werden. Ganz ähnlich wie das an der ETHZ schon für die Sprachen ALGOL und FORTRAN der Fall ist, würden diese Kurse es den interessierten Studenten gestatten, bis zum Ende ihres Studiums mit einer ganzen Reihe von Programmpaketen aus ihrem Fachbereich vertraut zu werden.

6. Computerunterstützter Unterricht

Noch auf einem ganz anderen Gebiet beginnt der Computer für die Ingenieurausbildung eine Rolle zu spielen, nämlich im Bereich des computerunterstützten Unterrichts. Wenn ein Student in einer Vorlesung nicht alles mitbekommen oder eine Vorlesungsstunde verpasst hat, sollte er die Möglichkeit haben, den Stoff auf einfache Art repetieren oder nacharbeiten zu können.

Im Computer gespeicherte Lehrprogramme bzw. Repetitorien gestatten es dem Studenten, sich zu der von ihm gewünschten Zeit, mit der von ihm gewünschten Geschwindigkeit in eine Materie einzuarbeiten und bei Bedarf diesen Prozess auch beliebig zu wiederholen.

Auch die Einarbeitung in ausgewählte neue Gebiete ist ohne weiteres möglich. So wird an der Universität von Bradford die obligatorische Einführung in die Digital-Schaltungstechnik in Form eines computerunterstützten Kurses gegeben. Beim heutigen Mangel an Assistenten wäre auch zu überlegen, ob nicht ein Teil des Übungsbetriebes auf Computerterminals verlagert werden könnte. Der Student würde vom Lehrprogramm seine Übungsaufgabe abholen, bei auftauchenden Problemen gewisse Hilfsprogramme abrufen und schliesslich die von ihm gefundene Lösung mit der vom Computer vorbereiteten vergleichen können.

Auch der Tatsache, dass in Studenten- und Mitarbeiterkreisen vielfach die Kenntnisse über die in der Rechenzentrumsbibliothek vorhandenen Routinen und den in diesen verwendeten Methoden und Algorithmen nur spärlich verbreitet sind, könnte mittels computerunterstützter Unterweisung abgeholfen werden. Zu diesem Zweck müsste eine Methoden-Datenbank aufgebaut und ein Lehrprogramm geschrieben werden, mit dessen Hilfe jeder Interessent in einem Konversationsmodus aufgrund seiner in passender Weise eingegebenen Problemschilderung auf das für seine spezifische Aufgabenstellung beste Programm hingewiesen wird.

Das Schreiben solcher Lehr- oder Auskunftsprogramme ist zwar sehr zeitaufwendig, trotzdem dürfte diese Unterrichtsmethode auch in der Schweiz immer mehr zum Zuge kommen. Zwar wird bereits am Institut für Informatik und im Mikrowellenlabor der ETHZ auf diesem Gebiet Pionierarbeit geleistet. Wahrscheinlich sollte aber schon heute jede Abteilung der ETHZ einen vollamtlichen Mitarbeiter ausbilden, der sich zum Nutzen der ganzen Abteilung vollamtlich mit computerunterstütztem Unterricht befasst.

7. Fachrichtung Computer-Ingenieurwesen

Es wäre gar nicht abwegig, die Einführung einer neuen Unter-Abteilung für Computer-Ingenieurwesen zu prüfen. Die Hälfte der 150 amerikanischen Hochschulen, an denen die erwähnte Untersuchung durchgeführt wurde, offeriert eine Computer-Ingenieur-Ausbildung. Ein Einwand, den man vielleicht vorbringen möchte, ist, dass die Schweiz keine bekannten Computerhersteller besitzt und demzufolge nur Ingenieure für das Ausland ausbilden würde. Dieser Einwand ist nicht stichhaltig, denn

- es sind in der Schweiz bereits Industriefirmen vorhanden, welche sehr gute Kleinrechner herstellen,
- der Bedarf an Ingenieuren, die etwas von Computer-Hardware verstehen, um Interfaces zwischen Experimenten und Prozessrechnern zu bauen, wird immer grösser,
- der Computer-Anschluss peripherer Eingabe-, Ausgabe- und Speichergeräte, welche nicht von einem Computerhersteller geliefert werden, ist die Aufgabe eines Ingenieurs,
- die sich immer mehr verbreitende Verwendung von Computern für nicht rein rechnerische Zwecke, u. a. Daten-Kommunikation, Informations-Verarbeitung usw., macht es notwendig, dass sich vermehrt Elektroingenieure mit der Computer-Hardware auseinandersetzen,
- wenn in Zukunft noch verschiedenartige Computer zu Netzen zusammengehängt werden, damit jeder Benützer bei Bedarf auf die Funktion eines jeden Rechners zugreifen kann, werden auch wieder Ingenieure die Aufgabe haben, die Verbindungen zu konstruieren.

