THE DEVELOPMENT OF A COMPUTERIZED MODEL FOR TEACHING ENGINEERING STATICS

A Dissertation
for the Degree of Ph. D.
MICHIGAN STATE UNIVERSITY
John William Johnson
1974

This is to certify that the

thesis entitled

THE DEVELOPMENT OF A COMPUTERIZED MODEL
FOR TEACHING ENGINEERING STATICS

presented by

JOHN WILLIAM JOHNSON

has been accepted towards fulfillment
of the requirements for

PHD. degree in HIGHER EDUCATION

George M. Van Dusen
Major professor

Date Oct. 16, 1974

O-7639

E-1418

that

rapic

analy

indiv

compu

a stu

stati

was g

opment

design

develo

course

to add

synthe

of the

ABSTRACT

THE DEVELOPMENT OF A COMPUTERIZED MODEL
FOR TEACHING ENGINEERING STATICS

By

John William Johnson

A means was sought in this study of demonstrating that a digital computer could be employed to make a rapid, thorough, and efficient check of a student's analysis and synthesis in problem-solving courses with individualized feedback.

It was the purpose of this study to develop a computer related instructional model capable of checking a student's analysis and synthesis of a broad range of statics problems on an individualized basis. The research was guided by the following objectives: (1) the development of the model would follow a systems analysis and design approach which would serve as a basis for the development of additional models for other problem-solving courses; (2) the student would be given the opportunity to address his attention solely to the analysis and synthesis portions of problem-solving by being relieved of the mechanics of calculations; (3) selection of the

computer

transpor

evaluate

The

educatio

in its c

an IBM 3

would ac

core ava

be used,

able for

required

and eithe

Extr

model to

and syste

(3) the p

(4) the p

statics.

The

statics p

was permi

for the s

impending

called th

computer and computer language would be based on maximum transportability of the model; (4) the model would be evaluated for proper operation and educational impact.

The model was developed so it could be used by any educational institution with a relatively low investment in its computer hardware system. It was developed on an IBM 360-22 computer, but any digital computer which would accept Basic Fortran IV and had 16K(decimal) of core available for the model plus system overhead could be used, with minor modifications, if means were available for overlaying subroutines. No terminals were required, any standard statics textbook could be used, and either the vector or scalar approach was permitted.

Extreme caution was taken in the development of the model to be certain that: (1) data capture was simple and systematic; (2) very little keypunching was required; (3) the program could easily be debugged by the student; (4) the program would not interfere with the learning of statics.

The model was a simulation of the solution of a statics problem. Any two- or three-dimensional problem was permitted if only one free-body diagram was required for the solution, except problems with friction at impending motion. The student entered pertinent data and called the appropriate subroutines to perform the required

calculati
gave the
in the fo
synthesis

It w
student w
and/or a
students
consiste
in an en
computer
populati
students
not com

St
instruc
pre-tes

En
(1) li
experi
use th
was ne
not in

culty

calculations. The computer then solved the problem and gave the student feedback (both positive and negative) in the form of diagnostics concerning his analysis and synthesis.

It was assumed that the model could be used by a student with no previous experience with a keypunch and/or a computer. Therefore, two populations of students were used in testing the model. One population consisted of eight sophomore engineering students, enrolled in an engineering statics class, who had completed a computer course in Fortran IV programming. The second population consisted of twenty-seven freshmen technology students, enrolled in a technology statics class, who had not completed a computer course in Fortran IV programming.

Students' attitudes toward the computer related instructional model were measured by administering a pre-test and post-test to both populations.

Engineering and technology students agreed that: (1) little keypunching was required; (2) no previous experience with a keypunch was necessary to effectively use the model; (3) previous experience with a computer was necessary or at least desirable; (4) the model did not interfere with the learning of statics.

The engineering students experienced little difficulty using the model, considered the diagnostics to be valuable,

the model

a valuabl

an effect

tematic i

not agree

items.

When

who taugh

were cert

from the

problems

students.

Addi

model has

friction

two or mo

experimen

an exper

ment effe

should b

area of

groups a

should b

achievem

valuable, found the program easy to debug, considered the model a good teaching tool, considered the model a valuable asset to learning, and considered the model an effective means of helping them to become more systematic in problem-solving. The technology students did not agree with the engineering students on any of these items.

When the experiment ended, both faculty members who taught the technology class emphasized that they were certain their students would have gained much more from the model if they had been required to turn in all problems assigned as was required of the engineering students.

Additional experiments should be conducted when the model has been further developed to include problems with friction at impending motion and problems which require two or more free-body diagrams for the solution. The experiments should be designed with a control group and an experimental group in such a way that the only treatment effect would be the use of the model. An instrument should be developed to measure achievement in the cognative area of learning which would be administered to both groups at the end of the experiments. The test results should be used to determine significant differences in achievement of the two groups.

A

sis. A

compare

using t

if the

from th

A further study should include a cost-benefit analysis. A cost analysis of the use of the model should be compared with increases in student achievement when using the model. The analysis would be used to determine if the added cost when using the model could be justified from the added benefits the students would realize.

Depa

THE DEVELOPMENT OF A COMPUTERIZED MODEL FOR TEACHING ENGINEERING STATICS

By

John William Johnson

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Administration and Higher Education

1974

Chapter

I

II

III

TABLE OF CONTENTS

Chapte

IV

V

TABLE OF CONTENTS

(continued)

Chapte

SELECT

GENERA

TABLE OF CONTENTS

(continued)

Table

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

LIST OF TABLES

Table
18
19
20
21

LIST OF TABLES

(continued)

Figur

1

2

3

4

LIST OF FIGURES

Appendi

A

B

LIST OF APPENDICES

many

of t

Sali

use

and

perf

payr

Anci

simp

func

can a

"auth

instr

CHAPTER I

THE PROBLEM

Background

The uses of the computer in support of education are many, and the field is rapidly expanding. Proliferation of terminology has been one result of this rapid growth. Salisbury has suggested three functional areas for the use of computers by educators: administrative; ancillary; and instructional. The administrative functions are those performed in direct support of the administrator such as payroll, record keeping, scheduling, and counseling. Ancillary applications are those in which the computer is simply a tool for problem-solving. The instructional functions serve the "learner" element of the system and can augment or replace the "materials," "monitor" and/or "author-teacher" elements. Salisbury further divides the instructional applications as follows:

a. Computer-Administered Instruction (CAI): A man-machine interaction in which the teaching function is accomplished by a computer system without intervention of a human instructor. Both training material and instructional logic are stored in computer memory. (Also referred to popularly as computer-assisted instruction.)

b. Computer-Supported Instruction (CSI): All computer applications in support of instruction in which the computer is used by a human instructor to assist him in the accomplishment

Comp

written i

tutorial;

and probl

defines d

> Use
> moni
> task
> a pr
> This
> vari
> unde

Salisbury

> The
> and
> mat
> stu
> gen
> rat
> dri
> tut
> com

The dial

defined

> Con
> est
> and
> his

---

[1]Ala
Agreemen
ber, 197

[2]Ala
Technolo

[3]Ibi

> of his instructional objectives; essentially all uses of the computer as a classroom training aid.[1]

Computer-Assisted Instruction (CAI) programs can be written in any of the following modes: drill and practice; tutorial; dialogue, conversational, or socratic; simulation; and problem-solving. The U. S. Continental Army Command defines drill and practice as:

> Use of the computer to guide, control and monitor by repetition a specific task or set of tasks. The purpose of this mode is to develop a predetermined level of proficiency in a skill. This proficiency may be changing under a wide variety of constantly changing conditions or under a single set of consistent conditions.[2]

Salisbury has described the tutorial mode as:

> The tutorial mode is more complex than the drill and practice mode in that more instructional material is presented and more sophisticated student responses are often called for. It is generally used for presenting original instruction rather than supplemental, as in the case of drill and practice. More than any other, the tutorial mode exemplifies the automation by computer of the programmed text.[3]

The dialogue, conversational, or socratic mode has been defined as:

> Conversational or socratic systems attempt to establish a two-way dialog between the student and the machine and allow the student to chart his own course through material made available

---

[1]Alan B. Salisbury, "Computers and Education: Toward Agreement on Terminology," Educational Technology, September, 1971, pp. 35-40.

[2]Alan B. Salisbury, "An Overview of CAI," Educational Technology, October, 1971, p. 48.

[3]Ibid., p. 48.

to

Wh

plays a

options

values

informa

the stu

Th

a mathe

algorit

the com

compute

CA

Gustave

to teac

1959, Do

of Illi

designe

that CA

tively,

and mor

traditi

---

[4]J.
Compute
Art," IE
Novembe

[5]La
Experime
cation C

to him by the computer.[4]

When the simulation mode is used, the computer displays an experiment of some real world situation with options for varying parameters. The student then specifies values of the parameters, and the computer processes the information and presents the results of the simulation to the student.

The problem-solving mode is used when the student has a mathematical problem to solve. The student writes an algorithm to solve the problem, stores the algorithm in the computer, and gives values of the variables. The computer then solves the problem and prints the solution.

CAI began in 1958 with the pioneering experiments of Gustave J. Rath and Nancy Anderson in which they attempted to teach binary arithmetic on an IBM 650 computer. In 1959, Donald L. Bitzer and his colleagues at the University of Illinois began to develop PLATO, a CAI system especially designed to meet the needs of instruction. It was thought that CAI might offer instruction more cheaply, more effectively, more patiently, and do so in a less regimented and more individualized way than instruction presented by traditional approaches.[5]

---

[4]J. A. Howard, P. F. Ordung, and R. C. Wood, "On-line Computer Systems for Engineering Education--State of the Art," IEEE Transactions on Education, Vol. E-14, No. 4, November, 1971, p. 210.

[5]Lawrence P. Grayson, "CAI: The Fifteen Million Dollar Experiment," Proceedings--Third Annual Frontiers in Education Conference, IEEE Cat. No. 73 CHO 720-3E, p. 357.

E
m
p
n
o
w
p
f
e
i
p
n

The fa

tance o

process

Ac

process

Diagnos

cause a

minatio

manner

systema

Sy

by techn

---

[6]Th

[7]E.
the Educ
ceedings
IEEE Cat

[8]A.
dent's A
Journal
1970, p.

[9]T.
Computer
and Bacon

> Enthusiasm for CAI quickly developed and remained high throughout the 1960's. Hundreds of programs were written by hundreds of authors at numerous centers in dozens of computer languages on a large variety of computers. Programs were written to teach a variety of subjects such as physics, chemistry, mathematics, engineering, foreign languages, psychology, statistics, economics, and many others. Sixteen years after its inception, however, the promise of CAI as a powerful and acceptable educational method had not been fulfilled.[6]

The failure of these early efforts to recognize the importance of linking computer applications to the learning process is well documented.[7,8]

According to Walker and Cotterman, "The learning process is, in large measure, the recognition of systems."[9] Diagnosis, simulation, decision-making, problem-solving, cause and effect, if-then, input-output, and the determination of how or why something would operate in a certain manner are all essential elements in the development of systematic problem-solving.

Systematic problem-solving is frequently encountered by technical students and includes analysis, synthesis,

---

[6]Ibid., p. 357.

[7]E. J. Anastasio and D. L. Alderman, "Evaluation of the Educational Effectiveness of PLATO and TICCIT," Proceedings--Third Annual Frontiers in Education Conference, IEEE Cat. No. 73 CHO 720-3E, p. 382.

[8]A. M. Mathis, T. Smith, and D. Hansen, "College Student's Attitudes Toward Computer-Assisted Instruction," Journal of Educational Psychology, Vol. 61, No. 1, February, 1970, p. 46.

[9]T. M. Walker and W. W. Cotterman, An Introduction to Computer Science and Algorithmic Processes, (Boston: Allyn and Bacon, 1970), p. 451.

and ca

one of

and te

ics an

rigid

troubl

I

purpos

free-b

forces

gram.

forces

to zer

T

are:

1

2

3

4

5

Steps

four i

F

seem t

libriu

experi

lems.

and calculations. Engineering mechanics is recognized as one of the core subject matter areas of most engineering and technology educational programs. The study of mechanics and particularly the application of problem-solving to rigid bodies at rest (statics) has proven to be extremely troublesome to students.

In the study of statics, a rigid body is selected for purposes of analysis, and a diagram or sketch, called a free-body diagram, is then drawn. All of the external forces and moments acting on the body are shown on the diagram. Equilibrium exists when the sum of the external forces and the sum of the external moments are both equal to zero.

The steps required to solve an equilibrium problem are:

1. read the problem
2. draw a free-body diagram
3. show all external forces and moments which act on the body
4. apply the principles of equilibrium
5. perform the necessary calculations

Steps one through three comprise the analysis, and step four is the synthesis of the problem.

From such a sketchy description of statics, it would seem that students would have little trouble solving equilibrium problems; however, students have traditionally experienced a great deal of difficulty working such problems. By far the greatest problem has been the drawing of a corr

and mo

was dr

from t

equili

studen

soluti

A

at a c

matel

a stud

know

most

the p

In ma

time-

ing t

consu

corre

or sy

orien

they

synth

in ca

witho

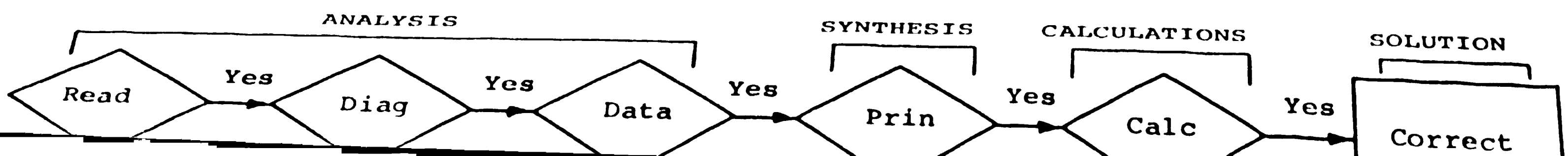
an es

the c

a correct free-body diagram showing all the external forces and moments acting on the body. If the free-body diagram was drawn incorrectly or if the correct data was not used from the free-body diagram when applying the principles of equilibrium, no amount of perserverance on the part of the student during the calculations would yield the correct solution to the problem.

According to Figure 1, there is only one way to arrive at a correct solution to a problem, but there are approximately twenty ways to arrive at an incorrect solution. If a student arrives at an incorrect solution, how does he know where to start checking for his error? Although the most desirable approach is to first check the analysis of the problem, most students first check their calculations. In many problems, performing the calculations is the most time-consuming part of the solution to the problem. Checking the calculations for the problem is still more time-consuming and does not help the student to arrive at the correct answer if an error had been made in the analysis or synthesis of the problem. Students have become so oriented to finding the correct answer to a problem that they have lost sight of the importance of the analysis and synthesis portions of problem-solving. Having expertise in calculations is commendable, but it is almost useless without a thorough understanding of analysis and synthesis.

The ability to perform calculations has always been an essential ingredient in problem-solving. However, when the calculations become lengthy and repetitive a great deal


ANALYSIS
SYNTHESIS
CALCULATIONS
SOLUTION
Read
Yes
Diag
Yes
Data
Yes
Prin
Yes
Calc
Yes
Correct

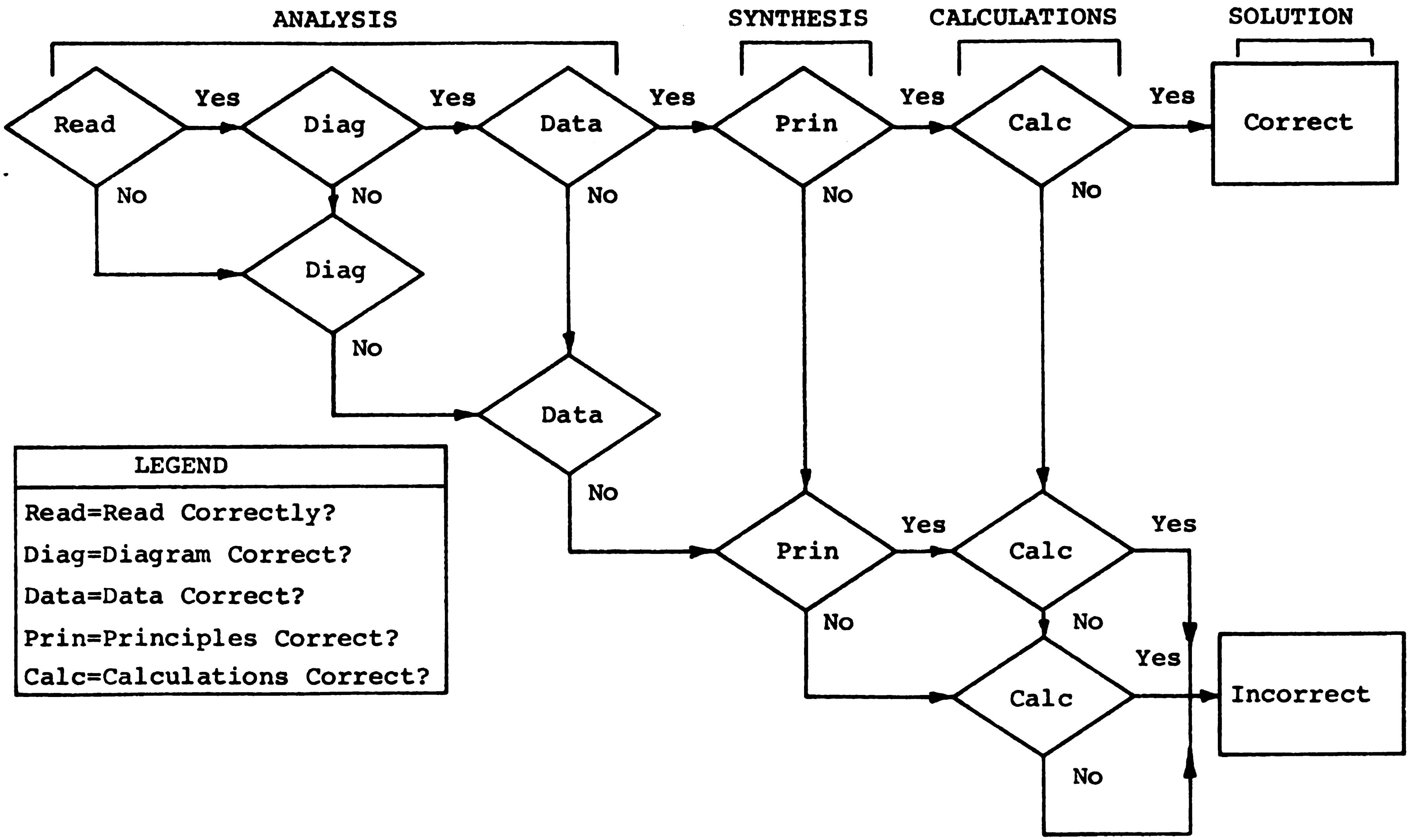


Figure 1. Flowchart of the Solution to a Problem

of time c

at consid

Rudh

stud
the

Acco

expe
is t
intu
ing
to t
the
the
lose
to t
the
an i
taug
able
prob
of r

Jame

nineteent

cert
when
What
mati
acqu
deal

---

[10] D
neering
November

[11] E
neering
Suitable
on Educa

[12] H
Nuemann,
p. 34.

of time can be wasted in performing routine calculations at considerable cost in learning.

Rudberg has stated:

> I think we have all found that the more the student is freed from the computational mechanics, the more insight he gains into the problem.[10]

According to Edwards:

> One of the main differences between the experienced professional and the fresh graduate is that the experienced man has acquired the intuitive ability to arrive at a workable engineering compromise. It is impossible (at the present) to teach intuition, so we must therefore convert the process to an analytic one if we are to close the gap. The "laborious calculation" approach loses the student among the trees of calculation to the extent that the forest goes unseen. Enter the computer. The student can now manually work an idealized example to acquire the concept being taught. With the aid of the computer he is then able to examine the possible solutions of real problems and thus acquire a better understanding of real system design.[11]

James Clerk Maxwell, a British physicist of the nineteenth century, said:

> The human mind is seldom satisfied, and is certainly never exercising its highest functions, when it is doing the work of a calculating machine. What the man of science, whether he is a mathematician or a physical inquirer, aims at is to acquire and develop clear ideas of the things he deals with.[12]

---

[10]D. A. Rudberg, "APL: A Natural Language for Engineering Education," IEEE Transactions on Education, November, 1971.

[11]E. M. Edwards, "APL: A Natural Language for Engineering Education Part II: The First Programming Language Suitable for Engineering Undergraduates," IEEE Transactions on Education, November, 1971, p. 179.

[12]H. H. Goldstine, The Computer from Pascal to von Nuemann, (Princeton: Princeton University Press, 1972), p. 34.

It is

more insig

clearer id

computer r

to relieve

to concent

It is

solving is

evidence t

into analy

computati

tation wi

are direc

analysis

will prov

ested in

The

velop a

checking

range o

followi

1.

It is essential that the students of the future gain more insight into the problems they must solve, and a clearer idea of the things with which they must deal. A computer related instructional model offers the potential to relieve the student of computational mechanics in order to concentrate on analysis and synthesis.

Statement of the Problem

It is generally recognized that analytical problem-solving is the basis for engineering analysis. There is evidence to indicate that students gain broader insights into analysis and synthesis when freed from detailed computational methods. There is also need for experimentation with computer related instructional models which are directly addressable to the learning process. An analysis and comparison of the method and the information will provide valuable input to engineering educators interested in applying innovative instructional techniques.

Purpose of the Study

The primary purpose of the proposed study is to develop a computer related instructional model capable of checking a student's analysis and synthesis of a broad range of statics problems on an individualized basis. The following objectives will guide the research:

1. The development of the model will follow a systems analysis and design approach which will serve as a basis for the development of additional models for other problem-solving courses.

2. The student will be given the opportunity to address his attention solely to the analysis and synthesis portions of problem-solving by being relieved of the mechanics of calculations.
3. Selection of the computer and computer language will be based on maximum transportability of the model.
4. The model will be evaluated for proper operation and educational impact.

The model will be evaluated at two levels. First, the basic computer program will be tested to verify that it will operate as specified. All computer calculations will have to function properly, the diagnostics will have to function properly when a student makes an error in analysis and/or synthesis, and the length of the program will have to meet the specifications.

Second, the model will be tested to obtain attitudinal information from two student populations who will use the model and from faculty members involved in the teaching. One student population will be composed of engineering students, and the other will be composed of technology students. A ten-item questionnaire for faculty members and a pre-test and post-test for the student populations will be used to obtain attitudinal information to determine the educational impact of the model.

In p
principle
student h
generally
would lik
errors.
Problems
reproduce
still sp

The
comprehe
of the p
whether
incorrec
analysis
calculat
step in

Fa
analysi
dents t
calcula
the ana
This co
transfo
would y
perform
howeve

Importance of the Study

In problem oriented courses, students are taught new principles and assigned problems for homework. If the student has anything with which to check his work, it is generally simply the answer to the problem. Many students would like someone to check their work and find their errors. This is seldom done because of the time required. Problems can be explained in class or solutions can be reproduced and given to the student, but the student must still spend time finding just where his error occurred.

The computer related instructional model will make a comprehensive check of the student's analysis and synthesis of the problem. The diagnostics will inform the student whether the solution is correct (positive feedback) or incorrect (negative feedback). If the student finds his analysis and synthesis are correct, he can perform the calculations for the problem and have a check for each step in the entire solution.

Faculty members are aware of the importance of analysis and synthesis. Many would like for their students to learn to set-up a problem without performing the calculations. Setting-up a problem consists of performing the analysis and synthesis portions of problem-solving. This consists of taking a problem written in words and transforming it into mathematical relationships which would yield a solution when appropriate calculations were performed. The faculty generally do not have the time, however, to analyze the solutions of all problems assigned for eac

check wh

and synt

Alt

portant

technolo

acceptan

Educatio

Des
ass
typ
imp
men
in
to

The

the affe

attitudi

Sys

related

among th

Inf

men, mac

of proce

a physic

Program

mation t

13Ana

for each student. They definitely do not have time to check what was right and what was wrong in the analysis and synthesis as well as the calculations.

Although cost and technical sophistication are important factors when considering the use of an innovative technology, the effects on achievement and educational acceptance are crucial. Anastasio and Alderman of the Educational Testing Service have noted:

> Despite substantial prior research in computer-assisted instruction, instructional systems typically lack detailed information regarding their impact upon the educational community. The development of delivery systems and course materials has, in most cases, proceeded without adequate attention to their educational effectiveness.[13]

The collection of detailed information concerning the affective domain of learning will provide important attitudinal information related to educational impact.

Operational Definitions

System describes any set of interacting and/or interrelated items, and the dynamic process of interaction among these system components.

Information system is the interrelationship between men, machines, operations, and documents for the purpose of processing or reporting information. It is a model of a physical system, and consists of data structures and program structures. The program structure is the information that defines the transformation to be performed

---

[13]Anastasio, p. 382.

on the i

related

form.

A d.

forms tr

An

solve a

procedur

A c

language

The

factors

system,

ships an

systems

Ana

whole in

Syn

elements

The

problem

definiti

systems

Tim

of a sir

Car

have bee

on the information, and the data structure is the problem-related information that the program structure is to transform.

A digital computer is a device which stores and performs transformations upon information systems.

An algorithm is a procedure which is utilized to solve a problem in a finite number of steps where the procedure consists of an ordered set of unambiguous rules.

A computer program is an algorithm coded in a computer language.

The systems approach takes into account all of the factors or interrelationships relevant to the subject, or system, under study. The recognition of critical relationships and significant variables is an important part of systems analysis and design.

Analysis consists of the separation of a substantial whole into constituents for individual study.

Synthesis consists of the combining of the separate elements to form a coherent whole.

The function of systems analysis and design includes problem recognition, the formulation of objectives, systems definition, analysis, design, planning, and control of systems implementation.

Time-sharing is the concurrent, effective utilization of a single computer by multiple users.

Canned subroutines refers to those subroutines which have been developed for a particular purpose and are stored c

needs th

The

is in su

referenc

An

cal loca

A r

instruct

will des

lations,

the data

be prese

study, a

for fur

Chapter

stored on the disk of a computer for use by anyone who needs them.

The output of a computer program is a hard copy if it is in such a form that a person can keep it for further reference.

An in-house computer is a computer at the same physical location as the user.

Overview

A review of literature relating to computer-assisted instruction will be presented in Chapter II. Chapter III will describe the design of the study including the populations, instrumentation, and a brief discussion of how the data will be analyzed. The analysis of the data will be presented in Chapter IV. Finally, the findings of the study, a discussion and recommendations, and implications for further development and research will be presented in Chapter V.

The

backgrou

tions to

Also inc

tion (NS

the futu

On-

with the

computer

operatio

tral com

control

memory,

system;

teaching

fying th

to be fo

provides

referred

managemen

CHAPTER II

REVIEW OF RELATED LITERATURE

The review of related literature includes a brief background of, problems associated with, and recommendations to improve Computer-Assisted Instruction (CAI). Also included is a discussion of a National Science Foundation (NSF) experiment now in progress which may determine the future of CAI.

Background of CAI

On-line computer-assisted instruction became feasible with the development of the high speed electronic digital computer in 1945 and the dataphone in 1958. The basic operational elements of an on-line CAI system are: a central computer which provides the executive communications control and which encompasses the logic, the rapid-access memory, and the main data-processing facility for the system; a computer software system for organizing various teaching, testing, and research strategies and for specifying the language in which directions to the computer are to be formulated; the individual student console which provides the "interface" between man and computer (also referred to as a student terminal or student station); management and other professional services in the computer based ed

telephon

between

nals.[14,

Com
wit
and
bir
Don
of
esp
It
che
in
ins

Ent
hig
wer
cen
var
tea
che
lan
man

CAI

which in

dialogue

problem-

observed

---

[14] G

[15] R
Approach
1970, pp

[16] D
Based Ed

[17] G

based education system; communication channels, such as telephone or microwave cables, which carry information between the computer and the individual student terminals.[14,15,16]

> Computer-assisted instruction (CAI) began in 1958 with the pioneering experiments of Gustave J. Rath and Nancy Anderson in which they attempted to teach binary arithmetic on an IBM 650 computer. In 1959, Donald L. Bitzer and his colleagues at the University of Illinois began to develop PLATO, a CAI system especially designed to meet the needs of instruction. It was thought that CAI might offer instruction more cheaply, more effectively, more patiently, and do so in a less regimented and more individualized way than instruction presented by traditional approaches.
>
> Enthusiasm for CAI quickly developed and remained high throughout the 1960's. Hundreds of programs were written by hundreds of authors at numerous centers in dozens of computer languages on a large variety of computers. Programs were written to teach a variety of subjects such as physics, chemistry, mathematics, engineering, foreign languages, psychology, statistics, economics, and many others.[17]

CAI programs may be written in any of several modes which include: drill and practice; inquiry; tutorial; dialogue, conversational, or socratic; simulation; and problem-solving. As late as 1971, however, Salisbury observed:

---

[14]Goldstine, p. 225.

[15]Richard T. Bueschel, "Time-Sharing - A Pragmatic Approach in the School," Educational Technology, March, 1970, pp. 21-23.

