

THE USE OF COMPUTERS IN CIVIL ENGINEERING EDUCATION

Harold J. Welch

**Department of Civil Engineering
The University of Michigan**

June 1, 1962

This material is distributed by The Ford Foundation Project on the Use of Computers in Engineering Education at The University of Michigan. This report appears in the library edition of the Final Report of the Project and is also issued as a separate booklet. Similar "Curriculum Reports" for other engineering disciplines are also available on request.

Table of Contents

	<u>Page</u>
I. Introduction	C3
II. The Civil Engineering Department of The University of Michigan: Faculty and Curricula	C4
III. Introduction of Civil Engineering Students to the Use of the Computer	C11
IV. Specific Uses of Computers in Various Fields of Study	C16
V. Conclusions	C20
VI. Example Problems Prepared by Project Participants	C21
76 Use of the Computer as a Teaching Machine for Plane Analytic Geometry	C23
77 Optimum Regulation of a Reservoir for Power Production	C35
78 Economical Size of Canal-Feeding Hydro-Power Plant	C41
79 Most Probable Number Method for Determining Bacterial Populations	C47
80 Iteration of Sum of Principle Stresses at Interior Points of Plane Rings	C56
81 Elastic Buckling Load for Columns of Non-Uniform Cross Section	C67
82 A MAD Program for Truss Analysis	C82
83 Stiffness Factors for a Flat Slab Column	C97
84 Vibration of Beams on Spring Supports	C104
VII. References	C113

USE OF COMPUTERS IN CIVIL ENGINEERING EDUCATION

ABSTRACT

The Civil Engineering Department of The University of Michigan has taken an active part in computing work at the University with a total of seven faculty members serving as participants in the Project on the Use of Computers in Engineering Education during the past two-year period. In addition, twenty-six civil engineering professors from other universities have participated in Project activities for periods varying from one week to a full semester.

Most undergraduate engineering students at the University are introduced to computer work through a required introductory course taught by Computing Center and Mathematics Department personnel. The Civil Engineering Department, however, has elected to introduce its students to computers via a departmental course, the second of the three required surveying courses. Details of the organization of this course are included as part of this report.

Digital and analog computer work has been assigned in more than 25 departmental courses during the past year, giving students an opportunity to gain practice in the application of computers in the solution of their engineering problems. The Civil Engineering curriculum, the areas where computer activity has been strongest, and a sampling of opinion of faculty members as to the effectiveness of computer usage in engineering instruction are included.

A selected set of example problems prepared by faculty Project participants is also included. These may be considered as a supplement to the seventy-five example engineering problems, including several related to Civil Engineering subject areas, which have been published previously by the Project.

I. INTRODUCTION

Civil Engineering, originally named to distinguish it from military engineering, has always covered a wide field of engineering practice. Civil engineers plan, design and supervise the construction of roads, railroads, harbors, buildings, tunnels, waterways, bridges, dams, airfields, canals, water supply and sewerage systems, and the many other facilities necessary for public works and industrial development. They plan the conservation, utilization, and control of water resources. They operate in the field of surveying and mapping.

In recent years, digital and analog computers have become important tools in many of these areas. With this in mind, the Civil Engineering Department of The University of Michigan has eagerly cooperated with the Ford Foundation Project on the Use of Computers in Engineering Education.

Since the Project was initiated, seven members of the Civil Engineering Department at Michigan and twenty-three visiting Civil Engineering faculty members have participated in the Project's program. In addition, other members of the faculty have become familiar with computer

philosophy and techniques as a result of weekly luncheon discussions and special lectures sponsored by the Project.

It has been found that problem solving by computer methods adds to the depth, scope, and logical understanding of many problems. It is believed that civil engineers of the future will profit from an introduction to computers as part of their undergraduate training.

II. THE CIVIL ENGINEERING DEPARTMENT OF THE UNIVERSITY OF MICHIGAN: FACULTY AND CURRICULA

The courses which comprise civil engineering may be divided into seven general areas: Construction, Geodetic, Hydraulic, Sanitary, Soil-Mechanics and Foundation, Structural, and Transportation Engineering. These areas are referred to as technical options and represent fields of specialization for the civil engineer. In addition, there are a few basic courses dealing with legal problems, professional conduct, ethics, the scientific method, etc., which are common to all fields of civil engineering.

Each of the general areas is described below. Members of the faculty primarily concerned with each field and specific courses are listed. Courses which are required for all civil engineering students are indicated with plus signs; courses in which the presentation is related to or supplemented by computers are indicated with asterisks. A more detailed description is given for those courses involving computer applications. Course numbers are given with each course to provide a convenient means of referring to them later. The number of semester credit hours is given in parentheses after each course.

A. Courses Common to All Fields of Civil Engineering

Lawrence C. Maugh, Professor of Civil Engineering, and Acting Chairman of the Department of Civil Engineering

- + CE400 Contracts, Specifications, Professional Conduct, and Engineering-Legal Relationships (2).
- CE500 Fundamentals of Experimental Research (2).
- CE501 Legal Aspects of Engineering (3).
- CE990 Civil Engineering Research (arranged).

B. Construction Engineering

The methods and techniques of modern construction; fundamental principles of construction applicable to all types of engineering structures; business and legal principles of contracting as applied in the field of construction.

Glen L. Alt, Professor of Civil Engineering (Advisor)
Frank E. Legg, Jr., Associate Professor of Construction Materials

- + CE350 Concrete Mixtures (1).
- CE531 Cost Analysis and Estimating (2).
- CE532 Construction Methods and Equipment (3).
- CE533 Estimating Practice (1).

+ Course required for all Civil Engineering students.

C. Geodetic Engineering

Theory and practical applications of surveying and surveying techniques; theoretical and applied geodesy, figure of the earth, local and extended control surveys; precise measurements and the adjustment of observations; design and execution of municipal surveys, topographic mapping projects, boundary surveys and land subdivision; industrial applications of surveying techniques; research in fields of instrumentation, computation, mapping and photogrammetry, and the problems of land surveying and route location.

Ralph Moore Berry, Professor of Geodetic Engineering (Advisor)
Harold J. McFarlan, Associate Professor of Geodetic Engineering
Harold J. Welch, Assistant Professor of Geodetic Engineering

+* CE260 Basic Surveying (3).

Use, care, and adjustment of basic surveying instruments; horizontal angle measurement, leveling, taping; circular curves, grades and vertical parabolic curves, profiles, earthwork; computation, significant figures, errors, desk calculators, U. S. Public Land System.

+* CE261 Surveying Computations (3).

Principles of horizontal control, unified system of geometric computations based on rectangular coordinates and the "Intersection Solution"; logical synthesis of computations for complicated geometrical engineering applications; introduction, applications, and use of electronic computers; principles of surveying astronomy, use of ephemeris and star catalog; elements of photogrammetry.

+* CE362 Advanced Surveying Measurements (4).

Precise observational methods for triangulation and control levels; use of theodolite and geodetic level, base-line measurements; electronic distance measurements; field assessment of errors and adjustments; astronomical observations for precise azimuth determinations; application of precise control to layout of engineering projects.

CE505 Boundary Surveys (3).

CE560 Photogrammetry (2).

CE561 Geodesy (3).

* CE562 Geodetic Field Methods (2).

Reconnaissance for geodetic triangulation; special observing methods for first-order horizontal and vertical control; Laplace stations and deflection of the vertical; actual observations, reduction and adjustment of results in an actual field situation. Term paper or project report required of each student.

* CE563 Adjustment of Geodetic Measurements (2).

Theory of least squares, applications to the adjustment of geodetic observations; arrangement for solution of complex adjustments on electronic computers; actual solution of selected problems.

* CE564 Special Problems in Advanced Surveying.

Special advanced work can be provided for those who have received credit in Advanced Surveying Measurements (CE362).

* CE565 Municipal Surveying (2).

Control surveys, methods and adjustments for use in municipal mapping and administration, surveys for streets, utilities, property lines, tax maps, subdivision control and development.

* CE960 Geodesy and Surveying Research.

Assigned work in geodesy, or other special field in surveying of interest to the student and approved by the professor of geodesy and surveying.

* Course involves computer applications.

D. Hydraulic Engineering

The application of the fundamental principles of hydraulics and hydrology to the development of water power, flood control, drainage, and improvement of rivers and harbors, and other hydraulic structures. Laboratory facilities and instruction are offered for students who wish to engage in research work in hydrology and hydraulics that will lead to advanced degrees.

Ernest F. Brater, Professor of Hydraulic Engineering (Advisor)
Victor L. Streeter, Professor of Hydraulics

+ CE420 Hydrology (3).

+* CE421 Hydraulics (2).

Hydrostatic stability; orifices and weirs; Venturi meters; cavitation; pump characteristics; flow in pipes and fittings; unsteady-uniform flow; steady-nonuniform flow. Lecture, laboratory, and computation.

* CE522 Hydraulic Transients (2).

Introduction to water power development, including selection of type of turbine; storage and pondage; surge in pipe lines; water hammer analysis; digital programming of unsteady flow situations.

CE523 Flow in Open Channels (3).

* CE524 Advanced Hydraulics (3).

Two-dimensional potential flow; the flow net; percolation and hydrostatic uplift; side-channel spillways; boundary-layer; hydraulic similitude; hydraulic models; stilling pools.

* CE526 Hydraulic Engineering Design (3).

Design of hydraulic structures such as diversion dams, head gates, control works, silt traps, siphon spillways, side-channel spillways, earth canals, and other structures involving accelerated flow, backwater, hydraulic jump, sedimentation, and erosion.

CE623 Applied Hydromechanics (2).

CE825 Seminar in Hydraulic Engineering.

CE920 Hydrological Research.

* CE921 Hydraulic Engineering Research.

Assigned work in hydraulic research; a wide range of matter and method permissible.

E. Sanitary Engineering

The planning, construction, and operation of water works, sewerage and drainage systems, water-purification plants, and works for the treatment and disposal of city sewage and industrial wastes; improvement and regulation of natural waters for purposes of sanitation; air sanitation; principles and standards of ventilation.

Jack A. Borchardt, Professor of Civil Engineering (Advisor)
Lloyd L. Kempe, Professor of Civil Engineering and of Chemical Engineering
Eugene A. Glysson, Associate Professor of Civil Engineering

+* CE480 Water Supply and Treatment (3).

Sources of water supply, quality and quantity requirements, design fundamentals of works for development, collection, purification, and distribution of water.

+* CE481 Sewerage and Sewage Treatment (2).

Requirements of residential and municipal sewerage systems, procedures for the design and construction of sewerage and sewage treatment works.

Use of Computers in Civil Engineering Education

- CE580 Microbiology (3).
- CE581 Sanitary Chemistry (2-3).
- CE582 Sanitary Engineering Design (3).
- CE583 Water Purification and Treatment (3).
- CE584 Waste Water Treatment and Disposal (3).
- CE585 Municipal and Industrial Sanitation (3).
- CE586 Industrial Waste Treatment (2).
- * CE587 Industrial Bacteriology (3).
Lectures and demonstrations to illustrate the application of microbiological principles and techniques to industrial processes.
- CE680 Microbiology II (2).
- CE681 Advanced Sanitary Chemistry (3).
- * CE682 Advanced Sanitary Engineering Design (3).
Functional design of sanitary engineering structures and typical plant layouts; drafting room and field studies; preparation of design reports.
- CE685 Special Problems in Sanitary Engineering I.
- CE686 Special Problems in Sanitary Engineering II.
- CE780 Public Water Supply (3).
- CE880 Sanitary Engineering Seminar (1).
- CE980 Sanitary Engineering Research.

F. Soil Mechanics and Foundation Engineering

Soil mechanics; evaluation of soil properties and environmental conditions in foundations of earth-supported structures; mass stability in excavations and subsurface construction. Soil engineering; use of soil characteristics and properties and soil classification in design and construction of highways, railways, airports, and other surface facilities.

William S. Housel, Professor of Civil Engineering (Advisor)
Robert O. Goetz, Instructor in Civil Engineering
Ulrich W. Stoll, Instructor in Civil Engineering

- + CE445 Engineering Properties of Soil (3).
- CE543 Soils in Highway Engineering (2).
- CE544 Airport Design and Construction (3).
- CE545 Foundation and Underground Construction (3).
- CE546 Soil Mechanics Laboratory (1).
- CE946 Soil Mechanics Research.

G. Structural Engineering

The theory, design, and construction of structures, such as bridges, buildings, dams, retaining walls, and reservoirs, involving the use of steel, reinforced concrete, and lumber; the testing and utilization of soils in foundations and subsurface construction.

Lawrence C. Maugh, Professor of Civil Engineering, and Acting Chairman of the Department of Civil Engineering (Advisor)
Glen V. Berg, Professor of Civil Engineering
Bruce G. Johnston, Professor of Structural Engineering
Leo M. Legatski, Professor of Civil Engineering
Robert B. Harris, Associate Professor of Civil Engineering
Wadi S. Rumman, Assistant Professor of Civil Engineering

- +* CE312 Theory of Structures (3).
Calculations of reactions, shears, and bending moments in simple, restrained, and continuous beams due to fixed and moving loads; simple trusses with fixed and moving loads; determinate frames; columns; tension members; girders; introduction to design.
- +* CE313 Elementary Design of Structures (3).
Design and details of simple beams, girders, columns, and trusses. Computations, drawing work, and laboratory experiments.
- +* CE415 Reinforced Concrete (3).
Properties of materials; stress analysis and design of reinforced concrete structures; introduction to prestressed concrete and ultimate strength analysis. Lectures, problems, and laboratory.
- CE310 Structural Drafting (2).
- CE511 Timber Construction (1).
- * CE512 Advanced Theory of Structures (3).
Stresses in subdivided panel trusses; principle of virtual displacements and virtual work; energy theorems; graphical methods; analysis of statically indeterminate trusses and frames.
- * CE513 Design of Structures (3).
Design of reinforced concrete and steel structures. Computations and drawing.
- * CE514 Rigid Frame Structures (3).
Analysis of rigid frames by methods of successive approximations and slope deflections; special problems in the design of continuous frames.
- CE515 Prestressed Reinforced Concrete (2).
- * CE516 Advanced Design of Structures (3).
Functional design of buildings; selection and analysis of structural elements; reinforced concrete flat slab design; digital computer applications. Lectures and computation laboratory.
- CE517 Bridge Engineering and Design (3).
- * CE611 Structural Dynamics (3).
Structural vibrations. Transient and steady state response to dynamic forces. Response beyond the elastic range. Response to nuclear explosions. Earthquake forces. Structural response to earthquake. The response spectrum. Seismic building codes and their relation to structural dynamics.
- CE612 Structural Members (3).
- * CE613 Structural Plate Analysis (2).
Stress analysis of flat plates loaded either in their plane or in bending. Numerical analysis. Applications to special problems in flat slab construction.
- * CE614 Advanced Problems in Statically Indeterminate Structures (3).
Continuous truss bents; hinged and fixed arches; rings; frames with curved members; flexible members including suspension bridges; frames with semi-rigid connections.
- CE615 Analysis and Design of Folded Plates, Domes and Shells (3).
- CE616 Plastic Analysis and Design of Frames (2).
- CE617 Mechanical Methods of Stress Analysis (1).
- * CE618 Computer Analysis of Structures (3).
The analysis of beams, frames, trusses, and arches by high speed digital computers. The general method of influence coefficients; matrix methods, algorithm development, flow charts, programming. Students will solve a sequence of problems on the high speed computer in the University Computing Center.
- CE810 Structural Engineering Seminar (1).
- CE910 Structural Engineering Research.

H. Transportation Engineering

Transportation Engineering is further subdivided into Highway, Railroad and Traffic Engineering. Highway Engineering involves the location, design, construction and maintenance of various types of roads and streets, including materials, surveys, plans, specifications, economics, and financing. Railroad Engineering involves the design, construction, and operation of railroad properties, including metropolitan terminals, statistical analysis of operating data, freight and passenger traffic, economics, financing, administration, and regulation. Traffic Engineering involves methods of increasing the efficiency and safety of traffic movement; street and off-street parking; traffic surveys, geometrical design of urban and rural highways, traffic control devices, and other means of regulating and controlling the use of highways.

John C. Kohl, Professor of Civil Engineering and Director of the Transportation Institute (Advisor for Traffic Engineering)

Ward K. Parr, Associate Professor of Highway Engineering

Bruce D. Greenshields, Lecturer in Transportation Engineering

Donald N. Cortright, Associate Professor of Civil Engineering (Advisor for Highway Engineering)

Clinton L. Heimbach, Lecturer in Civil Engineering (Advisor for Railroad Engineering)

+ CE370 Transportation Engineering (3).

CE550 Highway Materials (3).

CE551 Physical Properties of Concrete Masonry (2).

CE552 Bituminous Materials and Pavements (2).

CE570 Highway Traffic (2).

CE571 Traffic Engineering (3).

CE572 Highway Economics (2).

CE573 Highway Design (3).

CE574 Railroad Engineering (3).

* CE575 Terminal Design (3).

Design of railroad, highway, waterways, and airport terminals, joint terminals, layout of the various types of yards and traffic facilities.

CE576 Economics of Railroad Construction and Operation (2).

CE652 Advanced Bituminous Materials and Flexible Pavement Design.

* CE670 Transportation Planning (3).

Analysis of supply and demand for transportation services, transport relationships to land use and other elements of regional and urban planning, and planning techniques applied to transportation problems.

CE671 Advanced Highway Engineering (2).

CE672 Transportation (3).

CE673 Highway Transport (2).

CE674 Industrial Transport Management (4).

CE970 Highway Engineering Research.

CE971 Transportation Engineering Research.

A list of courses taken by Civil Engineering students is given in the table below. The table is arranged to indicate the semester in which the course is normally taken. Courses in which the presentation is related to or supplemented by computers are indicated by an asterisk (*).

LIST OF COURSES TAKEN BY CIVIL ENGINEERING STUDENTS

<u>Semester</u>	<u>Course Name and Number</u>	<u>Hours</u>	<u>Total Hours for Semester</u>
1	Analytic Geometry and Calculus I (Math 233)	4	15
	General and Inorganic Chemistry (Chem 104)	4	
	Engineering Drawing (Engr. Graphics 101)	3	
	Freshman Composition I (English 111)	3	
	Introductory Speech (English 121)	1	
2	Analytic Geometry and Calculus II (Math 234)	4	15
	Descriptive Geometry (Engr. Graphics 102)	3	
	Freshman Composition II (English 112)	2	
	English Elective	2	
	General and Inorganic Chemistry (Chem 106)	4	
3	Analytic Geometry and Calculus III (Math 371)	4	16
	Mechanics, Sound, and Heat (Physics 145)	5	
	Statics (Engr. Mech. 208)	3	
	Basic Surveying (CE260)	3	
	Concrete Mixtures (CE350)	1	
4	Analytic Geometry and Calculus IV (Math 372)	4	17
	Electricity and Light (Physics 146)	5	
	Mechanics of Material (Engr. Mech. 210)	4	
	Laboratory in Mechanics of Material (Engr. Mech. 212)	1	
	* Surveying Computations (CE261)	3	
Summer	* Advanced Surveying Measurements (CE362)	4	8
	Geology for Engineers (Geology 218)	4	
5	Fluid Mechanics (Engr. Mech. 324)	3	17
	* Theory of Structures (CE313)	3	
	Transportation Engineering (CE370)	3	
	Hydrology (CE420)	3	
	(*) Advanced Mathematics Elective	3	
	Elective in Humanities or Social Science	2	
6	Dynamics (Engr. Mech. 343)	3	17
	* Reinforced Concrete (CE415)	3	
	* Water Supply and Treatment (CE480)	3	
	* Hydraulics (CE421)	2	
	Engineering Properties of Soil (CE445)	3	
	Elective in Humanities or Social Science	3	
7	Specifications and Contracts (CE400)	2	16
	* Elementary Design of Structures (CE313)	3	
	Electrical Apparatus and Circuits (EE315)	4	
	* Sewerage and Sewage Treatment (CE481)	2	
	English Elective	2	
(*) Technical Elective	3		
8	Thermodynamics and Heat Transfer (Mech. Engr. 333)	4	17
	(*) Technical Electives	7	
	Modern Economic Society (Econ. 401)	3	
	Elective in Humanities or Social Science	3	

* Course Related to or supplemented by computers.

(*) Student may elect course related to or supplemented by computers.

III. INTRODUCTION OF CIVIL ENGINEERING STUDENTS TO THE USE OF THE COMPUTER

Surveying Computations (CE261) has been selected as the medium for introducing Civil Engineering students to the use of the digital computer. This course is the second in a series of three required courses in surveying and is normally taken in the second semester of the sophomore year.

Civil Engineering students are not required to take the one-hour course in introductory computing techniques (Math 373) which is required of students in most of the other fields of engineering. Surveying Computations has, therefore, been designed as an effective substitute.

A. Preparation of Input for the Computer

The first half of the semester is devoted to applications of analytic geometry for plane, rectangular coordinate computations. Emphasis is placed on five basic problem types. These types, described in detail in Example Problem No. 76, have the following names: Forward Computation, Inverse Computation, Line-Line Intersection, Line-Circle Intersection, and Circle-Circle Intersection.

The students are shown the recommended computing methods, supported by a mathematical derivation, for each problem type. After studying the assignment, listening to the lecture, and assimilating the methods taught, each student is assigned a set of fifteen individual problems of each type. The student is instructed initially to perform the necessary calculations for only the first five problems in each set and to prepare both the given information and the solution on punched IBM cards.

These cards are then submitted as data to a computer program prepared specifically for the course, called the "teaching machine program." This program, described in more detail in Example Problem No. 76, has access to all of the given data and computes the correct results for each of the students' problems. It compares all of the items on each of the students' cards with the known, correct values. If any single item in any one of the five solutions is found to be in error, the whole problem is considered incorrect. For each incorrect problem the student is required to correct all of the errors, and also solve an additional problem in the assignment. The original five problems (with errors corrected) plus the additional problems are then submitted in a second approach to the computer. Thus, for the second approach, the student may be required to enter from six to ten problems. The same rules apply again, for a third and final approach to the computer. The worst possible penalty would be that on the final approach to the computer the student would be required to present solutions to all fifteen of the assigned problems.

If all of the problems submitted in the final approach to the computer are not correct, the student is penalized in his grade. On the other hand, if a correct solution is obtained

to all of the problems submitted on any given approach to the computer, the student is given full credit regardless of whether success is achieved on the first, second or third trial.

It may be seen that the students quickly realize that it is to their advantage to obtain the correct results for submission on the first pass through the computer. In addition to the savings in their calculating time (on the desk calculator), by solving the first five problems of each type correctly and finishing the assignment, it has been noticed that a friendly spirit of rivalry generally develops between the students in which those who do not complete the assignment on the first pass receive friendly, but slightly barbed comments on their proficiency.

The "teaching machine" theory seems to be working quite well for these problems. The students quickly overcome their difficulties and by the time they have finished the five problem types, they show a marked proficiency in the use of trigonometric tables and the desk calculator. In order to meet the requirements for tolerance in the teaching machine program, the students learn by experience the necessity for using significant figures properly in their computations. For example, they are shown, by actually working problems both ways, that in an intersection of a line and a circle, it is necessary to interpolate fractions of seconds in order to obtain enough accuracy when the intersection angle is very small and the radius of the circle is very large. These are things which cannot be easily taught in classroom lectures but which can be illustrated very pointedly to the student during his laboratory work.

This three-semester-hour course is handled with a one-hour lecture per week and two three-hour laboratory periods per week. The first hour of each period is devoted to a lecture, followed by two hours of actual computation. During this time, the students solve problems in the laboratory room using desk calculators. They are encouraged to ask the instructor for help at any time on problems involving the theory of the computations, the mechanics of the desk calculators, the manipulation of trigonometric tables, or an explanation of lack of accurate checking of their completed problems.

The "teaching machine" idea serves also to introduce the students to the digital computer. They learn to use the keypunch to punch their own cards before they are actually assigned to write a program for the digital computer. They are strongly impressed with the capabilities of the computer when they see how it is able to solve their problems and check them at a rate of approximately two of their problems per second (on the IBM 709). They realize quite painfully that these problems cost them sometimes as much as an hour or more of desk calculator work. Because of this, when they are later permitted to begin programming for the computer, they do not have the reaction so often expressed by students who are given elementary problems, "I could have worked the problem faster on a desk calculator."

B. Geometric Analysis Using the Interpretive Language, AGILE

The writer has developed a computer program over the last several years which has recently been named "AGILE" (Analytic Geometry Interpretive Language for Engineers). A number of small programs to solve problems such as the five basic types described in Example Problem No. 76, have been combined with a reading, storing, and scheduling master program. In this way, any geometry problem (which may consist of a number of basic problems) can be solved with one pass through the computer.

After learning to compute the basic problems already described, the students are taught to recognize them when they occur in such applications as the design of a subdivision or a highway interchange. The students are also taught to describe these problems in a logical manner, using the AGILE language. A complete description of the AGILE language is being prepared. If sufficient interest develops, a manual describing the language and its use will be made available.

C. Computer Programming

Beginning with the first Wednesday night of the semester, a series of three weekly lectures of two hours each, on computer programming, is presented for the entire College of Engineering. These lectures were presented in 1961-1962 by Mr. Brice Carnahan, Assistant Director of the Project on Computers in Engineering Education. The lectures introduce the students to the philosophy of computers, and specifically, to the MAD language. The students are not required to attend these lectures, but are strongly urged to do so, being told that no substitute for them will be presented in classroom lectures.

Beginning immediately after the series of night lectures, the students are assigned their first problem which requires them to write a computer program for the course. This initial program is for a forward computation (the first problem type listed in Example Problem No. 76), and is handed to the students completely checked out and ready to run on the computer. They are required simply to prepare cards for communicating with the computer. They are also asked to punch data cards for a problem which is given to them. The form of the data cards is slightly different than the form in which the data are presented. In addition to the punched cards, the students are required to submit a flow diagram for the problem and a short discussion of the problem, its solution, and the method which a user must follow in preparing data for a run on the computer.

By this means the students are taken over a second, short step towards computer programming by being introduced as painlessly as possible to the computer itself. The submission of a program which has already been written for them may sound like a wasted step but it has been found that this is a step which, when combined with other complications, can become a very frustrating experience to the neophyte.

This first program is explained thoroughly to the students so that they understand what is done by the computer as a result of each instruction in the program. Two very elementary loops are included so that the student becomes familiar very quickly and painlessly with the concept of iterative techniques on the computer. The students are permitted to submit this program to the computer only once. It is felt that since everything has been handed to them they should be graded on their ability to follow instructions.

The second computer program is for the inverse computation. In this case, they are given a clear concept of the theory and the method of solution of the problem but are required to make up their own instructions in MAD for the solution of the problem. Again, they are required to submit a flow diagram, and a written report which includes an explanation of the method necessary to use the program successfully on the computer. In the solution of this problem the students are required to write their own Newton-Raphson iteration for the square root solution of the distance. They are also asked to solve for the distance by using either the sine or the cosine of the angle which has previously been computed from the arctangent.

This gives them their first introduction to the use of library subroutines, the formal part of MAD programming including the loop concept, and also gives them a chance to compare the relative accuracy by which they can obtain the distance by two completely independent methods.

In the third program which is assigned in this course, the students are instructed to write a computer program which will calculate the volume of a borrow pit. In this problem, the students are asked to bring in the "before" and "after" elevation readings over a grid pattern which is basically rectangular. They are asked to work with these two matrices, subtracting the "after" readings from the "before" readings to obtain a new matrix called the "excavation." This subtraction is done by the use of a double loop. Following this, the computations are performed by means of some single and double loops. This helps the student become familiar with methods of handling large sets of subscripted variables.

The fourth and last program is one which may vary considerably from semester to semester. It is designed to offer a challenge to the students. It is intentionally changed radically from one semester to the next in order to discourage any ideas of propagating information from one class to another. Examples of the problems which have been given previously are as follows:

- (1) Intersection of a straight line with a spiral envelope.
- (2) Approximate adjustment of a triangulation quadrilateral.
- (3) Progressive adjustment of interlocking traverses by the Bowditch (compass) Method.
- (4) Crandall method of adjustment of a traverse.

At the time of this writing it is strongly believed that the inclusion of computer programming in this course has not detracted from the course work in any way. In fact, it is felt that the blending image of the desk calculator, the digital computer, and surveying computation methods, has resulted in a course which has materially strengthened the students' understanding in each of the three fields mentioned.

Use of Computers in Civil Engineering Education

Some of the leading exponents of computer philosophy and education feel that geometry is one of the best vehicles for teaching computer philosophy. The student's ability to follow graphically and pictorially every step in his solution makes it possible for him to concentrate on the programming necessary to solve the problem. By the same token, his detailed computer solution of the problem brings him into a very intimate relationship with the problem itself. By the time he has successfully programmed a solution, he is completely familiar with every aspect of the problem.