Eine solche neue Ingenieurabteilung als integraler Bestandteil der Abteilung für Elektrotechnik müsste nicht beim Nullpunkt beginnen, denn im neuen Studienplan existiert bereits eine ganze Reihe von Veranstaltungen, insbesondere zwei auf Computertechnik ausgerichtete Fachzweige, welche zur Ausbildung des zukünftigen Computeringenieurs gehören würden. So werden u. a. bereits Vorlesungen über Elemente digitaler

COSINE (Committee in Computer Science in Electrical Engineering)
Tabelle VI

Recommended subjects for undergraduate computer engineering 1971	Semester hours
General background	
General Physics	6...9
Calculus, Differential Equations	9...12
Linear and Abstract Algebra	3
Probability Theory	3
Electric and Electronic Circuits	9...15
Introductory Computer Programming	3
	33...45
Basic subjects	
Switching Theory and Logical Design	6
Machine Structure and Machine Language	3
Computer Organization	3
Systems Programming, Operating Systems	3
	15
Electives	
Programming Languages, Translation	6
Numerical Analysis	3
Logic and Automata Theory	3
Communications Systems	3
Operations Research	3
Simulation and Modeling	3
Field Analysis	3

Rechner, Prozessrechner, Impuls-, Schaltungs- und Vermittlungstechnik gehalten. Vielleicht müsste für den Bereich der Computer-Architektur ein zusätzlicher Dozent an die ETHZ berufen werden, doch dürfte dafür in den USA sicher ein erfahrener Fachmann gefunden werden.

TH Stuttgart-Elektronik Tabelle VII

– Datenverarbeitung
– Rechnergestützter Entwurf analoger Schaltungen
– Rechnergesteuerte Vermittlungssysteme
– Schaltwerk- und Automatentheorie I + II
– Prozessautomatisierung I + II
– Problemorientierte Sprachen
– Compiler I + II
– Digitale Speicher
– Systemprogramme moderner Rechenanlagen 1 + 2
– Numerische Feldberechnungsverfahren
– Impuls- und Digitaltechnik

TU Berlin-Kybernetik Tabelle VIII

– Algorithmen I + II
– Programmiersprachen
– Theorie analoger und hybrider Komponenten
– Ausgewählte Probleme der Semantik
– Semantik algorithmischer Sprachen
– Algorithmen 1 + 2
– Compilerbau 1 + 2
– Rechnerorganisation I + II
– Rechenwerke und Mikroprogrammierung
– Computer Graphics I + II
– Hardware-Softwaresysteme für Graphics
– Interpretierer Lisp
– Formale Sprachen
– Dokumentenretrieval
– Einführung in Betriebssysteme 1 + 2
– Entwurf kommerzieller On-line-Systeme
– Probleme der kommerziellen EDV
– Programmierung von Kleinrechnern
– EDV I + II
– Prozessdatenverarbeitung I — IV
– Fertigungsorientiertes Planspiel
– Rechnergestütztes Konstruieren I + II
– Automatisierung im Konstruktionsbereich
– Computergestütztes Lernen
– Regelungstechnik und Prozessautomatisierung
– Computer und Gesellschaft I + II
26 Veranstaltungen 21 Dozenten

In Nordamerika wurde bereits 1965 ein Komitee unter dem Namen COSINE gebildet, welches Empfehlungen für die Ausbildung von Computer-Ingenieuren im Rahmen der elektrotechnischen Abteilungen ausarbeiten sollte. Die entsprechenden Vorschläge für das Undergraduate-Studium sind aus Tabelle VI zu ersehen. Sie umfassen die Grundausbildung mit total etwa 40 Semesterstunden, die auch alle anderen Elektroingenieur-Studenten absolvieren müssen. Dazu kommen die Basisfächer für Computer-Ingenieure mit total 15 Semesterstunden und schliesslich eine Reihe von empfohlenen Wahlfächern. Diese Vorlesungen sollen etwa über zwei Jahre verteilt sein und durch entsprechende Übungen und Labor-Praktika untermauert werden. Die Empfehlungen des COSINE-Berichts umfassen auch detaillierte Angaben über den im Rahmen dieser Veranstaltungen zu vermittelnden Stoff.

Aus Tabelle III ist ersichtlich, dass bereits 1973 viele Schulen diesen Empfehlungen weitgehend gefolgt sind. Die zahlreichen Computer-Veranstaltungen, welche insbesondere die elektrotechnischen Abteilungen offerieren, zeigten deutlich das Gewicht, welches schon heute dem Computer-Ingenieur zugemessen wird.

Dass man auch in Europa auf diesem Gebiet nicht untätig geblieben ist, erhellt Tabelle VII, in dem die Computer-Vorlesungen der Technischen Hochschule Stuttgart zusammengestellt sind. Insgesamt 11 verschiedene Vorlesungen werden durch die elektrotechnische Abteilung dem angehenden Computeringenieur angeboten. Besonders interessant sind die Verhältnisse an der Technischen Universität Berlin (Tabelle VIII). Die Ausbildung der Computer-Ingenieure erfolgt dort durch die Abteilung für Kybernetik. Insgesamt werden 26 Veranstaltungen von 21 verschiedenen Dozenten offeriert. Dabei existieren sogar Kurse, wie z. B. die Prozessdatenverarbeitung, die sich über vier Semester erstrecken.

Vor 19 Jahren wurde an der ETH-Zürich bereits auf hervorragende Art Computer-Ingenieurwesen betrieben, als seinerzeit die ERMETH entwickelt wurde. Vielleicht wäre es an der Zeit, an diese Tradition anzuknüpfen.

Adresse des Autors:

Dr. *J. Vogel*, Abteilung für technisch-wissenschaftliche Computeranwendungen, IBM Schweiz, Zürich.