[16]D. Alpert and D. L. Bitzer, "Advances in Computer-Based Education," Science, March, 1970, p. 1586.

[17]Grayson, p. 357.

Mos
the
rea
sic
rep
so-
wer

Alt

improve

Grayson

Now
pro
edu
A m
mos
dra
mat
wer
hav

Com
a c
ins
the
the
nev
bee
sch
but
of

The

many fac

---

[18]A
Technolo

[19]Gr

[20]Ke
CAI (An C
Third Ann
No. 73 Ch

> Most of the CAI systems currently in use (beyond the experimental or developmental phases) are in reality little more than advanced automated versions of programmed instruction. As such they represent a logical evolutionary step beyond the so-called teaching machines, just as the machines were a step beyond the simple programmed text.[18]

Although many groups continued developmental work to improve the state-of-the-art of CAI in the early 70's, Grayson and Stetten concluded in 1973:

> Now, fifteen years since its inception, the promise of CAI as a powerful and acceptable educational method still awaits fulfillment. A majority of the early experiments are over; most of the commercial organizations have withdrawn from the field; the sales market has not materialized; the large number of students that were projected to have taken CAI courses by 1973 have not even seen a CAI demonstration.[19]

> Computer-Assisted Instruction (CAI) has been a commercial failure. It has failed despite instructional research that has demonstrated the effectiveness of CAI, and at a time when the problems of traditional instruction have never been more apparent. CAI systems have been offered by large companies and small, and school systems have never had larger budgets, but the dollars flow toward continuing support of traditional instruction.[20]

Problems Associated with CAI

The commercial failure of CAI has been attributed to many factors. Stetten lists: an initial oversell of its

---

[18]Alan B. Salisbury, "An Overview of CAI," Educational Technology, October, 1971, p. 48.

[19]Grayson, p. 357.

[20]Kenneth J. Stetten, "Toward a Market Success for CAI (An Overview of the TICCIT Program)," Proceedings--Third Annual Frontiers in Education Conference, IEEE Cat. No. 73 CHO 720-3E, p. 371.

capabili
and unre
to the i
ized str
to tens
individu

Alp
not capa
practica
Present
two and
nal.[22,2

Acc
programm
and impl
forty-ei
expects
and scie
of purpo

Wit
cost-ben
to docum

---

[21]S

[22]A

[23]H

[24]K
Educatio

capabilities; poorly authored educational content; expensive and unreliable hardware; an educational bureaucracy resistant to the intrusion of computers in the classroom; the decentralized structure of the American educational system that leads to tens of thousands of school systems, each having to be individually sold on the idea.[21]

Alpert claims that the technology of the 1960's was not capable of making a significant and economically practical contribution to the nation's educational program. Present CAI systems entail total costs which range between two and eleven dollars per student-contact hour at a terminal.[22,23]

According to Zinn, the current state of instructional programming languages is characterized by proliferation and implicit assumptions. He presents a list of some forty-eight different CAI languages, and states that he expects less progress toward standards than in business and scientific programming, because of the great variety of purpose and process in instructional programming.[24]

With the increased interest in accountability and cost-benefit analysis, it has become increasingly important to document the benefits received when substantial amounts

---

[21]Stetten, p. 371.

[22]Alpert, p. 1586.

[23]Howard, p. 216.

[24]Karl L. Zinn, "Instructional Programming Languages," Educational Technology, March, 1970, pp. 43-46.

of money a

have point

Despi
Assis
lack
upon
deliv
cases
thei

Othe

CAI's ran

use of co

student d

hours wri

Man

be done

of instr

Nev
fil
mo
re
ac

25

26
What NS
Educati

2

2
by an
Octobe

2

of money are spent on innovations. Anastasio and Alderman have pointed out:

> Despite substantial prior research in Computer-Assisted instruction, instructional systems typically lack detailed information regarding their impact upon the educational community. The development of delivery systems and course materials has, in most cases, proceeded without adequate attention to their educational effectiveness.[25]

Other factors include: an uncertainty concerning CAI's range of application;[26] many people thought that the use of computers in instruction dehumanized the teacher-student dialogue;[27] authors are required to spend too many hours writing CAI programs.[28,29]

Recommendations to Improve CAI

Many suggestions have been made regarding what must be done if CAI is to become an economically feasible means of instruction. According to Zinn:

> New techniques for preparation of curriculum files must be developed, techniques which are more powerful in the sense of fewer author hours required to write and revise materials which achieve the subject matter objectives intended.

---

[25]Anastasio, p. 382.

[26]Erik D. McWilliams, "The $15M CAI Experiment--What NSF Expects," Proceedings--Third Annual Frontiers in Education Conference, IEEE Cat. No. 73 CHO 720-3E, p. 357.

[27]Howard, p. 210.

[28]Herbert S. Diamond, "The Writing of a CAI Program by an Author New to Computers," Educational Technology, October, 1971, p. 42.

[29]Zinn, p. 44.

Auth
divi
in e

Dia

for each

a course

language

that aut

used.[32]

when dev

twenty s

materia

psycholo

ists, a

sold li

Zi

more as

that be

compute

ways.

used m

B

to pro

3

3

3

3

3

> Authors cannot often afford the luxury of individually shaping or tailoring each line of text in each frame for each kind of student.[30]

Diamond has suggested that the same format be used for each lesson when developing and writing programs for a course.[31] Alpert suggests the use of an easily learned language such as Tutor used for the PLATO IV system, and that authors receive royalties when their programs are used.[32] Stetten suggests a split of strategy and content when developing courseware. His group has identified twenty strategies or logics of instruction. The educational material (courseware) is developed by teams of instructional psychologists, subject matter specialists, media specialists, and programmers. The entire package would then be sold like textbooks.[33]

Zinn further advises that projects use the computer more as a learning tool than a presentation device, and that benefits are apt to be considerably greater when the computer does things which could not be achieved in other ways. Problem-solving and games and simulation should be used more.[34]

Boblick suggests that computer simulations be used to provide learning experiences which might not be available

---

[30]Zinn, p. 44.

[31]Diamond, p. 42.

[32]Alpert, p. 1589.

[33]Stetten, p. 372.

[34]Zinn, pp. 43-46.

to stud

cost or

factors

Bit

educatio

estimate

capital

dent hou

of less

the TICC

The
det
pre
wit
com
in
has
fro
the
ser
Mit
hav
TIC
min
and

[35] Jo
the Teac
Vol. 54,

[36] Do
Tenczar,
Annual Fr
73 CHO 72

[37] St

to students because of factors such as safety, equipment cost or availability, prohibitive set-up time, or other factors of cost or convenience.[35]

Bitzer emphasizes that the cost of computer-based education must become far lower than it has been, and estimates that when PLATO IV is fully implemented that capital and operating costs will be fifty cents per student hour at the terminal.[36] Stetten estimates a cost of less than one dollar per student contact hour with the TICCIT system.[37]

NSF Experiment

> The National Science Foundation in an effort to determine the current problems and opportunities presented by CAI is supporting a major experiment within the limited, but specific confines of the community college setting, and to a lesser extent in elementary schools. The University of Illinois has received $5 million of NSF funds, and $5 million from other sources, to complete the development and then test PLATO, which, in its present design, will serve up to 4096 terminals simultaneously. The Mitre Corporation and Brigham Young University jointly have received $4 million from NSF to develop and test TICCIT, a CAI system that will serve up to 128 terminals. The total experiment, which will last four-and-one-half years and will be completed in 1976,

---

[35] John M. Boblick, "The Use of Computer Simulations in the Teaching of High School Physics," Science Education, Vol. 54, No. 1, Jan-Mar, 1970, pp. 77-81.

[36] Donald L. Bitzer, Bruce Arne Sherwood, and Paul Tenczar, "PLATO: Everyone's Answer," Proceedings--Third Annual Frontiers in Education Conference, IEEE Cat. No. 73 CHO 720-3E, p. 360.

[37] Stetten, p. 371.

wil
und
com

The

PLATO IV

widely s

large sc

The
dis
the
qua
dis
com
jec
For
res
inc
wri
whi
ful
and
ho

Th

institu

hardwar

and col

voice-a

display

the use

sary co

termina

38

39

> will be evaluated by the Educational Testing Service under a $1 million grant from NSF. . . . Its outcome may very well determine the future of CAI.[38]

The two systems differ significantly in many respects. PLATO IV is designed as a computing utility to serve 4096 widely scattered student terminals simultaneously from a large scientific computer system.

> The heart of the student terminal is the plasma display panel, a flat sheet of glass upon which the computer can light up or turn off any of a quarter-million dots (in a 512 by 512 grid) to display text, graphs, and line drawings. The computer can select color photographs to be projected on the back of the transparent panel. For technical reasons, this display device represents a major advance over previous technology, including the cathode-ray tube. . . . Authors write their own materials in the TUTOR language which is powerful yet easy to learn. . . . When fully implemented it is estimated that capital and operating costs will be $0.50 per student hour at a terminal.[39]

The TICCIT system is designed to serve a single institution by using relatively inexpensive minicomputer hardware. Major innovations include: the use of audio and color TV displays in the student terminals to provide voice-accompanied multicolored alphanumeric and graphic displays (200 by 256 grid), as well as full-color movies; the use of a pair of minicomputers to provide the necessary computer power in a self-contained system of 128 terminals; the capability to deliver CAI and other socially

---

[38]Grayson, p. 357.

[39]Bitzer, p. 360.

relevan

a new a

high-qu

control

hardwar

than on

Th

Educati

Th
po
wh
so
me
CA
ex
im
st
ed
ev
ed
th
co

4

4

relevant computer services via cable television to homes; a new authoring system styled to support the production of high-quality CAI; a new and innovative use of "learner control" in CAI; a projected commercial cost including hardware, equipment maintenance, and CAI programs of less than one dollar per student contact hour.[40]

The PLATO and TICCIT systems will be evaluated by the Educational Testing Service.

> The scope of these demonstrations will make possible the collection of detailed information which reflects not only the cost and technical sophistication, but also the effects on achievement and educational acceptance. Thus the NSF CAI project extends beyond a developmental exercise to a study of instructional technology's impact upon the educational institution, upon students, teachers, and administrators. The educational component of the PLATO and TICCIT evaluations will focus upon the consumers of educational innovations in order to determine the practical benefits and problems accompanying computer-based education.[41]

---

[40]Stetten, p. 371.

[41]Anastasio, p. 382.

CHAPTER III

DESIGN OF THE STUDY

The Design of the Study is presented in two sections. Section I describes the development of the computer related instructional model and the testing required to make the model operational. Section II describes the collection and analysis of student and instructor attitudes concerning the model.

Section I: Development of the Model

A systems analysis and design approach was used in the development of the model to narrow the scope of the project. (The computer related instructional model will be referred to hereafter as the information system, which is consistent with the terminology used in systems analysis and design literature.) The approach also provides the potential for developing similar models for other problem-solving courses.

The systems analysis and design approach was a modification of that proposed by Walker and Cotterman[42] and was divided into the following parts:

---

[42]Walker, pp. 451-477.

1.

2.

3.

4.

5.

It i

solving i

evidence

into anal

computati

mentation

which are

An analys

tion will

intereste

1. Problem recognition
2. Feasibility study
    a. goals and objectives
    b. information system's blueprint
    c. major equipment decision
    d. implementation planning
3. Traditional information system redesign
    a. scope and objectives
    b. analysis
    c. specifications
    d. design
    e. implementation
4. Information system mechanization
5. Information system modification

Problem Recognition

It is generally recognized that analytical problem-solving is the basis for engineering analysis. There is evidence to indicate that students gain broader insights into analysis and synthesis when freed from detailed computational methods. There is also need for experimentation with computer related instructional models which are directly addressable to the learning process. An analysis and comparison of the method and the information will provide valuable input to engineering educators interested in applying innovative instructional techniques.

The

parts: 9

print; m

The

determi

an info

check c

statics

were:

1

Feasibility Study

The feasibility study was composed of the following parts: goals and objectives; information system's blueprint; major equipment decision; and implementation planning.

Goals and Objectives

The basic goal of the feasibility study was to determine the feasibility of designing and implementing an information system which would make a comprehensive check of a student's analysis and synthesis when solving statics problems. The objectives which guided the study were:

1. The development of the information system would follow a systems analysis and design approach which would serve as a basis for the development of additional systems for other problem-solving courses.
2. The student would be given the opportunity to address his attention solely to the analysis and synthesis portions of problem-solving by being relieved of the mechanics of calculations.
3. Selection of the computer and computer language would be based on maximum transportability of the system.
4. The system would be evaluated for proper operation and educational impact.

The

general c

for the s

steps wer

along wit

One

the infor

items wer

courses v

ments on

with any

taught w:

system wa

become th

scientif:

guages ha

system wa

be avail.

of compu

from very

signed fo

be availa

Many sta

because c

tions whe

are the s

Information System's Blueprint

The blueprint of the information system documented the general design envisioned and provided a general outline for the systems analysis and design. General processing steps were determined, and data structures were specified along with an estimate of the volume of each.

One of the primary purposes of the study was to make the information system as transportable as possible. Six items were considered. Since textbooks used in statics courses vary from campus to campus and even between departments on the same campus, the system was designed for use with any standard statics textbook. Statics classes are taught with either the vector or scalar approach, so the system was designed for use with either. Fortran IV has become the most universally used computer language for scientific calculations, but a large number of CAI languages have been developed for use with a terminal. The system was designed using a computer language which would be available at many educational institutions. The sizes of computer installations in educational institutions vary from very small to extremely large. The system was designed for use on a relatively small computer so it would be available to a large number of educational institutions. Many statics courses are taught primarily in two dimensions because of the great amount of time required for calculations when using three dimensions. Since the principles are the same for either case, the system was designed for

use with
system re
so a new
permittin
practical
of perfor
system wa
statics p

The
to read a
external
rectangul
designate
the axes,
necessary
finally t
on comput

Many
of the or
the direc
leaving t

The
The first
would per
types of
the syste
The third

use with either two- or three-dimensional problems. The system required no calculations on the part of the student, so a new dimension could be added to statics courses by permitting the student to analyze and synthesize many practical three-dimensional problems without the necessity of performing lengthy and repetitive calculations. The system was also designed for use with a wide variety of statics problems.

The person using the information system was required to read a problem, draw a free-body diagram showing all external forces and moments acting on the body, show a rectangular coordinate system on the free-body diagram, designate the location of the origin and the direction of the axes, take the data from the free-body diagram necessary to arrive at a solution to the problem, and finally tabulate the data in suitable form to be punched on computer cards.

Many textbooks have problems showing the location of the origin of the rectangular coordinate system and the directions of the axes, so the system was designed leaving these items to the discretion of the student.

The information system was developed in three stages. The first stage was to develop an information system which would perform the necessary functions to solve the desired types of problems. The second stage consisted of modifying the system so that it could easily be debugged by students. The third stage consisted of modifying the system so it would mak

ysis and

through t

Stage I

The

followin

couples,

lations

the resu

point ar

forces a

moments

equatio

known c

Da

lines

the da

and ar

type c

Stage

stud

perf

The

to

would make a comprehensive check of the student's analysis and synthesis of the problem. Each stage was taken through the complete analysis and design process.

Stage I

The information system was required to perform the following functions: read data for forces, moments or couples, points, and lines; perform the necessary calculations to transform the data into vector form and store the results; calculate the moment of a force about a point and store the result; calculate the sum of the forces and store the result; calculate the sum of the moments and store the result; fill a matrix with the equations of equilibrium; solve the matrix for the unknown quantities; present the solution.

Data for forces, moments or couples, points, and lines could be given in various ways. Table 1 shows what the data was for, the types of data that were permitted, and an estimate of the elements of data required for each type of data.

Stage II

Several write statements were included to show the student just where in the program the execution was being performed in case the program was prematurely terminated. The data was read in and then written out for the student to check whether the data had been entered correctly.

E

Data for

Concentra
Distribut
Moments c
Points
Lines

Stage II

Seve

check th

The

Campus w

System I

Com

Lan

Loc

System I

Com

Lar

Loc

System I

Con

Lar

Loc

TABLE 1

ESTIMATES OF THE VOLUMES OF DATA REQUIRED

Data for	Types	Elements of data required for each type
Concentrated forces	6	7
Distributed forces	4	9
Moments or couples	3	3
Points	1	3
Lines	2	6

Stage III

Several diagnostics were built into the system to check the student's analysis and synthesis of the problem.

Major Equipment Decision

The computer facilities available at the Fort Wayne Campus were very good and included the following:

System I:

Computer--IBM 360-22

Language--Basic Fortran IV

Location--Fort Wayne, Indiana (In-house)

System II:

Computer--CDC 6500

Language--Fortran IV

Location--Purdue University, West Lafayette, Indiana

System III:

Computer--CDC 6600

Language--Fortran IV

Location--Indiana University, Bloomington, Indiana

System IV

Comp

Lang

Loca

System V

Com

Lar

Loc

System

Co

La

Lo

System

C

L

I

T

in al

excep

Fortr

many

to pr

CAI s

which

Sever

mathe

erati

System IV:

Computer--CDC 6500 (The PLATO system)

Language--Tutor

Location--University of Illinois, Champaign, Illinois

System V:

Computer--CDC 6500

Language--PLANIT

Location--Purdue University, West Lafayette, Indiana

System VI:

Computer--CDC 6600

Language--PLANIT

Location--Indiana University, Bloomington, Indiana

System VII

Computer--CDC 6600

Language--APL

Location--Indiana University, Bloomington, Indiana

The investigator has had experience writing programs in all of the computer languages of the seven systems except the tutor language for the PLATO system. Basic Fortran IV and Fortran IV have been used when writing many programs requiring scientific calculations, programs to process student records, and programs to plot output. CAI simulation programs concerning heat loss calculations, which required a mathematical model, were written in APL. Several CAI simulation programs, which did not require a mathematical model, concerning trouble-shooting of refrigeration systems were written in PLANIT. In the programs,

the stude
finding t
eration s
union app
trouble-s
an exciti
technicia

The
the adva
when deve
universa
people i
subrouti
many cal
cards ca
into th
compute
cards a

A
but th
main p
routin
progra
be en
termi
debu
prob
term

the student was permitted to practice his reasoning in finding the cause of a given malfunction in a refrigeration system. Instructors in the local steam fitters union apprentice program were very impressed with the trouble-shooting programs. They felt the programs added an exciting new dimension to the training of refrigeration technicians.

The following are the investigators views concerning the advantages and disadvantages of the computer languages when developing an information system. Fortran IV is the universal language for scientific calculations, and most people in the scientific field are familiar with it. subroutines can be called from the main program to perform many calculations. When a program is being developed, cards can be punched on a keypunch and the program fed into the computer. The program is debugged from the computer print-out, and corrections are made by repunching cards and feeding the program back through the computer.

APL is a completely different language from Fortran IV, but the programs are similar to Fortran programs since a main program and subroutines can be written with the subroutines called from the main program as needed. When the program is being developed, however, all instructions must be entered at a terminal, and the program debugged from the terminal. Although the APL program can be punched in, debugged, and tested for proper operation immediately, a problem can exist if it is difficult to gain access to a terminal.

PLA
writing
This can
can be p
computer
debugged
is a dis
Tut
very eas
the PLAT
an expen
system b
graphics
The
the deve
of this
PLATO sy
terminal
and acce
in the e
to use i
as the o
The
informat
accessab
the depa

PLANIT is another completely different language, and writing a program consists of writing a series of frames. This can be very awkward and time-consuming. The cards can be punched on a keypunch, the program fed into the computer without a terminal, and then the program can be debugged from the terminal or by repunching cards. This is a distinct advantage if terminal access is a problem.

Tutor, used with the PLATO system, is claimed to be very easy to learn. However, the greatest advantage to the PLATO system is its graphics capabilities. It requires an expensive special terminal, however, and the information system being developed may not require the outstanding graphics capabilities.

The use of PLATO, APL, or PLANIT posed a problem for the development and operation of the information system of this study. Only one terminal was available for the PLATO system, and it was used almost continuously. Two terminals were available for the APL and PLANIT systems, and access to them was limited. Also, the APL system was in the experimental stage, and it would have been impossible to use it for the development of an information system such as the one proposed in this study.

The selection of the computer system to use for the information system was based on five items: capabilities; accessability; transportability; hard copy; the cost to the department for the use of the system. The questions considere

Capa

1.

2.

Acce

1.

Tra

1.

2.

3.

Ha

1.

2

considered under each item were:

Capabilities:

1. Did the computer have the required core?
2. Did the computer have the capability to perform all of the transformations of information required?

Accessability at the Fort Wayne Campus:

1. Could the student gain access to the system without wasting a lot of time on such things as trying to gain access to a keypunch, waiting his turn for a terminal, and turn around time on the computer?

Transportability:

1. Would the computer system be the size that most educational institutions would have available?
2. Would the computer language be fairly universal?
3. Would the computer language be such that most students and faculty members would be familiar with it?

Hard Copy:

1. Would the output of the system be such that the student could take it with him for future reference?
2. If the output were a hard copy, would it be in a neat, systematic form?

Cost

1.

2.

Tabl

system on

received

informati

for use o

A va

implement

Stage I

1.

2.

3.

4.

Stage II

1.

Stage II

1.

2.

3.

4.

Cost to the department at the Fort Wayne Campus:

1. Would the department be charged anything for the use of the system?
2. What charges would be made to the department?

Table 2 shows how the investigator rated each computer system on the various items. The IBM 360-22 computer received the highest rating, so it was decided that the information system would be written in Basic Fortran IV for use on the IBM 360-22 computer.

Implementation Planning

A variety of major activities were necessary for implementation of the system which are listed by stages.

Stage I

1. detailed design of each subsystem
2. algorithm development and testing
3. installation of the information system
4. review and evaluation of the system

Stage II

1. modification of algorithms and testing

Stage III

1. detailed design of subsystems
2. algorithm development and testing
3. review and evaluation of the system
4. preparation of written procedures for faculty and students

TABLE 2

RATINGS OF COMPUTER SYSTEMS

SYSTEM	COMPUTER AND LANGUAGE	CAPABILITIES	ACCESSIBILITY	HARD COPY	DEPARTMENT CHARGED	TRANSPORTABILITY
I	IBM 360-22 Basic Fortran IV	Very good	Excellent	Yes	No	Excellent
II	DCD 6500 Fortran IV	Excellent	Good	Yes	Yes	Good
III	CDC 6600 Fortran IV	Excellent	Good	Yes	Yes	Good
IV	CDC 6500 Tutor(PLATO)	Excellent	Very poor	No	No	Very poor
V	CDC 6500 PLANIT	Good	Poor	Yes	Yes	Fair
VI	CDC 6600 PLANIT	Bad	Poor	Yes	Yes	Fair
VII	CDC 6600 APL	Bad	Poor	Yes	Yes	Fair

5. selection and training of faculty members whose students would use the system
6. training of the students who would use the system

Traditional Information System Redesign

The Traditional Information System Redesign included scope and objectives, analysis, specifications, design, and implementation.

Scope and Objectives

The feasibility study was concerned with the design of a broad overall system. At this point, the objectives were amplified into specific objectives as follows:

Stage I

1. The student would be given the opportunity to perform the analysis and synthesis of many problems which heretofore were impossible or very difficult due to lengthy calculations.
2. The system would consist of a very simple main computer program consisting of call statements for canned subroutines.
3. The student would gain experience in calling canned subroutines.
4. The system would keep the student involved with the computer.
5. The system could be used with any computer which accepted Basic Fortran IV and had a

maximum core available for the program of 16K(decimal). The system would thus be very transportable.

6. The student would be required to address his attention only to the analysis and synthesis portions of problem-solving.
7. A wide variety of statics problems could be used with the system.
8. Very little key-punching would be required of the student.
9. The system could be used by a person with little or no previous experience with a computer.
10. The system could be used by a person with little or no previous experience with a key-punch.
11. Students using vector or scalar statics could use the system.
12. Use of the system would not interfere with the learning of statics.
13. No calculations would be required on the part of the student.

Stage II

14. The system could be easily debugged by the student.

Stage III

15. The system would make a comprehensive check of the student's analysis and synthesis of each problem with no faculty time required.
16. The system would visually bring out points concerning simultaneous equations with several variables which would ordinarily only be discussed in a class or in a textbook.
17. The system would provide feedback to the student on an individualized basis.
18. The system would give several valuable diagnostic messages.
19. The system would be usable as a teaching tool.
20. The system would be a valuable asset to the student in the learning process.
21. The system would teach students to become more systematic in problem-solving.

Analysis

The analysis of the information system included a detailed determination of the nature of the existing information system used to check a student's analysis and synthesis of statics problems.

There were very few methods available to make a comprehensive check of a student's analysis and synthesis in problem-solving courses, and each had severe drawbacks. Probably the best method was for the instructor to collect homework and check each problem carefully and thoroughly. The method, however, required an enormous amount of the instructor's time.

A second method involved a discussion of each problem in class. Although the method can be very effective in teaching students how to analyze and synthesize problems, it required a large amount of class time, and the student still had to find just where he made an error. A third method consisted of duplicating the solutions to all problems and giving them to the students. Again, the student was required to check to find just where he made an error. The last two methods created a potential hazard, since some students would not even attempt to work problems if they knew they would be discussed in class or that the solutions would be handed to them.

In each of the above methods, the student was required to complete the calculations for each problem even though they were repetitive and very time-consuming.

The existing information system was, therefore, to be replaced by a completely new and innovative type of system. Walker and Cotterman have described such an approach as a clinical information system redesign. One of the most extreme approaches advocated the elimination of the analysis step. A more moderate variety, rep

sys

st

<u>St</u>

im

sy

th

re

da

fl

st

da

ar

represented by the ideal designs of effective and logical systems concept, reduced but did not eliminate the analysis step.[43]

Stage I

For the present study, the analysis was to play an important role in the development of the information system. The analysis was to include a description of the equations to be used for calculations, the data required, the calculations required to transform the data into the required form, the development of a function flowchart illustrating the proper sequence of the basic steps involved in the process, and a description of the data structures as noted on the function flowchart.

The vector equations of equilibrium used in statics are:

$$\sum \bar{F}=0 \qquad \sum \bar{M}=\bar{0}$$

The scalar equations of equilibrium are:

$$\sum F_x=0 \qquad \sum F_y=0 \qquad \sum F_z=0$$
$$\sum M_x=0 \qquad \sum M_y=0 \qquad \sum M_z=0$$

---

[43]Walker, pp. 472-475.

re

or
or
ti

u
da
sy
bu
Th
ty

di
mo
in
A
re

fa
fo
as
wa
ti
the

The summations include all external active and reactive forces and moments acting on the body.

The system was to be designed for use with any two- or three-dimensional statics problem which required only one free-body diagram. However, problems involving friction at impending motion were excluded.

Data was required for concentrated forces, distributed forces, moments or couples, lines, and points. This data could be given in a variety of ways. The information system was to be designed to be as general as possible, but the capacity of the computer imposed limitations. Therefore, it was necessary to make a decision as to the types of data the system would accept.

Six types of concentrated forces, four types of distributed forces, two types of lines, two types of moments or couples, and one point were selected as shown in the user's manual, Appendix B, pages 113 through 125. A number for identification, a diagram, and the data required are shown.

A point along the line of action of each concentrated force was selected by the student so the moments of the forces could be calculated. Distributed forces were given as a function in the form $F(x)=Ax^2+Bx+C+Dx^{.5}$, and each was replaced by an equivalent force system. The calculations determined the equivalent concentrated force and the point where the force intersected the axis. This

point was used to find the moment of the force about the given point.

All that was required of the information system at this point was that it should read data, perform the necessary calculations, and print out the solution to the problem. Figure 2 shows the function flowchart which indicates the data structures required. A description of the data structures is presented in Table 3.

Stage II

When a long computer program is written with many calculations, it is very difficult to debug the program if an error has been made in calculations. Such a situation can easily be remedied by inserting write statements after each calculation. In this manner, not only can the accuracy of the calculations be checked, but the point of termination can easily be located if the program is interrupted prematurely.

The same problem could exist with the information system of this study. Therefore, several write statements were included in the system so the student could easily locate the point in the program where termination occurred.