It has been found that many of the students are completely apathetic to computers and the philosophy of computation. These students manage to pass the course by literally hanging on for dear life. Other students in the middle category passively accept the work and perform the assignments satisfactorily but with little imagination. Finally, we have the group who become quite interested in computers and their possibilities and who go beyond the call of duty, polishing up their programs to achieve more than the assignment requires. Certainly these students are in the minority, but it is encouraging to notice that their number seems to be increasing. It is believed that this enthusiasm is growing because the work is passed down to succeeding generations by those who have run into advanced problems which required computer applications. Also, it is quite apparent to the students that many interviewers from industrial firms who come to the campus are now asking about the amount of computer work students have been exposed to. These students, especially those who are doing well in their other subjects (primarily mathematics), are encouraged to sign up for Mathematics 473, Introduction to Digital Computers, a comprehensive, senior-graduate level computer course.

It is not our intention to force the majority of our students into computer technology, but it is felt that all students should have some knowledge of computers and that some few students should have a specialized knowledge of computer programming techniques. In Surveying Computations, it is possible for the students to decide which of these paths they wish to follow. Upon completion of this course, no student will feel that a computer is a mysterious black box which possesses magical properties. On the contrary, he will realize that it is most simple for him to solve elementary problems and that it is not impossible for him to solve even the most difficult problems by using it.

IV. SPECIFIC USES OF COMPUTERS IN VARIOUS FIELDS OF STUDY

A. Geodetic Engineering

The required courses in surveying are designed to give the student a realistic appreciation for the theory of measurements, application of mathematics to geometric figures, logical analysis, and systematic computation procedures. These courses provide six of the first seven semester hours of contact between the Civil Engineering Department and the students.

The first course, Basic Surveying (CE260), does not lend itself to student use of computers. However, the instructor assigns a unique traverse computation problem to each student and compares his answers to those provided by the computer. Other similar uses are being explored.

The second required course, Surveying Computations (CE261), has been described in detail in Part III of this report.

The terminal required course, Advanced Surveying Measurements (CE362), is taught at Camp Davis, Jackson, Wyoming. This course is primarily one engaging in precise control surveys using theodolites, electronic distance measuring devices and invar tapes. Although no computers are available in the vicinity of the camp which can be used for the course purposes, programs have been written which will make it possible to send data from camp to the University Computing Center and receive the results back in time to use them for the purposes of the course. At the present time these programs are primarily concerned with reducing observation data, such as Tellurometer observations, to yield the actual measured values from the observations. These reductions, when performed by hand, are quite laborious. Even more important to the user, however, such reductions are filled with potential errors, which, because they deal with measurements, cannot be checked. The use of the computer for this purpose virtually eliminates all chance of such blunders in reductions.

Geodetic Field Methods (CE562) consists of unique problems which vary with the interests of the students and the program of the department. Usually there are several applications for computer solutions of problems which occur in this course.

Adjustment of Geodetic Measurements (CE563) depends very heavily on digital computer applications. Problems are programmed and solved by means of the Gauss-Seidel iterative technique. Matrix inversion and other methods of solving large systems of linear equations are also applied to the computer.

Special Problems in Advanced Surveying (CE564). Problems in this course again are quite variable and may or may not require computer applications.

Municipal Surveying (CE565) consists in part of several conducted tours to various private and municipal or governmental agencies. One private computing firm, the Bureau of Surveys of the City of Detroit, and the Michigan State Highway Department were three locations visited where computers were being used. Students observed the computers in operation and were told by the engineers in charge exactly what applications were being made in the specific fields of their needs.

Geodesy and Surveying Research (CE960) is another example of a course which may have varying interests or needs. Certainly many students in the future who take this course will be carrying on extensive computer research in the field of geodetic surveying.

B. Hydraulic Engineering

During the last three years a majority of the courses offered in hydraulics and hydraulic engineering have been completely re-examined, and now require use of the digital computer as a part of the course work.

The required course, Hydraulics (CE421), has in the past offered the student an opportunity to program one or two problems on a voluntary basis. Three or four students, out of about 40, usually wrote programs. Beginning with the spring semester (1962), however, two programs are required of all students: the first, the determination of critical depth, normal depth, water surface profiles, and location of a hydraulic jump in a trapezoidal channel; and the second, the determination of the best formula for a v-notch weir, from student data, by the method of least squares.

The course in Hydraulic Transients (CE522), has evolved from a hydropower course, primarily because of the great improvement in means of solving problems in unsteady flow. Problems in surge and water hammer have been programmed, and the instructor has set up the differential surge tank problem for class inspection. New methods for handling water hammer have been developed by graduate students as a result of this course and the availability of the digital computer.

The course in Advanced Hydraulics (CE524), relies heavily on the computer, and has used it for such problems as relaxation methods for computing flow under a dam and for applications of the momentum theory. One significant problem, the design of a side channel spillway, has been given very thorough study, and a new method has been developed to optimize the design. Two unknown parameters are needed: a and n in $V = ax^n$, for velocity at a cross section x distance along the spillway for minimum total excavation. A "gradient method" is used, to determine how a and n should be changed to move in the direction of minimum excavation.

The Hydraulic Design course (CE526), is built entirely on the use of the computer. At present it consists of several programs relating to a project involving construction of a gravity dam, mass diagram study of the inflow hydrograph, flood routing study, reservoir operation study for flood control, power and irrigation supply, and finally an economic study which determines the height of the dam to be constructed.

It is believed that the students in the hydraulic option are much better prepared to meet and adjust to the future demands of their profession as a result of their ability to think through the logical processes required in a program. It also gives them a definite competitive edge in securing interesting job assignments with greater growth potential.

C. Sanitary Engineering

The digital computer is being used for network analysis in Water Supply and Treatment (CE480), as an optional problem which has been worked out successfully by several students.

In Sewerage and Sewage Treatment (CE481), the design of a storm sewer system is at present being worked on as an optional assignment.

In Industrial Bacteriology (CE587), a problem has been developed requiring the solution on the computer of the MPN (Most Probable Number) equation for determining bacterial populations.

In Advanced Sanitary Engineering Design (CE682), the programming of the sewage network analysis is to be described and discussed but no actual programming is anticipated. However, the students may elect to do so if they desire.

D. Soil Mechanics and Foundation Engineering

The traditional aim of the courses in soil mechanics and foundation engineering at The University of Michigan has been to train the student to view problems not only in terms of mathematical formulations but also in relation to environmental associations. Particular emphasis has been placed on framing problems in terms of simplified basic relationships which can be conveniently handled by slide rule or desk calculator. One observes that the conscientious student develops a "feel" for the subject in this approach which is more readily adapted to variations in field conditions. It is desired to maintain this emphasis on the practical application of fundamental principles and, at the same time, to take full advantage of the spectacular developments in computational technique with present day electronic computers.

In this respect, the computer makes possible more exacting solutions of certain soil mechanics problems and eliminates compromising assumptions and approximations which have heretofore been necessary. Consequently, the soil mechanics teaching and research staff have re-examined a number of the more important problems that require extensive computations and have developed computer programs for the following:

1. Analysis of the plate loading tests to determine the bearing capacity of soil.
2. Determination of pressure distribution from large loaded areas.
3. Static equilibrium of sheet pile retaining walls under a variety of nonhomogeneous soil conditions.

Additional problems under investigation include computation of settlement in granular materials from volume-density relationships, a general solution for critical stresses under surface loading and the reproduction of pavement profiles for a variable length of base line reference. The solution of these more complex problems has not advanced to the stage where they can be utilized in undergraduate instruction. They are, however, being developed and used by graduate students working on these specialized problems.

The objective in presenting the use of computers to students of soil mechanics is to demonstrate the value of these more exacting solutions and familiarize the student with their application to soil mechanics problems. It is felt that the real value of computer techniques

to soil mechanics lies in the rigorous organization of the problem in a sequence of logical steps leading from fundamental principles and realistic assumptions to a useful practical solution. It is contemplated that the use of computers in soil mechanics will be included in the advanced design course.

E. Structural Engineering

Computers are used in the structures sequence both at undergraduate and graduate levels.

In Elementary Structural Design (CE313), some experiments have been carried on in the use of the computer in simple design problems. This has not been entirely successful, largely because the students have too little experience in the use of the computer in earlier work. There is not sufficient time to include programming in the course lectures.

In Theory of Structures (CE512), and Rigid Frame Structures (CE514), the computer has been used to solve problems in truss and beam deflections, using conventional procedures. The use of computers in these courses is still on an experimental basis.

In Structural Design (CE513), the computer is used regularly each semester and one computer problem is assigned. In a typical semester, the students are asked to prepare section modulus and shear connector tables for composite beams in accordance with the AISC 1962 Specifications. The problem is relatively simple but involves quite a number of logical decisions that require a more complete knowledge of composite beam behavior than would the comparable hand solution for only a few special cases. There is nothing artificial about the problem. It is the type of problem that one might expect to encounter not infrequently in practice.

In Structural Dynamics (CE611), the computer is regularly used for one or two problems each semester. Ordinarily the problems involve numerical solution of the differential equations of motion and constructing response spectrum curves for some given dynamic load.

In Structural Plate Analysis (CE613), one problem is assigned each semester in finding the deflections of a plate by finite difference methods. The problem is taken up in class at two levels; first the solution is obtained by hand for a coarse mesh, and later a finer mesh solution is prepared using the computer.

Computer Analysis of Structures (CE618), is a course serving the specific purpose of providing the background needed for using the computer efficiently in structural analysis. Matrix methods are of special value in computer analysis of structures, so the course is built on matrix techniques. Most of the course is devoted to formulating the theory of structures in the language of matrices. Assigned problems include small, near-trivial problems where all of the matrix operations are carried out by hand, and larger, non-trivial problems for which the students program the matrix operations for the machine. The smaller problems serve the purpose of familiarizing the student with the details of the procedures and also retaining a physical interpretation for all of the various matrix operations.

F. Transportation Engineering

The staff use of computers in the area of traffic and transportation is somewhat limited at the present time, but will undoubtedly increase in the future. In CE575, Transportation Terminal Design (highway, rail, water, and air), and CE670, Transportation Planning, the problems, insofar as possible, are placed in a local Ann Arbor context. Both the City of Ann Arbor and Washtenaw County have acquired voluminous data on local transportation, land use, and population. Much of this data is available on punched cards so that it can be sorted in any manner desired. Thus in design problems the computer is used to summarize and compile data on specific geographic areas interconnected by transportation facilities.

There are some iterative type solutions involved in layout and design of automobile parking lots and in the delineation of retail trade areas for shopping centers which lend themselves quite nicely to computer application. At the present, a study is being made of the extent to which it is desirable to require the student to use the computer in solving these problems. Transportation planning problems quite often are examined on a systems analysis basis. While the computer has not been used to any great extent in this field, there is a potential for widespread usage.

To summarize, the various areas in traffic and transportation courses and the extent to which it is desirable for the student to become involved with the computer are currently being examined. Use at the present time, however, is largely experimental.

V. CONCLUSIONS

As in the case of every important innovation, it will take several years to evaluate the ultimate benefits resulting from the introduction of computers into Civil Engineering education. At present, opinions of the faculty are varied, as evidenced by the following sampling of statements.

"The Civil Engineering Department at The University of Michigan has successfully initiated the use of computers into the curriculum and the faculty is gradually understanding the value of adapting their work to accept and include these new techniques."

"The real benefits of using the computer come not from teaching the student how to program, but rather from forcing him to make all of the logical decisions that are necessary to arrive at a suitable algorithm for solving an engineering problem in a more general formulation than he usually encounters. It is often said that a good way to learn a subject is to teach it. In a sense, the student is being put in a position of teaching the computer how to solve the problem at hand. In order to accomplish this he must necessarily understand the problem more completely than would be necessary for conventional hand solution, since all of the steps involving judgment that are at his disposal in hand methods must be analyzed and programmed when the machine is doing the work."

"While it is agreed that computers are here to stay and are a valuable tool in the civil engineering profession, we must be very careful not to sacrifice the teaching of fundamental engineering principles in order to teach the use of computers."

"The method of introducing Civil Engineering students at Michigan to computer techniques by integrating it with the teaching of surveying computations seems to be an excellent approach."

"In introducing the use of computers, it is necessary to maintain the emphasis on the practical application of fundamental principles and, at the same time, to take full advantage of the spectacular developments in computational technique with present-day electronic computers. The real value of computer techniques lies in the rigorous organization of the problem in a sequence of logical steps leading from fundamental principles and realistic assumptions to a useful, practical solution."

"We need to make fuller use of computers in all fields of Civil Engineering. In this way our graduates will be much better prepared to meet and adjust to the future demands of their profession as a result of their ability to think through the logical processes required to develop algorithms for the computer."

VI. EXAMPLE PROBLEMS PREPARED BY PROJECT PARTICIPANTS

Each participant prepared solutions for example problems which he felt would be useful in courses at the undergraduate or graduate level. Nine of these problems have been selected for inclusion in this report and are listed below.

LIST OF PROBLEMS

<u>Number*</u>	<u>Title</u>	<u>Author</u>	<u>Page</u>
76	Use of the Computer as a Teaching Machine for Plane Analytic Geometry	H. J. Welch	C23
77	Optimum Regulation of a Reservoir for Power Production	J. W. Howe	C35
78	Economical Size of Canal-Feeding Hydro-Power Plant	J. W. Howe	C41
79	Most Probable Number Method for Determining Bacterial Populations	L. L. Kempe	C47
80	Iteration of Sum of Principal Stresses at Interior Points of Plane Rings	W. Weaver, Jr.	C56
81	Elastic Buckling Load for Columns of Non-Uniform Cross Section	K. H. Chu	C67
82	A MAD Program for Truss Analysis	K. H. Chu	C82
83	Stiffness Factors for a Flat Slab Column	R. B. Harris	C97
84	Vibration of Beams on Spring Supports	S. S. Kuo	C113

* These problems may be considered as a supplement to problems presented in previous publications of the Project (3, 4, 5, 6) and are numbered accordingly.

All of these problems were solved on the digital computer using either the MAD or FORTRAN language. These languages are described in References 1 and 2.

It should be noted that there are several other sources of example computer problems dealing with Civil Engineering subjects. See, for example, Example Problem Numbers 8, 9, 10, 21, 22, 23, and 27 in Reference 3; Example Problem Numbers 48, 49, and 50 in Reference 4; and Example Problem Numbers 63 and 64 in Reference 6. A number of problems prepared by Project participants are abstracted in the Second Annual Report of the Project (4). Some of these (listed above) have been included in this report. Most of the remaining problems are available on microfilm from the Project.

Example Problem No. 76

USE OF THE COMPUTER AS A TEACHING MACHINE
FOR PLANE ANALYTIC GEOMETRY

by

Harold J. Welch

Department of Civil Engineering

The University of Michigan

Course: Surveying Computations

Credit Hours: 2

Level: Sophomore

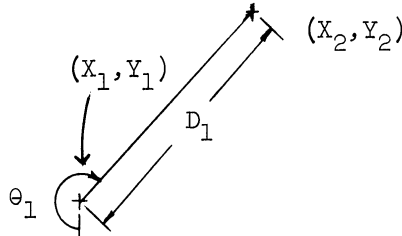
Statement of Problem

In learning surveying computations, students are required to solve a number of problems in plane, analytic geometry using a desk calculator and tables of trigonometric functions. The problems consist of several examples of each of five basic problem types illustrated below.

1. Forward Computation

Given: X_1, Y_1, D_1, θ_1

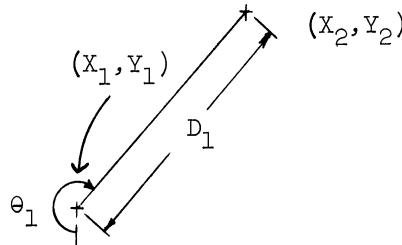
Find: X_2, Y_2



2. Inverse Computation

Given: X_1, Y_1, X_2, Y_2

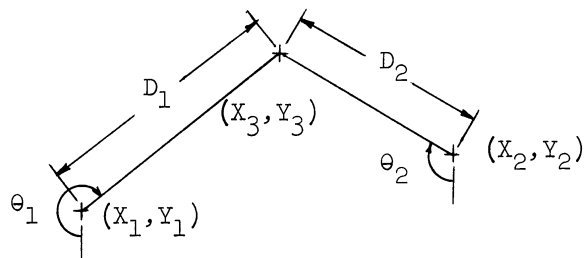
Find: D_1, θ_1



3. Line-Line Intersection

Given: $X_1, Y_1, \theta_1, X_2, Y_2, \theta_2$

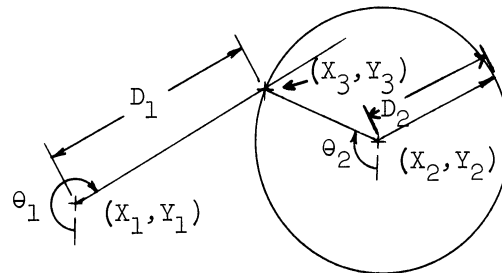
Find: X_3, Y_3, D_1, D_2



4. Line-Circle Intersection

Given: $X_1, Y_1, \theta_1, X_2, Y_2, D_2$

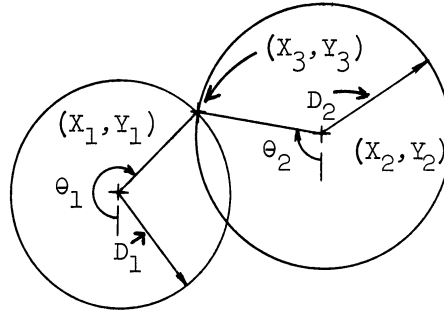
Find: X_3, Y_3, θ_2, D_1



5. Circle-Circle Intersection

Given: $X_1, Y_1, D_1, X_2, Y_2, D_2$

Find: $X_3, Y_3, \theta_1, \theta_2$



The only difference between two problems of the same type is that the numerical data are different. A computer program has been written to check the students' solutions to problems assigned in class and point out any errors. The program has been written to "penalize" students who have incorrect solutions by assigning them additional problems. On the other hand, students with correct solutions are "rewarded" by not being assigned additional problems. Because of this feature of penalties and rewards, the program which grades the student problems has been named the "Teaching Machine Program."

Use of the Teaching Machine Program

Each student is given an identification number and a set of basic data for each problem type. He then uses his identification number to modify the basic data. For example, if the student's number were 1234, selected distances and point coordinates would be changed by

$$\begin{array}{r} 4253.897 \\ 1.234 \\ \hline 4255.131 \end{array}$$

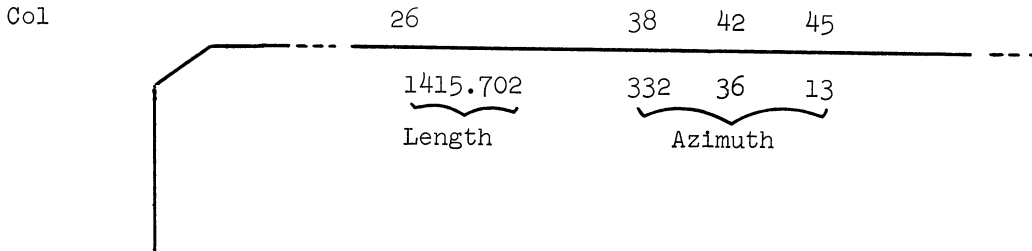
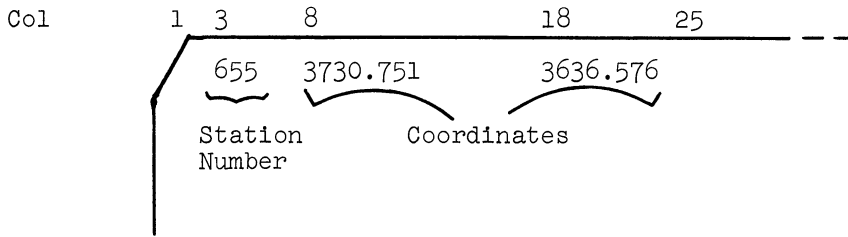
Azimuths (line directions) would be changed according to

$$\begin{array}{r} 290^{\circ} 47' 18'' \\ 12 \quad 34 \\ \hline 290^{\circ} 59' 52'' \end{array}$$

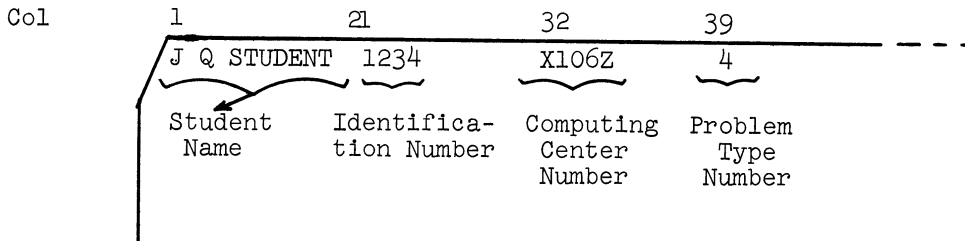
In this manner, all the students work essentially the same problem, but with slightly different data and results.

When the student has solved a problem to his satisfaction, he punches both his modified data and his computed results on IBM cards. On the first card, the station number and the coordinates of the first point are punched. The length and azimuth of the line between the first two points are punched on the second card. This sequence is repeated until all of the problem data have been given. Hence, for the Forward and Inverse Computations, each problem requires three cards. For each Line-Line Intersection problem, five cards are required. Examples of input cards are shown on the next page.

Example Problem No. 76



On the first run of each type of problem, the student must work 5 problems (3 for the Circle-Circle intersection). In front of the data cards, he places a card bearing his name, class number, computing center number, and the type number of the problems which follow. For example, a name card might be



The data for each type of problem must be submitted separately, with an individual name card for the set of data for each type.

At execution, the program first reads the current date to be used in output headings. Following this is the class roll. Each card in this deck contains a student's name and class number, the number of problems due and the number of the run to be made for each of the five types of problems. Next, the five sets of basic data are read and then the first student's data is considered.

The class roll, the sets of basic data, and each set of student data is followed by a card having an equal sign punched in column 1. Whenever this equal sign is read, control is transferred to the portion of the program which handles the type of data expected to follow.

In processing student data, the information on the student's name card is read and printed in a heading. Then the student's class number is checked to see if it is a legal number. The corresponding run number is checked to see if the student is attempting an extra run. Finally, the student's name is checked to see if it corresponds with the given class

number. If any of these tests fail, an appropriate comment is printed and the program searches for the name card of the next student. If the information is all legal, control is transferred to the section of the program which handles the type of problem given on the name card. Each of these five sections modifies the basic data in the same manner as the student and computes the correct solution to the problem. These results are then used by the internal function CHECK.

CHECK reads the student's data and compares it with the computed data, with a tolerance of .003 feet on distances and coordinates and 3 seconds on azimuths. The student's data are then printed, with asterisks flagging any incorrect values. Whenever an error occurs, a boolean flag is set, and at the end of the problem, the value of the flag is returned to the main program. If the flag has been set, the main program increments an error counter. Then the data for the next problem (of the same type) are examined, the correct solutions are computed, and CHECK is again called. This process continues until the required number of problems for this sequence have been considered or until an equal-sign card is read. Extra problems are ignored and simply scanned for an equal-sign card.

At the end of the set of data, if the error counter is not equal to zero, or if too few problems have been submitted, the number due is increased. On the next run, the student must resubmit all the correct problems, the problems which were incorrect or missing (hopefully corrected by this time), and an extra problem for each incorrect or missing problem. This new due number is printed following the data listing. If all the problems are correct, the due number is set to zero and a comment is printed.

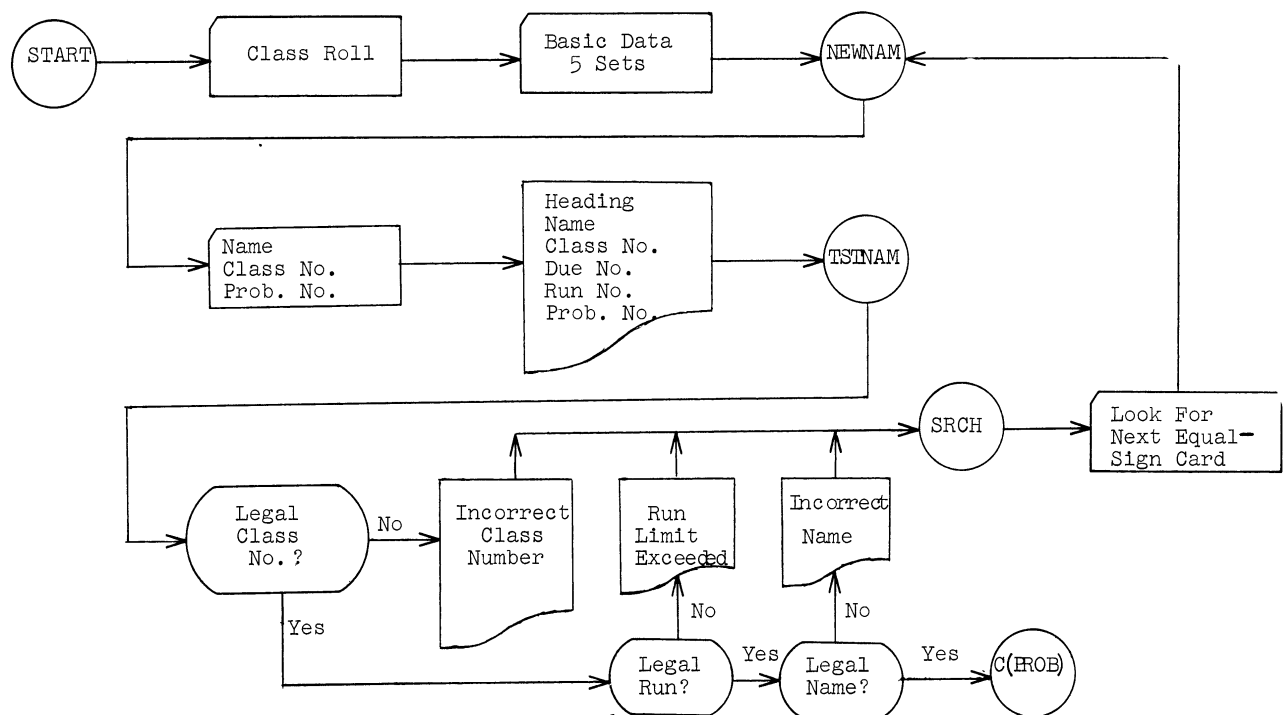
The run number is now incremented and the program goes on to the next student's data.

After all the problem sets have been processed, the updated class roll is punched for use in the next run and printed for the information of the instructor.

Flow Diagram

A general flow diagram of the Teaching Machine Program is shown below.

- C(1) - Forward Computation
- C(2) - Inverse Computation
- C(3) - Line-Line Intersection
- C(4) - Line-Circle Intersection
- C(5) - Circle-Circle Intersection



The Computer as a Teaching Machine

MAD Program, Continued

```

1,FS4(J),INT4(J),DT04(J),TO4(J),TOX4(J),TOY4(J)
L4      WHENEVER EQUALS,E,$=$,TRANSFER TO NEXT2
        PRINT COMMENT$4TOO MUCH DATA FOR PROBLEM 4$
        EXECUTE SYSTEM.
NEXT2   THROUGH L5, FOR J=1,1,J.G.25
        READ FORMAT RD5,EQUALS,FROM5(J),FRX5(J),FRY5(J),DFM5(J),INT5(
1J),DT05(J),TO5(J),TOX5(J),TOY5(J)
L5      WHENEVER EQUALS,E,$=$,TRANSFER TO NEWNAM
        PRINT COMMENT$4TOO MUCH DATA FOR PROBLEM 5$
        EXECUTE SYSTEM.
NEWNAM  ERR=0
        TER=0B
        READ FORMAT RDNAME,EQUALS,SNAME...SNAME(2),SCN,PROB
        WHENEVER EQUALS,E,$=$,TRANSFER TO NEWNAM
        PRINT FORMAT TITLEB,SNAME...SNAME(2),SCN,PROB,DATE,DATE1
1,DATE2
        THROUGH L1, FOR K=1,1,K.E.N
L1      WHENEVER SCN.E.CN(K),TRANSFER TO TSTRUN(PROB)
        PRINT COMMENT $0INCORRECT CLASS NUMBERS$
        TRANSFER TO SRCH
TSTRUN(1) WHENEVER RUN1(K).L.4,TRANSFER TO TSTNAM
        PRINT COMMENT $ORUN LIMIT EXCEEDED ON PROBLEM 1$
        TRANSFER TO SRCH
TSTRUN(2) WHENEVER RUN2(K).L.4,TRANSFER TO TSTNAM
        PRINT COMMENT $ORUN LIMIT EXCEEDED ON PROBLEM 2$
        TRANSFER TO SRCH
TSTRUN(3) WHENEVER RUN3(K).L.4,TRANSFER TO TSTNAM
        PRINT COMMENT $ORUN LIMIT EXCEEDED ON PROBLEM 3$
        TRANSFER TO SRCH
TSTRUN(4) WHENEVER RUN4(K).L.4,TRANSFER TO TSTNAM
        PRINT COMMENT$ORUN LIMIT EXCEEDED ON PROBLEM 4$
        TRANSFER TO SRCH
TSTRUN(5) WHENEVER RUN5(K).L.4,TRANSFER TO TSTNAM
        PRINT COMMENT$ORUN LIMIT EXCEEDED ON PROBLEM 5$
        TRANSFER TO SRCH
TSTNAM  WHENEVER NAME(K).NE.SNAME.OR.NAME1(K).NE.SNAME(1).OR.NAME2(K
1).NE.SNAME(2)
        PRINT COMMENT $0INCORRECT NAMES$
SRCH    READ FORMAT SEARCH,EQUALS
        WHENEVER EQUALS,E,$=$,TRANSFER TO NEWNAM
        TRANSFER TO SRCH
        END OF CONDITIONAL
        TRANSFER TO C(PROB)
C(1)   PRINT FORMAT TITLEC,PROB,DUE1(K),RUN1(K)
        THROUGH BIGLP1,FOR I=1,1,I.G.DUE1(K)
        STA=FROM(I)
        STA(1)=TO(I)
        X=FRX(I)
        Y=FRY(I)
        DIST=DTO(I)+SCN/1000.
        D1=D(I)
        M1=M(I)+SCN/100
        S1=S(I)+SCN-100*(SCN/100)
        SEC=3600*D1+60*M1+S1
        RA=.48481368E-5*SEC
        X(1)=FRX(I)-SIN.(RA)*DIST
        Y(1)=FRY(I)-COS.(RA)*DIST
        TER=CHECK.(THRU1)
        WHENEVER TER
        ERR=ERR+1
        TER=0B
        END OF CONDITIONAL
BIGLP1  WHENEVER ERR.E.0
        DUE1(K)=0
        PRINT FORMAT OK,PROB
        TRANSFER TO RK1
        END OF CONDITIONAL
        DUE1(K)=DUE1(K)+ERR
TEST1   WHENEVER DUE1(K).G.15,DUE1(K)=15
        PRINT FORMAT RESMIT,DUE1(K)
RK1     RUN1(K)=RUN1(K)+1
        TRANSFER TO SRCH
THRU1   DUE1(K)=2*DUE1(K)+1+ERR-I
        TRANSFER TO TEST1
C(2)   PRINT FORMAT TITLEC,PROB,DUE2(K),RUN2(K)
        THROUGH BIGLP2,FOR I=1,1,I.G.DUE2(K)
        STA=FROM2(I)

```

MAD Program, Continued

```

      X=FRX2(I)+SCN/1000.
      Y=FRY2(I)
      DX=TOX2(I)-SCN/1000.-X
      DY=TOY2(I)-FRY2(I)
      DIST=SQRT.(DX*DX+DY*DY)
      SEC=ATN1.(-DX,-DY)/0.48481368E-5+0.5
      STA(1)=TO2(I)
      X(1)=X+DX
      Y(1)=TOY2(I)
      TER=CHECK.(THRU2)
      WHENEVER TER
      ERR=ERR+1
      TER=0B
BIGLP2  END OF CONDITIONAL
      WHENEVER ERR.E.0
      DUE2(K)=0
      PRINT FORMAT OK,PROB
      TRANSFER TO RK2
      END OF CONDITIONAL
      DUE2(K)=DUE2(K)+ERR
TEST2   WHENEVER DUE2(K).G.15,DUE2(K)=15
      PRINT FORMAT RESMIT,DUE2(K)
RK2     RUN2(K)=RUN2(K)+1
      TRANSFER TO SRCH
THRU2   DUE2(K)=DUE2(K)*2+1+ERR-I
      TRANSFER TO TEST2
C(3)    PRINT FORMAT TITLEC,PROB,DUE3(K),RUN3(K)
      THROUGH BIGLP3, FOR I=1,1,I.G.DUE3(K)
      STA=FROM3(I)
      STA(1)=INT3(I)
      STA(2)=TO3(I)
      X=FRX3(I)
      Y=FRY3(I)
      X(2)=TOX3(I)
      Y(2)=TOY3(I)
      D1=FD3(I)
      D2=TD3(I)
      M1=FM3(I)+SCN/100
      M2=TM3(I)+SCN/100
      S1=FS3(I)+SCN-100*(SCN/100)
      S2=TS3(I)+SCN-100*(SCN/100)
      SEC=3600*D1+60*M1+S1
      SEC(1)=3600*D2+60*M2+S2
      RA=.48481368E-5*SEC
      RB=.48481368E-5*SEC(1)
      SA=-SIN.(RA)
      SB=-SIN.(RB)
      CA=-COS.(RA)
      CB=-COS.(RB)
      DX=X(2)-X
      DY=Y(2)-Y
      DA=(DX*CB-DY*SB)/SIN.(RA-RB)
      X(1)=X+DA*SA
      Y(1)=Y+DA*CA
      DB=(DY-DA*CA)/CB
      DIST=DA
      DIST(1)=DB
      TER=CHECK.(THRU3)
      WHENEVER TER
      ERR=ERR+1
      TER=0B
BIGLP3  END OF CONDITIONAL
      WHENEVER ERR.E.0
      DUE3(K)=0
      PRINT FORMAT OK ,PROB
      TRANSFER TO RK3
      END OF CONDITIONAL
      DUE3(K)=DUE3(K)+ERR

```

MAD Program, Continued

```

TEST3      WHENEVER DUE3(K).G.15,DUE3(K)=15
            PRINT FORMAT RESMIT,DUE3(K)
RK3        RUN3(K)=RUN3(K)+1
            TRANSFER TO SRCH
THRU3      DUE3(K)=2*DUE3(K)+1+ERR-I
            TRANSFER TO TEST3
C(4)       PRINT FORMAT TITLEC,PROB,DUE4(K),RUN4(K)
            THROUGH BIGLP4, FOR I=1,1,I.G.DUE4(K)
            STA=FROM4(I)
            STA(1)=INT4(I)
            STA(2)=TO4(I)
            X=FRX4(I)
            Y=FRY4(I)
            X(2)=TOX4(I)
            Y(2)=TOY4(I)
            D1=FD4(I)
            M1=FM4(I)+SCN/100
            S1=FS4(I)+SCN-100*(SCN/100)
            SEC=3600*D1+60*M1+S1
            WHENEVER DTO4(I).L.0.
            DIST(1)=DTO4(I)-SCN/1000.
            OTHERWISE
            DIST(1)=DTO4(I)+SCN/1000.
            END OF CONDITIONAL
            DX=X(2)-X
            DY=Y(2)-Y
            RA=.48481368E-5*SEC
            SA=-SIN.(RA)
            CA=-COS.(RA)
            Q=DX*SA+DY*CA
            H=DX*CA-DY*SA
            F=SQRT.(DIST(1)*DIST(1)-H*H)
            WHENEVER (DX*DX+DY*DY-DIST(1)*DIST(1)).L..0.OR.DIST(1).L..0
            DIST=Q+F
            OTHERWISE
            DIST=Q-F
            END OF CONDITIONAL
            X(1)=X+SA*DIST
            Y(1)=Y+CA*DIST
            RB=ATN1.(X(1)-X(2),Y(1)-Y(2))
            SEC(1)=RB/.48481368E-5+.5
            TER=CHECK.(THRU4)
            WHENEVER TER
            ERR=ERR+1
            TER=0B
BIGLP4     END OF CONDITIONAL
            WHENEVER ERR.E.0
            DUE4(K)=0
            PRINT FORMAT OK ,PROB
            TRANSFER TO RK4
            END OF CONDITIONAL
            DUE4(K)=DUE4(K)+ERR
TEST4      WHENEVER DUE4(K).G.15,DUE4(K)=15
            PRINT FORMAT RESMIT,DUE4(K)
RK4        RUN4(K)=RUN4(K)+1
            TRANSFER TO SRCH
THRU4      DUE4(K)=2*DUE4(K)+1+ERR-I
            TRANSFER TO TEST4
C(5)       PRINT FORMAT TITLEC,PROB,DUE5(K),RUN5(K)
            THROUGH BIGLP5, FOR I=1,1,I.G.DUE5(K)
            STA=FROM5(I)
            STA(1)=INT5(I)
            STA(2)=TO5(I)
            X=FRX5(I)
            Y=FRY5(I)
            X(2)=TOX5(I)
            Y(2)=TOY5(I)
            DIST=DFM5(I)+SCN/1000.
            DIST(1)=DTO5(I)+SCN/1000.
            DX=X(2)-X
            DY=Y(2)-Y
            DC=SQRT.(DX*DX+DY*DY)
            F=(DC*DC+DIST*DIST-DIST(1)*DIST(1))/(2.*DC)
            H=SQRT.(DIST*DIST-F*F)
            S3=DX/DC
            C3=DY/DC
            X(1)=X+F*S3-H*C3
            Y(1)=Y+F*C3+H*S3
            RA=ATN1.(X-X(1),Y-Y(1))

```

MAD Program, Continued

```

RB=ATN1.(X(1)-X(2),Y(1)-Y(2))
SA=-SIN.(RA)
SB=-SIN.(RB)
CA=-COS.(RA)
CB=-COS.(RB)
SEC=RA/.48481368E-5+.5
SEC(1)=RB/.48481368E-5+.5
TER=CHECK.(THRU5)
WHENEVER TER
ERR=ERR+1
TEK=0B
BIGLP5  END OF CONDITIONAL
WHENEVER ERR.E.0
DUE5(K)=0
PRINT FORMAT OK ,PROB
TRANSFER TO RK5
END OF CONDITIONAL
DUE5(K)=DUE5(K)+ERR
TEST5  WHENEVER DUE5(K).G.15,DUE5(K)=15
PRINT FORMAT RESMIT,DUE5(K)
RK5    RUN5(K)=RUN5(K)+1
TRANSFER TO SRCH
THRU5  DUE5(K)=2*DUE5(K)+1+ERR-I
TRANSFER TO TEST5
LAST   PRINT FORMAT PDATE,DATE,DATE1,DATE2
THROUGH DONE, FOR I=1,1,I.E.N
PRINT FORMAT PR1,NAME(I),NAME1(I),NAME2(I),CN(I),DUE1(I),DUE2
1(I),DUE3(I),DUE4(I),DUE5(I),RUN1(I),RUN2(I),RUN3(I),RUN4(I),
2RUN5(I)
DONE   PUNCH FORMAT PCH,NAME(I),NAME1(I),NAME2(I),CN(I),DUE1(I),DUE2
1(I),DUE3(I),DUE4(I),DUE5(I),RUN1(I),RUN2(I),RUN3(I),RUN4(I),
2RUN5(I)
EXECUTE SYSTEM.
INTERNAL FUNCTION (HERE)
STATEMENT LABEL HERE
BOOLEAN FLAG
ENTRY TO CHECK.
CK=0
FLAG=0B
NWCD   READ FORMAT NEWCAD,EQUALS,ST(1),ST(3),ST(5)
WHENEVER EQUALS.E.$=$,TRANSFER TO HERE
WHENEVER ST(1).E.STA(CK)
ST(2)=$ $
OTHERWISE
ST(2)=$***$
FLAG=1B
END OF CONDITIONAL
WHENEVER.ABS.(FST(3)-X(CK)).G..003
ST(4)=$***$
FLAG=1B
OTHERWISE
ST(4)=$ $
END OF CONDITIONAL
WHENEVER .ABS.(FST(5)-Y(CK)).G..003
ST(6)=$***$
FLAG=1B
OTHERWISE
ST(6)=$ $
END OF CONDITIONAL
PRINT FORMAT OUT2,ST(1)..ST(6)
WHENEVER CK.E.1.AND.PROB.L.3,TRANSFER TO FINI
WHENEVER CK.LE.1
READ FORMAT NEXCAD,EQUALS,ST(7),ST(9)..ST(11)
WHENEVER EQUALS.E.$=$,TRANSFER TO HERE
WHENEVER .ABS.(FST(7)-DIST(CK)).G..003
ST(8)=$***$
FLAG=1B
OTHERWISE
ST(8)=$ $
END OF CONDITIONAL
BIGSEC=3600*ST(9)+60*ST(10)+ST(11)

```

The Computer as a Teaching Machine

MAD Program, Continued

```

WHENEVER,ABS.(SEC(CK)-BIGSEC).G.3
ST(12)=$***$
FLAG=1B
OTHERWISE
ST(12)=$ $
END OF CONDITIONAL
PRINT FORMAT OUT3,ST(7)...ST(12)
CK=CK+1
TRANSFER TO NWCD
END OF CONDITIONAL
FINI FUNCTION RETURN FLAG
END OF FUNCTION
EQUIVALENCE (FST,ST)
BOOLEAN TER
BOOLEAN CHECK.
INTEGER ST,DATE,DATE1,N,EQUALS,NAME,NAME1,NAME2,CN,DUE3,DUE4,
1DUE5,RUN3,RUN4,RUN5,J,FROM3,FD3,FM3,FS3,INT3,TD4,TM3,TS3,TO3,
2FROM4,FD4,FM4,FS4,INT4,TO4,FROM5,INT5,TO5,ERR,SNAME,SCN,PROB,
3K,STA,I,D1,D2,M1,M2,S1,S2,SEC,CK,ST,BIGSEC
INTEGER DUE1,DUE2,RUN1,RUN2,FROM,TO,D,M,S,FROM2,TO2,DATE2
DIMENSION NAME(100),NAME1(100),NAME2(100),CN(100),DUE3(1 ),D
1UE4(100),DUE5(100),RUN3(100),RUN4(100),RUN5(100),FROM3(25),FR
2X3(25),FRY3(25),FD3(25),FM3(25),FS3(25),INT3(25),TD3(25),
3TM3(25),TS3(25),TO3(25),TOX3(25),TOY3(25),FROM4(25),FRX4(25),
4FRY4(25),FD4(25),FS4(25),FM4(25),INT4(25),DFO4(25),TO4(25),
5TOX4(25),TOY4(25),FROM5(25),FRX5(25),FRY5(25),DFM5(25),INT5(2
65),DFO5(25),TO5(25),TOX5(25),TOY5(25),SNAME(2),STA(2),X(2),
7Y(2),DIST(1),SEC(1),ST(12),FST(12)
8,DUE1(100),DUE2(100),RUN1(100),RUN2(100)
DIMENSION FROM(25),TO(25),FRX(25),FRY(25),DFO(25),D(25),M(25)
1,S(25),FROM2(25),FRX2(25),FRY2(25),TO2(25),TOX2(25),TOY2(25)
VECTOR VALUES RDATE=$3C6*$
VECTOR VALUES TITLEC=$13H-INTERSECTION13,16,26H PROBLEMS DUE
1 FOR RUN NO.13//*$
VECTOR VALUES PR1=$1H ,S20,3C6,S6 ,I4,S4,10I6 *$
VECTOR VALUES RDNAME=$C1,3C6,I5,S14,I1*$
VECTOR VALUES TITLEB=$1H1S5,3C6,S5,I4,S5,7HPROBLEMI3,S5,2C6*$
VECTOR VALUES SEARCH=$C1*$
VECTOR VALUES OK=$1H0S5,45HYOU HAVE DONE ALL THE PROBLEMS FOR
1 ASSIGNMENT I4,13H CORRECTLY./45H0SEE YOUR INSTRUCTOR FOR
2FURTHER ASSIGNMENTS.*$
VECTOR VALUES RESMIT=$1H0S5,55HYOU HAVE AT LEAST ONE ERROR (I
INDICATED BY ASTERISKS). /34HUCORRECT THE DATA AND RESUBMIT T
2HEI4,20H PROBLEMS NOW DUE.*$
VECTOR VALUES PCH=$S1,3C6,I5,I2,9I3*$
VECTOR VALUES RD=$C1,2I4,3F10.3,S5,3I3*$
VECTOR VALUES RD1=$C1,3C6,I5,I2,9I3*$
VECTOR VALUES RD2=$C1,2(I5,2F10.3)*$
VECTOR VALUES RD3=$C1,I3,2F.3,I5,2I3,2I5,2I3,I5,2F9.3*$
VECTOR VALUES RD4=$C1,I3,2F9.3,I5,2I3,15,F10.3,I6,2F9.3*$
VECTOR VALUES RD5=$C1,I3,3F9.3,I6,F10.3,I6,2F9.3*$
VECTOR VALUES NEXCAD=$C1,S25,F9.3,I6,2I3*$
VECTOR VALUES NEWCAD=$C1,I4,2F10.3*$
VECTOR VALUES OUT2=$1H S5,I4,C3,F10.3,C3,F10.3,C3*$
VECTOR VALUES OUT3=$1H S37,F10.3,C3,I5,2I3,C3 *$
VECTOR VALUES PDATE=$1H1,S19,65H PLANE GEOMETRY TEACHING
1MACHINE MODEL 3-62 PROGRAM 9C8 3C6//S20,12HSTUDENT NAME,
2S8,12HCLASS NUMBER,S3,58HDUE1 DUE2 DUE3 DUE4 DUE5 RUN1
3RUN2 RUN3 RUN4 RUN5//*$
END OF PROGRAM
$ DATA

```


Example Problem No. 76

Data

The following is an abbreviated set of data which might be submitted to the Teaching Machine Program. The class roll has been shortened to four names. Basic data for only one problem type are shown and only two sets of student's results are shown.

\$DATA

MAY 17, 1962

D J PIKE	4118	3	3	3	3	3	1	1	1	1	1
J A FOX	1949	3	3	3	3	3	1	1	1	1	1
H W ANDRES	1144	3	3	3	3	3	1	1	1	1	1
S H HAMMOND	2703	3	3	3	3	3	1	1	1	1	1

} Class Roll

=	147	89	5000.000	5000.000	253.897	341	17	59
	148	254	4569.374	4430.280	157.016	345	50	01
	161	162	3228.603	4458.202	93.000	266	56	35
	233	234	3325.440	4388.835	101.180	63	44	43
	168	227	3911.262	4533.392	108.887	00	28	34
	167	228	3803.796	4505.241	103.256	342	49	20
	234	161	3234.698	4344.077	114.288	176	56	34
	99	26	4896.323	4957.408	1875.690	157	39	59
	85	29	4511.209	4552.191	2165.140	171	17	46
	18	15	4303.953	4984.691	908.394	324	45	23
	19	20	3784.575	4539.076	780.284	272	27	17
	21	16	5110.979	4732.670	289.450	157	27	17
	45	41	4482.951	4623.752	899.568	145	12	50
	36	44	3539.884	4589.095	543.067	272	54	44
	46	1	4542.456	4538.092	650.158	224	43	41

} Data for First Problem Type

=								
=								
=								
=								

} Data for Other Four Problem Types Would be Inserted Here

H W ANDRES	1144	TOC3A	1
147	5000.000	5000.000	255.41
89	5080.946	4758.145	341 29 43
148	4569.374	4430.280	158.160
254	4607.558	4276.799	346 01 45
161	3228.603	4458.202	94.144
162	3322.630	4462.996	267 08 19

} First Student's Solution

D J PIKE	4118	TOC3P	1
147	5000.000	5000.000	258.15
89	5079.782	4574.630	341 59 17
148	4569.374	4430.280	161.134
254	4606.930	4273.584	346 31 19
161	3228.603	4458.202	97.113
162	3325.638	4462.216	267 37 53

} Second Student's Solution

The Computer as a Teaching Machine

Computer Output

The following is the computer output which corresponds to the data shown in the previous section.

H W ANDRES		1144	PROBLEM 1	MAY 17, 1962	
INTERSECTION	1	3	PROBLEMS DUE FOR RUN NO.	1	
147	5000.000	5000.000			
89	5080.946	4758.145	255.041	341 29 43	
148	4569.374	4430.280			
			158.160	346 1 45	
254	4607.558	4276.799			
161	3228.603	4458.202			
			94.144	267 8 19	
162	3322.630	4462.996***			
YOU HAVE AT LEAST ONE ERROR (INDICATED BY ASTERISKS).					
CORRECT THE DATA AND RESUBMIT THE 4 PROBLEMS NOW DUE.					

This is returned to first student.

D J PIKE		4118	PROBLEM 1	MAY 17, 1962	
INTERSECTION	1	3	PROBLEMS DUE FOR RUN NO.	1	
147	5000.000	5000.000			
89	5079.782	4574.630***	258.015	341 59 17	
148	4569.374	4430.280			
			161.134	346 31 19	
254	4606.930	4273.584			
161	3228.603	4458.202			
			97.118	267 37 53	
162	3325.638	4462.216			
YOU HAVE AT LEAST ONE ERROR (INDICATED BY ASTERISKS).					
CORRECT THE DATA AND RESUBMIT THE 4 PROBLEMS NOW DUE.					

This is returned to second student.

This summary is printed for use by the instructor:

INVERSE COMPUTATION TEACHING MACHINE MODEL 3-62 PROGRAM 9C8 MAY 17, 1962											
STUDENT NAME	CLASS NUMBER	DUE1	DUE2	DUE3	DUE4	DUE5	RUN1	RUN2	RUN3	RUN4	RUN5
D J PIKE	4118	4	3	3	3	3	2	1	1	1	1
J A FOX	1949	3	3	3	3	3	1	1	1	1	1
H W ANDRES	1144	4	3	3	3	3	2	1	1	1	1
S H HAMMOND	2703	3	3	3	3	3	1	1	1	1	1

Number due on next run for each problem type.
Run number of next run for each problem type.

This is the updated class roll which is used as input on the next run:

D J PIKE	4118	4	3	3	3	3	2	1	1	1	1
J A FOX	1949	3	3	3	3	3	1	1	1	1	1
H W ANDRES	1144	4	3	3	3	3	2	1	1	1	1
S H HAMMOND	2703	3	3	3	3	3	1	1	1	1	1

Critique

A complete discussion of the effectiveness of the use of the Teaching Machine Program is given in the section "Introduction of Civil Engineering Students to the Use of the Computer," in the first part of this report.

Example Problem No. 77

OPTIMUM REGULATION OF A RESERVOIR FOR POWER PRODUCTION

by

J. W. Howe

Department of Mechanics and Hydraulics

State University of Iowa

Course: Water Power Engineering II Credit Hours: 1-3 Level: Graduate

Statement of the Problem

It is desired to determine the best way of operating the turbines at a reservoir in order to produce the most power in the period studied. The following physical relations are assumed:

$$\text{Reservoir volume} = (Y)^{3.5/4} \text{ (acre feet)}$$

$$\text{Spillway discharge} = 120,000 h^{3/2} \text{ (acre feet/month)}$$

$$\text{Turbine discharge} = 12,000 H^{1/2} \text{ (acre feet/month)}$$

$$\text{Tailwater discharge} = 21,000 Z^{3/2} \text{ (acre feet/month)}$$

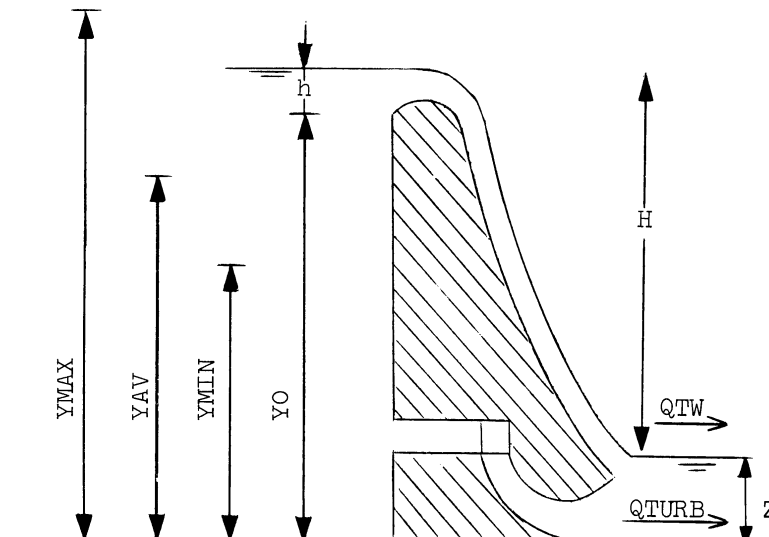
Reservoir spillway level at $Y = 63$ feet

Reservoir inflows: 12, 16, 90, 210, 47, 44, 27, 19, 45, 64, 31, 30, 10, 50, 199, 137, 59, 231, 307, 239, 75, 50, 34, 34 (thousand acre feet/month)

Determine the average power output per year if the turbines are shut down for various percentages of time when the upstream water level is below spillway level.

Solution

A cross-sectional diagram of the reservoir is shown below to illustrate the nomenclature.



Optimum Regulation of a Reservoir for Power Production

Initial levels Y1 and Z1 at beginning of month.

Final levels YAV and Z2 at end of month.

$$\text{VOLYAV} = \text{YAV}^{3.5}$$

$$\text{VOLO} = 63^{3.5}$$

$$\text{VOLYI} = \text{YI}^{3.5}$$

$$\text{QSTOR} = (\text{VOLYAV} - \text{VOLYI})/4$$

$$\text{YAV} = (\text{YMAX} + \text{YMIN})/2 = \text{level at end of month}$$

$$\text{AVHD} = (\text{YI} + \text{YAV})/2$$

STORAGE EQUATION:

$$\text{INFLOW} - \text{QSTOR} = \text{OUTFLOW} = \text{QSPLWY} + \text{QTURB} = \text{QTW}$$

$$\text{INFLOW} = \text{Q(I)} = \text{QTW} + \text{QSTOR} = \text{QTOT} \quad (1)$$

Initial values of YMIN and YMAX of 20 and 68 feet are assumed and YAV determined.

Using YAV in determining terms of Eq. (1), if $1000 \text{ Q(I)} > \text{QTOT}$, Y should be increased. This is accomplished by letting YAV replace YMIN. In the opposite situation, YAV becomes YMAX. In 15 iterations YAV is within a few thousandths of the value it must have to satisfy Eq. (1). It is assumed that the monthly discharge through the turbines or over the spillway can be secured by using the average of the initial and final water levels.