Stage III

When checking the analysis and synthesis of a problem, several items had to be considered:

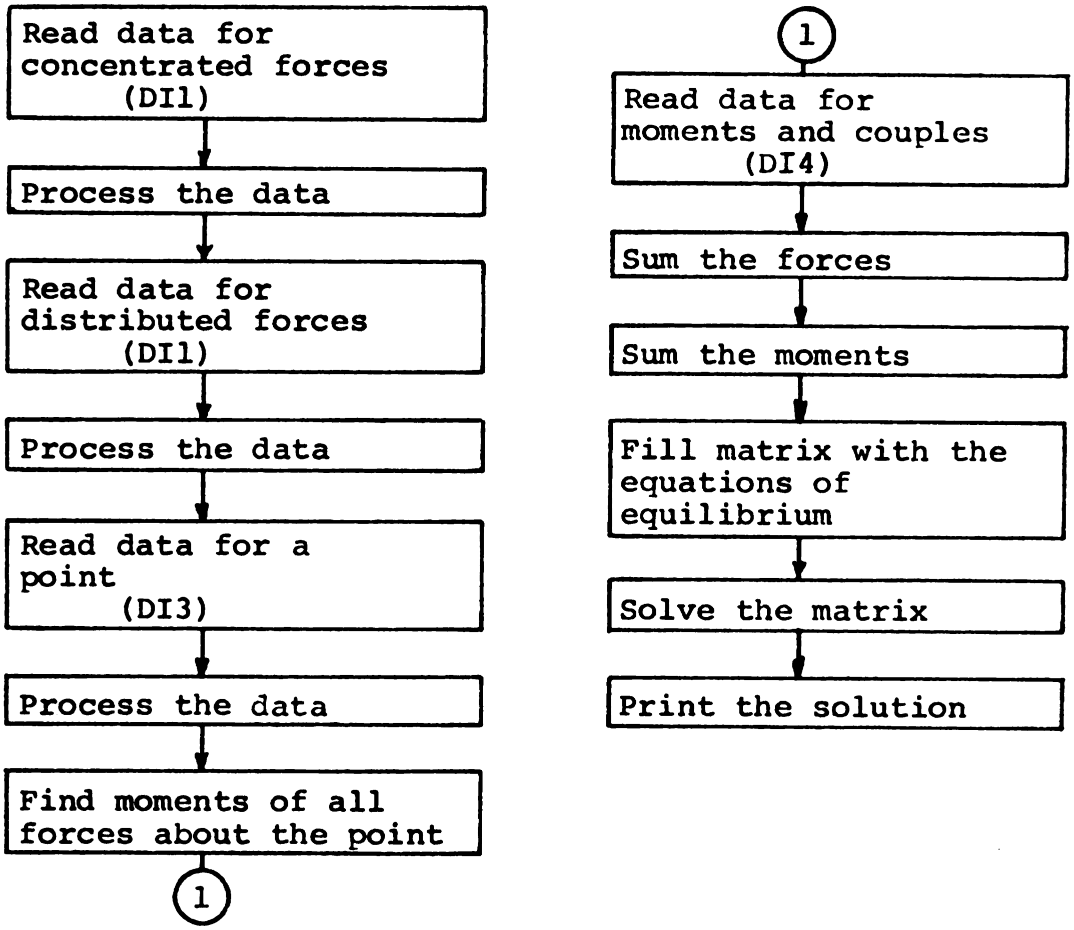


Figure 2. Function Flowchart for Stage I of the Analysis

TABLE 3

DATA STRUCTURES NOTED IN THE FUNCTION FLOWCHART

FOR STAGE I OF THE ANALYSIS

Data Structure	Data for	Content	Media
DI1	Concentrated forces	Alphanumeric	Computer card
DI2	Distributed forces	Alphanumeric	Computer card
DI3	Point	Alphanumeric	Computer card
DI4	Moments and couples	Alphanumeric	Computer card

Analysis:

1. Was the free-body diagram drawn correctly? (Were all active and reactive forces and moments shown correctly?)
2. Had the correct data been taken from the free-body to transform the forces and moments into vector form?
3. Had all forces and moments been included in the calculations?

Synthesis:

4. Had the correct method been used in calculations to transform the forces and moments into vector form?
5. Had the correct method been used to find the moments of the forces about a point?
6. Had the correct equations of equilibrium been used?

Calculations:

7. Were all calculations correct?
8. Was the answer correct?

Since all calculations were performed by the computer, only the analysis and synthesis were to be checked. The computer could not see the free-body diagram, so item 1 could not be checked. The computer calculated the moments of the forces about a point, so only items 2, 3, 4, and 5 were to be checked.

The primary purpose of this study was to force the student to go back and check his analysis and synthesis if his answer was incorrect. Therefore, the student was given the answer to the problems by the instructor, and the information system checked to see if the correct number of active and reactive forces and moments were included, and if the correct equations of equilibrium had been used. If the student received no error messages from the program, but his answer was incorrect, then he had either taken the data from the free-body diagram incorrectly or he had entered the data in the computer incorrectly. Thus, if an error existed, the student was forced to go back to the analysis of the problem and check for errors.

Specifications

Specifications were determined as needed in the development of the system.

Stage I

In the equations of equilibrium noted in the analysis, the summations referred to active and reactive forces and moments. The active forces and moments were known or fixed, but the reactive forces and moments were unknown or variables. This presented a problem when using the computer program, since variable quantities could not be fed directly into the computer. It was necessary, therefore, to revise the equations of equilibrium as follows:

$$\sum\bar{F}_{active} + \sum\bar{F}_{reactive} = \bar{0}$$
$$\sum\bar{M}_{active} + \sum\bar{M}_{reactive} = \bar{0}$$

$$\sum F_{x_{active}} + \sum F_{x_{reactive}} = 0 \; ; \sum M_{x_{active}} + \sum M_{x_{reactive}} = 0$$
$$\sum F_{y_{active}} + \sum F_{y_{reactive}} = 0 \; ; \sum M_{y_{active}} + \sum M_{y_{reactive}} = 0$$
$$\sum F_{z_{active}} + \sum F_{z_{reactive}} = 0 \; ; \sum M_{z_{active}} + \sum M_{z_{reactive}} = 0$$

The active forces and moments were fed directly into the computer, but the reactive forces and moments were fed in with magnitudes equal to one. The data was then transformed into vector form and the results placed in a matrix, the solution of which determined the values of the unknowns.

When the data for active forces and moments or couples was read, suitable calculations were performed to transform the data into vector form or scalar components. The results of the calculations were stored for future use. When the point was read, it was placed directly in storage. Separate storage space was required for forces, moments or couples, and points and lines.

When data for distributed forces was read, the type of calculations required was determined by the values given for the coefficients of the function. The results of the calculations were placed in the same storage space as the concentrated forces.

t
T
t

s
a

m
w

a
a
f
t

p
o
s

o
w
w
g
tr
nu
in
re

After all active forces and the point were stored, the moment of each force about the point was calculated. These moments were stored in the same storage space as the active moments.

When all active forces and moments were stored, the sum of the forces and the sum of the moments were calculated and stored.

Next, the storage spaces for active forces and moments were zeroed out, since the same storage space was used for reactive forces and moments.

Reading of and calculations for the reactive forces and moments were the same as that noted above for the active forces and moments. The moments of all the reactive forces were then calculated, but the sum of the forces and the sum of the moments were not calculated.

A matrix was then filled with the rectangular components of the reactive forces and moments, and the sums of the active forces and moments. The matrix was then solved for the unknown quantities.

The data required for concentrated forces, moments or couples, and points and lines included a name which was alphanumeric, a number to indicate the type which was integer, and from three to seven elements of data given which were numeric-real. The data required for distributed forces included a name which was alphanumeric, a number which described the location of the axes which was integer, and nine elements of data given which were numeric-real.

r

r

t

c

c

t

t

c

(S

e

c

t

i

w

co

eq

The storage spaces were limited to six forces, six moments or couples, and five points or lines.

Stage II

The first items printed for each problem were the chapter number and problem number. When each subroutine was called, the name of the subroutine was printed so that the student would know that the call statement was executed. If data was read in the subroutine, a write statement explained what the data was for and then printed the data as read. The student could then check the printout to see if the data was what he intended it to be. This was a good check to see if the data had been punched in the correct columns on the data card. When calculations were performed, the results were printed only after an explanation of what the results represented. If arrays were printed, the names of the arrays were printed first.

Stage III

The number of active and reactive forces and moments entered by the student were counted. These numbers were checked against the correct numbers entered by the instructor. The student was told how many of each he entered, and if he was correct. If he was incorrect, the correct number was given.

The type of a force system could be coplanar, coplanar-concurrent, three-dimensional, etc. The equations of equilibrium available were determined by the type of force system for any particular problem. The type of force system a student entered was checked against the correct type for the problem which was entered by the instructor. The student was told the type of force system he used, the equations of equilibrium he used, and whether or not he was correct. If he was incorrect, the correct type of force system was given.

Design

It was imperative that the data be tabulated in as simple and systematic form as possible. Therefore, the data was tabulated by concentrated forces, moments or couples, points and lines, and distributed forces. Tables 4 through 7 show the method established for the tabulation of data along with the designation used for each. Each designation was selected so that the name very nearly described the type of data being tabulated.

Since it was very important that the initial data capture be as simple as possible for the student, Tables 4 through 7 were placed on one sheet of paper and given to the student so that he would not have to keep looking through several pages of instructions to find them. Table 8 shows the sheet which was given to the students. The table shows the number to select for each type of data as well as the data required for each.

It was also necessary to provide the student with an easy method of tabulation. All the data for each type of force, moment or couple, point, or line was

TABLE 4

TABULATION OF DATA

FOR CONCENTRATED FORCES, DESIGNATED:

FORCES: CONCENTRATED

NAME	NO	F1	F2	F3	F4	F5	F6	F7
	1	F	l	m	n	x	y	z
	2	F	x_1	y_1	z_1	x_2	y_2	z_2
	3	F	x_s	y_s	z_s	x	y	z
	4	F	F_x	F_y	F_z	x	y	z
	5	F		l	m	x	y	
	6		F_x	F_y	F_z	x	y	z

TABLE 5

TABULATION OF DATA

FOR POINTS AND LINES, DESIGNATED:

PTLINE

Name	NO	F1	F2	F3	F4	F5	F6	F7
	1		l	m	n	x	y	z
	2		x_1	y_1	z_1	x_2	y_2	z_2
	3					x	y	z

TABLE 6

TABULATION OF DATA

FOR DISTRIBUTED FORCES, DESIGNATED:

DLOAD

NAME	NO	A	B	C	D	E	G	R	R_1	R_2

TABLE 7

TABULATION OF DATA

FOR MOMENTS AND COUPLES, DESIGNATED

COUPLE

NAME	No	F1	F2	F3	F4	F5	F6	F7
	1	C	l	m	n			
	2		C_x	C_y	C_z			
	3							

TABLE 8

SUMMARY OF DATA

FORCES: CONCENTRATED

NAME	NO	F1	F2	F3	F4	F5	F6	F7		
	1	F	l	m	n	x	y	z		
	2	F	x_1	y_1	z_1	x_2	y_2	z_2		
	3	F	x_s	y_s	z_s	x	y	z		
	4	F	F_x	F_y	F_z	x	y	z		
	5	F		l	m	x	y			
	6		F_x	F_y	F_z	x	y	z		

PTLINE:

NAME	NO	F1	F2	F3	F4	F5	F6	F7		
	1		l	m	n	x	y	z		
	2		x_1	y_1	z_1	x_2	y_2	z_2		
	3					x	y	z		

COUPLE

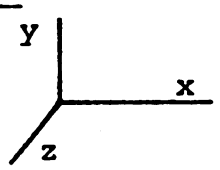
NAME	NO	F1	F2	F3	F4	F5	F6	F7		
	1	C	l	m	n					
	2		C_x	C_y	C_z					

DLOAD:

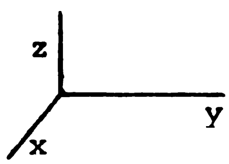
NAME	NO	A	B	C	D	E	G	R	R1	R2

FOR DLOAD ONLY:

NO=1



NO=2



included on one computer card. Table 9 shows the sheet given to each student for the tabulation of data. The table was divided into columns with a description at the top of each which matched those in Table 8. As noted in the specification, column 1 was to be alpha-numeric; column 2, numeric-integer; and columns 3-11, numeric-real. At the bottom of the table was listed the columns on the computer card which corresponded to the columns in the table along with a note that columns 3-11 had to have a decimal point.

This development made it possible for the student to look at the free-body diagram to select the data, and look at one sheet to see how the particular data was to be obtained for tabulation on the second sheet.

The next step was to punch the data on computer cards. It is very discouraging to have to keep record of exactly what column data is being punched in, so a card was punched for the drum of the keypunch and given to each student. Columns 1-4 were for alpha-numeric for the name, and after punching the name the skip key was struck which tabulated to column 8 where the number was punched as integer. When the number was punched, the card was in column 9 ready for data. The numeric key was not needed for this number or any further data on the card. The numbers were struck as seen on the keyboard, but the decimal point had to be used which was on the top row of the keyboard instead

TABLE 9

TABULATION OF DATA

NAME	NO	F1	F2	F3	F4	F5	F6	F7		
		A	B	C	D	E	G	R	R1	R2

1 45 89 1617 2425 3233 4041 4849 5657 6465 7273 80

|← DECIMAL POINT →|

of the one at the bottom. All numbers in columns 3-11 were left-justified and had to include the decimal point. When the student wanted another column, he merely hit the skip key which tabulated to the next column for another piece of data. All data was punched to be read with the F format.

Stage I

It was next necessary to specify algorithms which would read the data and perform the necessary calculations. Because of the limitation imposed by the size of the computer, subroutines were required to perform the functions. The process was broken into steps with the use of subroutines, the names of which described, as nearly as possible, the function being performed.

When data was read for concentrated forces, moments or couples, and points and lines, the name was entered as SAM, the number to describe the type as NO, and the data describing the particular type was entered in a one-dimensional array F(I) with seven elements. Several write statements were included to assist in checking the calculations performed by the subroutines during the developmental stage.

The following is a list of the subroutines which gives the name of the subroutine as well as the functions which they performed.

Subroutine FORCES read the data for concentrated forces and performed the calculations necessary to transform the data into vector form. The results were stored in two arrays called FNAM and FORC. FNAM was a one-dimensional array with six elements for the names of the forces, and FORC was a 6x6 two-dimensional array for storing the rectangular components of the force and a point the forced passed through. After all the concentrated forces were stored, the arrays FNAM and FORC were printed.

Subroutine COUPLE read the data for moments and couples and performed the calculations necessary to transform the data into vector form. The results were stored in two arrays called SMNAM and SMOM. SMNAM was a one-dimensional array with six elements for the names of the moments and couples, and SMOM was a 6x3 two-dimensional array for storing the rectangular components of the moments and couples. After all the moments and couples were stored, the arrays SMNAM and SMOM were printed.

Subroutine PTLINE read data for points and lines and performed the necessary calculations. The results were stored in two arrays called PTNAM and PTLN. PTNAM was a one-dimensional array with five elements for the names, and PTLN was a 5x6 two-dimensional array for the data. After all points and lines were read, the arrays PTNAM and PTLN were printed.

Subroutine DLOAD, which stood for distributed load, read the data for distributed forces or loads, and performed the necessary calculations to transform the data into vector form. The results were stored in arrays FNAM and FORC with the concentrated forces. After all distributed forces had been stored, the arrays FNAM and FORC were printed giving only the information for the distributed forces.

Subroutine MOMPT calculated the moment of each force in array FORC about the point in array PTLN and stored the results in arrays SMNAM and SMOM with the moments which had been read. The subroutine then printed the moment of each force about the point.

Subroutine SUMFOR summed all of the forces in array FORC and stored the results in a 1x3 one-dimensional array SM. The number of forces and the sum of the forces were then printed.

Subroutine SUMMOM summed all of the moments in array SMOM and stored the results in a 1x3 one-dimensional array SM. The number of moments and the sum of the moments were then printed.

Subroutine ZFORC zeroed out array FORC and blanked out array FNAM.

Subroutine ZSMOM zeroed out array SMOM and blanked out array SMNAM.

Subroutine ZPTLN zeroed out array PTLN and blanked out array PTNAM.

Subroutine ZARR zeroed out array ARR.

The solution of each problem required more than one subroutine. The first in the sequence was called SOLVE which filled a matrix with the equations of equilibrium. The array was called ARR and was a 6x7 two-dimensional array.

Subroutine EQUA presented the equations of equilibrium in equation form, the same form the student would arrive at if he had worked the problem himself. All terms which were zero were eliminated.

Subroutine MFGRR was the only subroutine in the information system which was not developed by the investigator. The solution of the 6x7 matrix required a method which was as efficient as possible to eliminate computer errors. The subroutine was included in a scientific package furnished by IBM for the IBM 360-22 computer. The subroutine performed the following calculations on the rectangular 6x7 array ARR:

1. It determined rank and linearly independent rows and columns of the matrix.
2. It expressed a submatrix of maximal rank as the product of triangular factors.
3. It expressed nonbasic rows in terms of basic ones.
4. It expressed basic variables in terms of free ones.

The rank was determined by the standard Gaussian elimination technique with complete pivoting. This implied that the rows and columns of the 6x7 matrix were interchanged at each elimination step if necessary. The interchange information was recorded in two integer permutation vectors IROW and ICOL. The results were returned in matrix ARR.

There were many possibilities of outcomes when the 6x7 matrix was solved. Therefore, subroutine DUMPIT was written so that all of the information from the solution in subroutine MFGRR could be written out to be certain the system was operating properly. It was very helpful in determining what had to be done in the next subroutine, FINISH, which determined which situation existed and printed the results.

Subroutine FINISH determined which possibility existed from the output of subroutine MFGRR. Figure 3 shows a flowchart of the possible outcomes when the 6x7 matrix was solved, which was used in the development of subroutine FINISH.

If there were the same number of equations as unknowns, the subroutine printed the solution to the problem. If there were linearly dependent equations, the subroutine showed the relationships among the equations. If there were more unknowns than equations, the subroutine showed some variables in terms of free variables.

There were six possible error messages which were given as ERROR I. An explanation of each follows:

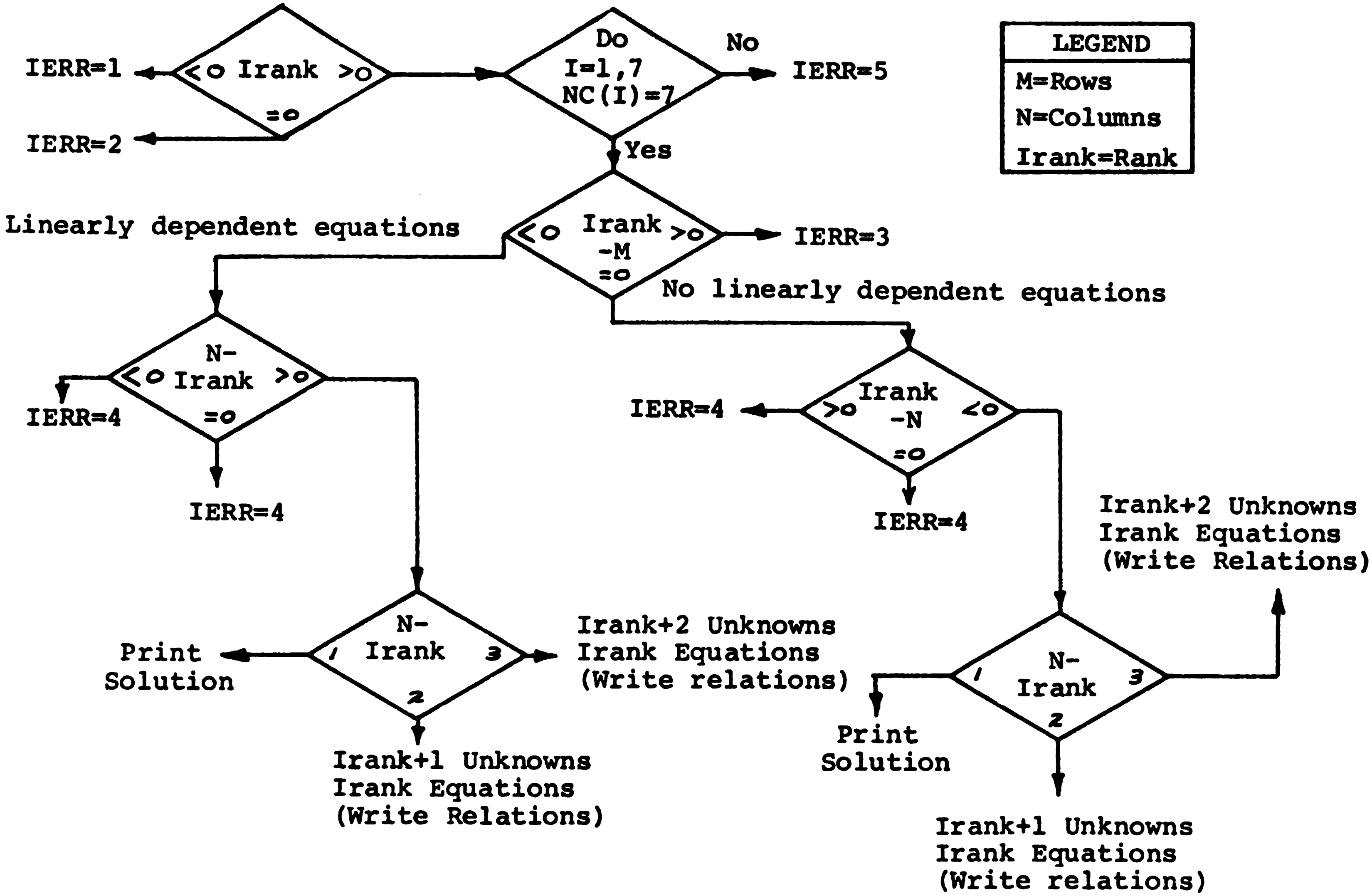


Figure 3. Flowchart of All Possibilities When Subroutine MFGRR Solves the Matrix

1. No rows or columns in the matrix.
2. The zero matrix.
3. The number of equations is less than the rank of the matrix.
4. More equations than unknowns.
5. All sums are zero for the active forces and moments. Therefore, the body was in equilibrium without any reactive forces or moments.
6. Inconsistent set of equations.

Figure 4 shows the operational flowchart for stage I of the design, and Table 10 lists the arrays used.

Stage II

All items noted in the specifications for Stage II to help make the system easily debugged by the student were included with Stage III.

Stage III

Five additional subroutines were necessary to make a check of the students' analysis and synthesis.

Subroutines FILKME, FILKMC, AND FILKMM were written so that answers to certain parts of the problem could be entered by the faculty members. Each subroutine contained the answers to five problems. For each problem, data was entered for chapter number, problem number, number of active forces, number of active moments, number of reactive forces, number of reactive moments, and a

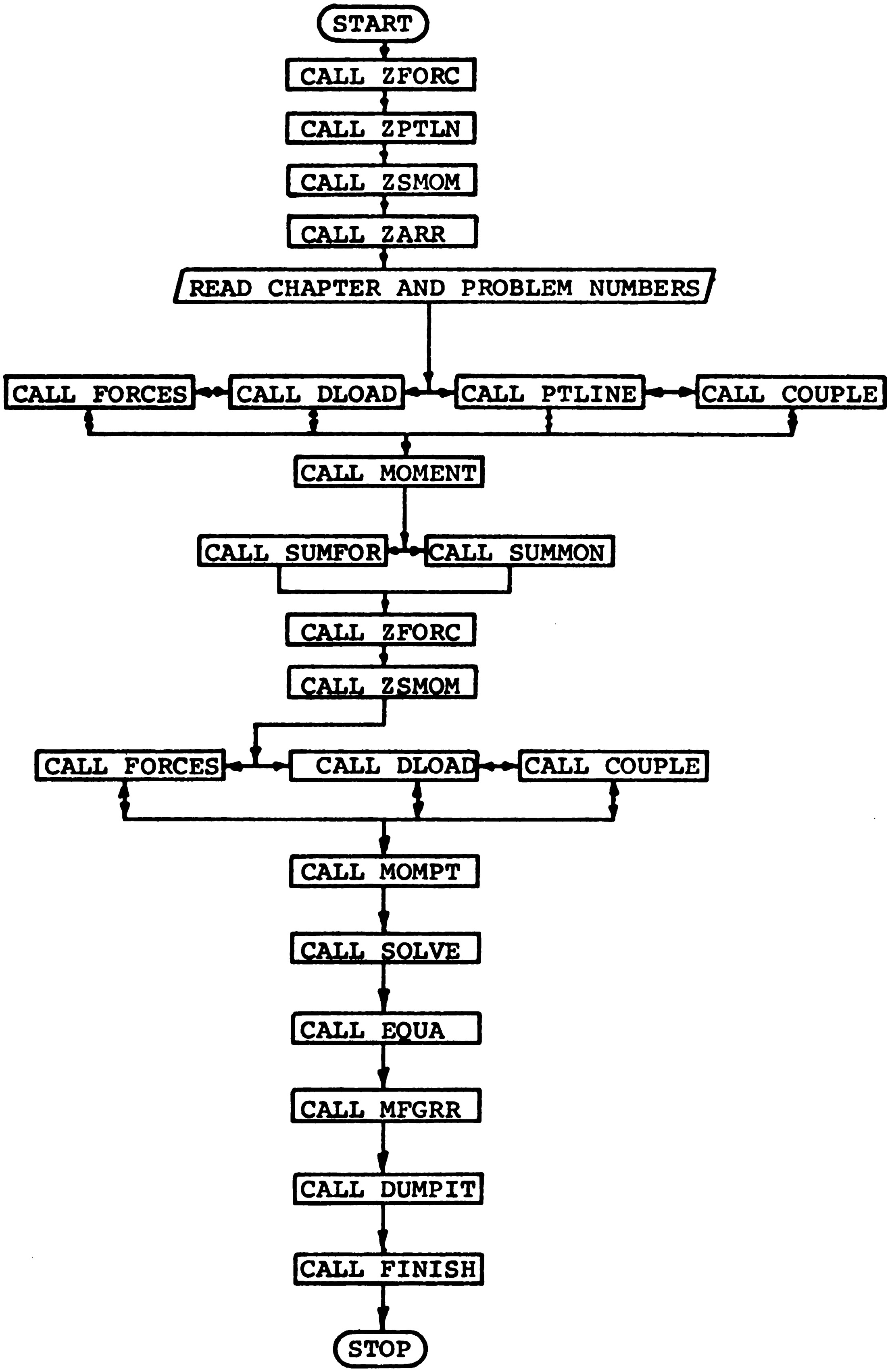


Figure 4. Operational Flowchart for Stage I of the Design

TABLE 10

ARRAYS USED IN THE INFORMATION SYSTEM

Name	For	Type	Dimensions	Elements
FORC	Forces	Numeric-real	6x6	36
FNAM	Forces-names	Alphanumeric	1x6	6
SMOM	Moments and couples	Numeric-real	6x6	36
SMNAM	Moments and couples-names	Alphanumeric	1x6	6
PTLN	Points and lines	Numeric-real	5x6	30
PTNAM	Points and lines-names	Alphanumeric	1x5	5
SF	Sum of forces	Numeric-real	1x3	3
SM	Sum of moments	Numeric-real	1x3	3
SUMS	Sums	Alphanumeric	1x6	6
ARR	Matrix	Numeric-real	6x7	42
ANAM	Names	Alphanumeric	1x6	6
KM	Answers	Numeric-integer	5x7	35
NUML	Numbers of forces and moments	Numeric-integer	2x2	4
F	Reading data	Numeric-real	1x7	7
NR	Rows of matrix	Numeric-integer	1x6	6
NC	Columns of matrix	Numeric-integer	1x7	7
IROWM	Rows of matrix	Numeric-integer	1x6	6
IROWC	Columns of matrix	Numeric-integer	1x7	7
IROW	Rows of matrix	Numeric-integer	1x6	6
ICOL	Columns of matrix	Numeric-integer	1x7	7
S	Matrix	Numeric-real	1x42	42
SCH	School	Alphabetic	1x1	1

number which represented the type of force system.

Subroutine TYPSYS was used to check the equations of equilibrium used by the student.

Subroutine CHECK was used to check the numbers of active and reactive forces and moments entered by the student.

Implementation

Stage I

When the information system was placed on the computer, it was necessary to use a system called LOADER. The system was developed by the computer center personnel, and its purpose was to overlay the subroutines so that only one subroutine and the main program were in operation at any given time. The subroutines were placed on the disc, and the maximum size of the system was determined by the size of the main program and the largest subroutine. It was not possible to call any subroutine from another subroutine, so the subroutines had to be called in the order in which they were needed. It was necessary to include an additional series of control cards when using LOADER.

Stage III

A user's manual, Appendix B, was developed for the students and faculty members. The objectives of the information system and some assumptions concerning problem-solving were given. The equations of equilibrium were presented in both vector and scalar form, and it was noted that it was necessary to separate the forces and moments into active and reactive when using the system. The types of data to be used were listed with examples of each, and the method of tabulating the data was specified. Simplified flow charts and diagrams showing the relationships between subroutines and arrays were presented. The main program was described as consisting of a series of call statements for subroutines which would read the data and perform the required calculations. The deck of cards which the students would be given was discussed, and a computer print-out of the deck was included. The students were instructed where to insert the main program and the data cards in the deck. The method of filling an array with answers to the problems was included for faculty members.

The deck of cards given the students included all the control cards needed; all necessary common, dimension, equivalence, read, and format statements for the main program; and all data cards required for the read statements. The deck of cards had no printing on them except a number. The complete deck was numbered in order so that if the student accidently dropped the deck it could easily be put back in order. The students were also given all of the required call cards for the subroutines and the read and format cards for reading the chapter and problem numbers.