In the case of a month having some spillway discharge in the month but not during the entire month, due to rising or falling water levels, the spillway discharge for a full month is multiplied by:

$(\text{VOLYAV} - \text{VOLO})/(\text{VOLYAV} - \text{VOLYI})$ for a rising stage and by

$(\text{VOLYI} - \text{VOLO})/(\text{VOLYI} - \text{VOLYAV})$ for a falling stage

$$\text{Power production/month} = \text{QTURB} \left(\text{AVHD} - \frac{\text{Z1} + \text{Z2}}{2} \right) \times \frac{0.746 \times 0.80}{8.814} \times \frac{43560}{3600} = 0.82 \text{ QTURB} \left(\text{AVHD} - \frac{\text{Z1} + \text{Z2}}{2} \right)$$

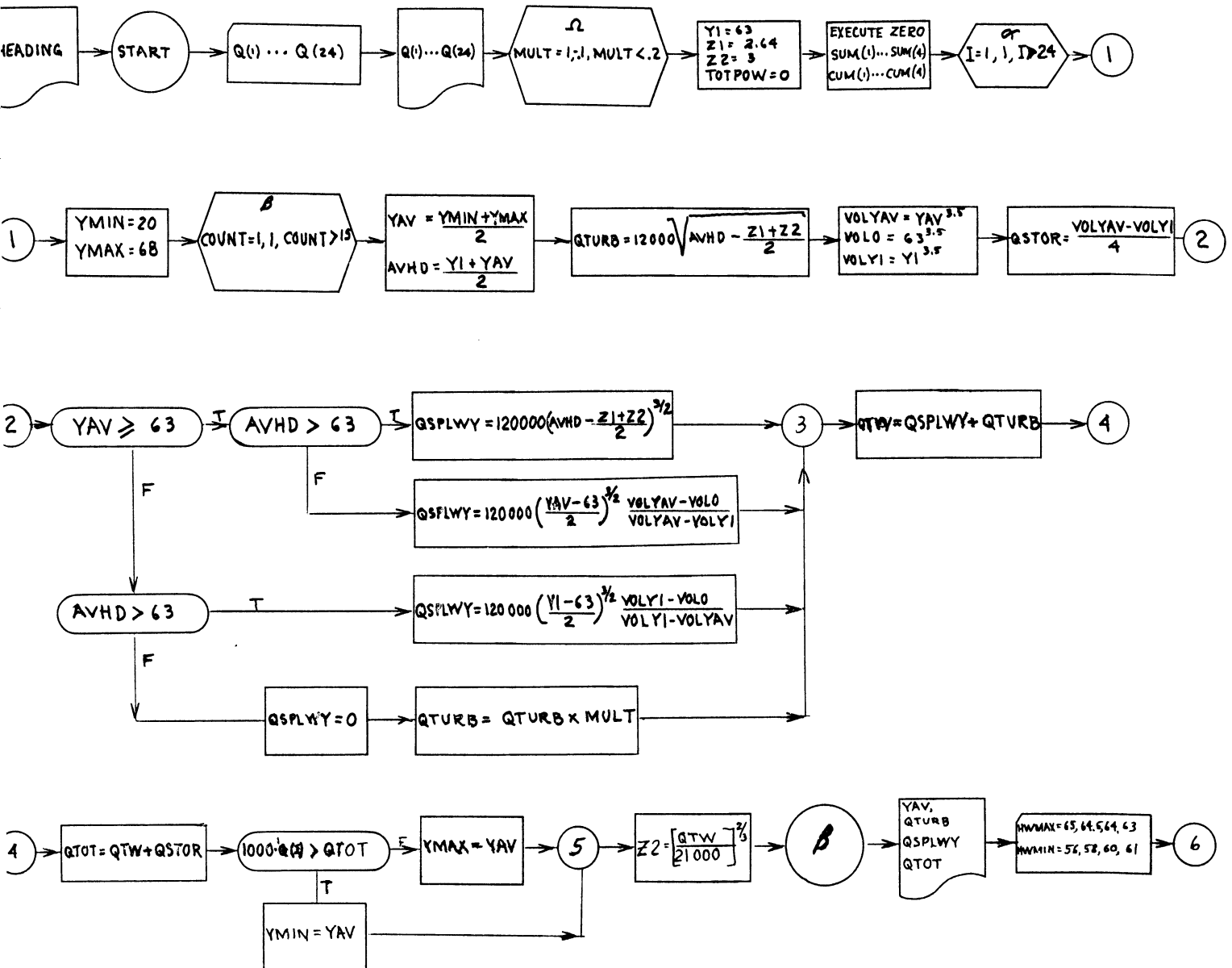
List of Symbols

Q(I)	Monthly flow, thousands of acre feet/month
Y1	Upstream depth at beginning of month, feet
Y2	Upstream depth at end of month, feet
Z1	Downstream depth at beginning of month, feet
Z2	Downstream depth at end of month, feet
YMIN	Minimum upstream depth at end of month, feet
YMAX	Maximum upstream depth at end of month, feet
YAV	Average upstream depth at end of month, feet
AVHD	Average upstream depth during month, feet

List of Symbols, Continued

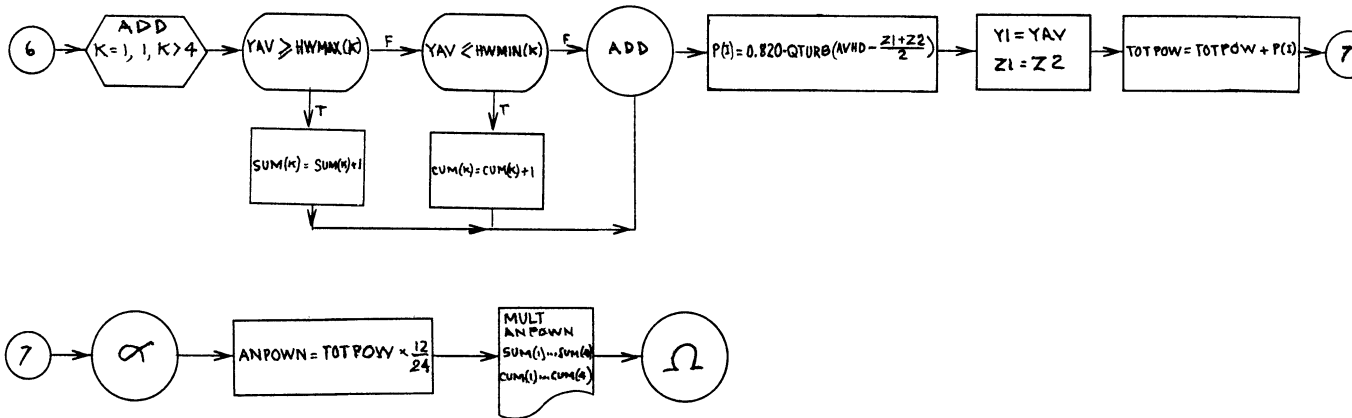
Q _{TURB}	Monthly flow through turbine, acre feet/month
Δ _{STOR}	Volume change in storage in one month, acre feet
Q _{SPLWY}	Monthly spillway discharge, acre feet/month
Q _{TW}	Monthly outflow, acre feet/month
P _(I)	Monthly power output, KWH/month
P _{NPOWN}	Annual power output, KWH/year
F _{MULT}	Fraction of time turbines are operated when upstream water level is below spillway level, dimensionless

Flow Diagram



Optimum Regulation of a Reservoir for Power Production

Flow Diagram, Continued



MAD Program and Data

```

J. W. HOWE          S164A          005  010  000  STOREG
J. W. HOWE          S164A          005  010  000  STOREG
$COMPILE MAD,EXECUTE,DUMP
  
```

```

R
ROPTIMUM REGULATION OF A POWER DEVELOPMENT AT A RESERVOIR
RASSUMED RELATIONSHIPS, VOLUME OF RESERVOIR=Y.P.3.5/4,
RQTW=21000*Z.P.1.5,QTURB=12000*SQRT.(H),
RQSPLWY=120000*(Y-63).P.1.5, POWER=Q*H*.820 KWH/MO
RQUANTITIES IN ACRE FEET/MO. SPILLWAY CREST AT Y=63FT.
R
DIMENSION Q(24), SUM(4),CUM(4),P(24)
INTEGER Q,COUNT,I,J,K,SUM,CUM
PRINT COMMENT $1$
READ FORMAT DATA,Q(1)...Q(24)
VECTOR VALUES DATA=$12I6*$
PRINT FORMAT ECHO,Q(1)...Q(24)
VECTOR VALUES ECHO=$1H 8I8*$
THROUGH OMEGA, FOR MULT=1,.-1, MULT.L..2
Y1=63.
Z1=2.64
Z2=3.
TOTPOW=0
EXECUTE ZERO.(SUM(1)...SUM(4),CUM(1)...CUM(4))
THROUGH ALPHA, FOR I=1,1,1.G.24
YMIN=20
YMAX=68.
THROUGH BETA, FOR COUNT=1,1, COUNT.G.15
YAV=(YMIN+YMAX)/2.
AVHD=(Y1+YAV)/2.
QTURB=12000.*SQRT.(AVHD-(Z1+Z2)/2.)
VOLYAV=YAV.P.3.5
VOL0=63..P.3.5
VOLY1=Y1.P.3.5
QSTOR=(VOLYAV-VOLY1)/4
WHENEVER YAV.GE.63.
WHENEVER AVHD.G.63.
QSPLWY=120000.*(AVHD-63.).P.1.5
OTHERWISE
QSPLWY=120000.*((YAV-63.)/2.).P.1.5*(VOLYAV-VOL0)/(VOLYAV-
1VOLY1)
END OF CONDITIONAL
QTW=QSPLWY+QTURB
OTHERWISE
QSPLWY=0
QTURB=QTURB*MULT
QTW=QTURB
END OF CONDITIONAL
END OF CONDITIONAL
QTOT=QTW+QSTOR
WHENEVER 1000*Q(I).G.QTOT
YMIN=YAV
OTHERWISE
  
```

MAD Program and Data, Continued

```

YMAX=YAV
END OF CONDITIONAL
BETA Z2=(QW/21000.)*P.667
PRINT FORMAT CHECK,YAV,QTURB,QSPLWY,QTOT
VECTOR VALUES CHECK=$1H0,F10.2,3F10.0*$
VECTOR VALUES HWMAX(1)=65.64,5,64.63.
VECTOR VALUES HWMIN(1)=56.58,60.61.
THROUGH ADD, FOR K=1,1,K.G.4
WHENEVER YAV.GE.HWMAX(K)
SUM(K)=SUM(K)+1
OR WHENEVER YAV.LE.HWMIN(K)
CUM(K)=CUM(K)+1
ADD END OF CONDITIONAL
P(I)=QTURB*(AVHD-(Z1+Z2)/2.)*.820
Y1=YAV
Z1=Z2
ALPHA TOTPOW=TOTPOW+P(I)
ANPOW=TOTPOW*12./24.
OMEGA PRINT FORMAT POWR,MULT,ANPOW,SUM(1)...SUM(4),CUM(1)...CUM(4)
VECTOR VALUES POWR=$1H0,6H MULT=,F5.2,12H ANNUAL KWH ,F11.1/
117H MONTHS ABOVE 65,I3/19H MONTHS ABOVE 64.5,I3/
217H MONTHS ABOVE 64,I3/17H MONTHS ABOVE 63,I3/
317H MONTHS BELOW 56,I3/17H MONTHS BELOW 58,I3/
417H MONTHS BELOW 60,I3/17H MONTHS BELOW 61,I3*$
END OF PROGRAM
$ DATA
012 016 090 210 047 044 027 019 045 064 031 030
010 050 199 137 059 231 307 239 075 050 034 034
    
```

Computer Output

12	16	90	210	47	44	27	19	} Q's
45	64	31	30	10	50	199	137	
59	231	307	239	75	50	34	34	
YAV	QTURB	QSPLWY	QTOT					
59.91	92015	0	11997					
56.68	89510	0	15981					
56.76	88262	0	90001					
61.81	90324	0	209970					
60.05	91600	0	47029					
58.08	90126	0	44001					
55.18	88189	0	26972					
51.53	85512	0	18985					
49.12	82963	0	45002					
47.90	81411	0	64014					
44.07	79182	0	31008					
39.49	75318	0	30002					
30.28	68527	0	9993					
27.21	61909	0	49998					
45.20	70035	0	199002					

Optimum Regulation of a Reservoir for Power Production

Computer Output, Continued

49.40	80462	0	137016
47.84	81526	0	59005
56.35	84506	0	231029
64.44	91118	14785	307008
63.98	93039	159229	239057
62.42	92167	26110	74969
60.81	91934	0	49989
58.46	90575	0	34027
55.94	88643	0	34014

MULT= 1.00 ANNUAL KWH 42415296.0
MONTHS ABOVE 65 0
MONTHS ABOVE 64.5 0
MONTHS ABOVE 64 1
MONTHS ABOVE 63 2
MONTHS BELOW 56 12
MONTHS BELOW 58 15
MONTHS BELOW 60 18
MONTHS BELOW 61 20

Discussion of Results

Only two years (24 months) of data were used which required 2.2 and 0.4 minutes, respectively, of execution and compiling time. Because of space limitations, the printed output from the computer is shown for only one percentage of turbine operation (i.e., the case of full-time turbine operation). The results indicate the desirability of decreasing the range of turbine operation examined and increasing the number of months studied. Eight years of record could be studied in five minutes of machine time with this modification.

The results revealed an unexpected advantage to an operation involving considerable drafting of the reservoir even at substantial reductions of head since such operation produced the greatest amount of energy conversion.

Critique

This problem has been used in a second course in Water Power Engineering but, because of the complexity of numerical work, full solutions have not been required previously. Since the computer makes a trial and error solution for the upstream and downstream water levels feasible, the problem can now be fully solved and will be assigned to students in the future.

Example Problem No. 78

ECONOMICAL SIZE OF CANAL FEEDING HYDRO-POWER PLANT

by

J. W. Howe

Department of Mechanics and Hydraulics

State University of Iowa

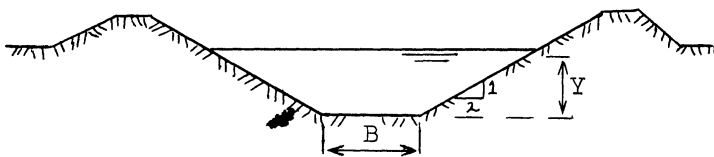
Course: Water Power Engineering Credit Hours: 3 Level: Senior and Graduate

Statement of the Problem

A canal having 1:2 side slopes is to feed a hydro-power plant. It is to have best hydraulic efficiency ($R = D/2$) when carrying the mean discharge of 1500 cubic feet per second, but capacity for the maximum discharge of 2000 cubic feet per second with a 2-foot freeboard. Excavation is limited to 60 per cent of the maximum depth with the excavated material forming berms on either side to provide the upper portion of the cross-section. Transverse slope of the ground is assumed to be zero but any longitudinal slope of the canal is possible by slight divergence from a contour. The following data are given: Canal roughness coefficient, $n = 0.025$; excavation cost, \$2.00/cubic yard; annual charge for maintenance, interest, and taxes, 10 per cent; cost of power, \$0.006/KWH; operation schedule, 16 hours/day, 300 days/year. Slopes varying from 0.00020 to 0.00046 are to be used. Determine the dimensions of the cross-section for which the annual cost is least.

Solution

A diagram of a typical canal cross-section is shown below to illustrate the important dimensions.



$$A = B \times Y + 2Y^2$$

$$R = A/P = \frac{BXY + 2Y^2}{B + 4.4722 \times Y} = \frac{Y}{2}$$

$$B = 0.4722 \times Y$$

For a series of slopes varying from 0.00020 to 0.00046 by 0.00002 increments, the Manning formula is solved by a trial and error process. Since the channel is to have best hydraulic efficiency at mean discharge ($R = Y/2$):

By the Manning formula

$$Q = \frac{1.49}{N} A R^{2/3} S^{1/2}$$

$$Q = \frac{1.49}{N} \frac{(BXY + 2Y^2)^{5/3}}{(B + 4.4722 \times Y)^{2/3}} S^{1/2}$$

Economical Size of Canal Feeding Hydro-Power Plant

Substituting $B = 0.4722Y$

$$Q = \frac{1.5}{N} \frac{(0.4722 Y^2 + 2Y^2)^{5/3}}{(0.4722Y + 4.4722Y)^{2/3}} S^{1/2} = \frac{(2.472 Y^2)^{5/3}}{(4.944 Y)^{2/3}}$$

$$Y = \frac{(Q N)^{3/8}}{1.375 S^{3/16}} \quad B = 0.4722 Y = \frac{0.4722 (Q N)^{3/8}}{1.375 S^{3/16}}$$

Hence, there is a particular bottom width for each slope which satisfies the condition $R = Y/2$ for the mean discharge.

The required depth for maximum discharge at each slope and corresponding bottom width are next found by trial and error starting with given minimum and maximum values of Y and using their mean as the depth. After 11 iterations the mean should have the correct value.

$$Q_{MAX} = \frac{1.5}{N} \frac{[B \times YAV + 2 \times (YAV)^2]^{5/3}}{(B + 4.4722 YAV)^{2/3}} S^{1/2}$$

$$L = (Q_{MAX} \times N) / 1.5 \times S^{1/2}$$

$$R = [B \times YAV + 2(YAV)^2]^{5/3} / (B + 4.4722 YAV)^{2/3}$$

Whenever $L > R$

$$Y_{MIN} = YAV$$

Whenever $L \leq R$

$$Y_{MAX} = YAV$$

$$\text{Depth} = 0.6 \times (\text{Final value of } Y_{MAX} + 2)$$

$$\text{Annual excavation cost/foot length} = C \times \text{INT} \times (\text{Vol. of Exc.} =$$

$$C \times \text{INT} \times [B \times 0.6 \text{ DEPTH} + 2(0.6 \text{ DEPTH})^2]$$

$$\text{Ann. Cost of Power loss/foot length} = Q_{MEAN} \times S \times H \times D \times \frac{P}{100} \times \frac{0.746}{11}$$

$$= Q_{MEAN} \times S \times H \times D \times \frac{P}{1475}$$

$$\text{Total Annual Cost} = \text{Annual charge for Excavation plus Annual Cost of Power Loss}$$

List of Symbols

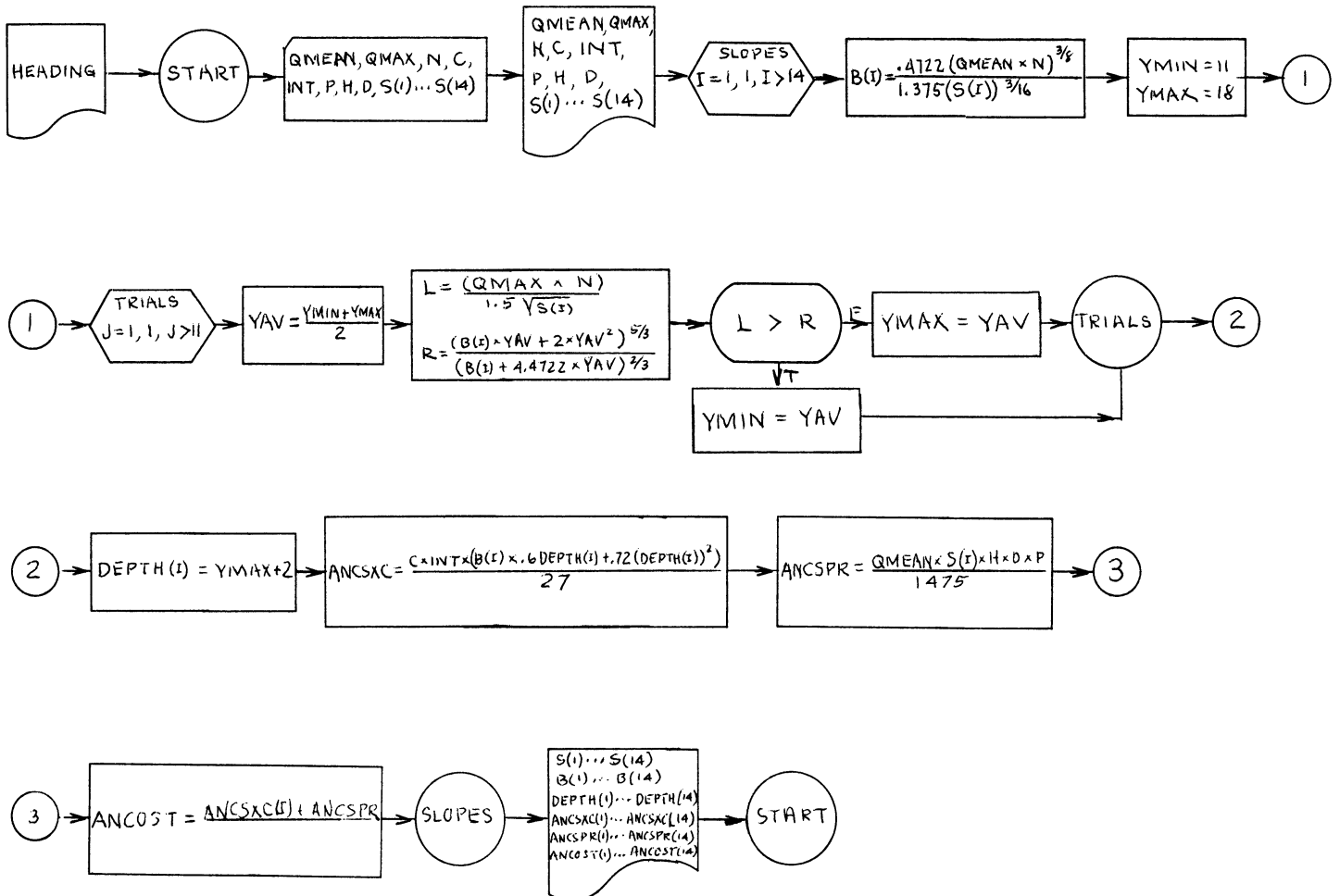
Q_{MEAN}	average discharge rate, ft^3/sec
Q_{MAX}	maximum discharge rate, ft^3/sec
A	cross-sectional area, ft^2
R	hydraulic radius, ft
N	roughness coefficient n in Manning formula
S	slope of bottom in direction of flow, dimensionless
B	bottom width of channel, ft
Y_{MIN}	lowest depth of water, ft
YAV	mean depth of water, ft

Example Problem No. 78

List of Symbols, Continued

- YMAX highest depth of water, ft
- DEPTH total depth of canal with 2-ft freeboard
- C cost of excavation, $\$/\text{yd}^3$
- INT annual fixed charges (interest, taxes, maintenance), dimensionless
- P value of power, cents/KWH
- H hours of operation/day, hrs
- D days of operation/yr, days
- ANCSXC annual cost of canal, $\$/\text{yr}$
- ANCSPR annual cost of power lost, $\$/\text{yr}$
- ANCOST total annual cost, $\$/\text{yr}$

Flow Diagram



Economical Size of Canal Feeding Hydro-Power Plant

MAD Program and Data

```

J. W. HOWE          S164A          001  005  000      CANAL
J. W. HOWE          S164A          001  005  000      CANAL
$ COMPILE MAD, EXECUTE, DUMP, PRINT OBJECT
R
RCANAL IN EARTH WITH 1/2SIDE SLOPES, MANNING ROUGHNESS N, BEST
RHYDRAYLIC RADIUS (0.4722*Y FOR QMEAN),QMEAN AND QMAX IN CFS,
REXCAVATED DEPTH 0.6(YMAX+2). COST OF EXCAVATION C $/CUYD,
RFIXED CHARGE INT, POWER COST P CENTS/KWH, OPERATION H HOURS/
RDAY, D DAYS/YR. ANY SLOPE AVAILABLE BY DIVERGENCE FROM
RCONTOUR. TRANSVERSE SLOPE ASSUMED ZERO.
R
  DIMENSION S(14),B(14),ANCSXC(14),ANCSPR(14),ANCCOST(14),
  1DEPTH(14)
  INTEGER I,J
  PRINT COMMENT $1$
  READ FORMAT DATA,QMEAN,QMAX,N,C,INT,P,H,D,S(1)...S(14)
  VECTOR VALUES DATA=$2F10.0,F10.3,3F8.2,2F8.0/(7F10.5)*$
  PRINT FORMAT ECHO,QMEAN,QMAX,N,C,INT,P,H,D,S(1)...S(14)
  VECTOR VALUES ECHO = $1H 2F10.0,F10.3,3F8.2,2F8.0/
  1 (1H07F10.5)*$
  THROUGH SLOPES, FOR I=1,1,I.G.14
  B(I)=.4722*(QMEAN*N).P..375/(1.375*S(I).P..1875)
  YMIN=11.
  YMAX=18.
  THROUGH TRIALS, FOR J=1,1,J.G.11
  YAV=(YMIN+YMAX)/2.
  L=(QMAX*N)/(1.5*SQRT.(S(I)))
  R=(B(I)*YAV+2*YAV*YAV).P.1.667/(B(I)+4.4722*YAV).P..667
  WHENEVER L.G.R
  YMIN=YAV
  OTHERWISE
  YMAX=YAV
  TRIALS  END OF CONDITIONAL
  DEPTH(I)=YMAX+2
  ANCSXC(I)=C*INT*(B(I)*.6*DEPTH(I)+.72*DEPTH(I)*DEPTH(I))/27
  ANCSPR(I)=QMEAN*S(I)*H*D*P/1475
  SLOPES  ANCCOST(I)=ANCSXC(I)+ANCSPR(I)
  PRINT COMMENT $4$
  PRINT FORMAT RESLTS,S(1)...S(14),B(1)...B(14),DEPTH(1)...DEPT
  1H(14),ANCSXC(1)...ANCSXC(14),ANCSPR(1)...ANCSPR(14),ANCCOST(1)
  2...ANCCOST(14)
  VECTOR VALUES RESLTS = $1H 7F10.5/1H07F10.5/(1H07F10.2)*$
  END OF PROGRAM
$ DATA
  1500.      2000.      .025      2.00      .10      .60      16.      300.
    .00020    .00022    .00024    .00026    .00028    .00030    .00032
    .00034    .00036    .00038    .00040    .00042    .00044    .00046

```

Example Problem No. 78

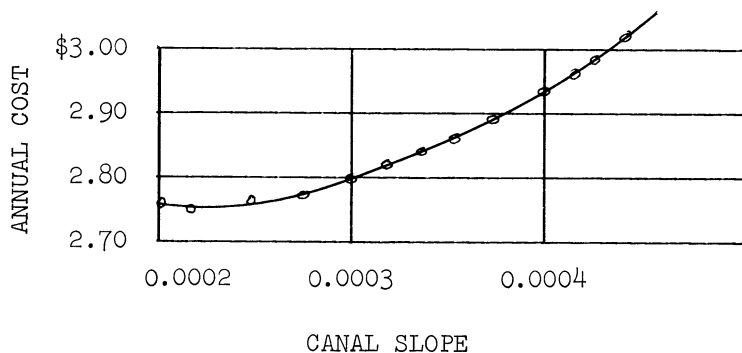
Computer Output

1500	2000	.025	2.00	.10	.60	16	300	} Input Data
.00020	.00022	.00024	.00026	.00028	.00030	.00032	.00036	
.00034	.00036	.00038	.00040	.00042	.00044	.00046		
.00020	.00022	.00024	.00026	.00028	.00030	.00032		} Slopes
.00034	.00036	.00038	.00040	.00042	.00044	.00046		
6.60	6.48	6.38	6.28	6.20	6.12	6.04		} Bottom Widths
5.98	5.91	5.85	5.80	5.74	5.69	5.65		
17.74	17.46	17.21	16.99	16.78	16.59	16.42		} Depths
16.25	16.10	15.96	15.83	15.70	15.58	15.47		
2.20	2.13	2.07	2.01	1.96	1.92	1.88		} Annual Cost of Canal
1.84	1.81	1.77	1.74	1.72	1.69	1.66		
.59	.64	.70	.76	.82	.88	.94		} Annual Cost of Power Lost
1.00	1.05	1.11	1.17	1.23	1.29	1.35		
2.79	2.77	2.77	2.78	2.78	2.80	2.82		} Total Annual Cost
2.84	2.86	2.89	2.92	2.95	2.98	3.01		

*

Discussion of Results

A plot is shown below of the total annual cost as a function of the canal slope. It is seen that a minimum cost of \$2.77 occurs for a slope of 0.00023.



The problem involves 0.3 minutes of compiling time and 0.7 minutes total time.

Critique

This is a problem regularly given to students in a Water Power course. Without the use of the computer, the students can obtain a satisfactory but laborious solution using special tables. Plotting of computed results is usually necessary for interpolation between points.

Economical Size of Canal Feeding Hydro-Power Plant

The use of the computer for obtaining a trial and error solution replaces the tables, makes it feasible to compute many more points and affords an easy way to secure solutions for different original data with little additional work.

The problem, though short, illustrates the advantage of the computer approach to an otherwise laborious problem and provides a good exercise for students who are learning to program.

Example Problem No. 79

MOST PROBABLE NUMBER METHOD FOR DETERMINING BACTERIAL POPULATIONS

by

L. L. Kempe

Sanitary Engineering

The University of Michigan

Course: Industrial Bacteriology

Credit hours: 3

Level: Senior or Graduate

Statement of Problem

Write a MAD program that will solve, using the Newton-Raphson method, the MPN (Most Probable Number) equation. The input of the program will be N , a_i , n_i , p_i . The output should be X , the most probable number of bacteria, and P , the probability associated with X .

Solution

The MPN equation is based on the extinction dilution method for finding the number of viable bacteria in a solution. The development of the MPN equation is well documented (1,2) and will only be reviewed here.

Assuming a negative exponential probability of growth, the probability of bacteria growing in n tubes of culture medium inoculated with " a " milliliters of a solution containing X bacteria per milliliter is

$$\frac{n!}{p!(n-p)!} (1 - e^{-aX})^p (e^{-aX})^{n-p}$$

If several different dilutions are used, the probability of having p_i tubes show growth out of a total of n_i tubes inoculated with a_i ml. each is

$$\prod_{i=1}^n \left(\frac{n_i!}{p_i!(n_i-p_i)!} \right) \left(1 - e^{-a_i X} \right)^{p_i} \left(e^{-a_i X} \right)^{n_i - p_i} \quad (1)$$

This gives an expression for the probability of n_i out of p_i tubes showing growth when the solution contains X bacteria per ml. Now an expression will be found for the most probable X given a set of a_i , n_i , p_i . To do this, it is noted that if P (the probability) is a maximum, then $\ln P$ will also be a maximum. Taking the natural logarithm of (1)

$$\ln P = \sum_{i=1}^N \left[\ln \left(\frac{n_i!}{p_i!(n_i-p_i)!} \right) - (n_i-p_i)a_i X + p_i \ln (1 - e^{-a_i X}) \right] \quad (2)$$

Thus

$$\frac{d}{dX} \ln P = \sum_{i=1}^N a_i \left(\frac{p_i}{1 - e^{-a_i X}} - n_i \right) = 0 \quad , \quad (3)$$

Method for Determining Bacterial Populations

which is an implicit equation in X and gives the most probable number (MPN) of bacteria per ml. of a solution used in an extinction dilution.