The classes selected for the experiment were an engineering statics class, taught by the investigator, and an engineering technology statics class taught by two other faculty members. Before the experiment began, a meeting was held with the two faculty members teaching the technology class for instructions concerning the use of the information system. The user's manual was thoroughly discussed, and they were shown some print-outs of the system. Each was given a user's manual and a deck of cards like the students would use. A statics problem was discussed, they punched the data cards, and the program was run through the computer.

During the second week of the semester, the investigator gave each class a brief introduction to the information system, discussing the purpose of the system and why it had been developed.

Later, when the students had studied forces in their statics classes and had worked several problems, each student was given a user's manual and a deck of cards. These items were discussed at length, and a few simple problems were discussed. The students were shown how to tabulate the data, select the call statements needed for the main program, and where to insert the main program and the data in the deck.

All that was discussed at the time was how to get concentrated forces into the computer and how to sum the forces. The students were then to punch the cards and run the program. As the students progressed through the classes, they were shown how to use the other subroutines. Table 11 shows the subroutines to which they were introduced according to what they were studying in class.

TABLE 11

STUDENT INTRODUCTION TO SUBROUTINES

Studying in class	Subroutines introduced
Concurrent forces	FORCES, SUMFOR
Moments and couples	PTLINE, COUPLE, MOMPT, COUPLE
Distributed loads	DLOAD
Equilibrium	The remainder of the subroutines

Information System Mechanization

The subroutines developed in the design, pages 59 through 68, were written in the Basic Fortran IV computer language and debugged. Table 12 lists the core required for the subroutines.

Information System Modification

The system was modified as follows:

Stage I

It was necessary to modify the information system to make it easily debugged by students.

Stage II

It was necessary to modify the information system so that it would make a comprehensive check of a student's

TABLE 12

CORE REQUIRED FOR SUBROUTINES
AND MAIN PROGRAM

Subroutine	Common Core	Core
ZFORC	1028	416
ZSMOM	1028	360
ZPTLN	1028	360
ZARR	1028	328
FORCES	1028	2148
DLOAD	1028	2400
PTLINE	1028	1500
COUPLE	1028	1216
MOMPT	1028	952
SUMFOR	1028	600
SUMMOM	1028	616
SOLVE	1028	2256
EQUA	1028	1584
FILKMC	1028	448
FILKME	1028	448
FILKMM	1028	448
CHECK	1028	1640
TYPSYS	1028	1704
MFGRR	0	2568
DUMPIT	1080	960
FINISH	1080	2600
MAIN PROGRAM	1080	1632

analysis and synthesis.

Stage III

When the students began using the system, they soon became aware that sending one problem through the computer at a time was much too time-consuming. The main program was, therefore, modified into a pat program for the students in the form of a do loop so that five problems could be sent through the computer at one time and still have their analysis and synthesis checked. A computer print-out of the revised main program was included in the user's manual, page 147 of Appendix B.

Section II: Student and Faculty Analysis

A description of the populations, instrumentation, procedures used in the data collection, and the analysis of the data are presented in this section.

Populations

Two populations were included in the study which consisted of students enrolled in statics classes during the 1974 spring semester at the Purdue University Regional Campus at Fort Wayne, Indiana. One population consisted of eight sophomore engineering students, enrolled in an engineering statics class, who had completed a computer course in Fortran IV programming. The second population consisted of twenty-seven freshman technology students, enrolled in a technology statics class, who had not completed a computer course in Fortran IV programming.

Instrumentation

Two versions of an instrument developed by Brown[43] were used to measure students' attitudes toward the computer related instructional model. The instrument was developed by Brown to measure expressed attitudes toward computer-

---

[43]B. R. Brown, "Experimentation with Computer-Assisted Instruction in Technical Education," (Semi-annual Progress Report, Project No. OEC-5-85-074), University Park, Pa., The Pennsylvania State University, December 31, 1966.

assisted instruction. The instrument was used in its original past tense form to measure reactions to the model for a post-test, Appendix A, and the contents were placed in the future tense so that a pre-test, Appendix A, could be given to measure prior attitudes of the students toward the model. Some of the items on the Brown scale were omitted and others were added by the investigator. The principal modification of the items was that Computer-Assisted Instruction was replaced with the Computer Program.

The original form of the Brown scale was reported as having an internal consistency reliability coefficient of .89. Mathis, Smith and Hansen[44] also used a modified version of the Brown scale and reported a Kuder-Richardson Formula 20 reliability of .82 for 158 Florida State undergraduates.

A ten-item faculty questionnaire, Appendix A, was developed by the investigator to be filled out by the two faculty members who would teach the technology class. The questionnaire added to the assessment of the model by providing input from faculty members who had not previously been exposed to the model.

---

[44]A. M. Mathis, T. Smith, and D. Hansen, "College Student's Attitudes Toward Computer-Assisted Instruction," Journal of Educational Psychology, Vol. 61, No. 1, February, 1970, pp. 46-51.

Data Collection

The pre-test was administered in class to both populations before the experiment began. The post-test and faculty questionnaire were administered in class at the end of the experiment. All items on all instruments were scored as follows: 1-strongly disagree; 2-disagree; 3-no opinion; 4-agree; 5-strongly agree.

Analysis of the Data

The specific objectives for the study, as listed on pages 38 through 40, are of two types. One type concerns the operation of the model, and the other concerns students' attitudes toward the model. The first type includes objectives 1, 2, 3, 4, 5, 6, 7, 13, 15, 16, and 17 which were satisfied when the model was operational. The results of the students' pre-test and post-test were used to determine if the second type had been satisfied.

The means for each item of both the pre-test and post-test were calculated for each population and indicated as being either positive or negative toward the model.

The items on the tests were then divided into twelve categories to check the second type of specific objectives. The categories were as follows:

Category 1 consisted of all 34 items of the tests and was used to determine the over-all attitude of the students toward the model.

Category 2 was used to determine the attitudes of the students concerning the value of the diagnostics in the model. Test items included were:

3. I was not concerned about missing a problem because I knew I would receive diagnostics describing errors in my analysis.

5. I knew whether my analysis was correct or not before I was told.

10. I felt as if I had a private tutor while using the COMPUTER PROGRAM.

11. I was aware of efforts to suit the diagnostics specifically to me.

13. Diagnostics were given in the COMPUTER PROGRAM which I felt were not relevant to the material.

29. I found the diagnostics given in the computer program to be very poor.

Category 3 was used to determine if the model could easily be used by students whether or not they used vector algebra in their statics class. Test items included were:

15. While using the COMPUTER PROGRAM I had a great deal of trouble keypunching.

17. I felt frustrated while using the COMPUTER PROGRAM.

28. I had a great deal of trouble finding my programming errors while using the COMPUTER PROGRAM.

30. I found it very confusing shuffling call cards for subroutines in the main program.

32. I feel the punched card for the keypunch drum was very helpful.

33. I found it difficult to organize the data when preparing to punch data cards.

34. I found it difficult to understand how to use the COMPUTER PROGRAM.

Category 4 was used to determine if the model interferred with the learning of statics. Test items used were:

2. I was concerned that I might not be understanding the material in the statics course because of the COMPUTER PROGRAM.

4. I tried to get the COMPUTER PROGRAM run rather than trying to learn statics.

8. I was more involved in understanding the COMPUTER PROGRAM than in understanding statics.

12. I found it difficult to concentrate on statics because of the COMPUTER PROGRAM.

Category 5 was used to determine the value of the model as a teaching tool. Test items included were:

14. The COMPUTER PROGRAM is an inefficient use of the student's time.

18. The COMPUTER PROGRAM approach was inflexible.

19. Even otherwise interesting material would be boring when using a COMPUTER PROGRAM.

21. In view of the amount I learned, I feel that the use of the COMPUTER PROGRAM is superior to traditional instruction.

22. With a course such as I am taking while using the COMPUTER PROGRAM, I would prefer the COMPUTER PROGRAM to traditional instruction.

23. I am not in favor of the COMPUTER PROGRAM because it is just another step toward depersonalized instruction.

25. The COMPUTER PROGRAM was boring.

Category 6 was used to determine if the students felt the model was a valuable asset to learning. Test items included were:

1. While using the COMPUTER PROGRAM I felt challenged to do my best work.

6. I guessed at the method of analysis when using the COMPUTER PROGRAM.

7. As a result of having used the COMPUTER PROGRAM, I am interested in trying to find out more about statics.

9. The COMPUTER PROGRAM made the learning too mechanical.

16. The COMPUTER PROGRAM made it possible for me to learn quickly.

21. In view of the amount I learned, I feel that the use of the COMPUTER PROGRAM is superior to traditional instruction.

26. The use of the COMPUTER PROGRAM made me more systematic in problem-solving.

Category 7 was used to determine if previous experience with a key-punch was necessary to use the model. Test item 24 was used.

24. Previous keypunching experience is necessary in order to perform easily while using the COMPUTER PROGRAM.

Category 8 was used to determine if the model taught the students to become more systematic in problem-solving. Test item 26 was used.

26. The use of the COMPUTER PROGRAM made me more systematic in problem-solving.

Category 9 was used to determine if previous experience with a computer was necessary to use the model. Test item 27 was used.

27. Previous experience with a computer is necessary if a student is to benefit from the COMPUTER PROGRAM.

Category 10 was used to determine if it was easy for students to debug the program. Test item 28 was used.

28. I had a great deal of trouble finding my programming errors while using the COMPUTER PROGRAM.

Category 11 was used to determine if the students felt that too much keypunching was required. Test item 31 was used.

31. I found there was too much keypunching required while using the COMPUTER PROGRAM.

Category 12 was used to determine if the students felt that the punched card for the keypunch drum was helpful. Test item 32 was used.

32. I feel the punched card for the keypunch drum was very helpful.

An overall mean was then calculated for each category. The Chi-square and Fisher tests were used to determine if there were significant differences in attitudes between pre-test and post-test for each population for each category, and between engineering students and technology students for each test for each category.

CHAPTER IV

ANALYSIS OF THE DATA

Two populations, engineering and technology students, were tested in the study, so there were four groups of data as follows: pre-test for engineering students; post-test for engineering students; pre-test for technology students; post-test for technology students. Each group of data consisted of thirty-four responses to the items on the pre-test or post-test. Each response consisted of a number from one through five as follows: 1-strongly disagree; 2-disagree; 3-no opinion; 4-agree; 5-strongly agree. All of the responses were coded so they were readily addressable to computer programs.

Each group of data was divided into twelve categories, as described in the Design of the Study, to check the specific objectives of the study. To facilitate the calculation of an over-all mean for each category of the items on the tests, it was necessary to adjust the student responses to some of the items. For some of the items a response of five was very positive toward the model, and for others a response of one was very positive. The responses were, therefore, adjusted as follows: 1-very positive; 2-positive; 3-no opinion; 4-negative; 5-very negative. As a result of the adjustment, all means for items and categories were positive if 1-mean 3 and negative if 3 mean-5.

The chi-square and Fisher tests were used to determine if there were significant differences in attitudes between pre-test and post-test for each population for each category, and between engineering students and technology students for each test for each category. When the tests were used, it was necessary to collapse the data into 2x2 contingency tables due to the small number of students. Therefore, the student responses were placed in one of two mutually exclusive classes, positive toward the model (for a response of one or two) and negative toward the model (for a response of four or five.)

A computer program was developed by the investigator to be used on an IBM 360-22 computer which sorted the data according to population, test, and category. Computer print-outs from the program are shown in Tables 13 through 16 for category 1, overall attitude, for engineering and technology students' pre-test and post-test. Student responses to each test item were tabulated, and the mean was given with a plus or minus sign indicating if the response was positive or negative toward the model. An asterisk indicated that the data had been adjusted. The number of students who were positive or negative was also listed for each item. The number of students tested, an overall mean for the category, the numbers of students with overall positive and negative

TABLE 13

PRE-TEST RESULTS FOR ENGINEERING STUDENTS IN CATEGORY 1

ITEM	ADJUSTED SCORES 1	2	3	4	5	MEAN	NUMBER POSITIVE	NEGATIVE
1	1	4	3	0	0*	2.250+	5	0
2	2	2	3	1	0	2.375+	4	1
3	0	3	3	2	0*	2.875+	3	2
4	1	3	1	3	0	2.750+	4	3
5	0	4	2	2	0*	2.750+	4	2
6	1	6	1	0	0	2.000+	7	0
7	0	1	5	2	0*	3.125-	1	2
8	0	4	3	1	0	2.625+	4	1
9	2	3	2	1	0	2.250+	5	1
10	0	0	6	1	1*	3.375-	0	2
11	0	2	6	0	0*	2.750+	2	0
12	1	3	3	1	0	2.500+	4	1
13	0	2	5	1	0	2.875+	2	1
14	3	3	1	1	0	2.000+	6	1
15	1	6	1	0	0	2.000+	7	0
16	0	2	3	3	0*	3.125-	2	3
17	1	4	2	1	0	2.375+	5	1
18	0	4	3	1	0	2.625+	4	1
19	0	6	2	0	0	2.250+	6	0
20	1	4	3	0	0*	2.250+	5	0
21	1	3	4	0	0*	2.375+	4	0
22	0	3	5	0	0*	2.625+	3	0
23	0	6	2	0	0	2.250+	6	0
24	0	4	3	1	0	2.625+	4	1
25	1	4	3	0	0	2.250+	5	0
26	0	6	2	0	0*	2.250+	6	0
27	1	3	4	0	0	2.375+	4	0
28	0	5	1	2	0	2.625+	5	2
29	0	1	6	1	0	3.000	1	1
30	1	4	2	1	0	2.375+	5	1
31	0	2	4	2	0	3.000	2	2
32	0	3	5	0	0*	2.625+	3	0
33	0	7	1	0	0	2.125+	7	0
34	0	7	1	0	0	2.125+	7	0

8 STUDENTS

8 POSITIVE 0 NEGATIVE

OVERALL MEAN=2.522

ATTITUDE +

TABLE 14

POST-TEST RESULTS FOR ENGINEERING STUDENTS IN CATEGORY 1

	ADJUSTED SCORES						NUMBER	
ITEM	1	2	3	4	5	MEAN	POSITIVE	NEGATIVE
1	3	3	2	0	0*	1.875+	6	0
2	1	6	0	1	0	2.125+	7	1
3	0	2	0	5	1*	3.625-	2	6
4	0	5	0	3	0	2.750+	5	3
5	2	2	4	0	0*	2.250+	4	0
6	5	3	0	0	0	1.375+	8	0
7	0	0	7	1	0*	3.125-	0	1
8	1	5	1	1	0	2.250+	6	1
9	2	5	1	0	0	1.875+	7	0
10	0	4	2	2	0*	2.750+	4	2
11	1	5	2	0	0*	2.125+	6	0
12	2	5	0	1	0	2.000+	7	1
13	3	5	0	0	0	1.625+	8	0
14	5	2	1	0	0	1.500+	7	0
15	1	6	0	1	0	2.125+	7	1
16	0	4	3	1	0*	2.625+	4	1
17	1	3	2	2	0	2.625+	4	2
18	2	5	0	1	0	2.000+	7	1
19	3	4	1	0	0	1.750+	7	0
20	3	5	0	0	0*	1.625+	8	0
21	0	4	3	1	0*	2.625+	4	1
22	3	1	2	2	0*	2.375+	4	2
23	3	5	0	0	0	1.625+	8	0
24	2	3	2	0	1	2.375+	5	1
25	1	7	0	0	0	1.875+	8	0
26	2	6	0	0	0*	1.750+	8	0
27	1	5	2	0	0	2.125+	6	0
28	2	6	0	0	0	1.750+	8	0
29	2	5	1	0	0	1.875+	7	0
30	5	3	0	0	0	1.375+	8	0
31	5	3	0	0	0	1.375+	8	0
32	5	2	0	1	0*	1.625+	7	1
33	3	5	0	0	0	1.625+	8	0
34	4	3	1	0	0	1.625+	7	0

8 STUDENTS

8 POSITIVE 0 NEGATIVE

OVERALL MEAN=2.059

ATTITUDE +

TABLE 15

PRE-TEST RESULTS FOR TECHNOLOGY STUDENTS IN CATEGORY 1

	ADJUSTED SCORES						NUMBER	
ITEM	1	2	3	4	5	MEAN	POSITIVE	NEGATIVE
1	5	19	2	1	0*	1.963+	24	1
2	3	12	10	1	1	2.444+	15	2
3	0	11	7	8	1*	2.963+	11	9
4	2	17	6	2	0	2.296+	19	2
5	0	7	16	4	0*	2.889+	7	4
6	5	12	9	1	0	2.222+	17	1
7	1	12	12	2	0*	2.556+	13	2
8	0	16	10	1	0	2.444+	16	1
9	2	17	6	1	1	2.333+	19	2
10	1	6	11	9	0*	3.037-	7	9
11	0	8	19	0	0*	2.704+	8	0
12	3	19	5	0	0	2.074+	22	0
13	0	8	18	1	0	2.741+	8	1
14	3	15	7	2	0	2.296+	18	2
15	4	8	15	0	0	2.407+	12	0
16	1	8	17	1	0*	2.667+	9	1
17	2	9	16	0	0	2.519+	11	0
18	0	7	19	1	0	2.778+	7	1
19	1	16	10	0	0	2.333+	17	0
20	4	17	6	0	0*	2.074+	21	0
21	1	4	18	3	1*	2.963+	5	4
22	1	1	20	3	2*	3.148-	2	5
23	4	19	4	0	0	2.000+	23	0
24	2	6	15	4	0	2.778+	8	4
25	3	9	15	0	0	2.444+	12	0
26	3	16	8	0	0*	2.185+	19	0
27	1	13	12	1	0	2.481+	14	1
28	0	7	20	0	0	2.741+	7	0
29	0	7	20	0	0	2.741+	7	0
30	1	6	19	1	0	2.741+	7	1
31	1	10	15	1	0	2.593+	11	1
32	1	6	20	0	0*	2.704+	7	0
33	0	5	18	4	0	2.963+	5	4
34	0	10	14	3	0	2.741+	10	3

27 STUDENTS

26 POSITIVE 1 NEGATIVE

OVERALL MEAN=2.558

ATTITUDE +

TABLE 16

POST-TEST RESULTS FOR TECHNOLOGY

STUDENTS IN CATEGORY 1

ITEM	ADJUSTED SCORES 1	2	3	4	5	MEAN	NUMBER POSITIVE	NEGATIVE
1	2	6	7	7	5*	3.259-	8	12
2	4	8	6	8	1	2.778+	12	9
3	0	6	11	10	0*	3.143-	6	16
4	7	11	3	5	1	2.333+	18	6
5	1	3	5	16	2*	3.556-	4	18
6	1	12	7	7	0	2.741+	13	7
7	0	2	8	13	4*	3.704-	2	17
8	3	13	4	5	2	2.630+	16	7
9	2	8	7	7	3	3.037-	10	10
10	0	1	7	14	5*	3.852-	1	19
11	0	5	19	3	0*	2.926+	5	3
12	1	10	5	5	6	3.185-	11	11
13	0	7	11	9	0	3.074-	7	9
14	2	5	7	8	5	3.333-	7	13
15	3	15	1	7	1	2.556+	18	8
16	0	0	7	20	0*	3.741-	0	20
17	0	4	7	12	4	3.593-	4	16
18	0	3	19	5	0	3.074-	3	5
19	1	9	12	3	2	2.852+	10	5
20	4	6	9	7	1*	2.815+	10	8
21	0	2	5	13	7*	3.926-	2	20
22	1	1	4	12	9*	4.000-	2	21
23	2	10	8	7	0	2.741+	12	7
24	2	13	1	9	2	2.852+	15	11
25	2	7	8	8	2	3.037-	9	10
26	0	4	10	13	0*	3.333-	4	13
27	1	10	7	6	3	3.000	11	9
28	0	7	4	13	3	3.444-	7	16
29	0	10	11	5	1	2.889+	10	6
30	1	10	4	10	2	3.074-	11	12
31	3	15	7	2	0	2.296+	18	2
32	3	12	3	6	3*	2.778+	15	9
33	1	4	5	12	5	3.593-	5	17
34	2	6	9	7	3	3.111-	8	10

27 STUDENTS

9 POSITIVE 16 NEGATIVE

OVERALL MEAN=3.125

ATTITUDE -

attitudes, and the overall attitude of the groups were listed at the bottom of the table.

A second computer program was developed by the investigator to be used on an IBM 360-22 computer which determined if there were significant differences in attitudes, at the .05 level of significance, between pre-test and post-test for each population for each category, and between engineering and technology students for each test for each category. The results of the calculations were included in Tables 17 through 20.

The overall attitudes of both engineering and technology students were positive toward the model on the pre-test. Engineering students became more positive on the post-test, but the change was not significant. Technology students became negative toward the model on the post-test, and the change was significant. Although the engineering students were more positive than the technology students on the pre-test, there was no significant difference. There was, however, a significant difference in attitudes on the post-test.

The attitudes of both engineering and technology students toward the value of the diagnostics built into the model were positive on the pre-test. Engineering students became more positive on the post-test, but the change was not significant. Technology students became negative on the post-test, and the change was significant. Although the technology students were more positive than the engineering students on the pre-test, there was no significant difference. There was, however, a significant difference in attitudes on the post-test.

The attitudes of both engineering and technology students were positive on the pre-test concerning the assumption that the model could easily be used by all students. Engineering students became more positive on the post-test, but the change was not significant. Technology students became negative on the post-test, and the change was significant. Engineering students were more positive than the technology students on the pre-test, but there was no significant difference. There was, however, a significant difference on the post-test.

The attitudes of both engineering and technology students were positive on the pre-test concerning the assumption that the model would not interfere with the learning of statics. Engineering students became more positive on the post-test, but the change was not significant. Technology students became less-positive on the post-test, and there was a significant difference. Technology students were more positive than engineering students on the pre-test, and there was a significant difference. Engineering students were more positive on the post-test, but there was no significant difference.

The attitudes of both engineering and technology students were positive on the pre-test concerning the value of the model as a teaching tool. Engineering students became more positive on the post-test, but the change was not significant. Technology students became negative on the post-test, and there was a significant difference. Although engineering students were more positive than the technology students on the pre-test, there was no significant difference. There was, however, a significant difference on the post-test.

The attitudes of both engineering and technology students were positive on the pre-test concerning the assumption that the model was a valuable asset to learning. Engineering students became more positive on the post-test, but the change was not significant. Technology students became negative on the post-test, and the change was significant. Technology students were more positive than engineering students on the pre-test, but the difference was not significant. There was, however, a significant difference on the post-test.

The attitudes of both engineering and technology students were positive on the pre-test concerning the assumption that previous experience with a keypunch was not necessary in order to use the model. Engineering students became more positive on the post-test, but the change was not significant. Technology students became less positive on the post-test, but the change was not significant. Although engineering students were more positive than technology students on the pre-test, the difference was not significant. The differences on the post-test were significant.

The attitudes of both engineering and technology students were positive on the pre-test concerning the assumption that the model would help students become more systematic in problem-solving. Engineering students became more positive on the post-test, but the change was not significant. Technology students became negative on the post-test, and the change was significant. Although technology students were more positive than engineering students on the pre-test, the difference was not significant. There was, however, a significant difference on the post-test.

The attitudes of both engineering and technology students were positive on the pre-test concerning the assumption that previous experience with a computer was not necessary in order to use the model. Engineering students became more positive on the post-test, but the change was not significant. Technology students became less positive on the post-test, and the change was significant. Although the engineering students were more positive than the technology students on the pre-test, the difference was not significant. There was also no significant difference on the post-test.

The attitudes of both engineering and technology students were positive on the pre-test concerning the assumption that the model could be easily debugged by students. Engineering students became more positive on the post-test, but the change was not significant. Technology students became negative on the post-test, and the change was significant. Engineering students were more positive than the technology students on the pre-test, but the difference was not significant. There was, however, a significant difference on the post-test.

The attitudes of engineering students were neutral and the attitudes of technology students were positive on the pre-test concerning the assumption that there would not be too much keypunching required. Engineering students became very positive on the post-test, but the change was not significant. Technology students became more positive on the post-test, but the change was not significant. Although technology students were more positive than engineering students on the pre-test, the difference was not significant. The difference on the post-test was also non-significant.

The attitudes of both engineering and technology students were positive on the pre-test concerning the value of the card for the keypunch drum. Engineering students became more positive on the post-test, but the change was not significant. Technology students became less positive on the post-test, but the change was not significant. Although engineering students were more positive than technology students on the pre-test, the difference was not significant. There was also no significant difference on the post-test.

A questionnaire was administered to the two faculty members who taught the technology class to receive input from faculty members who had not been previously exposed to the model. The results of the questionnaire are shown in Table 17.

TABLE 17

RESULTS OF FACULTY QUESTIONNAIRE

	Response				
Item	1	2	3	4	5
1	1	1			
2				2	
3			1		1
4					2
5	2				
6					2
7					2
8				2	
9			1	1	
10				2	

Item 1: Neither faculty member thought the model interferred with his teaching of statics.

Item 2: Both of the faculty members thought the model helped their students to become more systematic in problem-solving.

Item 3: One faculty member was not sure if he would use the model the next time he taught statics, but the other one definitely will.

Item 4: Both of the faculty members thought it was a good experience for their students to be exposed to a model such as the one in the study.

Item 5: Neither faculty member thought the model was too difficult for their students to understand or use.

Item 6: Both faculty members thought the model was a valuable teaching technique.

Item 7: Both faculty members thought the diagnostics were very good.

Item 8: Both faculty members thought their students were able to analyze problems which were heretofore impossible or very difficult.

Item 9: One faculty member noticed some resistance from his students concerning the use of the model, but the other had no opinion on the matter.

Item 10: Both faculty members thought the model helped students to better understand statics.

Summary

A summary of the analysis of the student data is given in Tables 18 through 21.

TABLE 18

SUMMARY OF THE ANALYSIS OF THE DATA FOR

ENGINEERING STUDENTS' PRE-TEST AND POST-TEST

	Pre-test			Post-test					
Category	P*	N*	Mean	P*	N*	Mean	Change in mean	Test	Significant differences
1	8	0	2.522+	8	0	2.059+	+	Fisher	No
2	4	2	2.938+	7	1	2.375+	+	Fisher	No
3	7	0	2.321+	8	0	1.821+	+	Fisher	No
4	5	3	2.563+	7	1	2.281+	+	Fisher	No
5	8	0	2.339+	8	0	1.964+	+	Fisher	No
6	8	0	2.482+	8	0	2.179+	+	Fisher	No
7	4	1	2.625+	5	1	2.375+	+	Fisher	No
8	6	0	2.250+	8	0	1.750+	+	Fisher	No
9	4	0	2.375+	6	0	2.125+	+	Fisher	No
10	5	2	2.625+	8	0	1.750+	+	Fisher	No
11	2	2	3.000	8	0	1.375+	+	Fisher	No
12	3	0	2.625+	7	1	1.625+	+	Fisher	No

*P=Positive N=Negative

TABLE 19

SUMMARY OF THE ANALYSIS OF THE DATA FOR

TECHNOLOGY STUDENTS' PRE-TEST AND POST-TEST

	Pre-test			Post-test					
Category	P*	N*	Mean	P*	N*	Mean	Change in mean	Test	Significant differences
1	26	1	2.558+	9	16	3.125-	-	Chi-square	Yes
2	17	7	2.846+	5	20	3.241-	-	Chi-square	Yes
3	17	1	2.688+	12	13	3.164-	-	Chi-square	Yes
4	23	1	2.315+	18	8	2.731+	-	Fisher	Yes
5	25	1	2.566+	7	18	3.280-	-	Chi-square	Yes
6	24	1	2.413+	6	19	3.392-	-	Chi-square	Yes
7	8	4	2.778+	15	11	2.852+	-	Fisher	No
8	19	0	2.185+	4	13	3.333-	-	Chi-square	Yes
9	14	1	2.481+	11	9	3.000	-	Fisher	Yes
10	7	0	2.741+	7	16	3.444-	-	Fisher	Yes
11	11	1	2.593+	18	2	2.296+	+	Fisher	No
12	7	0	2.704+	15	9	2.778+	-	Fisher	No

*P=Positive N=Negative

TABLE 20

SUMMARY OF THE ANALYSIS OF THE DATA FOR

ENGINEERING AND TECHNOLOGY STUDENTS' PRE-TEST

	Engineering			Technology					
Category	P*	N*	Mean	P*	N*	Mean	Most positive	Test	Significant differences
1	8	0	2.522+	26	1	2.558+	Engineering	Fisher	No
2	4	2	2.938+	17	7	2.846+	Technology	Fisher	No
3	7	0	2.321+	17	1	2.688+	Engineering	Fisher	No
4	5	3	2.563+	23	1	2.315+	Technology	Fisher	Yes
5	8	0	2.339+	25	1	2.566+	Engineering	Fisher	No
6	8	0	2.482+	24	1	2.413+	Technology	Fisher	No
7	4	1	2.625+	8	4	2.778+	Engineering	Fisher	No
8	6	0	2.250+	19	0	2.185+	Technology	Fisher	No
9	4	0	2.375+	14	1	2.481+	Engineering	Fisher	No
10	5	2	2.625+	7	0	2.741+	Engineering	Fisher	No
11	2	2	3.000	11	1	2.593+	Technology	Fisher	No
12	3	0	2.625+	7	0	2.704+	Engineering	Fisher	No

*P=Positive N=Negative

TABLE 21

SUMMARY OF THE ANALYSIS OF THE DATA FOR

ENGINEERING AND TECHNOLOGY STUDENTS' POST-TEST

	Engineering			Technology					
Category	P*	N*	Mean	P*	N*	Mean	Most positive	Test	Significant differences
1	8	0	2.059+	9	16	3.125-	Engineering	Fisher	Yes
2	7	1	2.375+	5	20	3.241-	Engineering	Fisher	Yes
3	8	0	1.821+	12	13	3.164-	Engineering	Fisher	Yes
4	7	1	2.281+	18	8	2.731+	Engineering	Fisher	No
5	8	0	1.964+	7	18	3.280-	Engineering	Fisher	Yes
6	8	0	2.179+	6	19	3.392-	Engineering	Fisher	Yes
7	5	1	2.375+	15	11	2.852+	Engineering	Fisher	Yes
8	8	0	1.750+	4	13	3.333-	Engineering	Fisher	Yes
9	6	0	2.125+	11	9	3.000	Engineering	Fisher	No
10	8	0	1.750+	7	16	3.444-	Engineering	Fisher	Yes
11	8	0	1.375+	18	2	2.296+	Engineering	Fisher	No
12	7	1	1.625+	15	9	2.778+	Engineering	Fisher	No

*P=Positive N=Negative

CHAPTER V

SUMMARY, CONCLUSIONS, RECOMMENDATIONS, AND IMPLICATIONS FOR FURTHER DEVELOPMENT AND RESEARCH

Chapter V includes a summary of the results of the student pre-test and post-test, conclusions concerning the objectives of the study, recommendations for modifications to improve the model, and implications for further development and research.

Summary

The primary purpose of the study was accomplished when the computer related instructional model was operational. The model was capable of making a comprehensive check of a student's analysis and synthesis of a broad range of statics problems on an individualized basis.

A secondary purpose of the study was to evaluate the model for educational impact. Attitudinal information was gathered by administering a pre-test and a post-test to the engineering and technology student populations.

The engineering students were positive toward the model in all categories of the pre-test except the one concerning keypunching. These students became more positive toward the model in all categories of the post-test, but none of the changes were statistically significant.

The technology students were positive toward the model in all categories of the pre-test. These students became less positive or negative in all categories on the post-test, and all changes were statistically significant except those concerning previous experience being required on a keypunch, too much keypunching being required when using the model, and the value of the card for the keypunch drum.

No significant differences were found between engineering and technology students on the pre-test except the category concerning the model interfering with the learning of statics.

Significant differences were found between engineering and technology students on the post-test in all categories except those concerning the model interfered with the learning of statics, previous experience with a computer was necessary, too much keypunching was required when using the model, and the value of the card for the keypunch drum.

Conclusions

The development of the model, as outlined in the Design of the Study, follows a systems analysis and design approach which will serve as a valuable basis for the development of additional models for other problem-solving courses.

The model is written in Basic Fortran IV, requires less than 16K of core including systems overhead, can be used with any standard statics textbook, and requires no terminals or telephone lines. These features insure transportability to other computer systems.

The model provides an opportunity for students to focus solely on the analysis and synthesis portions of problem-solving by being relieved of the mechanics of calculations. The output of the model is a hard copy in a neat, systematic form which checks the student's analysis and synthesis. Students may perform the analysis and synthesis of many three-dimensional problems which were heretofore impossible or very difficult due to lengthy calculations. The model visually brings out important points concerning simultaneous equations with as many as six equations and six unknowns, and provides feedback to the student on an individualized basis.

The model does not interfere with the learning of statics. Very little keypunching is required, so no previous experience with a keypunch is necessary to effectively use the model. However, some previous exposure to a computer is necessary or at least desirable.

The main computer program is a pat program capable of working five problems with each computer run. It consists of call statements for canned subroutines, thus giving the students experience in calling such subroutines and keeping them involved with the computer.

Approximately 2.5 minutes are required on the IBM 360-22 computer for each computer run.

Engineering students experienced little difficulty using the model, considered the diagnostics to be valuable, and found the program easy to debug. They also considered the model a good teaching tool, a valuable asset to learning, and an effective means of helping them to become more systematic in problem-solving. Technology students did not agree with the engineering students on any of these items.

The design could have been strengthened through the employment of control and experimental groups.

Recommendations

The student user's manual should be reduced in length. Flow charts of subroutines and the relationships between subroutines and arrays should not be included. An independent study package should be designed to complement the manual which would include the use of a slide projector and tape recorder to provide an explanation of the model. It would be preferable to have the package consist of two or three short presentations to be used as the class progresses through the course. The package would offer the advantage of reducing the class time required to introduce the model, and the student could study the material at his convenience, repeating the material as many times as desired.

Instructors using the model should require the students to work the problems and hand them in to be checked.

The technology class was selected on very short notice since it was the only one available for the study, and the technology faculty members were introduced to the model during the beginning of a new semester. It is recommended that when the model is used again, the faculty member or members have a much longer lead time to become familiar with and experiment with the model before attempting to use it with a class. They should also be thoroughly familiar with the strategy for using the model.

The pat main program should be made a subroutine and put on the disk, thus reducing the compiling time significantly. It is estimated that this could reduce the time required to work five problems from 2.5 minutes to a little less than a minute on the IBM 360-22 computer. Also, a few of the arrays could be reduced in size or eliminated, thus cutting down on the total core required.

When an error in analysis or synthesis is detected in a problem, a diagnostic message is given and the program is terminated. The model will be modified so that when such an error occurs the particular problem is terminated, but the program will continue to the remaining problems included in the program.

The sheet for tabulating the data will be revised so that the actual width and columns correspond in length to those on a computer card. Then when a student punches a card, he can lay it down on the data sheet to make certain he punched data in the desired columns.

<u>Implications for Further Development and Research</u>

The potential for a computer related instructional model, such as the one in this study, is almost unlimited. Any subject area which involves lengthy and/or repetitive calculations is a possible candidate. Furthermore, many computer programs have been written over the years for use by students which relieve the student of the calculations, but do not have the means for checking the student's analysis and synthesis with appropriate diagnostics. A thorough analysis of many of these programs would probably disclose that means could be incorporated to perform this added function, thus making them much more valuable as a teaching tool.

Plans are already in progress to explore the possibility of modifying the model so its usefulness will be greatly expanded. Problems with friction at impending motion will be considered first. Then, the possibility of including problems which require the free-body diagram to be broken into two or three additional free-bodies will be considered. The incorporation of these two items would increase the power of the model considerably.

The model was developed to be used as a supplement to a regular statics class. Possibilities are being explored to extend its use to independent study courses, freshman engineering design courses, courses for the professional development of graduate engineers, and engineering review or refresher courses intended to prepare graduate engineers for professional tests required for licensing.

Other areas being considered for instructional models are dynamics, thermodynamics, strength of materials, engineering design, and cost-analysis for design courses.

An experiment should be conducted with the model when it has been further developed as noted in the recommendations above. A large population of students should be selected for the study, all of whom would be enrolled in the same statics class with one instructor. Control and experimental groups should be randomly selected for the experiment in which only the experimental group would use the model. An instrument should be developed to measure achievement in the cognative area of learning which would be administered to both groups at the end of the experiment. The test results should be used to determine significant differences in achievement of the two groups.

A further study should include a cost-benefit analysis. A cost analysis of the use of the model should be compared with increases in student achievement when using the model. The analysis could be used to determine if the added cost when using the model could be justified from the added benefits the students would realize.

SELECTED BIBLIOGRAPHY

Alpert, D., and Bitzer, D. L. "Advances in Computer-Based Education." Science, (March, 1970), 360, 1586.

Anastasio, E. J., and Alderman, D. L. "Evaluation of the Educational Effectiveness of PLATO and TICCIT." Proceedings: Third Annual Frontiers in Education Conference. New York: IEEE Cat. No. 73 CHO 720-3E, (April, 1973), 382.

Bitzer, Donald L.; Sherwood, Bruce A.; and Tenczar, Paul. "PLATO: Everyone's Answer." Proceedings--Third Annual Frontiers in Education Conference. New York: IEEE Cat. No. 73 CHO 720-3E, (April, 1973), 360-370.

Boblick, John M. "The Use of Computer Simulations in the Teaching of High School Physics." Science Education, Vol. 54, No. 1, (Jan-Mar, 1970), 77-81.

Brown, B. R. "Experimentation with Computer-Assisted Instruction in Technical Education." Semi-annual Progress Report, Project No. OEC-5-85-074. University Park, Pa.: The Pennsylvania State University, 1966.

Bueschel, Richard T. "Time-Sharing--A Pragmatic Approach in the School." Educational Technology, (March, 1970), 21-23.

Diamond, Herbert S. "The Writing of a CAI Program by an Author New to Computers." Educational Technology, (October, 1971), 42.

Edwards, E. M. "APL: A Natural Language for Engineering Education, Part II: The First Programming Language Suitable for Engineering Undergraduates." IEEE Transactions on Education, (November, 1971), 179.

Goldstine, H. H. The Computer from Pascal to von Nuemann. Princeton: Princeton University Press, 1972

Grayson, Lawrence P. "CAI: The Fifteen Million Dollar Experiment." Proceedings--Third Annual Frontiers in Education Conference. New York: IEEE Cat. No. 73 CHO 720-3E, (April, 1973), 357.

Howard, J. A.; Ordung, P. F.; and Wood, R. C. "On-line Computer Systems for Engineering Education--State of the Art." IEEE Transactions on Education, Vol. E-14, No. 4, (November, 1971), 210,216.

Mathis, A. M.; Smith, T.; and Hansen D. "College Students' Attitudes Toward Computer-Assisted Instruction." Journal of Educational Psychology, Vol. 61, No. 1, (February, 1970), 46-51.

McWilliams, Erik D. "The Fifteen Million CAI Experiment--What NSF Expects." Proceedings--Third Annual Frontiers in Education Conference. New York: IEEE Cat. No. 73 CHO 720-3E, (April, 1973), 357.

Rudberg, D. A. "APL: A Natural Language for Engineering Education." IEEE Transactions on Education, (November, 1971), 183-184.

Salisbury, Alan B. "An Overview of CAI." Educational Technology, (October, 1971), 48.

Salisbury, Alan B. "Computers and Education: Toward Agreement on Terminology." Educational Technology, (September, 1971), 35-40.

Stetten, Kenneth J. "Toward a Market Success for CAI (An Overview of the TICCIT Program)." Proceedings--Third Annual Frontiers in Education Conference. New York: IEEE Cat. No. 73 CHO 720-3E, (April, 1973), 371-372.

Walker, T. M., and Cotterman, W. W. An Introduction to Computer Science and Algorithmic Processes. Boston: Allyn and Bacon, 1970.

Zinn, Karl L. "Instructional Programming Languages." Educational Technology, (March, 1970), 43-46.

GENERAL REFERENCES

Boblick, John M. "Computer-Based Simulation of an Acid-Base Titration." School Science and Mathematics, Vol. 71, (January, 1971), 49-54.

Boocock, Sarane S. "Using Simulation Games in College Courses." Simulation and Games, Vol. 1, No. 1, (March, 1970), 67-77.

Burke, J. Bruce; O'Neill, Julia; and Welsch, Kay. "A Humanized Model of a Computer Managed Instructional System." Educational Technology, (November, 1972), 31-36.

Deep, Donald. "The Computer Can Help Individualize Instruction." The Elementary School Journal, Vol. 70, No. 7, (April, 1970), 351-358.

Gaunt, Roger, N. "Benchmarks May be More Important Than Trademarks in Selecting Computers." College and University Business, (June, 1971), 68-70.

Gaunt, Roger N. "Computer Choices: Ignore it, Hire it, Buy it, or Get Together to Use it." College and University Business, (August, 1971), 20-27.

Gaunt, Roger N. "Everything You Always Wanted to Know About Computers, But Didn't Know Whom to Ask." College and University Business, (October, 1971). 64-66.

Gaunt, Roger N. "Selection of Computer System Must Start with Understanding of Hardware-Software Capabilities." College and University Business, (April, 1971), 74-78.

Hansen, Duncan N. and Harvey, William L. "Impact of CAI on Classroom Teachers." Educational Technology, Vol. X, No. 2, (February, 1970), 46-48.

Jacobson, Milton D. and MacDougall, Mary Ann. "Computer Management of Information and Structure in Computer-Supported Instructional Materials." Educational Technology, Vol. X, No. 3, (March, 1970), 39-42.

Jamison, D. and Suppes P. "Estimated Costs of Computer Assisted Instruction for Compensatory Education in Urban Areas." Educational Technology, (September, 1970), 49-57.

Kerr, Eugene G.; Ting, T. C.; Walden, William E. "An Instructional System for Computer Assisted Instruction on a General Purpose Computer." Educational Technology, Vol. X, No. 3, (March, 1970), 28-30.

Kooi, Beverly Y., and Geddes, Cleone. "The Teacher's Role in Computer Assisted Instructional Management." Educational Technology, Vol. X, No. 2, (February, 1970), 42-45.

Proceedings of a Conference on Computers in the Under-Graduate Curricula. The University of Iowa, Iowa City, Iowa, (June, 1970).

Swartz, G. Boyd. "Using a Small Computer in the College Mathematics Curriculum." Educational Technology, Vol. X, No. 3, (March, 1970), 31-32.

APPENDIX A

STUDENT AND FACULTY

INSTRUMENTS

POST-TEST

1. While using the COMPUTER PROGRAM I felt challenged to do my best work.
2. I was concerned that I might not be understanding the material in the statics course because of the COMPUTER PROGRAM.
3. I was not concerned about missing a problem because I knew I would receive diagnostics describing errors in my analysis.
4. I tried to get the COMPUTER PROGRAM run rather than trying to learn statics.
5. I knew whether my analysis was correct or not before I was told.
6. I guessed at the method of analysis when using the COMPUTER PROGRAM.
7. As a result of having used the COMPUTER PROGRAM, I am interested in trying to find out more about statics.
8. I was more involved in understanding the COMPUTER PROGRAM than in understanding statics.
9. The COMPUTER PROGRAM made the learning too mechanical.
10. I felt as if I had a private tutor while using the COMPUTER PROGRAM.
11. I was aware of efforts to suit the diagnostics specifically to me.
12. I found it difficult to concentrate on statics because of the COMPUTER PROGRAM.
13. Diagnostics were given in the COMPUTER PROGRAM which I felt were not relevant to the material.
14. The COMPUTER PROGRAM is an inefficient use of the student's time.
15. While using the COMPUTER PROGRAM I had a great deal of trouble keypunching.
16. The COMPUTER PROGRAM made it possible for me to learn quickly.
17. I felt frustrated while using the COMPUTER PROGRAM.

18. The COMPUTER PROGRAM approach was inflexible.

19. Even otherwise interesting material would be boring when using a COMPUTER PROGRAM.

20. In view of the effort I put into it, I am satisfied with what I learned while using the COMPUTER PROGRAM.

21. In view of the amount I learned, I feel that the use of the COMPUTER PROGRAM is superior to traditional instruction.

22. With a course such as I am taking while using the COMPUTER PROGRAM, I would prefer the COMPUTER PROGRAM to traditional instruction.

23. I am not in favor of the COMPUTER PROGRAM because it is just another step towards depersonalized instruction.

24. Previous keypunching experience is necessary in order to perform easily while using the COMPUTER PROGRAM.

25. The COMPUTER PROGRAM was boring.

26. The use of the COMPUTER PROGRAM made me more systematic in problem-solving.

27. Previous experience with a computer is necessary if a student is to benefit from the COMPUTER PROGRAM.

28. I had a great deal of trouble finding my programming errors while using the COMPUTER PROGRAM.

29. I found the diagnostics given in the COMPUTER PROGRAM to be very poor.

30. I found it very confusing shuffling call cards for subroutines in the main program.

31. I found there was too much keypunching required while using the COMPUTER PROGRAM.

32. I feel the punched card for the keypunch drum was very helpful.

33. I found it difficult to organize the data when preparing to punch data cards.

34. I found it difficult to understand how to use the COMPUTER PROGRAM.

PRE-TEST

1. While using the COMPUTER PROGRAM I will feel challenged to do my best work.
2. I am concerned that I might not be understanding the material in the statics course because of the COMPUTER PROGRAM.
3. I am not concerned about missing a problem because I know I will receive diagnostics describing errors in my analysis.
4. I feel I will try to get the COMPUTER PROGRAM run rather than try to learn statics.
5. I will know whether my analysis is correct or not before I am told.
6. I will guess at the method of analysis when using the COMPUTER PROGRAM.
7. As a result of having used the COMPUTER PROGRAM, I will be interested in trying to find out more about statics.
8. I will be more involved in understanding the COMPUTER PROGRAM than in understanding statics.
9. The COMPUTER PROGRAM will make the learning too mechanical.
10. I will feel as if I have a private tutor while using the COMPUTER PROGRAM.
11. I will be aware of efforts to suit the diagnostics specifically to me.
12. I will find it difficult to concentrate on statics because of the COMPUTER PROGRAM.
13. Diagnostics will be given in the COMPUTER PROGRAM which I will feel are not relevant to the material.
14. The COMPUTER PROGRAM will be an inefficient use of the student's time.
15. While using the COMPUTER PROGRAM I will have a great deal of trouble keypunching.
16. The COMPUTER PROGRAM will make it possible for me to learn quickly.
17. I will feel frustrated while using the COMPUTER PROGRAM.

18. The COMPUTER PROGRAM approach will be inflexible.

19. Even otherwise interesting material will be boring when using a COMPUTER PROGRAM.

20. In view of the effort I will put into it, I will be satisfied with what I will learn while using the COMPUTER PROGRAM.

21. In view of the amount I will learn, I feel that the use of the COMPUTER PROGRAM is superior to traditional instruction.

22. With a course such as I am taking while using the COMPUTER PROGRAM, I would prefer the COMPUTER PROGRAM to traditional instruction.

23. I am not in favor of the COMPUTER PROGRAM because it is just another step towards depersonalized instruction.

24. Previous keypunching experience is necessary in order to perform easily while using the COMPUTER PROGRAM.

25. The COMPUTER PROGRAM will be boring.

26. The use of the COMPUTER PROGRAM will make me more systematic in problem-solving.

27. Previous experience with a computer is necessary if a student is to benefit from the COMPUTER PROGRAM.

28. I will have a great deal of trouble finding my programming errors while using the COMPUTER PROGRAM.

29. I will find the diagnostics given in the COMPUTER PROGRAM to be very poor.

30. I will find it very confusing shuffling call cards for subroutines in the main program.

31. I will find there will be too much keypunching required while using the COMPUTER PROGRAM.

32. I feel the punched card for the keypunch drum will be very helpful.

33. I will find it difficult to organize the data when preparing to punch data cards.

34. I will find it difficult to understand how to use the COMPUTER PROGRAM.

FACULTY QUESTIONNAIRE

1. The COMPUTER PROGRAM interferred with my teaching of statics.

2. The COMPUTER PROGRAM helped the students to become more systematic in setting-up problems.

3. I will definitely consider using the COMPUTER PROGRAM when I next teach statics.

4. It is good experience for the students to be exposed to the use of a computer with a COMPUTER PROGRAM such as this one.

5. It is too difficult for the students to understand the COMPUTER PROGRAM and how to use it.

6. The COMPUTER PROGRAM is a valuable teaching technique.

7. The diagnostics in the COMPUTER PROGRAM were very good.

8. My students were able to analyze problems which were heretofore impossible or very difficult.

9. I noticed a great deal of resistance by the students concerning the use of the COMPUTER PROGRAM.

10. I feel that the COMPUTER PROGRAM helped the students to better understand statics.

COMMENTS:

APPENDIX B

USER'S MANUAL

USER'S MANUAL

OBJECTIVES

1. Check a student's analysis and synthesis of a problem with suitable diagnostic messages.
2. Require no calculations on the part of the student.
3. Be sure the computer does not interfere with the learning of statics.
4. Afford the students an opportunity to analyze and synthesize many practical problems which heretofore were impossible or very difficult due to lengthy calculations.
5. Give instant feedback to the student on an individualized basis.
6. Require a very simple main program consisting of call statements for canned subroutines.
7. Make the program so that it can easily be debugged by the student.
8. Require very little keypunching on the part of the student.
9. Be usable by a person with little or no previous experience with a computer or keypunch.
10. Give the student experience in calling canned subroutines.
11. Visually bring out points concerning simultaneous equations with several variables which are ordinarily only discussed in class or in a textbook.
12. Keep students involved with a computer.

ASSUMPTIONS

1. The analysis and synthesis are the most important parts of problem-solving.
2. Many of the errors in problem-solving occur during the analysis of the problem.
3. Given enough time, most students can make accurate calculations.

EQUATIONS OF EQUILIBRIUM

Vector form: $\sum \bar{F}=\bar{0}$ $\sum \bar{M}=\bar{0}$

Scalar form: $\sum F_x=0$ $\sum M_x=0$

$\sum F_y=0$ $\sum M_y=0$

$\sum F_z=0$ $\sum M_z=0$

For this program, the student must distinguish between active and reactive forces and moments.

Vector form: $\sum \bar{F}_{active}+\sum \bar{F}_{reactive}=\bar{0}$

$\sum \bar{M}_{active}+\sum \bar{M}_{reactive}=\bar{0}$

Scalar form: $\sum F_{x_A}+\sum F_{x_R}=0$ $\sum M_{x_A}+\sum M_{x_R}=0$

$\sum F_{y_A}+\sum F_{y_R}=0$ $\sum M_{y_A}+\sum M_{y_R}=0$

$\sum F_{z_A}+\sum F_{z_R}=0$ $\sum M_{z_A}+\sum M_{z_R}=0$

DATA

I. Forces:

A. Concentrated

B. Distributed

II. Points:

III. Lines:

IV. Moments:

TABULATION OF DATA

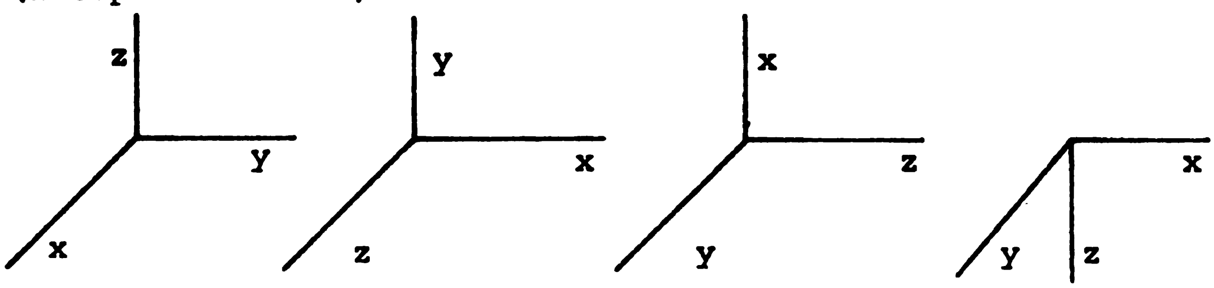
I. Forces:

A. Concentrated

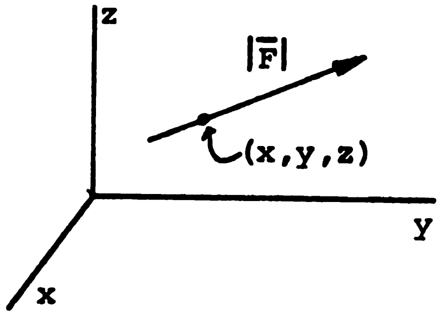
Name	NO.	F1	F2	F3	F4	F5	F6	F7
	1	F	l	m	n	x	y	z
	2	F	x_1	y_1	z_1	x_2	y_2	z_2
	3	F	x_s	y_s	z_s	x	y	z
	4	F	F_x	F_y	F_z	x	y	z
	5	F		l	m	x	y	
	6		F_x	F_y	F_z	x	y	z
A4	I4	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0
1-4	5-8	9-16	17-24	25-32	33-40	41.48	49-56	57-64

You have a free choice of the coordinate axes.

(Except for NO 5.)



No. 1



Given:

Direction cosiness: l,m,n
Magnitude: $|\bar{F}|$
Point: x,y,z

Example:

Name=F10
$|\bar{F}|$ =100.lb$_f$
l =-0.716
m = 0.275
n = 0.642

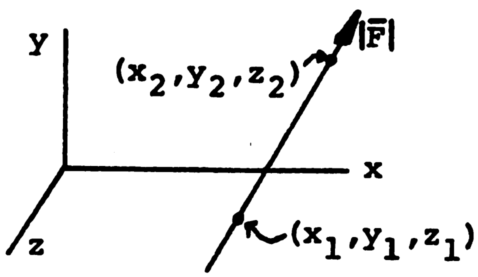
x=3.
y=0.
z=4.

DATA CARD:

Name	NO.	F1	F2	F3	F4	F5	56	F7
F10	1	100.	-0.716	0.275	.642	3.		4.

NOTE: F1-F7 must have a decimal point
NO has no decimal point.
If a value is zero, leave the columns blank
such as y=0 (F6 above)

No. 2



Given:

Magnitude: $|\bar{F}|$
Two points: From (x_1,y_1,z_1)
to (x_2,y_2,z_2)

Example:

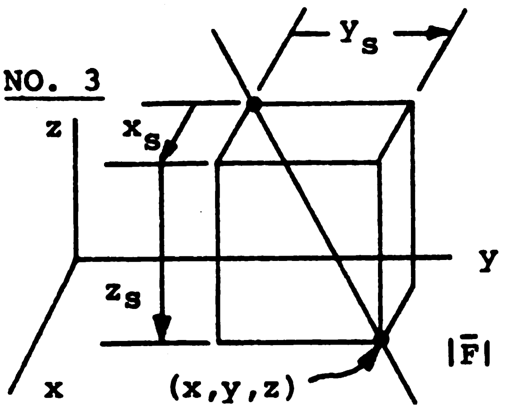
Name= F4
$|\bar{F}|$ = 500.lb$_f$
x_1 = 2
y_1 = -1.
z_1 = 8.

x_2=-5.
y_2=6.
z_2=0.

DATA CARD:

Name	NO.	F1	F2	F3	F4	F5	F6	F7
F4	2	500.	2.	-1.	8.	-5.	6.	

NO. 3



Given:

Magnitude: $|\bar{F}|$
Slope: x_s, y_s, z_s
Point: x,y,z

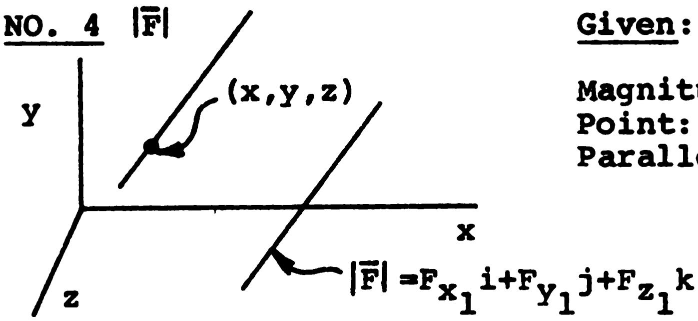
Example:

Name=F11
$|\bar{F}|$ =300.lb$_f$
x_s =2.
y_s =3.
z_s =-4.
x=3.
y=-1.
z=4.

DATA CARD:

Name	NO.	F1	F2	F3	F4	F5	F6	F7
F11	3	300.	2.	3.	-4.	3.	-1.	4.

NO. 4



Given:

Magnitude: $|\bar{F}|$
Point: x,y,z
Parallel to F_1: $F_{x_1}, F_{y_1}, F_{z_1}$

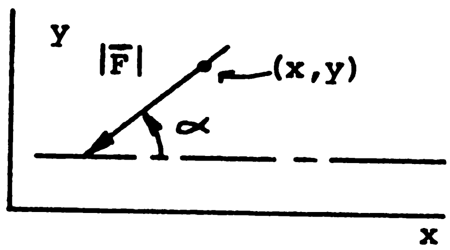
Example:

Name=F
$|\bar{F}|$ =250.lb$_f$
F_{x_1} =3.
F_{y_1} =2.
F_{z_1} =-10.
x=0
y=0
z=0

DATA CARD:

Name	No.	F1	F2	F3	F4	F5	F6	F7
F	4	250.	3.	2.	-10.			

NO. 5 (For two-dimensional only)



Given:

Magnitude: |F̄|
Point: x,y
Smallest angle F makes with the x-axis: ∝ (always positive)
Directions in the x and y directions: l,m

Example:

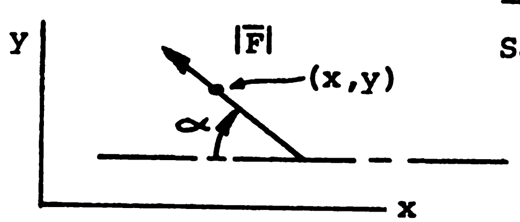
Name=F3
∝ =30°
l =-1.
m =-1.