The approach to the problem solution is to use the Newton-Raphson method to find the X that satisfies the MPN equation. Then, given this X, the probability associated with it is calculated using equation (2). The probability will give a measure of the reliability of the experiment. A low probability would say that the experimental results were not reliable.

Several extra features were incorporated in the solution. These include the ability to do a single calculation or a table of calculations. An automatic estimation of the initial value of X is also included. Any number of dilutions from 3 to 9 may be used for input data. The output format is automatically adjusted to the number of dilutions.

This program is much more elegant than required, but illustrates the power of a reasonably short computer program. Further information is included as remarks in the program.

List of Symbols

Problem Notation

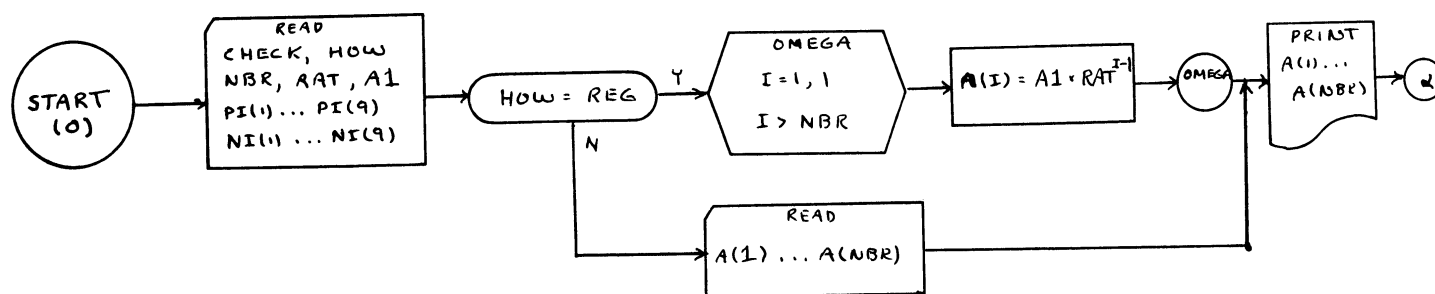
Program

N	number of different dilutions used	NBR
a_1	number of ml. used for inoculation	A(I)
n_1	total number of tubes inoculated	N(I)
p_1	number of tubes inoculated which show growth	P(I)
X	most probable number of bacteria	X
$P(\%)$	probability associated with X	PPER

Important additional variables:

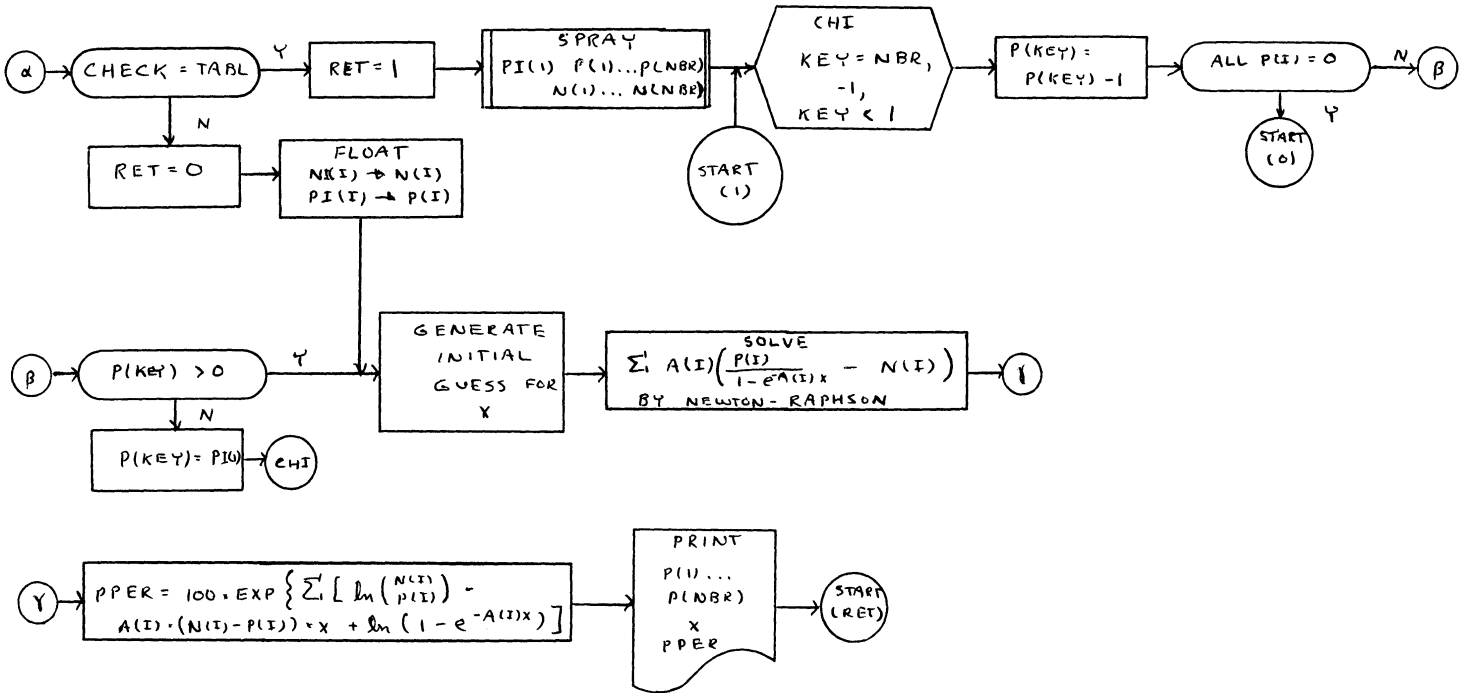
G	=	MPN function
GP	=	$\frac{dG}{dX}$
DELX	=	Δx

Flow Diagram



Example Problem No. 79

Flow Diagram (continued)



MAD Program and Data

SCOMPILE MAD,PRINT OBJECT,PUNCH OBJECT,EXECUTE,DUMP

MPN 001

R
R PROGRAM FOR DETERMINING BACTERIAL POPULATION
R USING THE MPN EQUATION

R
R
R CODED BY W.J. SANDERS 4/3/62

R
R
R THIS PROGRAM USES A METHOD DEVELOPED BY HALVORSON AND
R RZIEGLER (J. BACT.,25,101-121(1933)) AND PROGRAMMED FOR THE
R RIBM 650 BY NORMAN AND KEMPE (BIOTECHNOLOGY AND BIOENGINEERING
R RII-2,157-163(1960)). IT INVOLVES THE COMPUTATION OF THE
R RMOST PROBABLE NUMBER OF VIABLE BACTERIA IN A SAMPLE OF
R RATERIAL. INPUT DATA ARE THE RESULTS OF AN EXTINCTION
R RDILUTION EXPERIMENT AND ARE ARRANGED AS FOLLOWS

R
R
R RCARD TYPE I
R
R COL VAR
R
R 1-4 CHECK
R 5-7 HOW
R 8-10 NBR
R 11-14 RAT
R 15-22 A1
R 23,24 P(1) 25,26 P(2) ... 39,40 P(9)
R 41,42 N(1) 43,44 N(2) ... 57,58 N(9)

R
R
R RCARD TYPE II
R
R
R 1-8 A(1) 9-16 A(2) ... 65-72 A(9)

Method for Determining Bacterial Populations

MAD Program and Data (Continued)

```

R
R
R
R MOST NOTATION FOLLOWS THAT OF HALVORSON AND ZIEGLER.
R
R   NBR = NUMBER OF DILUTIONS
R   N(I) = NUMBER OF SAMPLES OF VOLUME A(I) ML. (I=1,...,NBR)
R   P(I) = NUMBER OF FAILURES OUT OF N(I) TRIALS. FAILURE
R           MEANS GROWTH IN TUBES
R
R
R
R THERE ARE TWO METHODS OF INPUT OF THE A(I). THEY DEPEND
R UPON THE SETTING OF THE VARIABLE HOW.
R
R HOW = REG
R   THIS OPTION IS USED WHEN A REGULAR DILUTION RATIO
R   (E.G., DECIMAL) IS USED. IT IS ONLY NECESSARY TO PROVIDE
R   THE FOLLOWING ADDITIONAL DATA
R
R       A1 = A(1)
R       RAT = DILUTION RATIO
R
R   THE FORMULA  $A(I) = A1 * RAT * P_{i-1}$  IS THEN USED TO
R   COMPUTE THE A(I). CARD TYPE II IS NOT NEEDED.
R
R HOW = VAR
R   THIS OPTION IS USED WHEN THERE IS NO REGULAR DILUTION
R   RATIO. VARIABLES A1 AND RAT ARE IGNORED AND AN
R   ADDITIONAL CARD OF TYPE II IS REQUIRED.
R
R
R
R THERE IS ALSO THE OPTION OF DOING A SINGLE CALCULATION OR THE
R PRINTING OF A TABLE OF CALCULATIONS. THIS DEPENDS UPON THE
R SETTING OF THE VARIABLE CHECK.
R
R CHECK = (BLANKS)
R   A SINGLE CALCULATION IS MADE USING THE VALUES OF N(I)
R   AND P(I) ON THE CARD.
R
R CHECK = TABL
R   A TABLE IS PRINTED. ONLY P(I) IS READ. THIS IS TAKEN
R   AS THE UPPER LIMIT FOR ALL OF THE N(I) AND P(I). ALL
R   FOLLOWING NUMBERS ON THE CARD ARE IGNORED.
R
R
R
R RESTRICTIONS ON VARIABLES
R   NBR MUST BE AN INTEGER BETWEEN 3 AND 9
R   N(I) AND P(I) ARE INTEGERS
R   A1, RAT, AND THE A(I) MUST BE FLOATING POINT.
R   (IF THERE IS DIFFICULTY IN OBTAINING A SOLUTION,
R   THE A(I) SHOULD BE RESCALED TO A VALUE CLOSER TO 1.0)
R
R
R
R SAMPLE INPUT CARDS ARE
R
R   VAR 3           8 8 7           8 8 8
R1.    .1          .01
R
R   TABLREG 3 .1 1.      8

```

MAD Program and Data (continued)

```

R
R
R
R THE OUTPUT OF THE PROGRAM CONSISTS OF A LISTING OF THE INPUT
R VARIABLES PLUS THE VARIABLES X AND P. THE VARIABLE X IS THE
R MOST PROBABLE NUMBER OF BACTERIA PER ML. OF THE
R SOLUTION. P IS THE PERCENTAGE PROBABILITY THAT THE
R COMBINATION OF THE N(I) AND P(I) WOULD HAVE RESULTED IF X
R WERE THE TRUE NUMBER OF BACTERIA.
R
R
R IT SHOULD BE NOTED THAT ALL CALCULATIONS THAT YIELD A P
R LESS THAN .01 ARE SUPPRESSED WHEN A TABLE IS PRINTED.
R
R
R
R DIMENSION A(10),P(10),PI(10),N(10),NI(10)
R INTEGER CHECK, HOW, NBR,PI,I,NI,COUNT,RET,KEY
R
R BOOLEAN ZEROS,TABL
R VECTOR VALUES BCD(3)=$36$, $48$, $60$, $72$, $84$, $96$, $108$
R EXECUTE SETERR.(START)
R
R RREAD IN CARD TYPE I
START
R
R READ FORMAT IN, CHECK, HOW, NBR,RAT,A1,PI(1),...,PI(9),NI(1)
R 1,...,NI(9)
R VECTOR VALUES IN=$C4,C3,I3,F4.3,F8.6,18I2*$
R
R RCHECK NBR
R
R WHENEVER NBR.L.3.OR.NBR.G.9
R PRINT COMMENTS$4NBR NOT IN RANGES
R TRANSFER TO START
R
R RREGULAR DILUTION RATIO
R
R OR WHENEVER HOW.E.$REGS
R THROUGH OMEGA, FOR I=1,1,I.G.NBR
OMEGA
R A(I)=A1*RAT.P.(I-1)
R PRINT FORMAT REG,NBR,A1,RAT
R VECTOR VALUES REG=$20H4NUMBER OF GROUPS IS,I3,9H. A(1) IS,
R 1F8.4,2H ./18HODILUTION RATIO IS,F6.4,2H .*$
R
R RVARIABLE DILUTIONS
R
R OR WHENEVER HOW.E.$VARS
R
R RREAD IN CARD TYPE II
R
R READ FORMAT IN1,A(1),...,A(NBR)
R VECTOR VALUES IN1=$9F8.2*$
R PRINT FORMAT VAR,A(1),...,A(NBR)
R VECTOR VALUES VAR=$2H4A,S3,9E12.3*$
R OTHERWISE
R PRINT COMMENTS$4INPUT ERRORS
R EXECUTE SYSTEM.
R END OF CONDITIONAL
R
R RPOSITION OUTPUT FORMAT
R
R SLIDE(2)=BCD(NBR)
R OUT(2)=BCD(NBR)
R PRINT FORMAT SLIDE
R VECTOR VALUES SLIDE=$1H /1H+,S9,S ,1HX,S6,5HP 0/0*$
R
R RSINGLE CALCULATION
R
R WHENEVER CHECK.E.$ $
R TABL=0B
R RET=0

```

Method for Determining Bacterial Populations

MAD Program and Data (continued)

```

R
R FLOAT N(I) AND P(I)
R
PSI THROUGH PSI, FOR I=1,1,I.G.NBR
N(I)=NI(I)
P(I)=PI(I)
PRINT FORMAT NOUT,NI(1)....NI(NBR)

VECTOR VALUES NOUT=$2H N,S3,9(I7,S5)*$
TRANSFER TO GRIND
R
RTABLE
R
OR WHENEVER CHECK.E.$STABLS
TABL=1B
RET=1
P = PI(1)
EXECUTE SPRAY.(P,N(1)....N(NBR),P(1)....P(NBR))
PRINT FORMAT NOUT,PI(1)
PRINT COMMENTS$OS
R
RCALCULATION OF P(I)
R
START(1) THROUGH CHI, FOR KEY=NBR,-1,KEY.L.1
P(KEY)=P(KEY)-1.
R
RALL P(I) = 0, I.E. FINISHED
R
WHENEVER ZEROS.(P,NBR).AND.P(2).E.0...AND.P(1).E.0..
1TRANSFER TO START
WHENEVER P(KEY).GE.0.
R
R FIX P(I) FOR PRINTING
R
PHI THROUGH PHI, FOR I=1,1,I.G.NBR
PI(I)=P(I)
TRANSFER TO GRIND
END OF CONDITIONAL
CHI P(KEY)=P
OTHERWISE
PRINT COMMENTS$4INPUT ERRORS
EXECUTE SYSTEM.
END OF CONDITIONAL
R
RCALCULATION OF INITIAL GUESS FOR X
R
GRIND THROUGH UPS, FOR I=1,1,I.G.NBR
WHENEVER P(I).G.0...AND.P(I).L.N(I)
X=(A(NBR/2+1)*P(I))/(N(I)*A(I))
TRANSFER TO TAU
UPS END OF CONDITIONAL
X=5.
TAU DELX=1.
R
RNEWTON-RAPHSON SOLUTION FOR X
R
THROUGH ALPHA, FOR COUNT=1,1,COUNT.G.50.OR..ABS.DELX.L..0005
G=0.
GP=0.
THROUGH BETA, FOR I=1,1,I.G.NBR
EXPON=1.-EXP.(-A(I)*X)
G=G+A(I)*(P(I)/EXPON-N(I))
GP=GP-A(I)*A(I)*P(I)*(1.-EXPON)/(EXPON*EXPON)
DELX=G/GP
X=X-DELX
R
R(SOLUTION UNSTABLE FOR X LESS THAN 0)

```

MAD Program and Data (Continued)

```

ALPHA      R
           R WHENEVER X.L.0.,X=.0000001
           R
           R ITERATION LIMIT EXCEEDED WITHOUT REDUCING DELX TO LESS THAN
           R .0005
           R
           R WHENEVER COUNT.G.50
           R PRINT FORMAT CMPTBL,PI(1)....PI(NBR)
           R PRINT FORMAT XCED
           R VECTOR VALUES XCED=$1H ,S113,13HITER EXCEEDED*$
           R TRANSFER TO START(RET)
           R END OF CONDITIONAL
           R
           R COMPUTATION OF P
           R
           R LOGEP=0.
           R THROUGH GAMMA, FOR I=1,1,I.G.NBR
           R LOGEP=LOGEP+ELOG.(BINOM.(N(I),P(I))-A(I)*(N(I)-P(I))*X
           R 1+P(I)*ELOG.(1.-EXP.(-A(I)*X))
           R PPER=100.*EXP.(LOGEP)
           R WHENEVER PPER.L..01.AND.TABL,TRANSFER TO START(RET)
           R PRINT FORMAT CMPTBL,PI(1)....PI(NBR)
           R VECTOR VALUES CMPTBL=$1H /2H+P,S3,9(I7,S5)*$
           R
           R PRINT ANSWERS
           R
           R PRINT FORMAT OUT,X,PPER
           R VECTOR VALUES OUT=$1H ,S4.S           ,E10.3,F7.2*$
           R TRANSFER TO START(RET)
           R END OF PROGRAM
$COMPILE MAD,PRINT OBJECT,PUNCH OBJECT           ZEROS001
           R
           R PROGRAM FOR CHECKING TO SEE IF P(2),....,P(NBR) ARE ZERO
           R
           R
           R
           R EXTERNAL FUNCTION (P,NBR)
           R INTEGER I,NBR
           R ENTRY TO ZEROS.
           R THROUGH ALPHA, FOR I=2,1,I.G.NBR
           R WHENEVER P(I).G.0.,FUNCTION RETURN 0B
           R FUNCTION RETURN 1B
           R END OF FUNCTION
           R END OF PROGRAM
$COMPILE MAD,PRINT OBJECT,PUNCH OBJECT           BINOM001
           R
           R PROGRAM FOR COMPUTING THE BINOMIAL COEFFICIENT
           R
           R
           R
           R EXTERNAL FUNCTION (UP,DWN)
           R ENTRY TO BINOM.
           R FACT=1.
           R THROUGH ALPHA, FOR TOP=DWN+1.,1.,TOP.G.UP
           R FACT=FACT*TOP
           R THROUGH BETA, FOR BOT=2.,1.,BOT.G.(UP-DWN)
           R FACT=FACT/BOT
           R FUNCTION RETURN FACT
           R END OF FUNCTION
           R END OF PROGRAM
$DATA
  VAR 3           8 8 7           8 8 8
  1. .1           .01
  TABLREG 3 .1 1. 3

```

Method for Determining Bacterial Populations

Computer Output

A	.100E 01	1.000E-01	1.000E-02		
N	8	8	8	X	P 0/0
P	8	8	7	.208E 03	39.27

NUMBER OF GROUPS IS 3. A(1) IS 1.0000 .
 DILUTION RATIO IS .1000 .

N	3			X	P 0/0
P	3	3	2	.110E 03	44.44
P	3	3	1	.462E 02	42.77
P	3	3	0	.240E 02	36.59
P	3	2	3	.292E 02	.24
P	3	2	2	.215E 02	2.47
P	3	2	1	.149E 02	12.51
P	3	2	0	.933E 01	32.82
P	3	1	3	.159E 02	.03
P	3	1	2	.115E 02	.65
P	3	1	1	.749E 01	6.58
P	3	1	0	.427E 01	37.43
P	3	0	2	.636E 01	.16
P	3	0	1	.385E 01	3.10
P	2	3	1	.360E 01	.02
P	2	3	0	.286E 01	.22
P	2	2	1	.276E 01	.17
P	2	2	0	.211E 01	2.32
P	2	1	2	.268E 01	.02
P	2	1	1	.205E 01	.63
P	2	1	0	.147E 01	11.96
P	2	0	2	.199E 01	.02
P	2	0	1	.143E 01	1.12
P	2	0	0	.918E 00	31.93
P	1	3	0	.157E 01	.03
P	1	2	1	.154E 01	.03

Computer Output (continued)

P	1	2	0	•114E 01	•63
P	1	1	1	•112E 01	•18
P	1	1	0	•736E 00	6.45
P	1	0	1	•723E 00	•62
P	1	0	0	•357E 00	39.20
P	0	2	0	•619E 00	•16
P	0	1	1	•611E 00	•05
P	0	1	0	•305E 00	3.36

Discussion of Results

The output shown above was designed to be illustrative rather than useful. In practice more than three tubes would be used in each dilution group.

Compiling time for the program was 1.1 minutes. Computing the results shown above took 0.5 minutes.

Critique

This problem is very appropriate for student solution. It gives the student an opportunity to become familiar with numerical solution of complex equations as well as an opportunity to become familiar with the use of the MPN equation. Solution of the MPN equation by hand is extremely tedious, so that the tables prepared by the instructor's program will be useful in later work.

References

1. Halvorson, H. O. and Ziegler, N. R. J. Bact., 25, 101-121 (1933).
2. Norman, R. L. and Kempe, L. L., Biotechnology and Bioengineering, II, No. 2, 157-163 (1960).

Example Problem No. 80
 ITERATION OF SUM OF PRINCIPAL STRESSES
 AT INTERIOR POINTS OF PLANE RINGS

by

William Weaver, Jr.

Department of Civil Engineering

Stanford University

Course: Experimental Stress Analysis Credit Hours: 3 Level: Graduate

Statement of the Problem

Write a computer program which will evaluate the sum of principal stresses at interior points of plane rings for which the boundary stresses are known from photo-elastic analysis. The recommended procedure is to iterate the Laplace difference equation applied to an x-y grid network of approximately square elements obtained by the conformal transformation of the R-theta polar network using the relationships:

$$x = \ln (R)$$

$$y = \text{theta}$$

The program should be flexible enough to accommodate a whole ring or any segment of a ring for which the stresses on the radial boundaries are known to be equal by symmetry. Divide the central angle of the polar network into 5° increments. Make the elements of the transformed rectangular grid as nearly square as possible. The number of radial increments will depend upon the ratio of the outer and the inner radii of the ring.

Limit the number of storage locations required by the grid to about 3,000. Iterate the sum of principal stresses to within 3 decimal point accuracy. Limit the number of iteration cycles to 50.

Solution

The Laplace difference equation for the sum of principal stresses at a lattice point 0 and its four neighboring points 1, 2, 3, 4 (see Fig. 1) may be written:

$$\frac{\frac{S_3 - S_0}{\Delta x_3} - \frac{S_0 - S_1}{\Delta x_1}}{\frac{\Delta x_1 + \Delta x_3}{2}} + \frac{\frac{S_2 - S_0}{\Delta y_2} - \frac{S_0 - S_4}{\Delta y_4}}{\frac{\Delta y_2 + \Delta y_4}{2}} = 0 \quad (1)$$

For the present problem let $\Delta x_1 = \Delta x_3 = p\Delta y_2 = p\Delta y_4$

Solving for S_0 in equation (1),

$$S_0 = \frac{1}{2(p^2+1)} (S_1 + S_3) + \frac{p^2}{2(p^2+1)} (S_2 + S_4) \quad (2)$$

in which, S = Sum of principal stresses at a point

p = Ratio of x-intervals to y-intervals.

Equation (2) may be used for the iterative evaluation of S at the lattice points of the transformed grid.

In the photoelastic analysis of plane rings the principal stresses at the inner and outer edges are determined experimentally at even intervals of the central angle. A practical minimum interval is 5° , or $\pi/36$ radians.

By conformal transformation:

$$x = \ln (R); \quad y = \theta$$

Let $\Delta y = \Delta \theta = \pi/36$. Define M as the number of angular increments, i.e., increments in the y-direction. M is given by

$$M = \frac{\text{total angle of ring}}{\Delta \theta}$$

Thus, for a full ring and $\Delta \theta = 5^\circ$, $M = 360^\circ/5^\circ = 72$. For a quarter ring $M = 90^\circ/5^\circ = 18$.

In order to obtain an exactly square grid, the number of subdivisions in the radial direction must be:

$$N_1 = \frac{x_N - x_0}{\pi/36} = \frac{\ln(R_N) - \ln(R_0)}{\pi/36} = \frac{36}{\pi} \ln(R_N/R_0)$$

in which,

R_0 = Inner radius of ring.

R_N = Outer radius of ring.

x_0 = Transformed inner radius.

x_N = Transformed outer radius.

In general, N_1 will not be an integer, but an approximately square grid may be obtained by computing the actual number of intervals N used in the radial direction as the next higher integer value of N_1 .

Iteration of Sum of Principal Stresses

$$N = \left[\frac{36}{\pi} \ln(R_N/R_0) \right] \text{ Rounded up} \quad (3)$$

The dimensionless ratio, p, may be computed from:

$$p = \frac{\Delta x}{\Delta y} = \frac{N_1}{N} = \frac{36}{\pi N} \ln(R_N/R_0) \quad (4)$$

Radii of interior network circles are determined by the relationship:

$$R_j = e^{(x_j)} = e^{(x_0 + j\Delta x)} = R_0 (e^{\ln(R_N/R_0)})^{j/N} = R_0 (R_N/R_0)^{j/N} \quad (5)$$

in which, R_j = Radius of jth network circle.

j = Radial index.

Assuming that a storage space of about 3,000 locations is available for the $M \times (N+1)$ matrix of lattice points required for the solution,

$$M \times (N+1) \leq 3000. \quad (6)$$

For a full ring the maximum value of M is $360^\circ/5^\circ = 72$, and the value of N is given by substitution of equation (3) into equation (6).

$$72 \left\{ \left[\frac{36}{\pi} \ln(R_N/R_0) \right] \text{ Rounded up} + 1 \right\} \leq 3000 \quad (7)$$

Equation (7) yields the maximum values of N and the corresponding ratio of radii.

$$(N)_{\max} = 41; \quad (R_N/R_0)_{\max} \approx 35.0$$

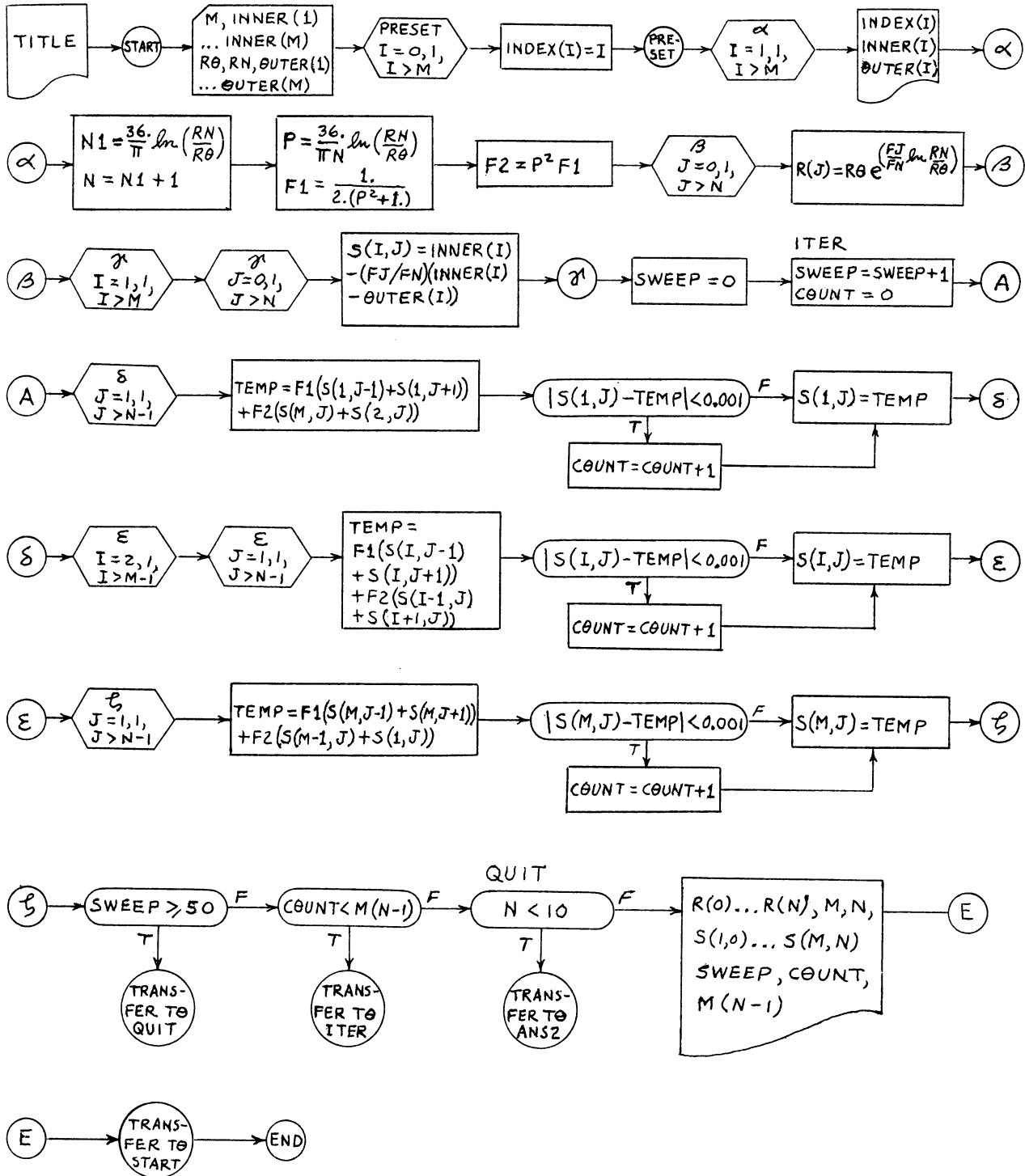
The minimum non-trivial values for these parameters correspond to a network of only two intervals in the radial direction.

$$(N)_{\min} = 2; \quad (R_N/R_0)_{\min} \approx 1.19$$

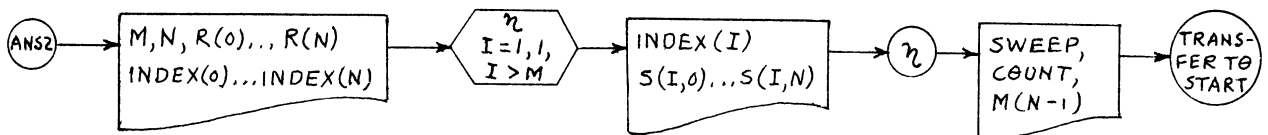
Figures 2 and 3 show the polar and transformed lattic networks for a quarter ring having $R_N/R_0 = 4$, $M = 18$, and $N = 16$. This configuration, accompanied by a set of hypothetical boundary stresses, constitutes the first of two trial problems submitted to the computer. The second problem, for which no figure is given, consists of a quarter ring having $R_N/R_0 = 4/3$, $M = 18$, and $N = 4$.