$|\bar{F}|$ =1000.lb$_f$
x =2.
y =-4.

DATA CARD:

Name	NO.	F1	F2	F3	F4	F5	F6	F7
F3	5	1000.	30.	-1.	-1.	2.	-4.	

NO. 5 (Continued)



Given:

Same as above

Example:

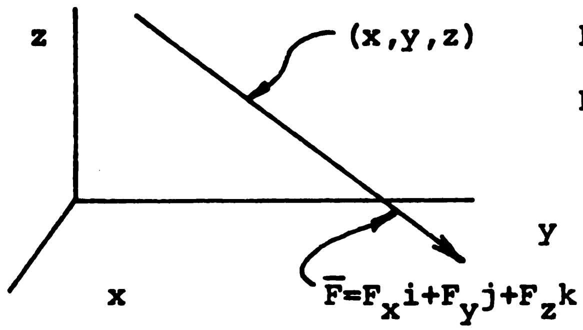
Name=F7
∝ =35°
l =-1.
m =1.

$|\bar{F}|$ =100.lb$_f$
x =4.
y =3.

DATA CARD:

Name	NO.	F1	F2	F3	F4	F5	F6	F7
F7	5	100.	35.	-1.	1.	4.	3.	

NO. 6



Given:

Force $\bar{F}$: F_x, F_y, F_z

Point: x,y,z

Example:

Name=FO
F_x =3.
F_y =-7.
F_z =12.

x=2.5
y=-1.8
z=8.

DATA CARD:

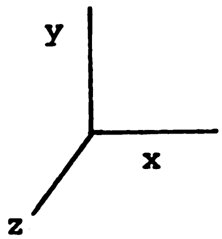
Name	NO.	F1	F2	F3	F4	F5	F6	F7
FO	6		3.	-7.	12.	2.5	-1.8	8.

I. Forces:

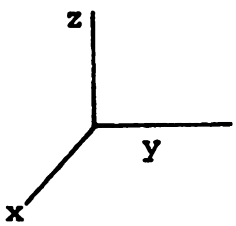
B. Distributed

LIMITATION on coordinate axes:

NO. 1



NO. 2



For a given problem, the axes must be the same for distributed forces, forces, moments, points and lines.

w(x) is the distributed load:

$w(x)=F(x)=Ax^2+Bx+C+Dx^{.5}$

In the computer:

Func(x)=A*(x-R)**2+B*(x-R)+C+D*(x-R)**.5

<u>LIMITATION</u> on the origin of the axes:

The origin must be at the left end of the member.

<u>LIMITATION</u> on the coefficients of func(x). (A,B,C,D)

All coefficients must be zero except one for each distributed load.

<u>LIMITATION</u> on R, R1, and R2 shown below:

R, R1, and R2 must always be positive or zero as measured from the origin to the right.

Data Cards:

Name	NO.	A	B	C	D	E	G	R	R1	R2
A4	I4	F7.0	F7.0	F7.0	F7.0	F7.0	F7.0	F7.0	F7.0	F7.0
1-4	5-8	9-16	17-24	25-32	33-40	41-48	49-56	57-64	65-72	73-80

<u>Note</u>:

R is the distance the curve has been translated to the right from standard position.

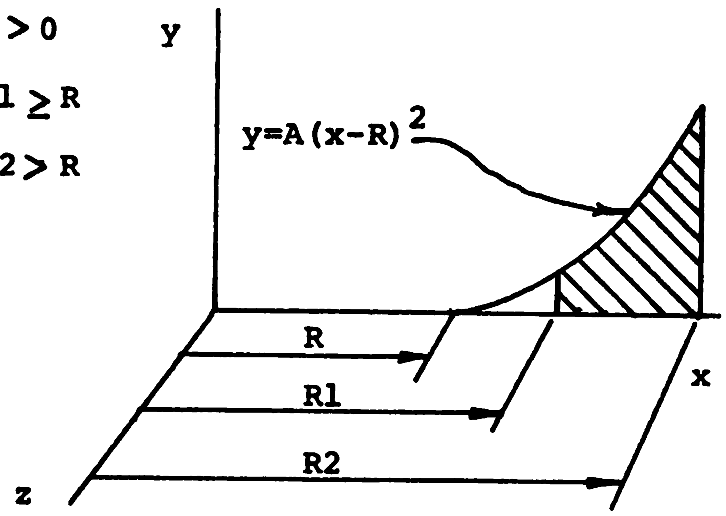
EXAMPLES OF DLOAD

CASE I:

A > 0

R1 ≥ R

R2 > R



Given:

Name=D1
NO=1
A=2.
B=0
C=0
D=0
C=0
G=0
R=10.
R1=11.
R2=15.

DATA CARD:

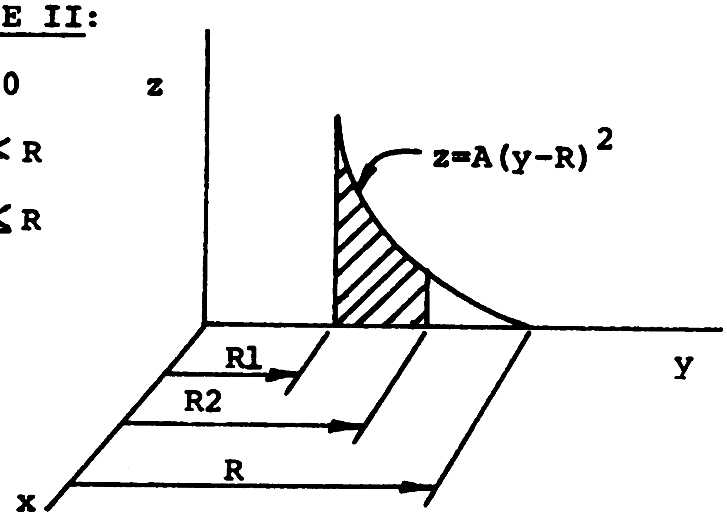
Name	NO.	A	B	C	D	E	G	R	R1	R2
D1	1	2.						10.	11.	15.

CASE II:

A > 0

R1 < R

R2 ≤ R



Given:

Name=D2
NO=2
A=1.
B=0
C=0
D=0
E=0
G=0
R=8.
R1=4.
R2=6.

DATA CARD:

Name	NO.	A	B	C	D	E	G	R	R1	R2
D2	2	1.						8.	4.	6.

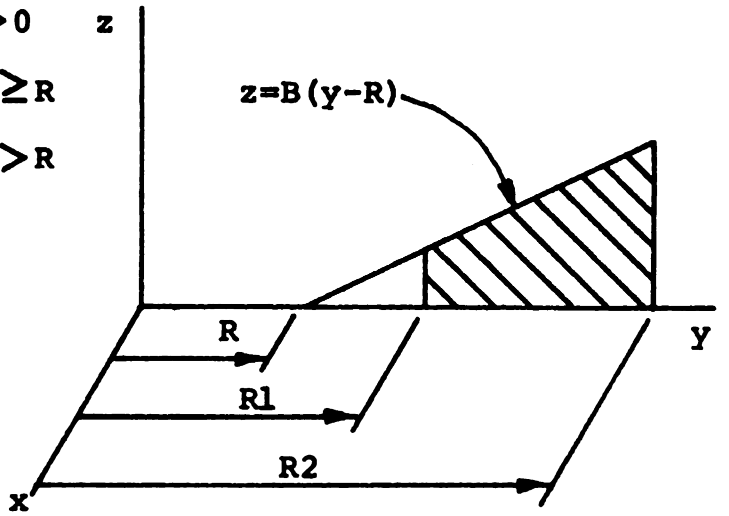
NOTE: A CANNOT BE LESS THAN ZERO.

CASE III:

B > 0

R1 ≥ R

R2 > R



Given:

Name=D3
NO=2
A=0
B=.5
C=0
D=0
E=0
G=0
R=4.
R1=6.
R2=10.

DATA CARD:

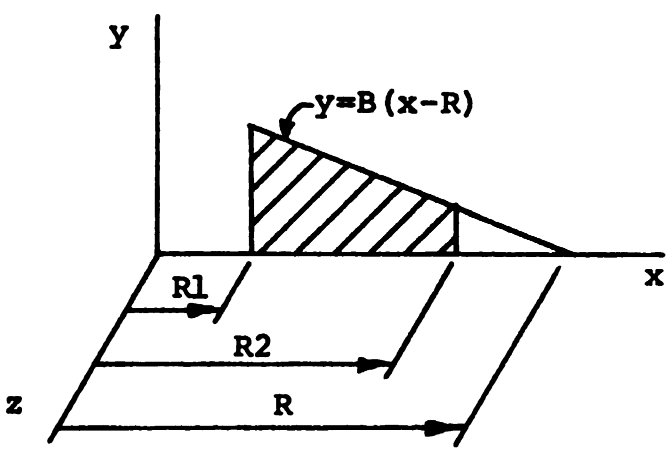
Name	NO.	A	B	C	D	E	G	R	R1	R2
D3	2		.5					4.	6.	10.

CASE IV:

B < 0

R1 < R

R2 ≤ R



Given:

Name=D4
NO=1
A=0
B=-1.
C=0
D=0
E=0
G=0
R=12.
R1=3.
R2=10.

DATA CARD:

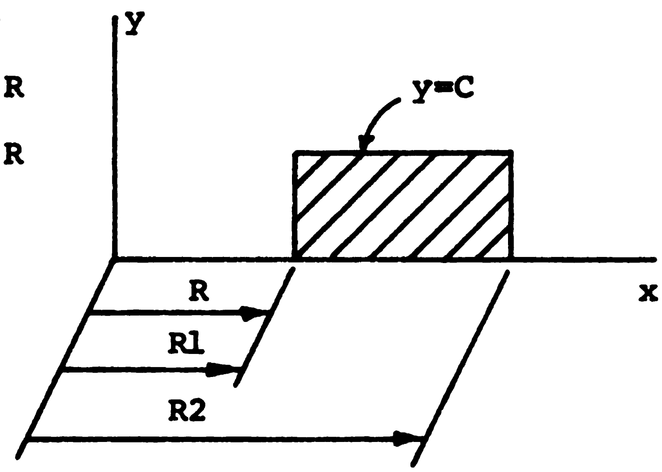
Name	NO.	A	B	C	D	E	G	R	R1	R2
D4	1		-1.					12.	3.	10.

CASE V:

$C > 0$

$R1 \geq R$

$R2 > R$



Given:

Name=D5
NO=1
A=0
B=0
C=25.
D=0
E=0
G=0
R=10.
R1=10.
R2=20.

DATA CARD:

Name	NO.	A	B	C	D	E	G	R	R1	R2
D5	1			25.				10.	10.	20.

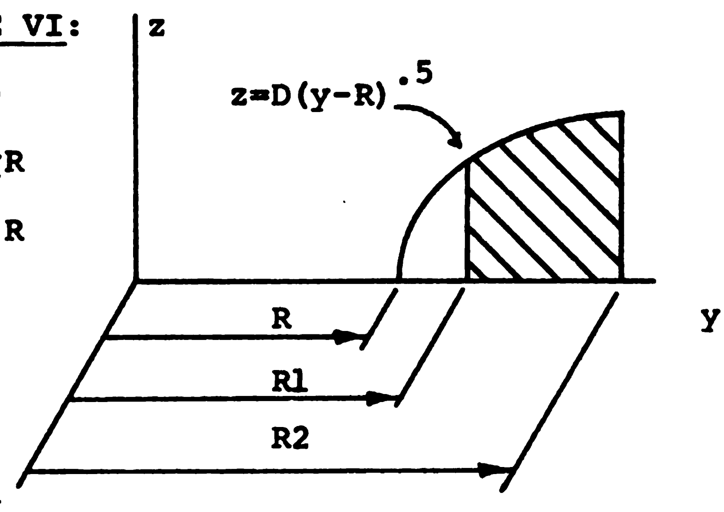
NOTE: C CANNOT BE LESS THAN ZERO.

CASE VI:

$D > 0$

$R1 \geq R$

$R2 > R$



Given:

Name=D6
NO=2
A=0
B=0
C=0
D=2.
E=0
G=0
R=10.
R1=11.
R2=17.

DATA CARD:

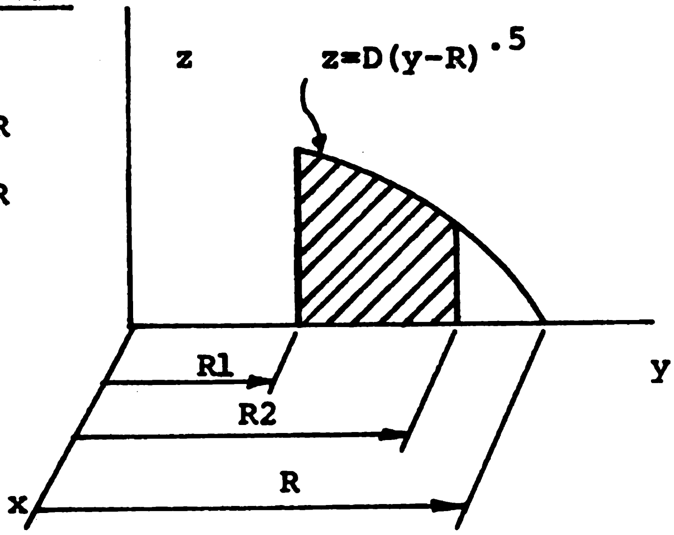
Name	NO.	A	B	C	D	E	G	R	R1	R2
D6	2				2.			10.	11.	17.

CASE VII:

$D < 0$

$R1 < R$

$R2 \leq R$



Given:

Name=D7
NO=2
A=0
B=0
C=0
D=-2.
E=0
G=0
R=10.
R1=5.
R2=8.

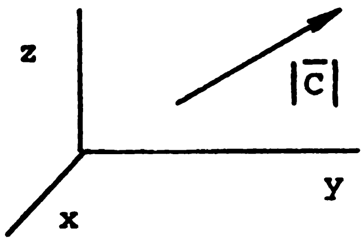
DATA CARD:

Name	NO.	A	B	C	D	E	G	R	R1	R2
D7	2				-2.			10.	5.	8.

IV. Moments: (Moments or couples)

Name	NO.	F1	F2	F3	F4	F5	F6	F7
	1	$\|\bar{C}\|$	l	m	n			
	2		C_x	C_y	C_z			
A4	I4	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0
1-4	5-8	9-16	17-24	25-32	33-40	41-48	49-56	57-64

NO. 1



Given:

Magnitude: $|\bar{C}|$
Direction cosines: l,m,n

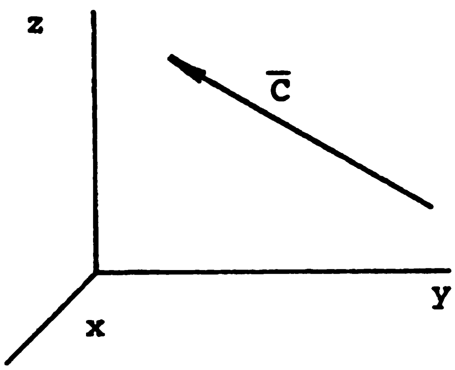
Example:

Name=C3 $|\bar{C}|$=300. l=-.716 m=.275 n=.642

DATA CARD:

Name	NO.	F1	F2	F3	F4	F5	F6	F7
C3	1	300.	-.716	.275	.642			

NO. 2



Given:

$\bar{C}$: C_x, C_y, C_z

Example:

Name=C1 C_x=3. C_y=4. C_z=-5.

DATA CARD:

Name	NO.	F1	F2	F3	F4	F5	F6	F7
C1	2		3.	4.	-5.			

II. **Points**:

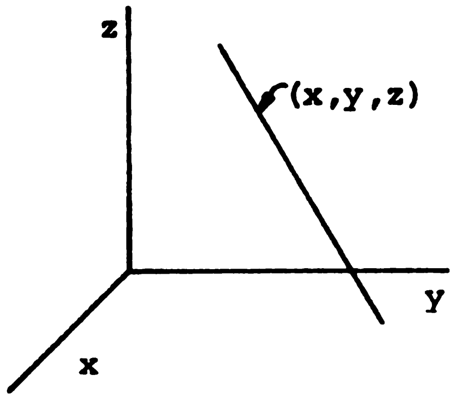
III. **Lines**:

Name	NO.	F1	F2	F3	F4	F5	F6	F7
	1		l	m	n	x	y	z
	2		x_1	y_1	z_1	x_2	y_2	z_2
	3					x	y	z

A4 I4 F8.0 F8.0 F8.0 F8.0 F8.0 F8.0 F8.0

1-4 5-8 9-16 17-24 25-32 33-40 41-48 49-56 57-64

NO. 1(Line)



Given:

Direction cosines: l,m,n
Point: x,y,z

Example:

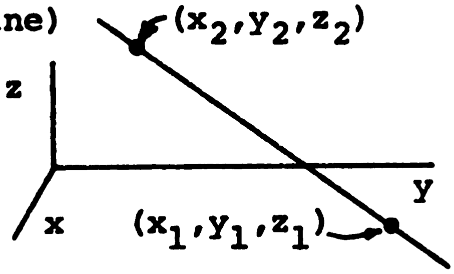
Name=L1
NO =1
l =-.716
m =.275
n =.642

x=2.
y=4.
z=-3.

DATA CARD:

Name	NO.	F1	F2	F3	F4	F5	F6	F7
L1	1		-.716	.275	.642	2.	4.	-3.

NO. 2 (Line)



Given:

Two points:
From (x_1, y_1, z_1)
to (x_2, y_2, z_2)

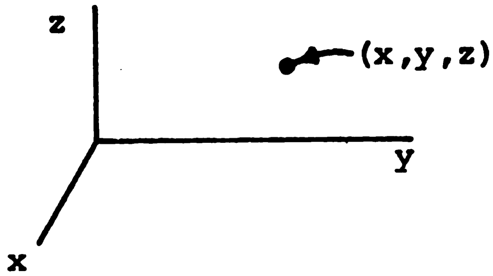
Example:

Name=L4
NO =2
x_1 =5.
y_1 =3.
z_1 =0.
x_2=-5.
y_2=3.
z_2=-1.

DATA CARD:

Name	NO.	F1	F2	F3	F4	F5	F6	F7
L4	2		5.	3.		-5.	3.	-1.

NO. 3 (Point)



Given:

Point: x,y,z

Example:

Name=P1
NO =3
x =5.
y =2.
z =-3.

DATA CARD:

Name	NO.	F1	F2	F3	F4	F5	F6	F7
P1	3					5.	2.	-3.

The preceding material has been condensed for your convenience. All of the data has been placed on one sheet. You will be given a loose sheet of this form for use when solving problems. Another special form has been developed to assist you in tabulating the data. Loose copies of this sheet will be provided to you as you need them. Copies of the forms follow.

You will be provided with a special card for the keypunch drum which will cause the keypunch to tabulate to the desired column when you strike the skip key.

SUMMARY OF DATA

FORCES: CONCENTRATED

NAME	NO	F1	F2	F3	F4	F5	F6	F7		
	1	F	l	m	n	x	y	z		
	2	F	x_1	y_1	z_1	x_2	y_2	z_2		
	3	F	x_s	y_s	z_s	x	y	z		
	4	F	F_x	F_y	F_z	x	y	z		
	5	F		l	m	x	y			
	6		F_x	F_y	F_z	x	y	z		

PTLINE:

NAME	NO	F1	F2	F3	F4	F5	F6	F7		
	1		l	m	n	x	y	z		
	2		x_1	y_1	z_1	x_2	y_2	z_2		
	3					x	y	z		

COUPLE:

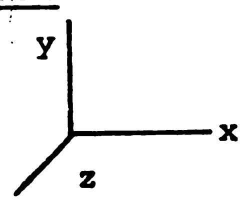
NAME	NO	F1	F2	F3	F4	F5	F6	F7		
	1	C	l	m	n					
	2		C_x	C_y	C_z					

DLOAD:

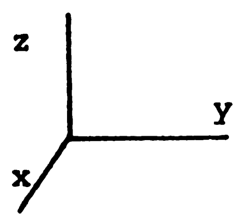
NAME	NO	A	B	C	C	E	G	R	R1	R2

FOR DLOAD ONLY:

NO=1



NO=2



TABULATION OF DATA

NAME	NO	F1	F2	F3	F4	F5	F6	F7		
		A	B	C	D	E	G	R	R1	R2

1 45 89 1617 2425 3233 4041 4849 5657 6465 7273 80

|← DECIMAL POINT →|

HOW TO ENTER DATA

I. Forces:

A. Concentrated:

Data is read into the computer by calling

SUBROUTINE FORCES

The subroutine continues reading forces (one force/card) until it reaches a BLANK CARD.

LIMITATION: Only SIX FORCES may be read at a time. This includes both concentrated and distributed forces.

B. Distributed:

Data is read into the computer by calling

SUBROUTINE DLOAD

The subroutine continues reading forces (one force/card) until it reaches a BLANK CARD.

II. Points:

III. Lines:

Point and line data is read into the computer by calling

SUBROUTINE PTLINE

The subroutine continues reading data (one point or line/card) until it reaches a BLANK CARD.

LIMITATION: Only five points and/or lines can be read at a time.

IV. Moments:

Data is read into the computer by calling

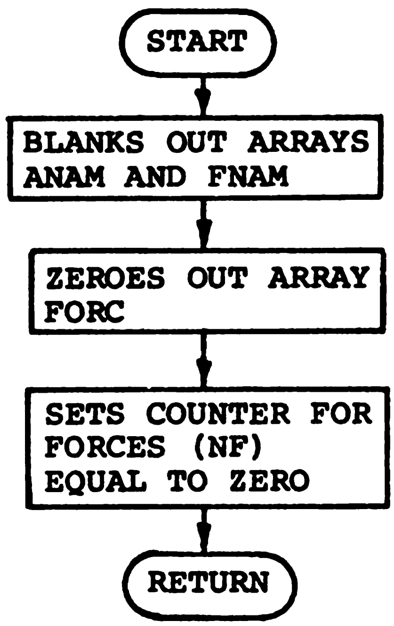
SUBROUTINE COUPLE

The subroutine continues reading data (one moment/ card) until it reaches a BLANK CARD.

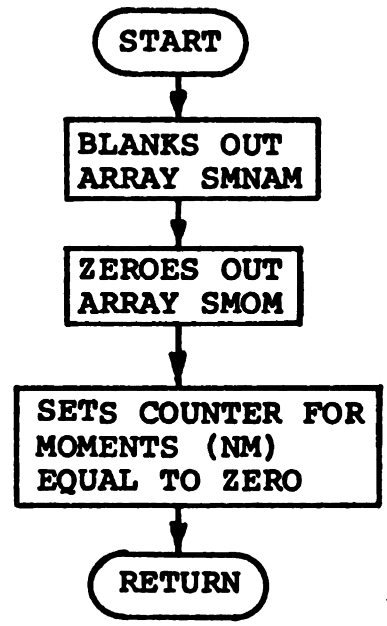
LIMITATION: Only six moments can be read at at time.

The following are flow diagrams of the subroutines and diagrams showing the relationships between subroutines and arrays.