The first row of the transformed grid may be omitted because the stresses must be the same as in the last row.

Flow Diagram



Special Output Form when $N < 10$:



- INNER(I) = Boundary stress at inner edge.
- OUTER(I) = Boundary stress at outer edge.
- F1 = Weighting factor from first term in equation (2).
- F2 = Weighting factor from second term in equation (2).
- FJ = J expressed as a floating point number.
- FN = N expressed as a floating point number.

Iteration of Sum of Principal Stresses

MAD Program

```

W. WEAVER          D046N          005   005   000
W. WEAVER          D046N          005   005   000
SCOMPILE MAD, EXECUTE
R
RTHIS PROGRAM IS INTENDED FOR USE IN EVALUATING THE
R  SUM OF PRINCIPAL STRESSES AT INTERIOR POINTS OF
R  PLANE RINGS ANALYZED EXPERIMENTALLY BY THE
R  METHOD OF PHOTOELASTICITY. THE PROCEDURE CONSISTS OF
R  THE ITERATION OF THE LAPLACE DIFFERENCE EQUATION
R  APPLIED TO AN X-Y GRID NETWORK OF APPROXIMATELY
R  SQUARE ELEMENTS OBTAINED BY THE CONFORMAL
R  TRANSFORMATION OF THE R-THETA NETWORK USING
R  THE RELATIONSHIPS
R    X=LN(R)
R    Y=THETA
R  THE PROGRAM IS APPLICABLE FOR A WHOLE RING OR
R  ANY SEGMENT THEREOF, THE RADIAL BOUNDARIES OF
R  WHICH LIE MIDWAY BETWEEN ADJACENT RADII OF SYMMETRY.
R  THE CENTRAL ANGLE IS DIVIDED INTO FIVE DEGREE
R  INCREMENTS, RESULTING IN M=72 INCREMENTS FOR A
R  FULL RING, M=36 INCREMENTS FOR A HALF RING, M=18
R  FOR A THIRD RING, ETC. THE NUMBER OF RADIAL INCRE-
R  MENTS DEPENDS UPON THE RATIO OF THE OUTER AND THE
R  INNER RADII OF THE RING. IN ORDER TO LIMIT THE NUMBER
R  OF STORAGE LOCATIONS REQUIRED BY THE GRID TO ABOUT
R  3,000, THE MAXIMUM RATIO OF RADII PERMITTED IS 35.
R  THE MINIMUM NON-TRIVIAL RATIO IS 1.19.
R  THE CALCULATIONS ARE CARRIED TO 3 DECIMAL PLACE ACCURACY
R  FOR 50 ITERATION CYCLES, WHICHEVER OCCURS FIRST.
R
R  PRINT FORMAT TITLE
R  VECTOR VALUES TITLE=$1H1,29HITERATION OF SUM OF PRINCIPAL/
R  11H0,42HSTRESSES AT INTERIOR POINTS OF PLANE RINGS*$
R
R  READ AND ECHO CHECK DATA
R
START  READ FORMAT DATA1,M,INNER(1)...INNER(M)
        READ FORMAT DATA2,RO,RN,OUTER(1)...OUTER(M)
        VECTOR VALUES DATA1=$12,S18,5F10.3/(7F10.3)*$
        VECTOR VALUES DATA2=$2F10.4,5F10.3/(7F10.3)*$
        THROUGH PRESET, FOR I=0,1,I.G.M
PRESET  INDEX(I)=I
        PRINT FORMAT DDATA1
        PUNCH FORMAT DDATA1
        VECTOR VALUES DDATA1=$1H0,S5,4HDATA,S15,5HINDEX,S5,5HINNER,S5
        1,5HOUTER*$
        THROUGH ALPHA, FOR I=1,1,I.G.M
ALPHA  PRINT FORMAT DDATA2,INDEX(I),INNER(I),OUTER(I)
        THROUGH ALPHA1, FOR I=1,1,I.G.M
ALPHA1 PUNCH FORMAT DDATA2,INDEX(I),INNER(I),OUTER(I)
        VECTOR VALUES DDATA2=$1H ,S27,I2,2F10.3*$
R
R  COMPUTE ITERATION FACTORS
R
        N1=(36./3.14159)*ELOG.(RN/RO)
        N=N1+1
        P=(36.*ELOG.(RN/RO))/(3.14159*N)
        F1=1./(2.*(P*P+1.))
        F2=P*P*F1
R
R  COMPUTE INTERMEDIATE RADII
R  (USING FLOATING POINT NF=N AND JF=J)
R
        NF=N
        THROUGH BETA, FOR J=0,1,J.G.N
        JF=J
BETA   R(J)=RO*EXP.((JF/NF)*ELOG.(RN/RO))
R
R  RESET STARTING VALUES FOR S USING LINEAR APPROXIMATION
R  TO ESTIMATE INTERIOR VALUES BASED ON BOUNDARY VALUES.
R  (USING FLOATING POINT NF=N AND JF=J)
R
        DIMS(2)=N+1
        THROUGH GAMMA, FOR I=1,1,I.G.M
        THROUGH GAMMA, FOR J=0,1,J.G.N
        JF=J

```

MAD Program, Continued

```

GAMMA      S(I,J)=INNER(I)-(JF/NF)*(INNER(I)-OUTER(I))
           R
           R INITIALIZE COUNTERS FOR NO. OF CYCLES AND ACCURACY
           R
           SWEEP=0
           R
           R ITERATE FIRST ROW, USING VALUES FROM FIRST, LAST, AND
           R SECOND ROWS.
ITER        SWEEP=SWEEP+1
           COUNT=0
           THROUGH DELTA, FOR J=1,1,J.G.(N-1)
           TEMP=F1*(S(1,J-1)+S(1,J+1))+F2*(S(M,J)+S(2,J))
           WHENEVER.ABS.(S(1,J)-TEMP).L.0.001,COUNT=COUNT+1
DELTA      S(1,J)=TEMP
           R
           R ITERATE INTERMEDIATE ROWS
           R
           THROUGH EPSIL, FOR I=2,1,I.G.(M-1)
           THROUGH EPSIL, FOR J=1,1,J.G.(N-1)
           TEMP=F1*(S(I,J-1)+S(I,J+1))+F2*(S(I-1,J)+S(I+1,J))
           WHENEVER.ABS.(S(I,J)-TEMP).L.0.001,COUNT=COUNT+1
EPSIL      S(I,J)=TEMP
           R
           R ITERATE LAST ROW, USING VALUES FROM FIRST, LAST, AND NEXT-
           R TO-LAST ROWS.
           THROUGH ZETA, FOR J=1,1,J.G.(N-1)
           TEMP=F1*(S(M,J-1)+S(M,J+1))+F2*(S(M-1,J)+S(1,J))
           WHENEVER.ABS.(S(M,J)-TEMP).L.0.001,COUNT=COUNT+1
ZETA      S(M,J)=TEMP
           R
           R TEST FOR NO. OF CYCLES AND ACCURACY
           R
           WHENEVER SWEEP.GE.50,TRANSFER TO QUIT
           WHENEVER COUNT.L.(M*(N-1)),TRANSFER TO ITER
           WHENEVER N.L.10
           TRANSFER TO ANS2
           OTHERWISE
           R
           R OUTPUT FORM FOR N GREATER THAN OR EQUAL TO 10.
R          PRINT FORMAT FORM1,R(0)...R(N)
           VECTOR VALUES FORM1=$1H0,11HVALUES OF R/(7F10.4)*$
           PRINT FORMAT FORM1A,M,N
           VECTOR VALUES FORM1A=$1H0,15HITERATED VALUES,S10,2HM=,I2,S10,
12HN=,I2//*$
           PRINT FORMAT FORM1B,S(1,0)...S(M,N)
           VECTOR VALUES FORM1B=$(7F10.3)*$
           PRINT FORMAT FORM1C,SWEEP,COUNT,M*(N-1)
           VECTOR VALUES FORM1C=$1H0,14HNO. OF CYCLES=,S11,I2/1H0,23HNO.
1 OF ACCURATE VALUES=,I4/1H0,23HNO. OF ITERATED VALUES=,I4*$
           END OF CONDITIONAL
           TRANSFER TO START
R          RSPECIAL OUTPUT FORM FOR N LESS THAN 10.
R          PRINT FORMAT FORM2,M,N,R(0)...R(N)
           VECTOR VALUES FORM2=$1H0,15HITERATED VALUES,S10,2HM=,I2,S10,
12HN=,I2/1H0,S3,6HVALUES /1H0,S5,2HOF,S7,2HR=,S3,10F10.4*$
           PRINT FORMAT FORM2A,INDEX(0)...INDEX(N)
           VECTOR VALUES FORM2A=$1H0,S4,5HTHETA,S3,7HINDICES,
1S6,I1,9I10*$
ETA        THROUGH ETA, FOR I=1,1,I.G.M
           PRINT FORMAT FORM2B,5*INDEX(I),INDEX(I),S(I,0)...S(I,N)
           VECTOR VALUES FORM2B=$S5,I3,S7,I2,S3,10F10.3*$
           PRINT FORMAT FORM2C,SWEEP,COUNT,M*(N-1)
           VECTOR VALUES FORM2C=$1H0,14HNO. OF CYCLES=,S11,I2/1H0,
123HNO. OF ACCURATE VALUES=,I4/1H0,23HNO. OF ITERATED VALUES=
2I4*$
           TRANSFER TO START
           INTEGER M,N,I,J,SWEEP,COUNT,INDEX
           DIMENSION INNER(72),OUTER(72),INDEX(72),R(42)
           DIMENSION S(3000,DIMS)
           VECTOR VALUES DIMS=2,1,42
           END OF PROGRAM

```

Iteration of Sum of Principal Stresses

Data

\$DATA							
18			-5000	-3222	-1800	-457	613
	1711	2641	3416	3800	4000	3800	3416
	2641	1711	613	-457	-1800	-3222	
	15000	60000	3000	7	-2697	-5002	-6896
	-8005	-8985	-9625	-9900	-10000	-9900	-9625
	-8985	-8005	-6896	-5002	-2697	7	
18			-5000	-3222	-1800	-457	613
	1711	2641	3416	3800	4000	3800	3416
	2641	1711	613	-457	-1800	-3222	
	45000	60000	3000	7	-2697	-5002	-6896
	-8005	-8985	-9625	-9900	-10000	-9900	-9625
	-8985	-8005	-6896	-5002	-2697	7	

Computer Output

DATA	INDEX	INNER	OUTER			
	1	-5.000	3.000			
	2	-3.222	0.007			
	3	-1.800	-2.697			
	4	-0.457	-5.002			
	5	0.613	-6.896			
	6	1.711	-8.005			
	7	2.641	-8.985			
	8	3.416	-9.625			
	9	3.800	-9.900			
	10	4.000	-10.000			
	11	3.800	-9.900			
	12	3.416	-9.625			
	13	2.641	-8.985			
	14	1.711	-8.005			
	15	0.613	-6.896			
	16	-0.457	-5.002			
	17	-1.800	-2.697			
	18	-3.222	0.007			
VALUES OF R						
1.5000	1.6358	1.7838	1.9453	2.1213	2.3133	2.5227
2.7510	3.0000	3.2715	3.5676	3.8905	4.2426	4.6266
5.0454	5.5020	6.0000				
ITERATED VALUES		M=18	N=16			
-5.000	-3.075	-2.290	-1.982	-1.920	-1.999	-2.160
-2.365	-2.584	-2.790	-2.953	-3.034	-2.977	-2.688
-1.988	-0.471	3.000	-3.222	-2.498	-2.049	-1.860
-1.852	-1.961	-2.142	-2.362	-2.595	-2.816	-2.998
-3.108	-3.098	-2.899	-2.404	-1.464	0.007	-1.800
-1.642	-1.545	-1.556	-1.666	-1.851	-2.085	-2.348
-2.620	-2.883	-3.119	-3.304	-3.412	-3.411	-3.271
-2.994	-2.697	-0.457	-0.725	-0.933	-1.153	-1.406
-1.691	-2.001	-2.325	-2.654	-2.979	-3.290	-3.578
-3.836	-4.063	-4.278	-4.543	-5.002	0.613	0.132
-0.307	-0.717	-1.114	-1.507	-1.901	-2.297	-2.693
-3.089	-3.484	-3.882	-4.290	-4.727	-5.234	-5.897
-6.896	1.711	0.945	0.288	-0.293	-0.824	-1.321
-1.799	-2.266	-2.730	-3.197	-3.674	-4.174	-4.713
-5.318	-6.031	-6.909	-8.005	2.641	1.649	0.807
0.079	-0.566	-1.154	-1.706	-2.238	-2.762	-3.291
-3.840	-4.425	-5.068	-5.799	-6.659	-7.699	-8.985
3.416	2.202	1.210	0.370	-0.363	-1.022	-1.632
-2.213	-2.785	-3.363	-3.966	-4.614	-5.332	-6.149
-7.103	-8.241	-9.625	3.800	2.529	1.461	0.554
-0.233	-0.936	-1.583	-2.196	-2.798	-3.407	-4.044
-4.730	-5.491	-6.357	-7.360	-8.533	-9.900	4.000
2.653	1.549	0.618	-0.188	-0.905	-1.564	-2.188
-2.800	-3.420	-4.068	-4.767	-5.544	-6.425	-7.443
-8.628	-10.000	3.800	2.530	1.463	0.557	-0.230
-0.931	-1.577	-2.190	-2.791	-3.401	-4.038	-4.724
-5.487	-6.354	-7.358	-8.532	-9.900	3.416	2.204
1.214	0.376	-0.355	-1.013	-1.621	-2.202	-2.773
-3.351	-3.955	-4.604	-5.323	-6.142	-7.098	-8.239
-9.625	2.641	1.652	0.812	0.087	-0.556	-1.142
-1.692	-2.222	-2.746	-3.275	-3.825	-4.411	-5.056
-5.790	-6.653	-7.696	-8.985	1.711	0.948	0.294
-0.285	-0.812	-1.308	-1.783	-2.249	-2.712	-3.178

Example Problem No. 80

Computer Output, Continued

-3.657	-4.158	-4.700	-5.308	-6.024	-6.905	-8.005
0.613	0.135	-0.302	-0.708	-1.102	-1.494	-1.886
-2.280	-2.675	-3.071	-3.467	-3.866	-4.276	-4.716
-5.227	-5.893	-6.896	-0.457	-0.722	-0.928	-1.146
-1.397	-1.680	-1.987	-2.310	-2.639	-2.964	-3.275
-3.564	-3.824	-4.054	-4.272	-4.540	-5.002	-1.800
-1.640	-1.542	-1.551	-1.660	-1.843	-2.076	-2.337
-2.609	-2.872	-3.108	-3.294	-3.403	-3.404	-3.267
-2.991	-2.697	-3.222	-2.497	-2.047	-1.858	-1.849
-1.957	-2.137	-2.357	-2.590	-2.811	-2.993	-3.103
-3.094	-2.896	-2.402	-1.462	0.007		

NO. OF CYCLES= 50

NO. OF ACCURATE VALUES= 211

NO. OF ITERATED VALUES= 270

DATA	INDEX	INNER	OUTER
	1	-5.000	3.000
	2	-3.222	0.007
	3	-1.800	-2.697
	4	-0.457	-5.002
	5	0.613	-6.896
	6	1.711	-8.005
	7	2.641	-8.985
	8	3.416	-9.625
	9	3.800	-9.900
	10	4.000	-10.000
	11	3.800	-9.900
	12	3.416	-9.625
	13	2.641	-8.985
	14	1.711	-8.005
	15	0.613	-6.896
	16	-0.457	-5.002
	17	-1.800	-2.697
	18	-3.222	0.007

ITERATED VALUES M=18 N= 4

VALUES

OF	R=	4.5000	4.8356	5.1962	5.5836	6.0000
THETA	INDICES	0	1	2	3	4
5	1	-5.000	-2.999	-1.644	-0.018	3.000
10	2	-3.222	-2.524	-1.845	-1.042	0.007
15	3	-1.800	-2.023	-2.226	-2.430	-2.697
20	4	-0.457	-1.551	-2.607	-3.723	-5.002
25	5	0.613	-1.132	-2.901	-4.775	-6.896
30	6	1.711	-0.675	-3.039	-5.463	-8.005
35	7	2.641	-0.248	-3.092	-5.974	-8.985
40	8	3.416	0.114	-3.086	-6.297	-9.625
45	9	3.800	0.326	-3.064	-6.447	-9.900
50	10	4.000	0.414	-3.050	-6.493	-10.000
55	11	3.800	0.326	-3.064	-6.447	-9.900
60	12	3.416	0.113	-3.086	-6.297	-9.625
65	13	2.641	-0.248	-3.092	-5.974	-8.985
70	14	1.711	-0.674	-3.039	-5.462	-8.005
75	15	0.613	-1.130	-2.899	-4.774	-6.896
80	16	-0.457	-1.549	-2.606	-3.722	-5.002
85	17	-1.800	-2.022	-2.225	-2.429	-2.697
90	18	-3.222	-2.524	-1.844	-1.042	0.007

NO. OF CYCLES= 11

NO. OF ACCURATE VALUES= 54

NO. OF ITERATED VALUES= 54

Iteration of Sum of Principal Stresses

Discussion of Results

The first problem submitted to the computer involved 270 iterations in each cycle. After 50 cycles, 211, (or about 78%) of the values were accurate to three decimal places. Because N was large, values of R and S were printed sequentially according to the general format labeled QUIT.

The second problem involved 54 iterations in each cycle, and all values were accurate to three decimal places after 11 cycles. Because N was, in this case, less than 10, values of R and S were printed in rows and columns corresponding to Figure 3, according to the special format labeled ANS2.

Critique

This problem is very well suited for digital computation. The analysis of plane rings is a recurring problem in photoelastic experiments, and the iteration of the sum of principal stresses at interior points is an essential, though tedious, step in the total analysis. The fineness of the lattice network is not significantly limited by the storage space of a computer of reasonable size. A more important factor in limiting fineness is the practical difficulty of measuring boundary stresses accurately at the inner edges of plane rings for small increments of the central angle. Increments of 5° are therefore used as a practical minimum in establishing the lattice network.

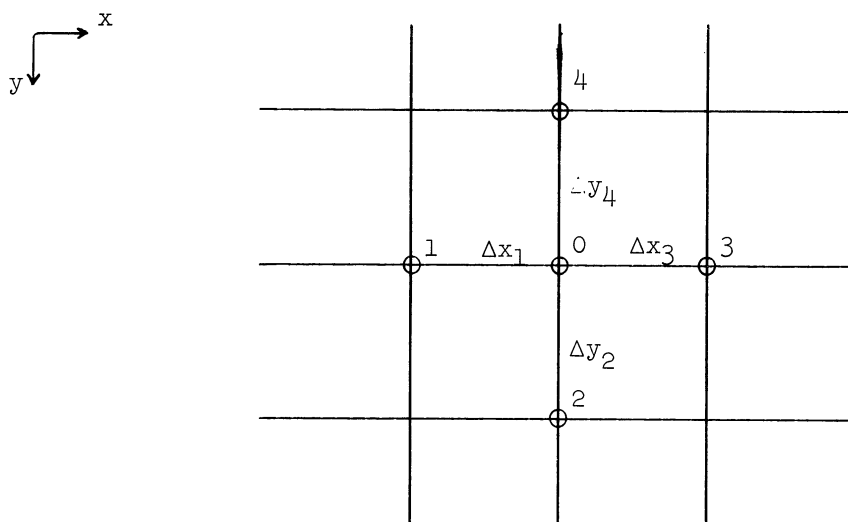


Figure 1.

Lattice Point 0 and its Four Neighboring Points 1,2,3,4

Example Problem No. 80

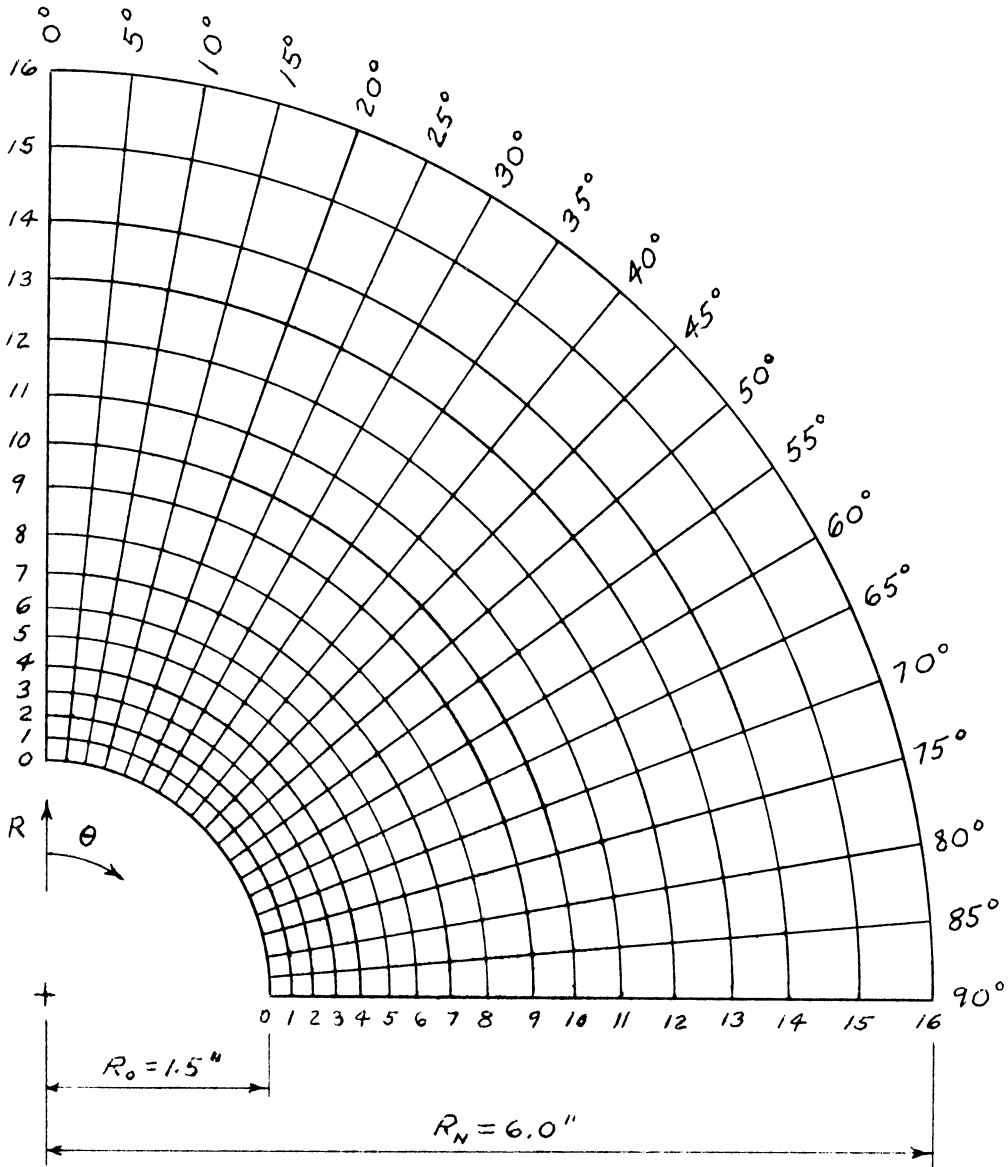


Figure 2.

Quarter Ring: $R_N/R_0 = 4$

$M = 18$ $N = 16$

Iteration of Sum of Principal Stresses

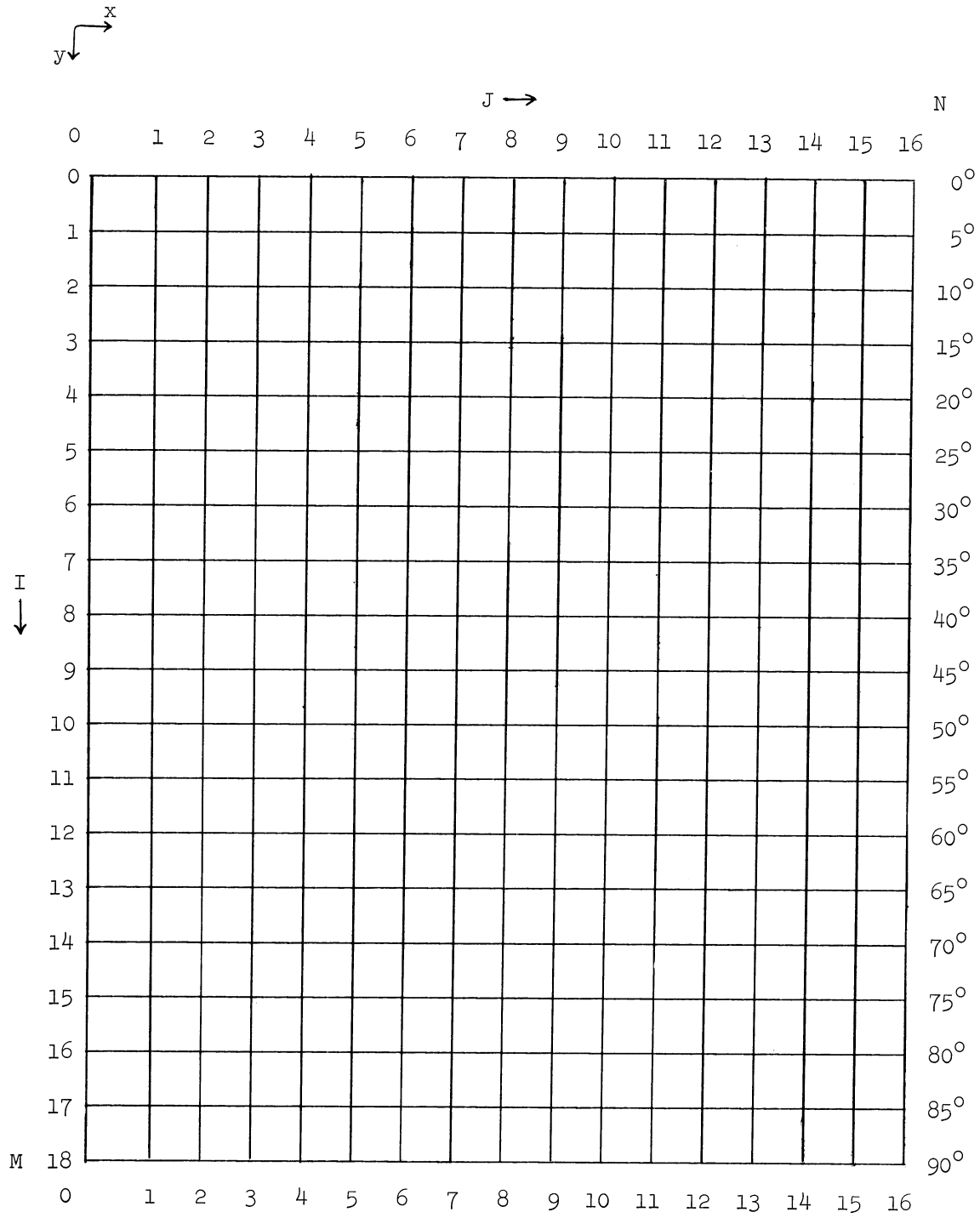


Figure 3.