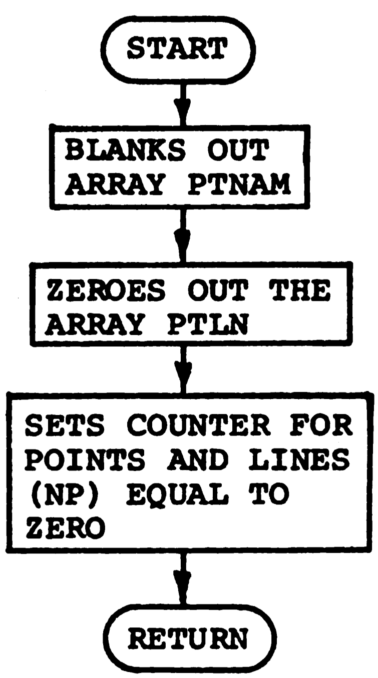
SUBROUTINE ZFORC



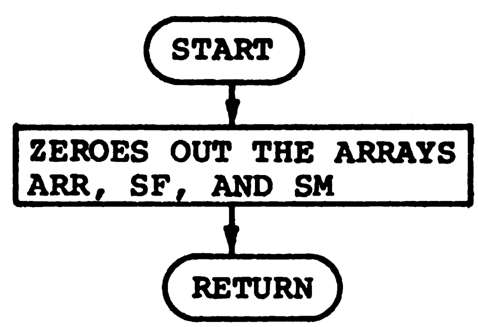
SUBROUTINE ZSMOM



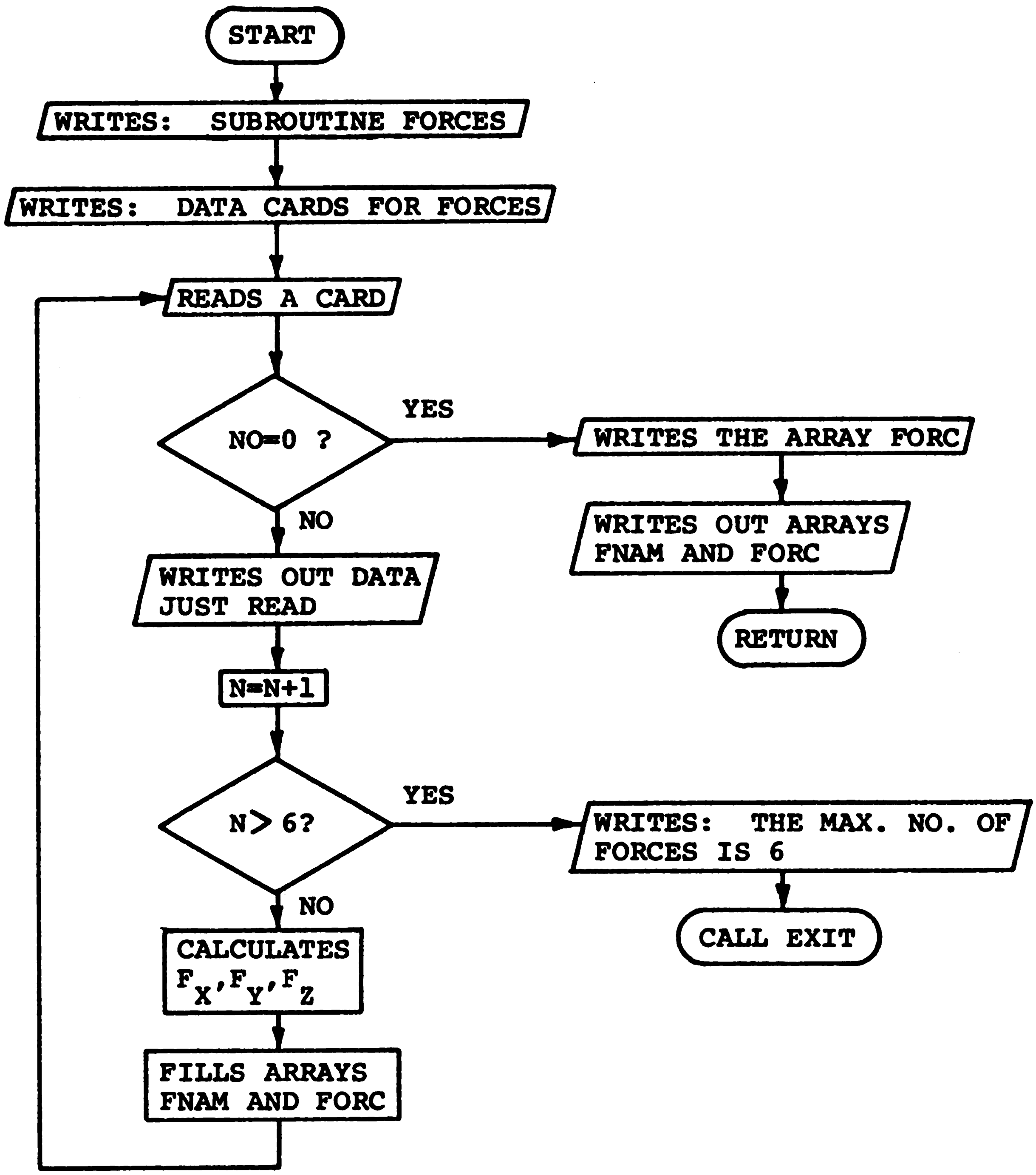
SUBROUTINE ZPTLN



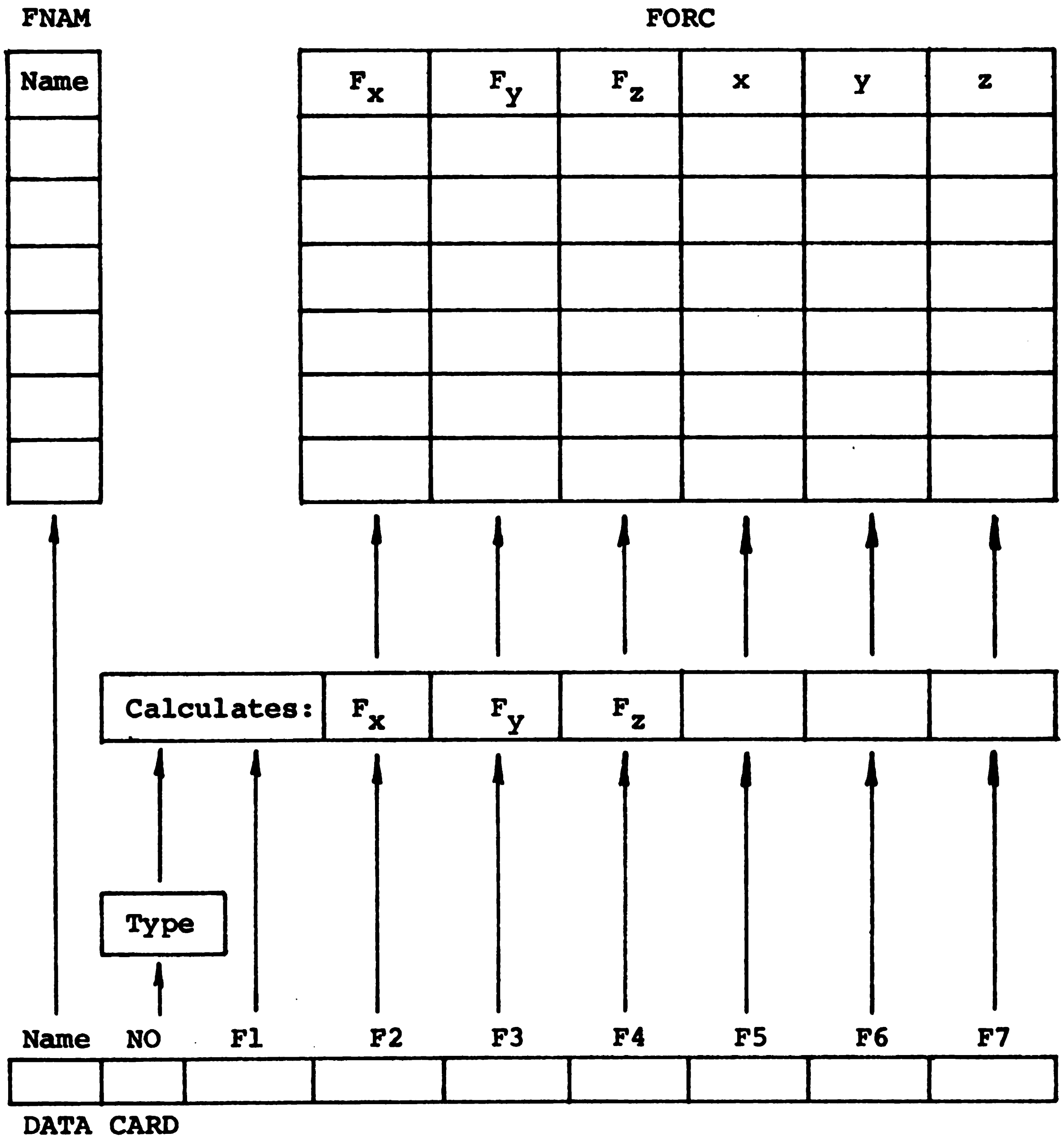
SUBROUTINE ZARR



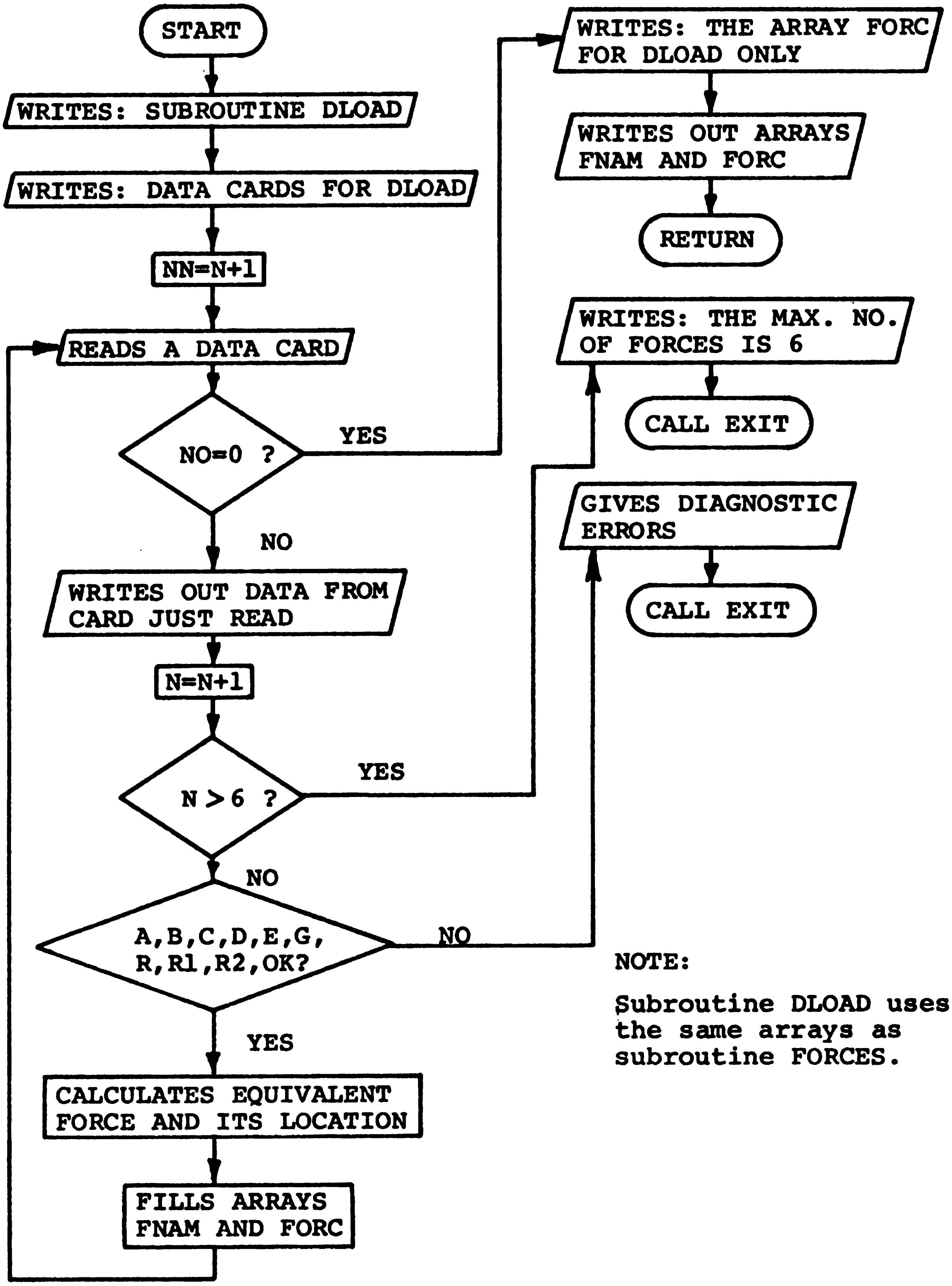
SUBROUTINE FORCES



ARRAYS USED WITH SUBROUTINE FORCES



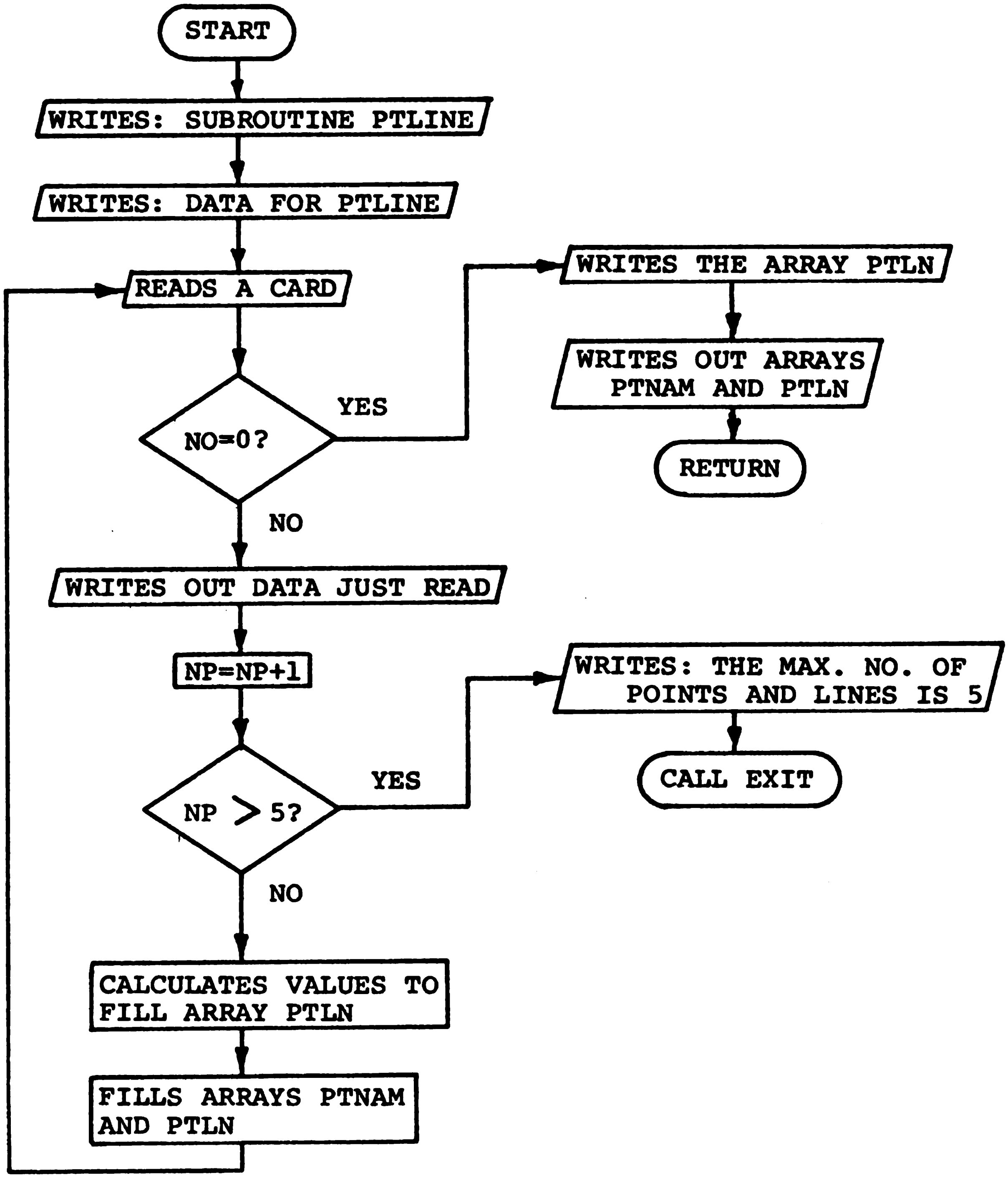
SUBROUTINE DLOAD



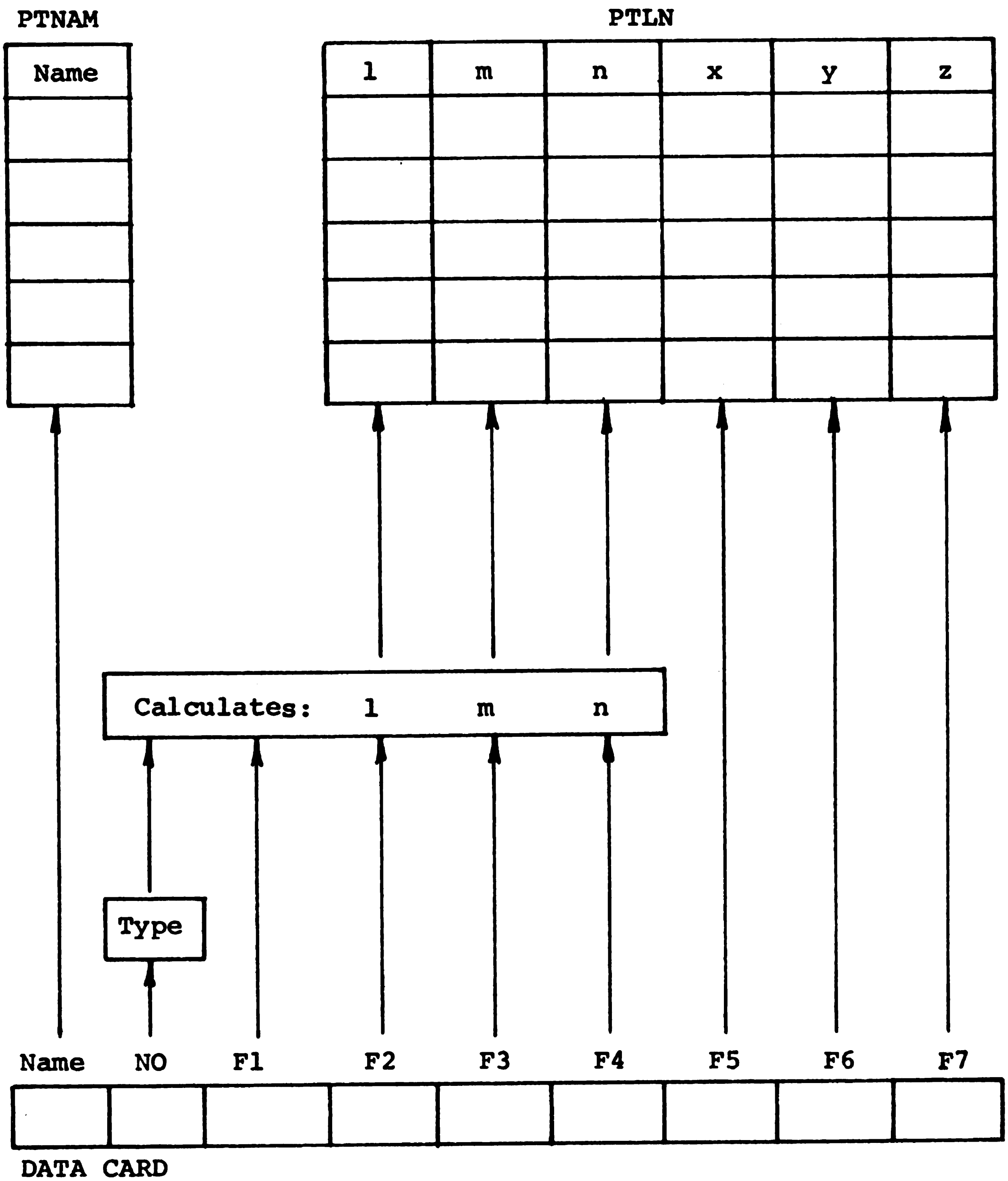
NOTE:

Subroutine DLOAD uses the same arrays as subroutine FORCES.

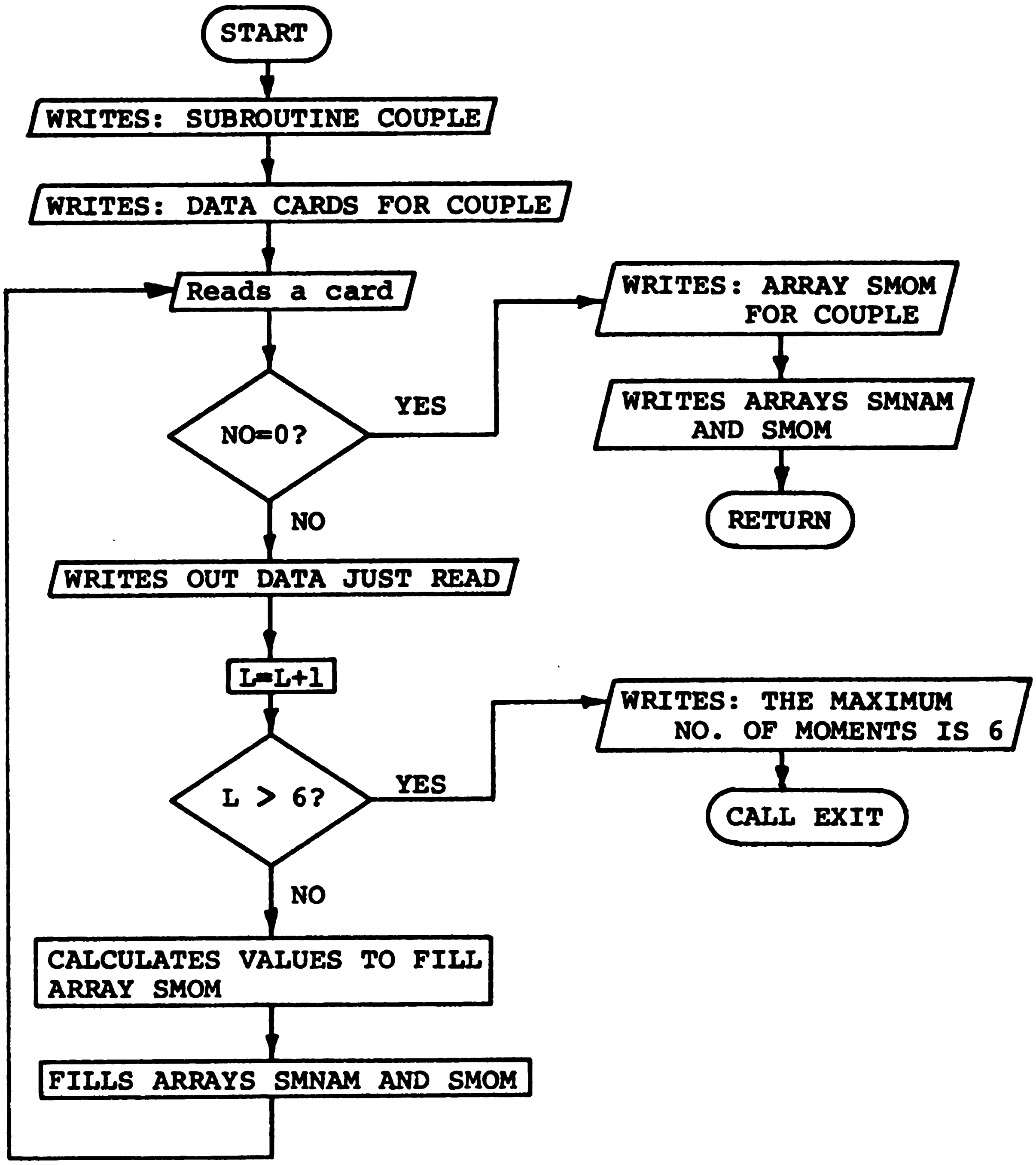
SUBROUTINE PTLINE



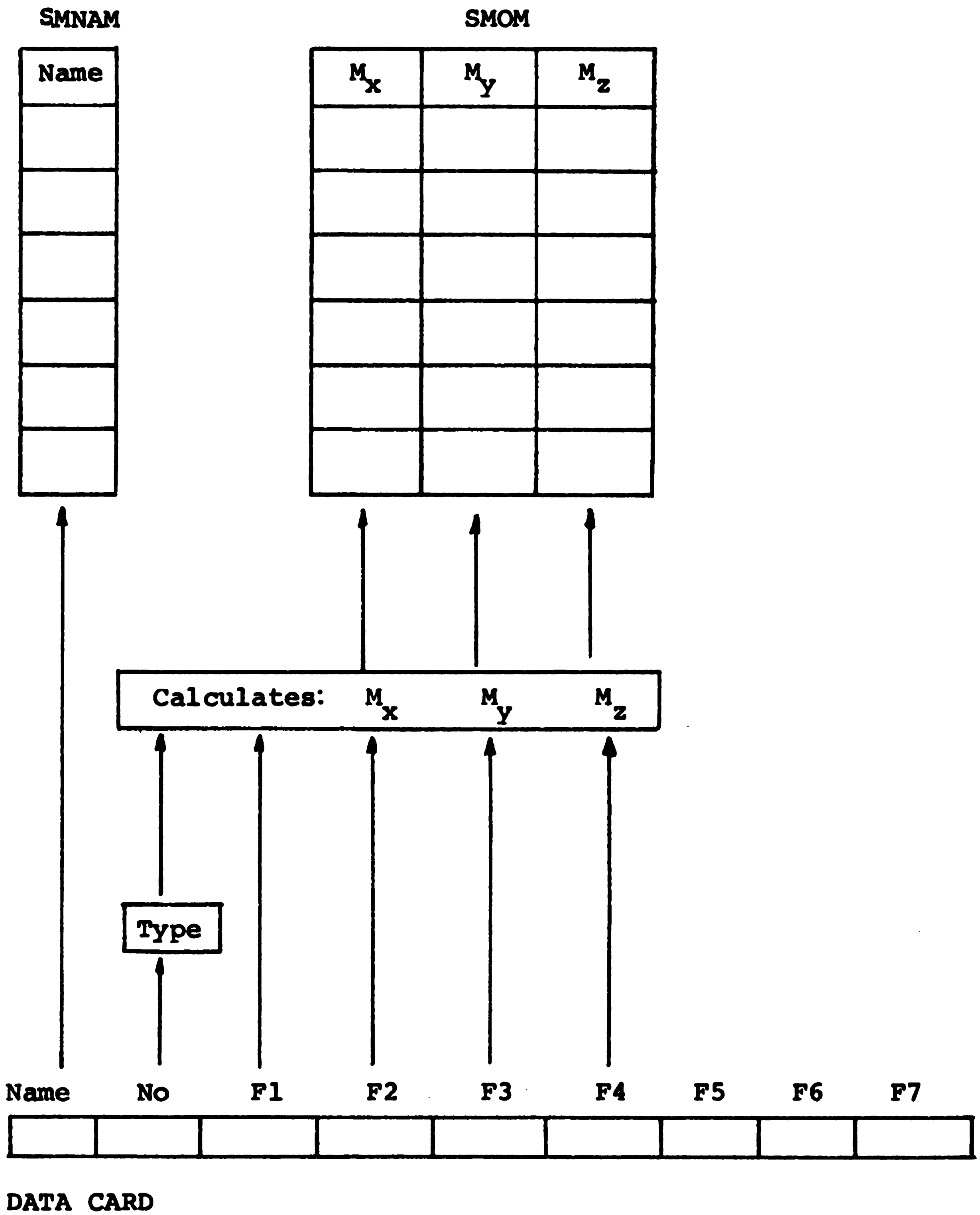
ARRAYS USED WITH SUBROUTINE PTLINE



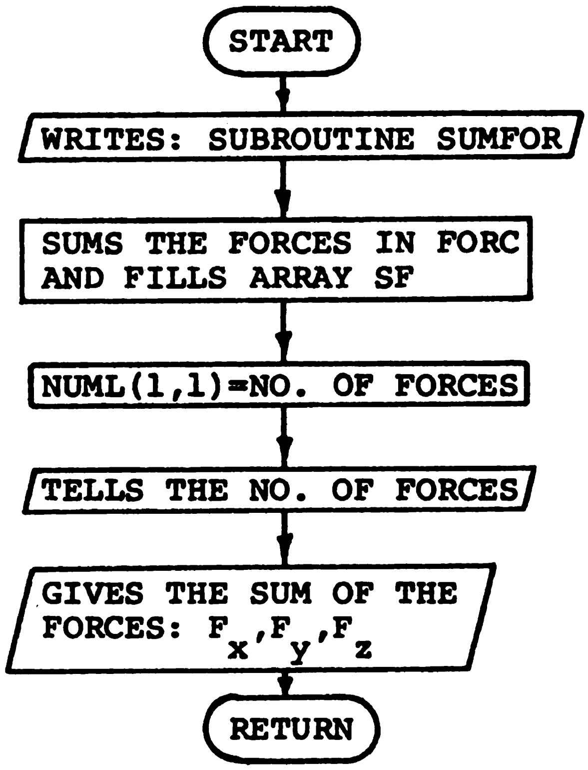
SUBROUTINE COUPLE



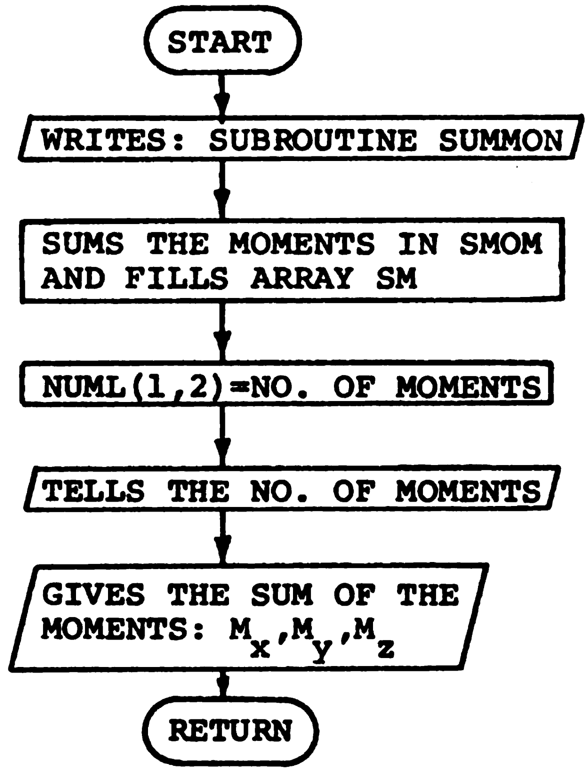
ARRAYS USED WITH SUBROUTINE COUPLE



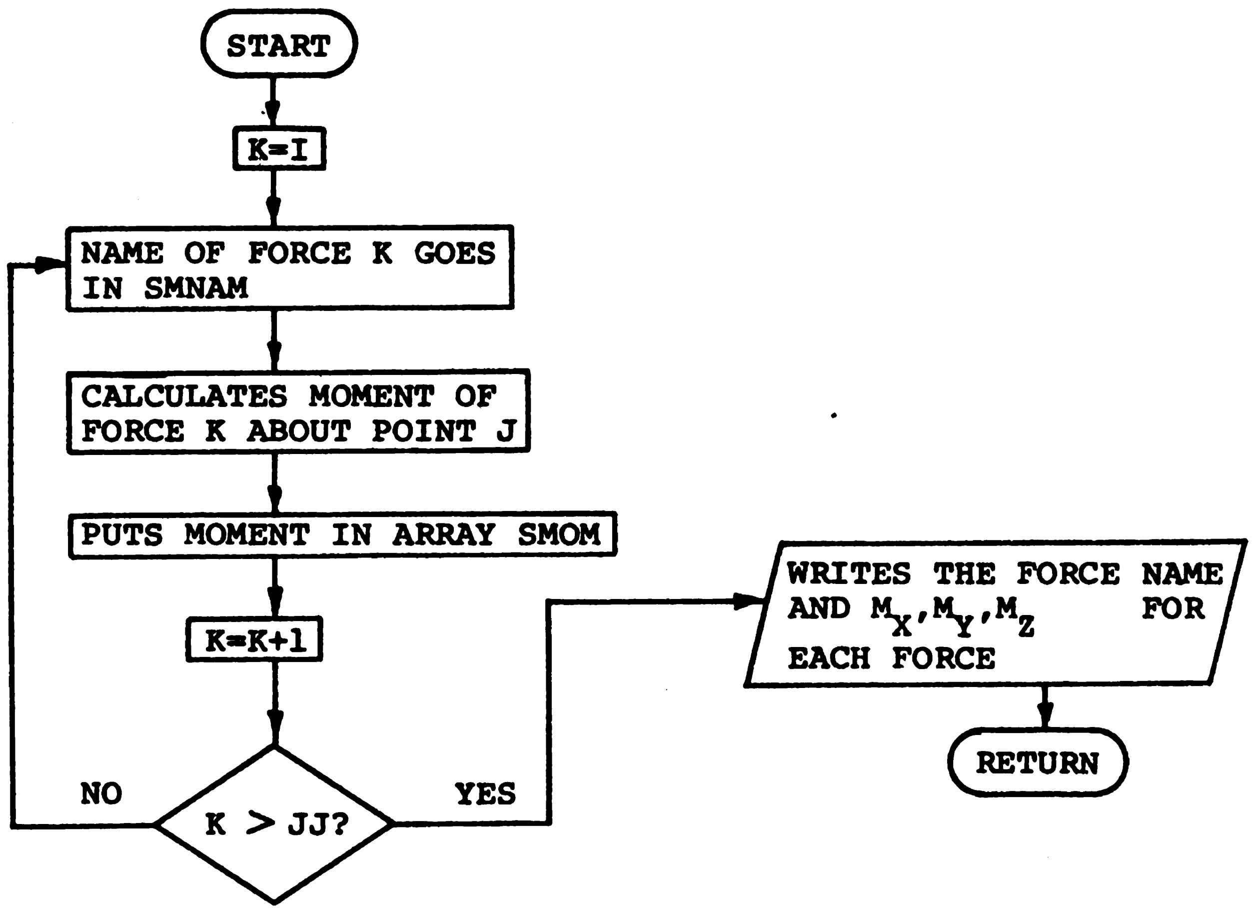
SUBROUTINE SUMFOR



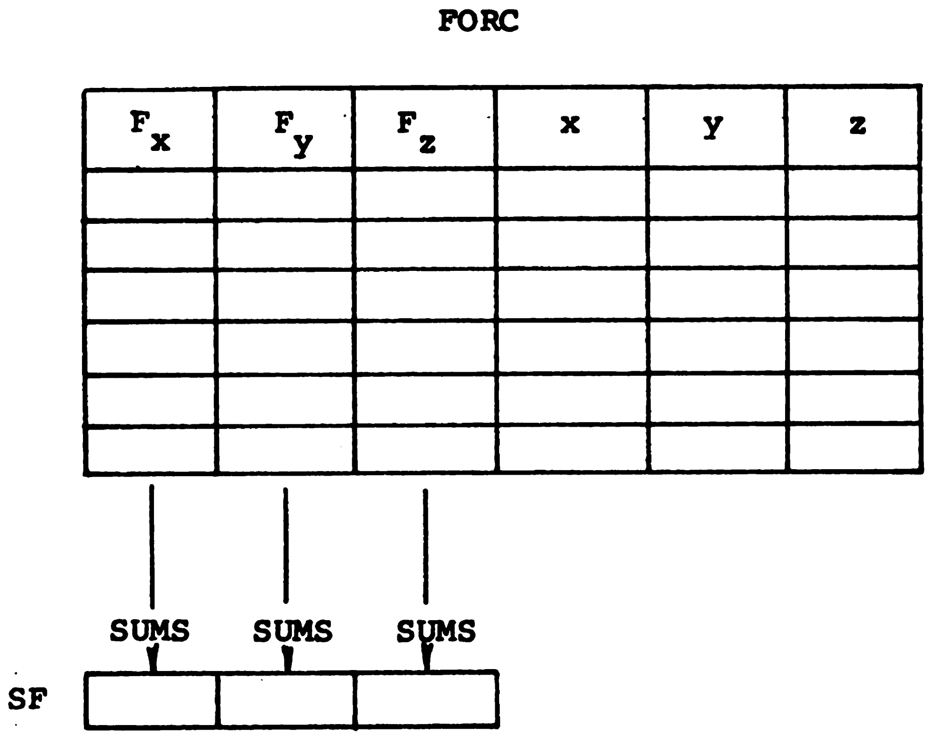
SUBROUTINE SUMMON



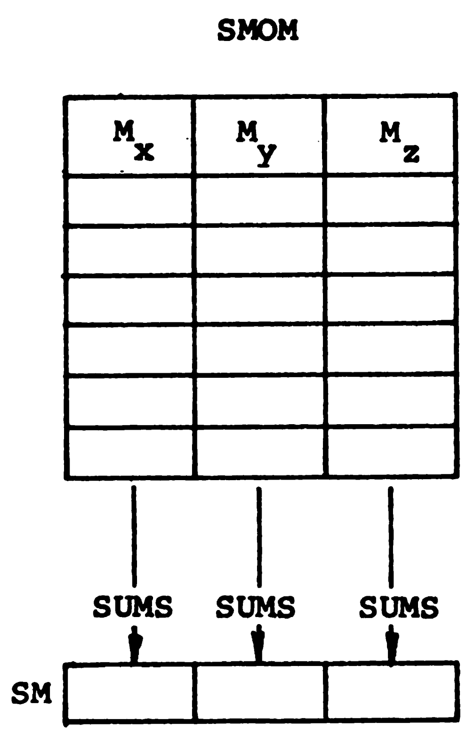
SUBROUTINE MOMPT



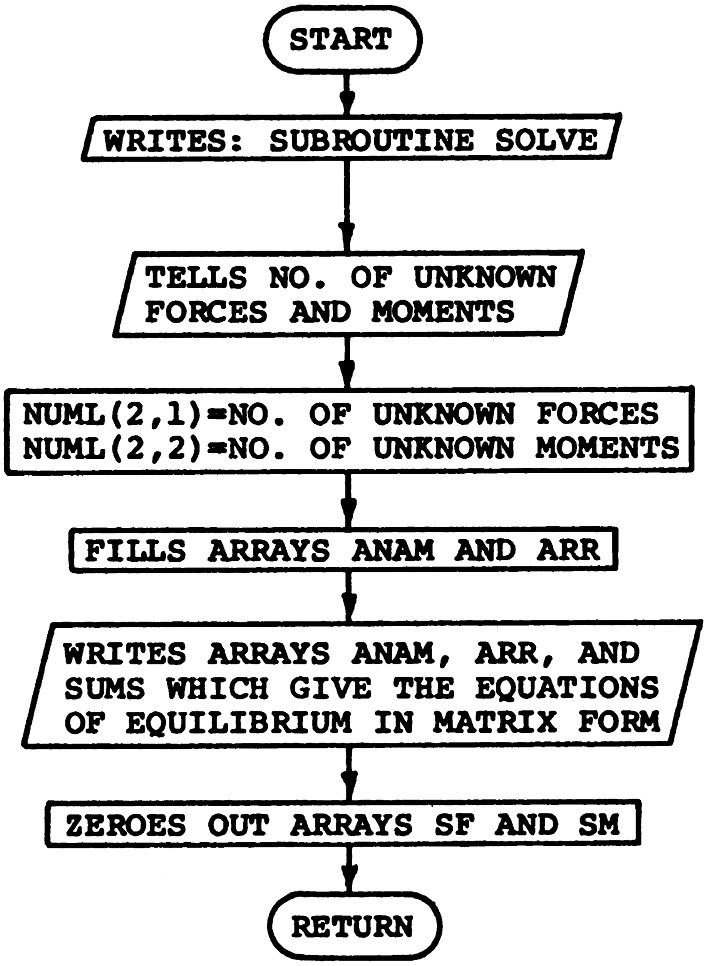
ARRAYS USED WITH SUMFOR



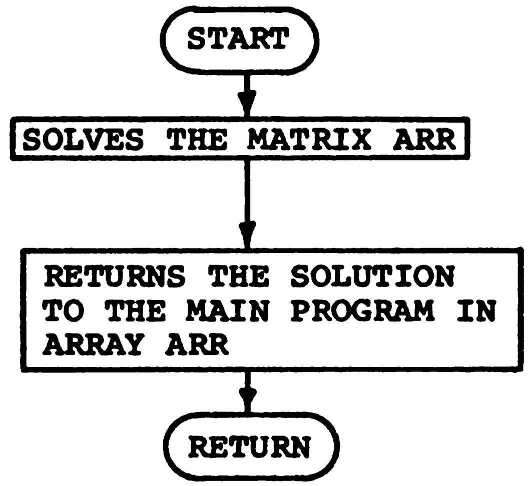
ARRAYS USED WITH SUBROUTINE SUMMOM



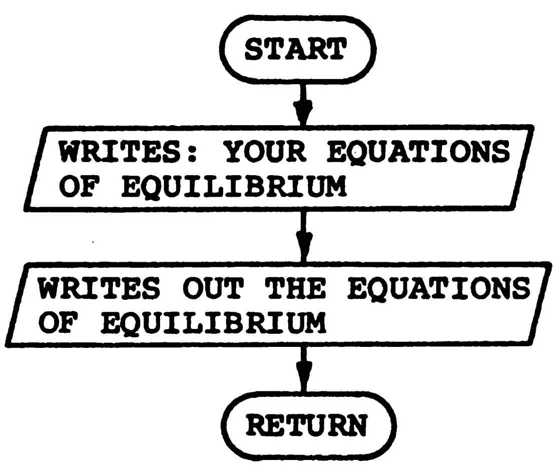
SUBROUTINE SOLVE



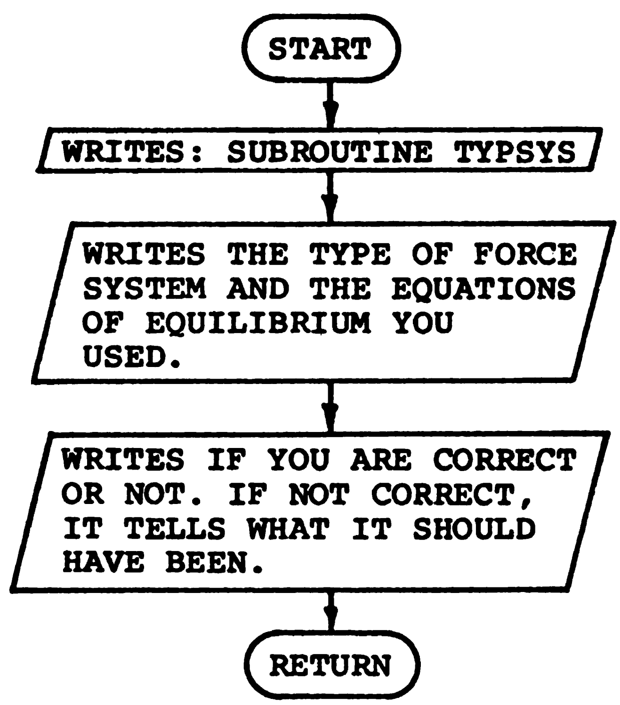
SUBROUTINE MFGRR



SUBROUTINE EQUA



SUBROUTINE TYPSYS



ARRAYS USED WITH SUBROUTINE SOLVE

FNAM

Name
F1
F2
F3

FORC

F_x	F_y	F_z	x	y	z
Fx_1	Fy_1	Fz_1	x	y	z
Fx_2	Fy_2	Fz_2	x	y	z
Fx_3	Fy_3	Fz_3	x	y	z

SMNAM

Name
M1
M2
F1
F2
F3

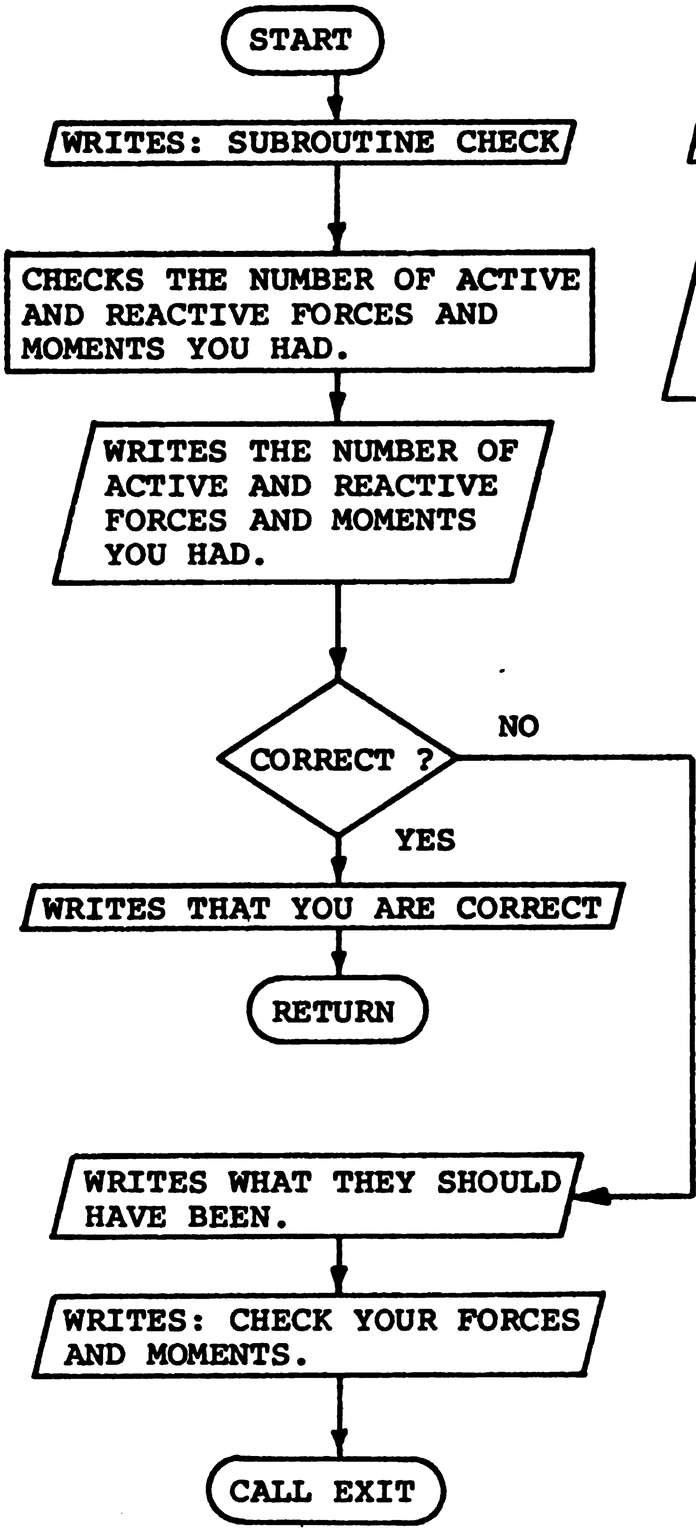
SMOM

M_x	M_y	M_z
$M1_x$	$M1_y$	$M1_z$
$M2_x$	$M2_y$	$M2_z$
$MF1_x$	$MF1_y$	$MF1_z$
$MF2_x$	$MF2_y$	$MF2_z$
$MF3_x$	$MF3_y$	$MF3_z$

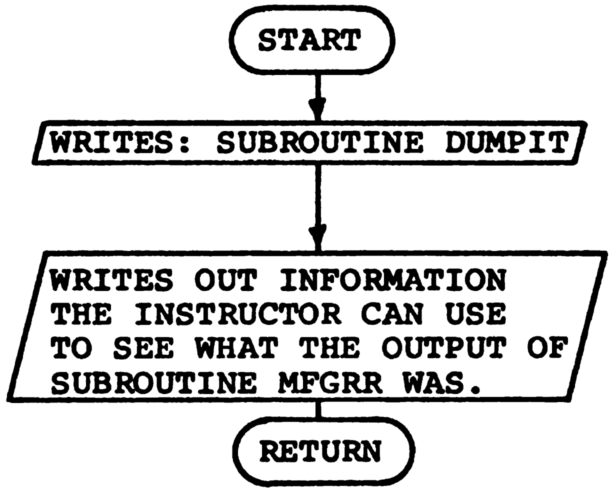
SUMS	SFX	SFY	SFZ	SMX	SMY	SMZ
SF	SF(1)	SF(2)	SF(3)			
SM	SM(1)	SM(2)	SM(3)			

							SF and SM	SUMS
ANAM	F1	F2	F3	M1	M2			
ARR	Fx_1	Fx_2	Fx_3				SF(1)	SFX=0
	Fy_1	Fy_2	Fy_3				SF(2)	SFY=0
	Fz_1	Fz_2	Fz_3				SF(3)	SFZ=0
	$MF1_x$	$MF2_x$	$MF3_x$	$M1_x$	$M2_x$		SM(1)	SMX=0
	$MF1_y$	$MF2_y$	$MF3_y$	$M1_y$	$M2_y$		SM(2)	SMY=0
	$MF1_z$	$MF2_z$	$MF3_z$	$M1_z$	$M2_z$		SM(3)	SMZ=0

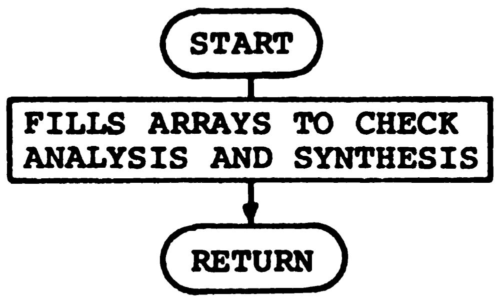
SUBROUTINE CHECK



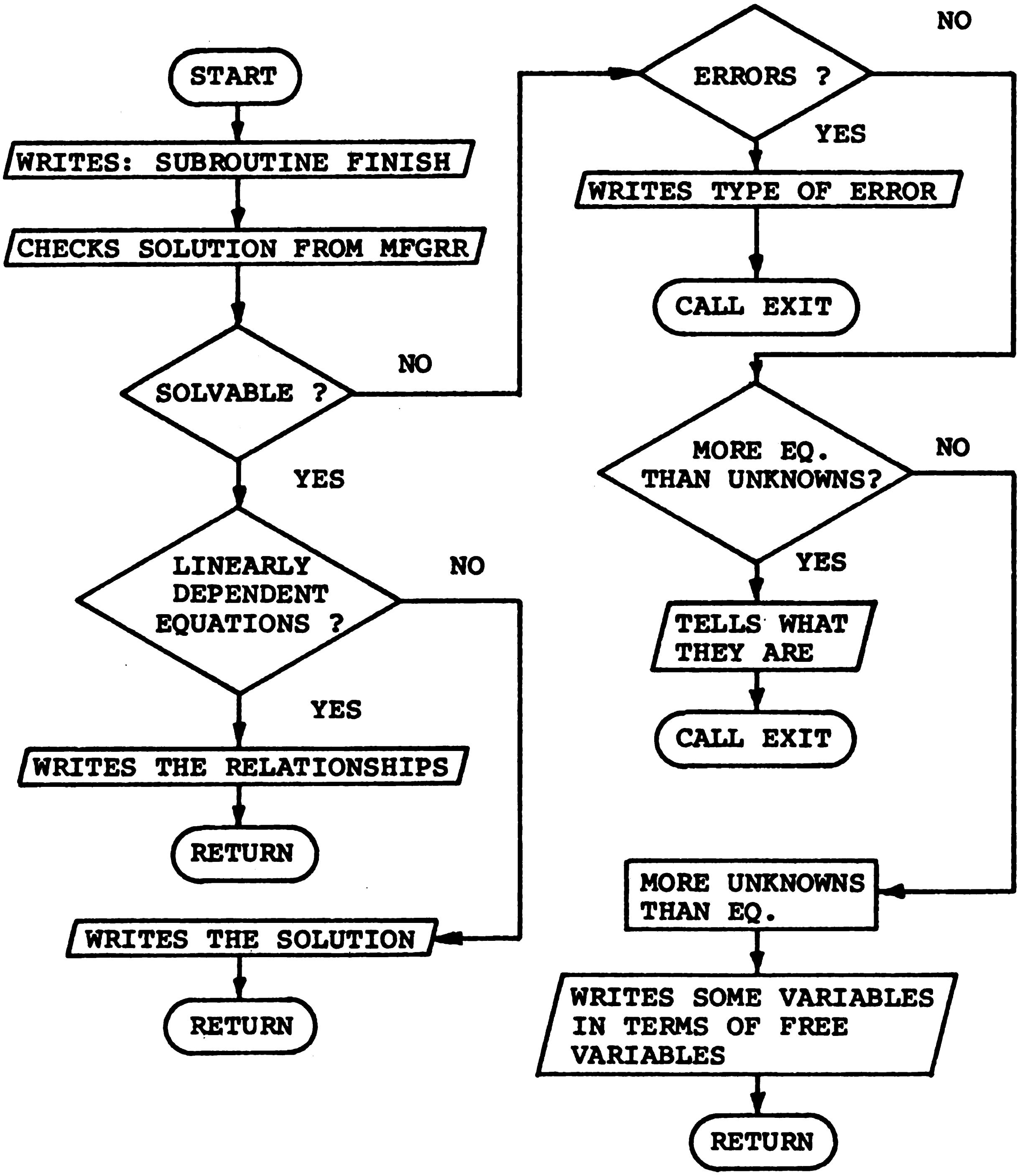
SUBROUTINE DUMPIT



SUBROUTINES FILKMC, FILKME,FILKMM



SUBROUTINE FINISH



MAIN PROGRAM

The main program consists of call statements for subroutines. All cards for the program will be given to you. A print-out of the main program, which is to be used for the solution of five problems, follows.

DECK

A deck of computer cards will be given to you which includes all cards needed to use the computer program except the calls for subroutines and the data cards. The cards in the deck will have nothing typed on them except a number. They will be numbered sequentially from 2-84 so that you may easily put them back in order in case you drop the deck. A computer print-out of the deck follows which shows you what is punched on each card.

The portion of the main program which consists of call statements for subroutines is to be placed between cards 26 and 27 and the data cards between cards 83 and 84 of the deck.

There are six possible error messages which will be given as ERROR __. The following is an explanation of what each error means:

1. No rows or columns in the matrix.
2. The zero matrix.
3. The number of equations is less than the rank of the matrix.
4. More equations than unknowns.
5. All sums are zero for the active forces and moments. Therefore, the body was in equilibrium without any reactive forces or moments.
6. Inconsistent set of equations.

COMPUTER PRINT-OUT OF MAIN PROGRAM

```
      COMMON FORC(6,6),SMOM(6,3),SF(3),SM(3),KM(5,7),NUML(2,2)             000007
       COMMON PTLN(5,6),PTNAM(5),F(7),SYSTEM(6,6),NR(6),NC(7)              000008
      COMMON ARR(6,7),FNAM(6),SMNAM(6),SUMS(6),SCH(1),ANAM(6)              000009
      COMMON IROWM(6),ICOLM(7)                                             909
      DIMENSION IROW(6),ICOL(7),S(42)                                      10
      EQUIVALENCE (ARR(1,1),S(1)),(IROWM(1),IROW(1)),(ICOLM(1),ICOL(1))    000011
      DOUBLE PRECISION DUMPIT,FORCES,SUMFOR,PTLINE,MOMPT,SUMMOM,ZSMOM      12
      DOUBLE PRECISION ZPTLN,ZFORC,SOLVE,ZARR,COUPLE,DLOAD,MFGRR,EQUA      000013
      DOUBLE PRECISION TYPSYS,CHECK,FILKME,FILKMC,FILKMM,FINISH            000014
      READ(1,1)BLANK,(SUMS(I),I=1,6),SCH                                   000015
    1 FORMAT(A4,6A3,A1)                                                    000016
      READ(1,25)((SYSTEM(I,J),J=1,6),I=1,6)                                000017
   25 FORMAT(3(6A4))                                                       000018
      READ(1,5)ZFORC,SOLVE,COUPLE,DLOAD,MFGRR,EQUA                         000019
      READ(1,5)        FORCES,SUMFOR,PTLINE,MOMPT,SUMMOM,ZSMOM,ZPTLN,ZARR  000020
      READ(1,5)TYPSYS,CHECK,FILKME,FILKMC,FILKMM,FINISH,DUMPIT             21
    5 FORMAT(10A8)                                                         000022
      DO 50 KKK=1,5
      CALL LOADER(ZFORC,NF,BLANK)                                          000023
      CALL LOADER(ZSMOM,NM,BLANK)                                          000024
      CALL LOADER(ZPTLN,NP,BLANK)                                          000025
      CALL LOADER(ZARR)                                                    000026
      READ(1,10)K,M
   10 FORMAT(8X,I2,6X,I3)
      WRITE(3,66)K,M
   66 FORMAT(1H1,5X,'CHAPTER',I4,5X,'PROBLEM',I5)
   11 CALL LOADER(FORCES,NF)
   21 CALL LOADER(DLOAD,NF)
   23 CALL LOADER(SUMFOR,NF)
   31 CALL LOADER(PTLINE,NP)
      CALL LOADER(MOMPT,1,1,NF,NM)
   41 CALL LOADER(COUPLE,NM)
   43 CALL LOADER(SUMMOM,NM,NF)
   44 CALL LOADER(ZFORC,NF,BLANK)
      CALL LOADER(ZSMOM,NM,BLANK)
   51 CALL LOADER(FORCES,NF)
   61 CALL LOADER(DLOAD,NF)
   63 CALL LOADER(MOMPT,1,1,NF,NM)
   71 CALL LOADER(COUPLE,NM)
   91 CALL LOADER(FILKME)
   72 CALL LOADER(SOLVE,NF,NM,NOR,NOC,BLANK)
      CALL LOADER(EQUA,NOT)
   94 CALL LOADER(TYPSYS,K,M,NOR)
      CALL LOADER(CHECK,K,M,MM,NOT)
   83 CALL LOADER(MFGRR,ARR,6,7,6,7,1.0E-7,IRANK,IROWM,ICOLM,S)
      CALL LOADER(DUMPIT,NOR,NOC,IRANK)
      CALL LOADER(FINISH,NOR,NOC,IRANK)
   50 CONTINUE
      CALL EXIT                                                            000027
      END                                                                  000028
```

COMPUTER PRINT-OUT OF THE DECK

```
// BCSS OUTPUT=2C00
// JCB FORCES
// OPTION LINK
 PHASE FORTMAIN,ROOT
// EXEC FORTRAN
      COMMON FORC(6,6),SMOM(6,3),SF(3),SM(3),KM(5,7),NUML(2,2)
       COMMON PTLN(5,6),PTNAM(5),F(7),SYSTEM(6,6),NR(6),NC(7)
      COMMON ARR(6,7),FNAM(6),SMNAM(6),SUMS(6),SCH(1),ANAM(6)
      DIMENSION IROW(6),ICOL(7),S(42),IROWM(6),ICOLM(7)
      EQUIVALENCE (ARR(1,1),S(1)),(IROWM(1),IROW(1)),(ICOLM(1),ICOL(1))
      DOUBLE PRECISION        FORCES,SUMFOR,PTLINE,MOMPT,SUMMOM,ZSMOM
      DOUBLE PRECISION ZPTLN,ZFORC,SOLVE,ZARR,COUPLE,DLOAD,MFGRR,EQUA
      DOUBLE PRECISION TYPSYS,CHECK,FILKME,FILKMC,FILKMM,FINISH
      READ(1,1)BLANK,(SUMS(I),I=1,6),SCH
 1    FORMAT(A4,6A3,A1)
      READ(1,25)((SYSTEM(I,J),J=1,6),I=1,6)
 25   FORMAT(3(6A4))
      READ(1,5)ZFORC,SOLVE,COUPLE,DLOAD,MFGRR,EQUA
      READ(1,5)        FORCES,SUMFOR,PTLINE,MOMPT,SUMMOM,ZSMOM,ZPTLN,ZARR
      READ(1,5)TYPSYS,CHECK,FILKME,FILKMC,FILKMM,FINISH
 5    FORMAT(10A8)
      CALL LOADER(ZFORC,NF,BLANK)
      CALL LOADER(ZSMOM,NM,BLANK)
      CALL LOADER(ZPTLN,NP,BLANK)
      CALL LOADER(ZARR)
      CALL EXIT
      END
/*
 INCLUDE IJTARXI
 INCLUDE IJTSSQT
 INCLUDE IJTSSCN
 INCLUDE IJTSLOG
 INCLUDE IJTEXPN
 INCLUDE IJTARXR
 PHASE COUPLE,*
 INCLUDE COUPLE
 PHASE SOLVE,COUPLE
 INCLUDE SOLVE
 PHASE ZFORC,COUPLE
 INCLUDE ZFORC
 PHASE ZARR,COUPLE
 INCLUDE ZARR
 PHASE ZSMOM,COUPLE
 INCLUDE ZSMOM
 PHASE ZPTLN,COUPLE
 INCLUDE ZPTLN
 PHASE SUMMOM,COUPLE
 INCLUDE SUMMOM
 PHASE MOMPT,COUPLE
 INCLUDE MOMPT
 PHASE PTLINE,COUPLE
 INCLUDE PTLINE
 PHASE SUMFOR,COUPLE
 INCLUDE SUMFOR
 PHASE FORCES,COUPLE
 INCLUDE FORCES
 PHASE MFGRR,COUPLE
 INCLUDE MFGRR
 PHASE DLOAD,COUPLE
 INCLUDE DLOAD
 PHASE EQUA,COUPLE
 INCLUDE EQUA
 PHASE TYPSYS,COUPLE
 INCLUDE TYPSYS
 PHASE CHECK,COUPLE
 INCLUDE CHECK
 PHASE FILKME,COUPLE
 INCLUDE FILKME
 PHASE FILKMC,COUPLE
 INCLUDE FILKMC
 PHASE FILKMM,COUPLE
 INCLUDE FILKMM
 PHASE FINISH,COUPLE
 INCLUDE FINISH
// EXEC LNKEDT
// EXEC
    SFXSFYSFZSMXSMYSMZC
COPLANAR-CONCURRENT     CONCURRENT-3DIMENSIONAL COPLANAR
PARALLEL-3DIMENSIONAL   COPLANAR-PARALLEL       3-DIMENSIONAL
ZFORC   SOLVE   COUPLE  DLOAD   MFGRR   EQUA
FORCES  SUMFOR  PTLINE  MOMPT   SUMMOM  ZSMOM   ZPTLN   ZARR
TYPSYS  CHECK   FILKME  FILKMC  FILKMM  FINISH
/*
```

FOR FACULTY MEMBERS ONLY

HOW TO ENTER ANSWERS:

Answers are stored in Array KM(5,7) for five problems as follows:

ARRAY KM(5,7)

Chapter Number	Problem Number	F_a	M_a	F_r	M_r	Type System

F_a=Number of active forces

M_a=Number of active moments

F_r=Number of reactive forces

M_r=Number of reactive moments

Type System:

1. Coplanar-concurrent
2. Concurrent-three-dimensional
3. Coplanar
4. Parallel-three-dimensional
5. Coplanar-parallel
6. Three-dimensional

These answers are entered in subroutines FILKME, FILKMC, and FILKMM. Each time a new set of five problems is used, 35 computer cards must be punched, and the subroutine put back on the disk.