Transformed Quarter Ring

$$R_N/R_0 = 4 \quad M = 18 \quad N = 16$$

Example Problem No. 81

ELASTIC BUCKLING LOAD FOR COLUMNS OF NON-UNIFORM CROSS SECTION

by

K. H. Chu

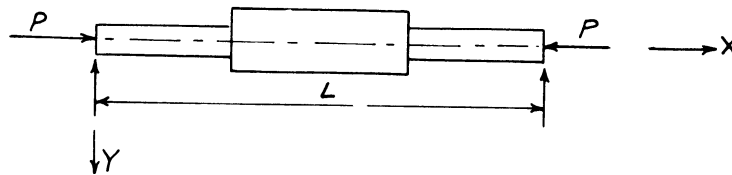
Civil Engineering Department

Illinois Institute of Technology

Course: Advanced Structural Theory and Design Credit hours: 3 Level: Seniors
or First Year
Graduates

Statement of the Problem

The differential equation for elastic buckling of a column as shown below is as follows:



$$\frac{d^2 Y}{dX^2} = - \frac{M}{E \cdot IA} = - \frac{P \cdot Y}{E \cdot IA} = - \frac{P \cdot Y}{E \cdot IO \cdot I} \quad (1)$$

where X is the coordinate along the longitudinal axis of the column, Y is the deflection of the column, M is the bending moment, P is the axial load, E is the modulus of elasticity, IA is the moment of inertia of the cross-section about the Z axis normal to the XY plane at the point under consideration, IO is the same at a certain reference point and $I = IA/IO$.

The buckling load, PCR, is defined as the smallest value of P which will satisfy the above equation with appropriate boundary conditions. The value of PCR is to be determined by an iterative procedure using a digital computer.

Solution

Equation (1) is analogous to the following equation

$$\frac{d^2 M'}{dX^2} = W \quad (2)$$

where W is the distributed load on a beam whose longitudinal axis coincides with the X axis and M' is the bending moment due to W. W is positive when it

is acting upward. M' is positive when it causes compression in the upper fiber of a beam.

If Y_Q is a solution of Eq. (1), then using $\frac{-Y_Q}{I}$ as elastic weight W , the bending moment obtained will be equal to

$$M' = Y_R = - \iint \frac{Y_Q}{I} * dX * dX + A * X + B \quad (3)$$

where M' is renamed as Y_R and A and B are arbitrary constants determined by the boundary conditions.

But, from Eq. (1)

$$Y_Q = \frac{P C R}{E * I O} * Y_R \quad (4a)$$

Therefore

$$\frac{Y_Q}{Y_R} = \frac{P C R}{E * I O} \quad (4b)$$

or

$$P C R = \frac{Y_Q}{Y_R} * E * I O \quad (4c)$$

The procedure of obtaining the buckling load then is as follows: The column length L is divided into N equal spacings* of length λ . Assume an initial set of deflections $\tilde{Y}_O(J)$ for each point at $X=J*\lambda$. Let $I(J)$ be the I at the same point. Then take $-\frac{\tilde{Y}_O(J)}{I(J)}$ as the distributed elastic weight W at the point $X=J*\lambda$. Obtain the bending moment $Y(J)$ at each of the equally spaced points. If the ratio

$$\frac{\tilde{Y}_O(J)}{\tilde{Y}(J)} = C \quad \text{for } 0 < J < N \quad (5)$$

where C is a constant (within a tolerable limit) then

$$P C R = C * E * I O \quad (6)$$

If not, then take the $\tilde{Y}(J)$ 'S as the new values for $\tilde{Y}_O(J)$ 'S and repeat the procedure. Note that the real deflection $Y(J)$ due to $P=1$ is given by the following equation:

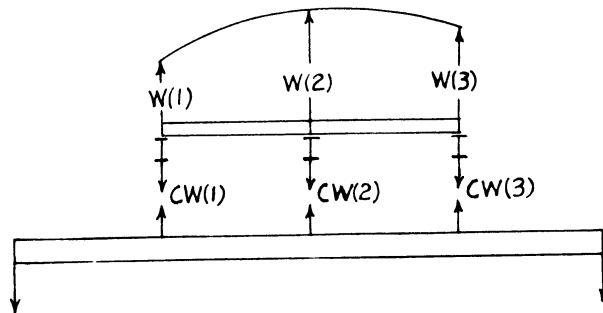
$$Y(J) = \frac{P * \tilde{Y}(J)}{E * I O} = \frac{\tilde{Y}(J)}{E * I O} \quad (\text{for } P = 1) \quad (7)$$

* Spacings do not have to be equal, but computations will be simpler if equal spacing is used.

Example Problem No. 81

The problem is essentially an eigenvalue problem and it can be proved mathematically that the procedure will always converge to the eigenvalue which corresponds to the smallest value of P, i.e. PCR.

Newmark (reference 1) has shown that the numerical integration of equations (1) and (2) can be facilitated by the following method. If W is a distributed load, then we may pass a parabola through three points of W, say W(1), W(2) and W(3), which are equally spaced at a distance λ and convert the distributed load to concentrated loads by the formulas as shown below.



$$CW(1) = [7*W(1) + 6*W(2) - W(3)] * \frac{\lambda}{24} \quad (8a)$$

$$CW(2) = [W(1) + 10*W(2) + W(3)] * \frac{\lambda}{12} \quad (8b)$$

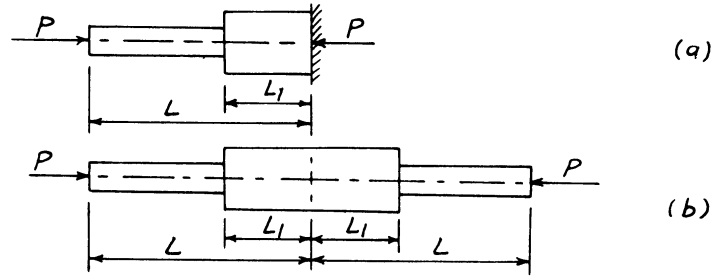
$$CW(3) = [7*W(3) + 6*W(2) - W(1)] * \frac{\lambda}{24} \quad (8c)$$

The shear and moment due to a system of concentrated loads at equally spaced points can be easily obtained without further comment.

The solution to the problem consists of writing a program using the above procedure which will determine the elastic buckling load of a column with simply supported ends and with either uniform or non-uniform cross-section. For columns of non-uniform cross-sections, the cross-sections may be varying continuously or discontinuously with steps occurring at any of the equally spaced points along the column. If the discontinuity occurs between two of the neighboring equally spaced points, the column will be treated as if the cross-section varies continuously between these two points.

It should be noted that the solution technique, with slight modification, can be applied to a cantilever column with one end free and the other end fixed. For example, the cantilever column and the simply supported column as shown below will each have the same buckling load.

Elastic Buckling Load for Columns of Non-Uniform Cross Section



The following suggestions may be helpful in programming the solution. Let us define the following quantities:

$$K\tilde{Y}(J) = \frac{Y(J) * E * I_0}{\lambda^2} = \frac{\tilde{Y}(J)}{\lambda^2} \quad (9)$$

$$CPCR = \frac{L^2 * \tilde{Y}_0(\frac{N}{2})}{E * I_0 * Y(\frac{N}{2})} = \frac{L^2 * \tilde{Y}_0(\frac{N}{2})}{\tilde{Y}(\frac{N}{2})} = \frac{L^2 * \tilde{Y}_0(\frac{N}{2})}{\lambda^2 * K\tilde{Y}(\frac{N}{2})} = \frac{N^2 * \tilde{Y}_0(\frac{N}{2})}{K\tilde{Y}(\frac{N}{2})} \quad (10) *$$

$$T = \frac{N^2 * \tilde{Y}_0(J)}{(CPCR) * K\tilde{Y}(\frac{N}{2})} \quad (11)$$

Let "EPS" be a preassigned tolerance value. Then the condition for constancy of the ratio $\frac{\tilde{Y}_0(J)}{\tilde{Y}(J)}$ (Eq. (5)) may be restated as follows:

$$|T - 1| < EPS \text{ and } T > 0 \quad \left. \begin{array}{l} \text{(for all } N-1 \text{ points)} \\ \text{(with } 0 < J < N \end{array} \right\} \quad (12)$$

The formula for the critical load (Eq. (6)) may be rewritten into the following form

$$PCR = (CPCR) * \frac{E * I_0}{L^2} \quad (13)$$

If the number of points satisfying the above condition (Eq. (12)) is Q, then if $Q < N - 1$, we shall take the $K\tilde{Y}$'s as the new values of \tilde{Y}_0 's and repeat the process. In order to prevent the machine from doing an endless number of iterations, a maximum number of cycles should be specified.

Moreover, in order to prevent the value of \tilde{Y}_0 's from becoming very large after many iterations, we may take the new \tilde{Y}_0 's in terms of the maximum value of \tilde{Y}_0

* Note that N is the number of divisions and "integer division" is assumed for evaluating the quantity $\frac{N}{2}$.

or some value of \tilde{Y}_0 which is close to the maximum, say the \tilde{Y}_0 at a point near the midpoint ($J = \frac{N}{2}$) of the column. In other words, we shall define a new variable Y_0 , such that

$$Y_0(J) = \frac{\tilde{Y}_0(J)}{\tilde{Y}_0(\frac{N}{2})} \quad \text{for initial values} \quad (14a)$$

$$Y_0(J) = \frac{KY(J)}{KY(\frac{N}{2})} \quad \text{for subsequent iterations} \quad (14b)$$

In Eq. (14b)

$$KY(J) = \frac{K\tilde{Y}(J)}{\tilde{Y}_0(\frac{N}{2})} \quad KY(\frac{N}{2}) = \frac{K\tilde{Y}(\frac{N}{2})}{\tilde{Y}_0(\frac{N}{2})} \quad (15)$$

The KY 's are obtained by taking

$$- \frac{Y_0(J)}{I(J)} = W(J) \quad (16)$$

$$EW(J) = \frac{CW(J)}{\lambda} \quad (17)$$

and

$$KY(J) = \frac{M'}{\lambda} \quad \text{where } M' \text{ is the bending moment caused by } EW(J) \quad (18)$$

From the definitions of Y_0 and KY , we have

$$\frac{Y_0(\frac{N}{2})}{KY(\frac{N}{2})} = \frac{\tilde{Y}_0(\frac{N}{2}) / \tilde{Y}_0(\frac{N}{2})}{K\tilde{Y}(\frac{N}{2}) / \tilde{Y}_0(\frac{N}{2})} = \frac{\tilde{Y}_0(\frac{N}{2})}{K\tilde{Y}(\frac{N}{2})} \quad (19)$$

Therefore, the formulas for $\dot{C}PCR$ and T may be rewritten as follows

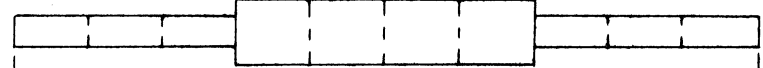
$$CPCR = \frac{N^2 * Y_0(\frac{N}{2})}{KY(\frac{N}{2})} \quad (20)$$

$$T = \frac{N^2 * Y_0(J)}{(CPCR) * KY(\frac{N}{2})} \quad (21)$$

Elastic Buckling Load for Columns of Non-Uniform Cross Section

If the moment of inertia varies discontinuously at any of the equally spaced points, then one way of solving the problem is as follows: Let the number of points with a step change in moment of inertia be M . For $M > 0$, we will have two sets of index numbers for the division points as shown below.

$M = 2$



1	Dummy Index S	0		1				2				M+1		
2	Index No. Set (1)	0	1	2	3	4	5	6	7	8	9	10		
3	STPO(S)	0			3				7			10		
4	Y0	Y0(0)	Y0(1)	Y0(2)	Y0(3)	Y0(4)	Y0(5)	Y0(6)	Y0(7)	Y0(8)	Y0(9)	Y0(10)		
5	Index No. Set (2)	0	1	2	3	4	5	6	7	8	9	10	11	12
6	STPP(S)	-1			3				8					12
7	I	I(0)	I(1)	I(2)	I(3)	I(4)	I(5)	I(6)	I(7)	I(8)	I(9)	I(10)	I(11)	I(12)
8	Y1	Y1(0)	Y1(1)	Y1(2)	Y1(3)	Y1(4)	Y1(5)	Y1(6)	Y1(7)	Y1(8)	Y1(9)	Y1(10)	Y1(11)	Y1(12)
9	Initial Values of Y1	Y0(0)	Y0(1)	Y0(2)	Y0(3)	Y0(4)	Y0(5)	Y0(6)	Y0(7)	Y0(8)	Y0(9)	Y0(10)		
10	Values of Y1 after first move	Y0(0)	Y0(1)	Y0(2)	Y0(3)	Y0(3)	Y0(4)	Y0(5)	Y0(6)	Y0(7)	Y0(8)	Y0(9)	Y0(10)	
11	Values of Y1 after 2nd move	Y0(0)	Y0(1)	Y0(2)	Y0(3)	Y0(3)	Y0(4)	Y0(5)	Y0(6)	Y0(7)	Y0(7)	Y0(8)	Y0(9)	Y0(10)

The index number set (1) will have one number for each division point while the index number set (2) will have two numbers at each step. The I's are stored according to the index number set (2). (See line 7 of the figure.) At each step in I, the index number in set (1) of the point with the step is stored in the vector* "STPO". (The two ends are also considered as steps.) For example, the index number 3 will be in the location for STPO(1). (See line 3 of the figure.)

The values of $Y_0(J)$ are first stored in the vector $Y_1(J)$ with J corresponding to the index number set (1). (See line 9 of the figure.) Then the step numbers, each corresponding to the index number in set (2) on the left side of a step, are computed and stored in the vector "STPP". For example, the index number 8 will be in the location STPP(2). (See line 6 of the figure.) Then starting from the right end, the value of $Y_1(J)$ is put into the location of $Y_1(J+1)$ until the index number of STPP(1) is reached. (See line 10 of the figure.) The process is then repeated until STPP(2) is reached and so on until STPP(M). (See line 11 of the figure.)

* The "vector" is a linear array of numbers as defined in modern algebra and used in computer language.

Example Problem No. 81

Next, the concentrated elastic weights (EW) at the ends are computed. Then the elastic weights at two neighboring points at each step and the elastic weights between the steps are computed. Then the two concentrated elastic weights at the two neighboring points of each step are combined into one value and the values EW(J) are moved back to the locations with index number set (1). The procedure is similar to the one for Y1(J) described above but in reverse.

With the elastic weights known, the procedure will be the same as that for columns of continuously varying moment of inertia.

Table of Symbols

In the following, the definitions of the variables which have been introduced previously and will be used in the program are recapitulated. Some new variables needed for the programming are also defined.

N	number of divisions
M	number of division points with step change in moment of inertia
EPS	an assigned tolerance value
STPO(0)...STPO(M + 1)	Index numbers of points with steps corresponding to index number Set (1). The two end points are also considered as steps.
I(0)...I(M + N)	The ratio of the actual moment of inertia (IA) about an axis normal to the plane of bending to the moment of inertia at a reference point (IO). The I's carry the index numbers of Set (1) for continuous variation and they carry the index number of Set (2) for discontinuous variation at the division points.
YO(0)...YO(N)	The ratio of the initial deflection at a point $\tilde{Y}_0(J)$ to the same at a reference point near the center of the span $\frac{Y_0(N)}{2}$. The YO's initially read in are assumed. For subsequent iterations, $Y_0(J) = \frac{KY(J)}{KY(\frac{N}{2})}$. The YO's carry the index numbers of Set (1).
STPP(0)...STPP(M + 1)	Index numbers of the steps, each corresponding to the index number in Set (2) on the left side of a step.
Y1(0)...Y1(M + N)	YO's carrying the index numbers of Set (2).
EW(0)...EW(M + N)	Concentrated elastic weights due to the distributed elastic load $-\frac{Y(J)}{I(J)}$ with division length λ as a unit. These elastic weights carry the index numbers of Set (2).
EWNN	refers to the set of elastic weights EW(0)...EW(M + N).

Elastic Buckling Load for Columns of Non-Uniform Cross Section

EW(0)...EW(N) Similar to the weights in the set EWNN except that the weights carry index numbers of Set (1). The two weights at each stepping point of the Set EWNN have been combined into one.

EW refers to the set of elastic weight EW(0)...EW(N).

RL, RR respectively left and right reactions due to EW

SW shear due to EW

KY(0)...KY(N) deflections with $\frac{\lambda^2 * YO(\frac{N}{2})}{E * IO}$ as a unit. They are equal to the bending moments (with division length λ as a unit) due to the concentrated elastic weights EW.

$$CPCR = \frac{N^2 * YO(\frac{N}{2})}{KY(\frac{N}{2})}$$

$$T = \frac{N^2 * YO(J)}{(CPCR) * KY(\frac{N}{2})}$$

CYL = number of cycles of iterations

The data to be read in are as follows:

N, M, EPS
 STPO(0)...STPO(M + 1)
 I(0)...I(M + N)
 YO(0)...YO(N) (assumed)

The data read in should be printed out immediately as an echo check.

The output should be as follows:

CYL the final number of iteration when convergence is obtained.
 CPCR the coefficient of the critical load.

The critical load is understood as given by Eq. 13

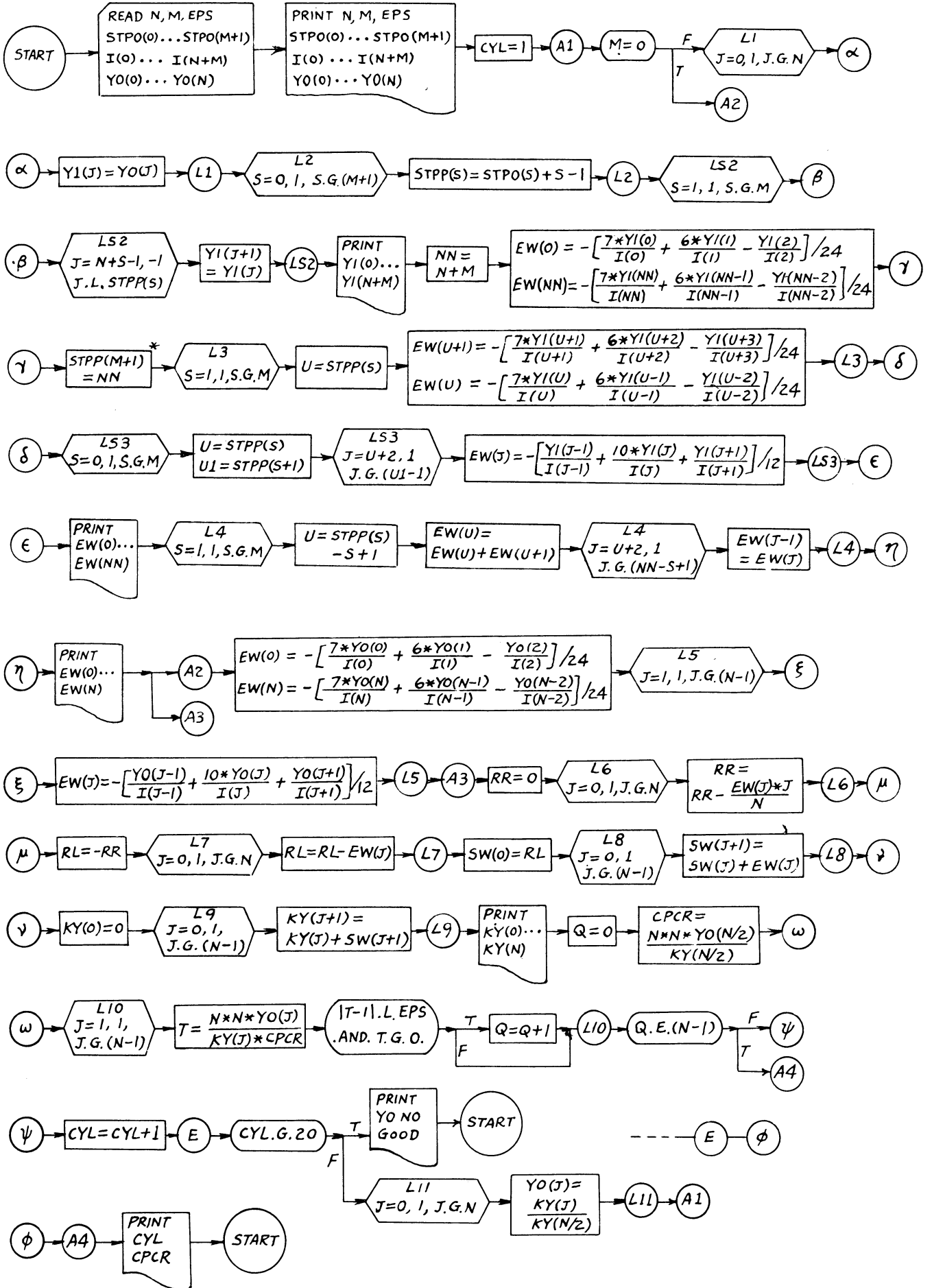
$$PCR = (CPCR) * \frac{E * IO}{L^2}$$

Where E is the modulus of elasticity, IO is the moment of inertia at a reference point and L is the length of the column as defined previously.

In order to check the results by manual computations, the following intermediate steps shall be printed out for each iteration:

Y1(0)...Y1(N + M)
 EWNN [EW(0)...EW(N + M)]
 EW [EW(0)...EW(N)]
 KY(0)...KY(N)

Flow Diagram



Elastic Buckling Load for Columns of Non-Uniform Cross Section

MAD Program

```

KUANG-HAN CHU          D028N          001  010  000
KUANG-HAN CHU          D028N          001  010  000
SCOMPILE MAD, EXECUTE
R  ELASTIC BUCKLING LOAD FOR COLUMNS
R  OF NON-UNIFORM CROSS SECTION
  DIMENSION YO(50),I(100),EW(100),SW(50),KY(50),
1  STPO(50),STPP(100),Y1(100)
  INTEGER N,J,CYL,Q,M,NN,S,U,U1,STPO,STPP
START
  READ FORMAT DATA1,N,M,EPS
  VECTOR VALUES DATA1=$2I4,F7.5*$
  READ FORMAT DATA2,STPO(0)...STPO(M+1)
  VECTOR VALUES DATA2=$(18I4)*$
  READ FORMAT DATA3,I(0)...I(N+M)
  VECTOR VALUES DATA3=$(8F9.3)*$
  READ FORMAT DATA4,YO(0)...YO(N)
  VECTOR VALUES DATA4=$(8F9.3)*$
  PRINT FORMAT ECHO1,N,M,EPS
  VECTOR VALUES ECHO1=$1H1,S9,3HN =,I4,S3,3HM =,I3,S4,
1  4HEPS=,F6.4*$
  PRINT FORMAT ECHO2,STPO(0)...STPO(M+1)
  VECTOR VALUES ECHO2=$1H0,S9,8HSTEPS AT/(S10,12I5)*$
  PRINT FORMAT ECHO3,I(0)...I(N+M)
  VECTOR VALUES ECHO3=$1H0,S9,10HRELATIVE I/(S10, 6F10.3)*$
  PRINT FORMAT ECHO4,YO(0)...YO(N)
  VECTOR VALUES ECHO4=$1H0,S9,2HYO/(S10, 6F10.3)*$
  CYL=1
A1  WHENEVER M.E.0,TRANSFER TO A2
  THROUGH L1,FOR J=1,1,J.G.N
L1  Y1(J)=YO(J)
  THROUGH L2,FOR S=0,1,S.G.M+1
L2  STPP(S)=STPO(S)+S-1
  THROUGH LS2,FOR S=1,1,S.G.M
  THROUGH LS2,FOR J=N+S-1,-1,J.L.STPP(S)
LS2 Y1(J+1)=Y1(J)
  PRINT FORMAT CHECK1,Y1(0)...Y1(N+M)
  VECTOR VALUES CHECK1=$1H0,S9,2HY1/(S10, 6F10.3)*$
  NN=N+M
  EW(0)=- (7*Y1(0)/I(0)+6*Y1(1)/I(1)-Y1(2)/I(2))/24
  EW(NN)=- (7*Y1(NN)/I(NN)+6*Y1(NN-1)/I(NN-1)-Y1(NN-2)/I(NN-2))/24
1  +6*Y1(NN-1)/I(NN-1)-Y1(NN-2)/I(NN-2))/24
  STPP(M+1)=NN
  THROUGH L3,FOR S=1,1,S.G.M
  U=STPP(S)
  EW(U+1)=- (7*Y1(U+1)/I(U+1)
1  +6*Y1(U+2)/I(U+2)-Y1(U+3)/I(U+3))/24
L3  EW(U)=- (7*Y1(U)/I(U)
1  +6*Y1(U-1)/I(U-1)-Y1(U-2)/I(U-2))/24
  THROUGH LS3,FOR S=0,1,S.G.M
  U=STPP(S)
  U1=STPP(S+1)
  THROUGH LS3,FOR J=U+2,1,J.G.(U1-1)
LS3  EW(J)=- (Y1(J-1)/I(J-1)+10*Y1(J)/I(J)+Y1(J+1)/I(J+1))/12
  PRINT FORMAT CHECK2,EW(0)...EW(NN)
  VECTOR VALUES CHECK2=$1H0,S9,4HEWNN/(S10, 6F10.3)*$
  THROUGH L4,FOR S=1,1,S.G.M
  U=STPP(S)-S+1
  EW(U)=EW(U)+EW(U+1)
  THROUGH L4,FOR J=U+2,1,J.G.(NN-S+1)
L4  EW(J-1)=EW(J)
  PRINT FORMAT CHECK3,EW(0)...EW(N)
  VECTOR VALUES CHECK3=$1H0,S9,2HEW/(S10, 6F10.3)*$
  TRANSFER TO A3
A2  EW(0)=- (7*YO(0)/I(0)+6*YO(1)/I(1)-YO(2)/I(2))/24
  EW(N)=- (7*YO(N)/I(N)+6*YO(N-1)/I(N-1)-YO(N-2)/I(N-2))/24
  THROUGH L5, FOR J=1,1,J.G.(N-1)
L5  EW(J)=- (YO(J-1)/I(J-1)+10*YO(J)/I(J)+YO(J+1)/I(J+1))/12
A3  RR=0
  THROUGH L6,FOR J=0,1,J.G.N
L6  RR=RR-EW(J)*J/N
  RL=-RR
  THROUGH L7,FOR J=0,1,J.G.N

```

MAD Program, Continued

```

L7      RL=RL-EW(J)
        SW(0)=RL
        THROUGH L8, FOR J=0,1,J.G.(N-1)
L8      SW(J+1)=SW(J)+EW(J)
        KY(0)=0
        THROUGH L9, FOR J=0,1,J.G.(N-1)
L9      KY(J+1)=KY(J)+SW(J+1)
        PRINT FORMAT CHECK4,KY(0)...KY(N)
        VECTOR VALUES CHECK4=$1H0,S9,2HKY/(S10, 6F10.3)*$
        Q=0
        CPCR = N*N*YO(N/2)/KY(N/2)
        THROUGH L10, FOR J=1,1,J.G.(N-1)
        T=N*N*YO(J)/(KY(J)*CPCR)
L10     WHENEVER (.ABS.(T-1).L.EPS).AND.(T.G.0.),Q=Q+1
        WHENEVER Q.E.(N-1)
        TRANSFER TO A4
        OTHERWISE
        CYL=CYL+1
        END OF CONDITIONAL
        WHENEVER CYL.G.20
        PRINT FORMAT NGD
        VECTOR VALUES NGD=$1H0,S9,10HYO NO GOOD*$
        TRANSFER TO START
        OTHERWISE
        THROUGH L11, FOR J=1,1,J.G.N
L11     YO(J)=KY(J)/KY(N/2)
        TRANSFER TO A1
        END OF CONDITIONAL
A4      PRINT FORMAT CYN,CYL
        VECTOR VALUES CYN=$1H0,S9,15HNO. OF CYCLES =,I4*$
        PRINT FORMAT CRL,CPCR
        VECTOR VALUES CRL=$1H0,S9,15HCOEFF. OF PCR =,F7.2*$
        TRANSFER TO START
        END OF PROGRAM
    
```

Data

\$DATA									
10	2	.005							
0	3	7 10							
	.2	.2	.2	.2	1.	1.	1.	1.	
	1.	.2	.2	.2	.2				
	0.	.4	.6	.8	.9	1.	.9	.8	
	.6	.4	0.						
10	0	.005							
0	10								
	.25	.36	.49	.64	.81	1.	.81	.64	
	.49	.36	.25						
	0.	.4	.6	.8	.9	1.	.9	.8	
	.6	.4	0.						
12	0	.005							
0	12								
	.2	.2	.2	.2	1.	1.	1.	1.	
	1.	.2	.2	.2	.2				
	0.	.4	.6	.8	.9	.95	1.	.95	
	.9	.8	.6	.4	0.				

Elastic Buckling Load for Columns of Non-Uniform Cross Section

Computer Output

N = 10 M = 2 EPS=0.0050

STEPS AT

0 3 7 10

RELATIVE I

0.200	0.200	0.200	0.200	1.000	1.000
1.000	1.000	1.000	0.200	0.200	0.200
0.200					

YO

0.000	0.400	0.600	0.800	0.900	1.000
0.900	0.800	0.600	0.400	0.000	

Y1

0.000	0.400	0.600	0.800	0.800	0.900
1.000	0.900	0.800	0.800	0.600	0.400
0.000					

EWNN

-0.375	-1.917	-3.000	-1.833	-0.417	-0.900
-0.983	-0.900	-0.417	-1.833	-3.000	-1.917
-0.375					

EW

-0.375	-1.917	-3.000	-2.250	-0.900	-0.983
-0.900	-2.250	-3.000	-1.917	-0.375	

KY

0.000	8.558	15.200	18.842	20.233	20.725
20.233	18.842	15.200	8.558	0.000	

Y1

0.000	0.413	0.733	0.909	0.909	0.976
1.000	0.976	0.909	0.909	0.733	0.413
0.000					

EWNN

-0.363	-2.026	-3.607	-2.157	-0.468	-0.973
-0.996	-0.973	-0.468	-2.157	-3.607	-2.026
-0.363					

EW

-0.363	-2.026	-3.607	-2.624	-0.973	-0.996
-0.973	-2.624	-3.607	-2.026	-0.363	

KY

0.000	9.728	17.429	21.524	22.995	23.493
22.995	21.524	17.429	9.728	0.000	

Y1

0.000	0.414	0.742	0.916	0.916	0.979
1.000	0.979	0.916	0.916	0.742	0.414
0.000					

EWNN

-0.363	-2.034	-3.646	-2.177	-0.470	-0.975
-0.996	-0.975	-0.470	-2.177	-3.646	-2.034
-0.363					

EW

-0.363	-2.034	-3.646	-2.647	-0.975	-0.996
-0.975	-2.647	-3.646	-2.034	-0.363	

KY

0.000	9.801	17.568	21.689	23.162	23.661
23.162	21.689	17.568	9.801	0.000	

NO. OF CYCLES = 3

COEFF. OF PCR = 4.23

Example Problem No. 81

Computer Output, Continued

N = 10 M = 0 EPS=0.0050

STEPS AT
0 10

RELATIVE I						
0.250	0.360	0.490	0.640	0.810	1.000	
0.810	0.640	0.490	0.360	0.250		

YO						
0.000	0.400	0.600	0.800	0.900	1.000	
0.900	0.800	0.600	0.400	0.000		

KY						
0.000	5.104	9.180	12.039	13.662	14.171	
13.662	12.039	9.180	5.104	0.000		

KY						
0.000	5.257	9.570	12.587	14.289	14.804	
14.289	12.587	9.570	5.257	0.000		

KY						
0.000	5.243	9.555	12.574	14.277	14.793	
14.277	12.574	9.555	5.243	0.000		

NO. OF CYCLES = 3

COEFF. OF PCR = 6.76

N = 12 M = 0 EPS=0.0050

STEPS AT
0 12

RELATIVE I						
0.200	0.200	0.200	0.200	1.000	1.000	
1.000	1.000	1.000	0.200	0.200	0.200	
0.200						

YO						
0.000	0.400	0.600	0.800	0.900	0.950	
1.000	0.950	0.900	0.800	0.600	0.400	
0.000						

KY						
0.000	11.183	20.450	26.717	29.325	30.771	
31.267	30.771	29.325	26.717	20.450	11.183	
0.000						

KY						
0.000	11.604	21.445	28.056	30.756	32.236	
32.735	32.236	30.756	28.056	21.445	11.604	
0.000						

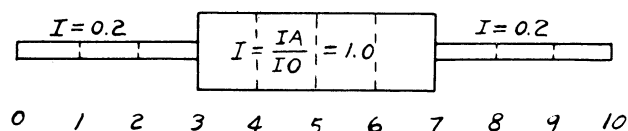
KY						
0.000	11.610	21.470	28.095	30.799	32.280	
32.778	32.280	30.799	28.095	21.470	11.610	
0.000						

NO. OF CYCLES = 3

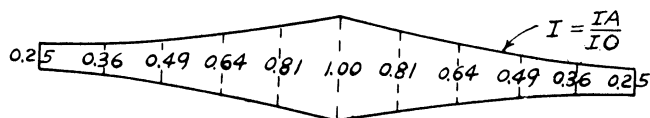
COEFF. OF PCR = 4.39

Discussion of Results

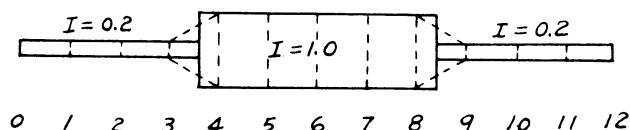
Three example problems are shown. The first example is a column shown below which is the same as shown in the previous figure. With 10 division points and values of EPS and Y0 as shown in the program, the procedure converges in 3 cycles and gives the answer of (CPCR) = 4.23. The coefficient given by exact solution of the differential equation is 4.22 (see reference 2).



The second example is for a column as shown below. The coefficient determined by the numerical method is 6.76, as compared with the value 6.72 which is interpolated from a table given in reference 3.



The third example for the column shown below is the same as the first, except that it uses 12 divisions and any discontinuity of moment of inertia between two neighboring division points is ignored and the column is treated as if it were a column with moment of inertia varying continuously. The value of (CPCR) obtained is 4.39 as compared with the true value 4.22.



It should be noted that although the examples shown are for columns with moment of inertia varying symmetrically with respect to the mid-point of the column, the program is perfectly general and is applicable to columns with moment of inertia varying unsymmetrically.

Critique

The program is a natural one for the computer since the computer is most suitable for any iterative procedure which requires us to do the same computation process over and over again.

References

1. N. M. Newmark, "Numerical Procedure for Computing Deflections Moments and Buckling Loads," ASCE Transactions, 1943.
2. S. Timoshenko, Theory of Elastic Stability.
3. S. Timoshenko & J. M. Lessels, Applied Elasticity.

Example Problem No. 82
A MAD PROGRAM FOR TRUSS ANALYSIS

by

K. H. Chu
Civil Engineering Department
Illinois Institute of Technology

Course: Structural Theory Credit hours: 3 Level: Junior

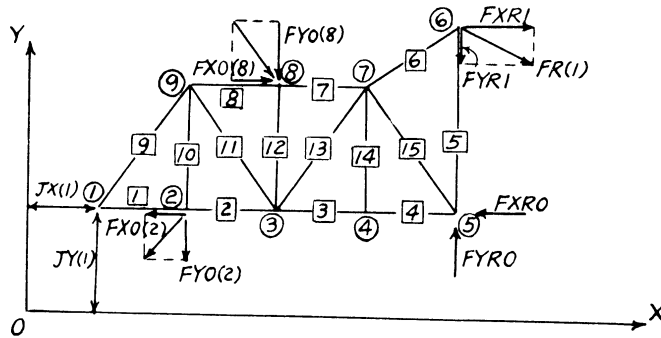
Statement of the Problem

The problem is to write a program which will determine stresses in members of trusses where the method of joints is applicable. The problem is of repetitive type which is well adapted for computers. Also it is of fundamental importance because if it is used as a subroutine, then a series of problems in truss analysis can be solved easily; for example, influence lines for trusses, deflection of truss by virtual work, stresses in indeterminate trusses, etc.

The method of joints is chosen instead of setting up a matrix and solving a system of simultaneous equations because there will be too many zero elements in the matrix. (The number of simultaneous equations the computer can solve by elimination is usually limited, say to 30.) Therefore, there is a need to develop a program which will not only conserve storage space but also will do the problem as we do it manually, i.e., the machine would select the joint to be solved and proceed to a joint which can be solved next.

Solution

Let us consider the truss shown on the next page. We will number the joints and members in any fashion and set up an arbitrary but convenient coordinate system.



The data to be read in are as follows:

N the total number of joints

M the total number of members

EPS1 a small number to guard against the case $|D| \approx 0$ (explained later)

JT(1)...JT(N) index number of the joints

JX(1)...JX(N)
JY(1)...JY(N) X and Y coordinates of the joints

FXO(1)...FXO(N) the X and Y components of the force acting on each joint. Positive
FYO(1)...FYO(N) if the component is in the positive direction of the coordinate
axis, zero if there is none.

UNMB(1)...UNMB(N) the number of members meeting at each joint. Their stresses
are unknowns to be determined. For example, in the figure, the
value of UNMB(3) will be equal to 5.

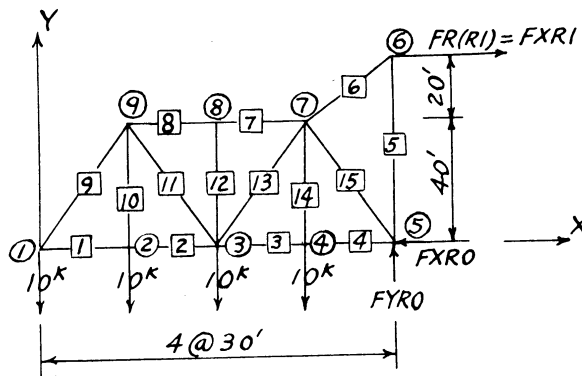
MB(1)...MB(M) the index number of the members and the corresponding index
JN(1)...JN(M) number of the joints at the ends of each member. It does not
JF(1)...JF(M) matter which end is referred to as the near joint (JN) and which
end is referred to as the far joint (JF). For example, in the figure
the value of MB(15) will be 15 and the values of JN(15) and
JF(15) may be 5 and 7 respectively or 7 and 5 respectively.

A MAD Program for Truss Analysis

So far as reactions are concerned, we need to read in the following data:

- JRO The joint number where the reaction of unknown magnitude and direction is applied. For example in the figure the value of JRO is 5.
- R1 This value is equal to 1. The purpose of reading in this number is for extending the program to the case of finding reactions along 3 given lines of applications. The reactions will be identified as FR(R1), FR(R2), FR(R3) with R1, R2, R3 being equal to 1, 2, 3 respectively. For the sample problem under consideration, it serves no useful purpose except to identify the reaction with given line of application as FR(R1) or FR(1).
- JR1 The joint number where the reaction of given line of application is applied. For example, in the figure the value of JR1 is 6.
- LXR1, LYR1, LR1 These are the numbers which specify the given line of application in the following way. The line of action is assumed to be a vector directed away from the joint and length of the vector (LR1) may be taken as unity or any other value. The X and Y components of this vector are denoted as LXR1 and LYR1 respectively.

In order that we understand the correlation of the various data which have been read in, a concrete example is shown below. This is the same truss as shown earlier except that the dimensions and loadings are specified and the origin of the coordinate axes is at joint 1 and the X axis coincides with the lower chord. The data for JT, JX, JY, FXO, FYO and UNMB are given in Table 1, and the data for MB, JN and JF are given in Table 2. The data for the reactions are given in Table 3. In the tables, there are many items which have not been defined. They will be explained later.



Having read in all the data, the first step is to compute the following quantities:

- LX(1)...LX(M)
 - LY(1)...LY(M)
 - L(1)...L(M)
- where L's are the length of the members and LX and LY are respectively the components of the vector of length L which is directed from the joint (JN) toward the joint (JF). The computed values of LX, LY, and L for the members of the truss shown are given in Table 2.

Example Problem No. 82

Next we shall compute the reactions with the definition of terms as follows:

SFX, SFY, SMO are respectively the sum of the X components of the external forces, the sum of the Y components of the external forces and the sum of moments of the external forces about the joint (JRO).

FR(R1) is the force of the reaction of known line of action at the joint (JR1). It will be positive if it is directed away from the joint. This quantity is denoted as a subscripted variable with the intention of writing a subroutine (or external function) for computing 3 reactions with given lines of action. The method of section is a possible application of this subroutine.

FXR1, FYR1 are respectively the X and Y components of FR(R1). The signs of FXR1 and FYR1 are positive if they are in the positive directions of the coordinate axes.

FXO(JR1), FYO(JR1) The external forces acting on the joint (JR1) must be modified to include the reaction components just computed.

FXRO, FYRO are values at the joint (JRO) corresponding to those as FXO(JRO), FYO(JRO) defined for the joint (JR1).

The answers for the values of FR(R1), FXR1, FYR1, FXRO, FYRO for the example shown are given in Table 3. The values of FXO, FYO at joints 5 and 6 after the reactions have been computed are as shown by the circled values in lines 7 and 8 of Table 1.

Table 1

Line	Dummy Index I	1	2	3	4	5	6	7	8	9
1	JT(I)	1	2	3	4	5	6	7	8	9
2	JX(I)	0.	30.	60.	90.	120.	120.	90.	60.	30.
3	JY(I)	0.	0.	0.	0.	0.	60.	40.	40.	40.
4	FXO(I)	0.	0.	0.	0.	0.	0.	0.	0.	0.
5	FYO(I)	-10.	-10.	-10.	-10.	0.	0.	0.	0.	0.
6	UNMB(I)	2	3	5	3	3	2	5	3	4
7	After Solving	FXO	0.	0.	0.	0.	-50.	+50.	0.	0.
8	the reactions	FYO	-10.	-10.	-10.	-10.	+40.	0.	0.	0.
9	After	UNMB	①	②	5	3	3	2	5	3
10	solving the	FXO	0.	+750.	0.	0.	-50.	+50.	0.	0.
11	first Joint	FYO	-10.	-1000.	-10.	-10.	+40.	0.	0.	-1000.

Table 3

Data	Answers
JRO 5	FR(R1) 50.
JR1 6	FXR1 50.
LXR1 1.00	FYR1 0.
LYR1 0.00	FXRO -50.
LRI 1.00	FYRO 40.

Table 2

Line	Dummy Index J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	MB(J)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	JN(J)	1	2	3	4	5	6	7	8	9	2	3	3	3	4	5
3	JF(J)	2	3	4	5	6	7	8	9	1	9	9	8	7	7	7
4	LX(J)	30.	30.	30.	30.	0.	-30.	-30.	-30.	-30.	0.	-30.	0.	30.	0.	-30.
5	LY(J)	0.	0.	0.	0.	60.	-20.	0.	0.	-40.	+40.	+40.	+40.	40.	40.	40.
6	L(J)	30.	30.	30.	30.	60.	36.06	30.	30.	50.	40.	50.	40.	50.	40.	50.
7	JTS(J)	1								1						
8	valves used	LX(J)	+30.							+30.						
9	for the joint	LY(J)	0.							+40.						
10	Eqs. for JT(I)	JN(J)	9999							9999						
11	After	JF(J)	9999							9999						
12	solving	FX(J)	0.00							+7.50						
13	the first	FY(J)	-7.50							+10.00						
14	Joint	F(J)	-7.50							+12.50						

A MAD Program for Truss Analysis

Next we shall compute the stresses in the members. We shall use the following symbols:

JT(I)	is the index number of the joint to be solved as identified by the dummy index I.
MS(1), MS(2)	are the index numbers of the two members meeting at JT(I).
F(MS(1)), F(MS(2))	are the two forces in the members to be solved. They are positive if they are directed away from the joint JT(I).
JTS(MS(1)), JTS(MS(2))	both store the value of JT(I) in order that in the output, we know the forces in the members MS(1) and MS(2) are the results of the solution of that particular joint.
FX(MS(1)), FY(MS(1)) FX(MS(2)), FY(MS(2))	are the components of the forces acting on JT(I) in the members MS(1) and MS(2). They are positive if they are along the positive direction of the coordinate axes.
IS(1), IS(2)	are respectively the joint numbers of the other ends of the members MS(1) and MS(2) which meet at the joint JT(I).
FXO(IS(1)), FYO(IS(1)) FXO(IS(2)), FYO(IS(2))	The external forces FXO, FYO acting on the joints IS(1) and IS(2) shall be revised in order to include the components FX(MS(1)), FY(MS(1)), etc. as known forces. It should be noted that the signs of FX(MS(1)), FY(MS(1)), etc. should be reversed since they are originally considered as applied at the other ends of the members (i.e. at the joint JT(I)).

The joint equations can be written very easily. In writing the joint equations, it should be noted that the joint under consideration JT(I) is taken as the near joint (JN). The forces are acting on the joint JT(I) and shall be considered as positive when they are directed away from the joint JT(I). We should also guard against the case where the denominator of the joint equation is nearly equal to zero. This is really the case when the joint is unstable.

The remaining problem is how the machine can tell which joint is to be solved first and which to be solved next. The following suggestions may be helpful. Have the machine examine a list of UNMB. If there is a joint with UNMB=2, the index number I of the joint JT(I) is picked up. First look into the list for JN. If at the index number J, the value of JN(J) is equal to JT(I), then the index number J is picked up and stored in the location MS(1). Notice that this number J identifies the member MB(J) (See Table 1) which is renamed as MS(1), since this is the first member to be solved for at the joint under consideration. Then, in order to prevent the number in JN(J) and JF(J) from being picked up in future examination of the list, we change the values of JN(J) and JF(J) into some unreasonable number, say 9999.

Example Problem No. 82

Then look further down the list of JN. If we find that at another value of J (say J'), JN(J') is equal to JT(I), then we store J' in MS(2) and the value of JF(J') of the number MB(J') in IS(2). Also we change the values of JN(J') and JF(J') into 9999.

If we examine the list of JN and find either none of the values or only one of the values is equal to JT(I), then we must examine the list of JF to find either both of the members MS(1) and MS(2) or only the member MS(2). The procedure is similar to the one described above with the following exceptions. Suppose that we find the value of JT(I) in the location of JF(J). We should store the value of JN(J) in the location of IS and we should reverse the signs of the values LX(MB(J)) and LY(MB(J)). The sign change is necessary because the joint equation is written for the joint JT(I) which is considered as the near joint (JN). Since we find the value of JT(I) in the list of far joints (JF), the vector which represents the positive direction of the member MB(J) must be reversed. (Note that the value of JT(I) is identical with the index number I and the value of MB(J) is identical with the index number J.)

Having found the two members meeting at the joint JT(I) ready to be solved, we end the search. We may set up a counter P to see whether P reaches the value of 2. Also we would like to update the number in the list of UNMB for the two members meeting at JT(I). The value of UNMB(I) corresponding to the joint solved JT(I) should be reduced by 2. Each of the two values of UNMB corresponding to the far ends of the 2 members solved as identified by the joint numbers corresponding to IS(1) and IS(2) shall be reduced by 1.

After going through the above procedure of searching and updating, we can go through the following steps: (1) Applying the joint equations to find the forces in the two members MS(1) and MS(2), (2) Obtaining the components of these forces, and (3) Modifying the forces applied at the joints at the far end of the member to include the components obtained as known forces.

As a concrete example of the above procedure, we examine the list of UNMB in Table 1 and find that UNMB=2 at I=1. Thus JT(1) is the joint to be solved. Next we examine the list for JN in Table 2 and find that JN(J)=JT(I)=I=1 at J=1. Thus we have MS(1)=1, noting that MS(1) really represents the member MB(1). We set IS(1)=JF(1)=2 and change the values of JN(1) and JF(1) to 9999.

A MAD Program for Truss Analysis

Looking further down the list of JN, we do not find any value equal to $JT(I)=I=1$. When we look into the list of JF, we find $JF(J)=JT(I)=I=1$ at $J=9$. Thus $MS(2)=MB(9)=9$. We set $IS(2)=JN(9)=9$ and change the values $JN(9)$ and $JF(9)$ to 9999 (See Table 2, lines 10 and 11). Since the number $JT(1)$ is found in the list of JF, we change the signs of $LX(9)$ and $LY(9)$ (See lines 8 and 9 in Table 2).

We will stop the search since we have already found the two members to be solved. We then reduce the value of $UNMB(JT(1))$ by 2 and the values of $UNMB(IS(1))$ and $UNMB(IS(2))$ each by 1 (See circled numbers in line 9 of Table 1). Since the joint solved is $JT(1)$, the values of $JTS(1)$ and $JTS(9)$ for the members $MB(1)$ and $MB(9)$ respectively are equal to $JT(1)=1$. (See line 7 of Table 2).

The forces $F(MS(1))$ and $F(MS(2))$ are then determined by applying the joint equations to the joint $JT(1)$. (See line 14 of Table 2.) Their components with respect to the joint $JT(1)$ are then determined. (See lines 12 and 13 of Table 2.) The forces FXO and FYO applied at joints $IS(1)$ and $IS(2)$ are then modified by adding the force components of $MS(1)$ and $MS(2)$ with sign reversed to the external forces acting at these joints. (See lines 10 and 11 in Table 1).

After the first joint has been solved, we re-examine the list of $UNMB$ (line 9 of Table 1) and repeat the whole procedure. We shall stop the process when all stresses in all the M members are solved. Since after each application of the joint equation, the total number of unknowns is reduced by 2, we may set up a counter, say Q , which will be increased by 2 after each cycle. Since M may be either even or odd, finally we may have $(M-Q)$ equal to 0 or 1. If it is equal to zero, the stresses in all the members have been determined. If it is equal to 1, we shall find the remaining member using a procedure similar to the one described previously. The force in the remaining member can easily be determined. It should be noticed, however, that in order to maintain accuracy, the force should be determined from the component with absolute value larger than the other.

We would like to print out the data as originally read in. They are: $JT, JX, JY, FXO, FYO, UNMB, MB, JN, JF, JRO, R1, JR1, LX1, LY1$ and $LR1$. The answers to be printed out are: $FR(R1), FXR1, FYR1, FXRO, FYRO, FXO(JRO), FYO(JRO), FXO(JR1), FYO(JR1), LX, LY, L, JTS, FX, FY$ and F .

With the detailed instructions as given above, the programming becomes more or less straight forward. From statement labeled "Start" to the statement just preceding the through loop L3, the data are read in and printed out in the specific form as stated in the instructions.

The values of LX, LY and L are computed in the loop L3. The fifteen statements following the statement SFX=0 are essentially for determining the reactions. The following quantities are computed: SFX, SFY, SMO, FR(R1), FXR1, FYR1, FXO(JR1), FYO(JR1), FXRO, FYRO, FXO(JRO) and FYO(JRO). The formula for SMO may be written in the following form:

$$SMO = \sum_{I=1}^N \{ [JX(I) - JX(JRO)] * FYO(I) - [JY(I) - JY(JRO)] * FXO(I) \} \quad (1)$$

With computer language, in a THROUGH loop for I, the equation for SMO becomes

$$SMO = SMO + [JX(I) - JX(JRO)] * FYO(I) - [JY(I) - JY(JRO)] * FXO(I) \quad (1a)$$

The formula for FR(R1) is written in the following form:

$$FR(R1) = \frac{SMO * LR1}{[JY(JR1) - JY(JRO)] * LXR1 - [JX(JR1) - JX(JRO)] * LYR1} \quad (2)$$

The searching for the members to be solved and updating various items by the procedure as described in the instructions are performed from twenty-seven statements after and including the statement labeled C3.

The joint equations are solved and the quantities F(MS(1)), FX(MS(1)), FY(MS(1)), FXO(IS(1)), FYO(IS(1)), F(MS(2)), FX(MS(2)), FY(MS(2)), FXO(IS(2)), and FYO(IS(2)) are computed from the statement (D = ...) to the statement (FYO(IS(2)) = ...). The joint equations are written in the following form:

$$F(MS(1)) = \frac{-FXO(JT(I)) * \frac{LY(MS(2))}{L(MS(2))} + FYO(JT(I)) * \frac{LX(MS(2))}{L(MS(2))}}{D} \quad (3)$$

$$F(MS(2)) = \frac{FXO(JT(I)) * \frac{LY(MS(1))}{L(MS(1))} - FYO(JT(I)) * \frac{LX(MS(1))}{L(MS(1))}}{D} \quad (4)$$

where

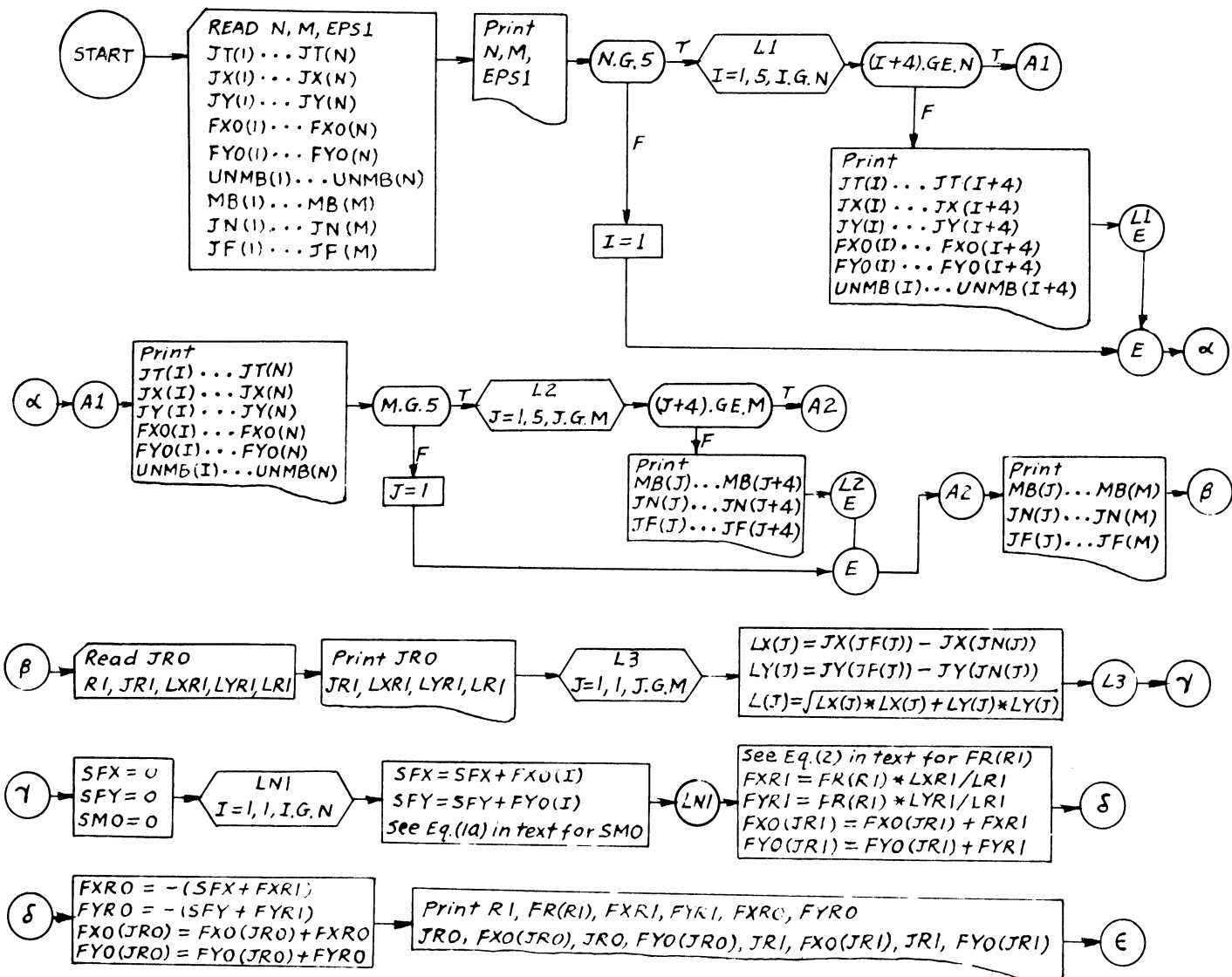
$$D = \frac{LX(MS(1)) * LY(MS(2))}{L(MS(1)) * L(MS(2))} - \frac{LY(MS(1)) * LX(MS(2))}{L(MS(1)) * L(MS(2))} \quad (5)$$

A MAD Program for Truss Analysis

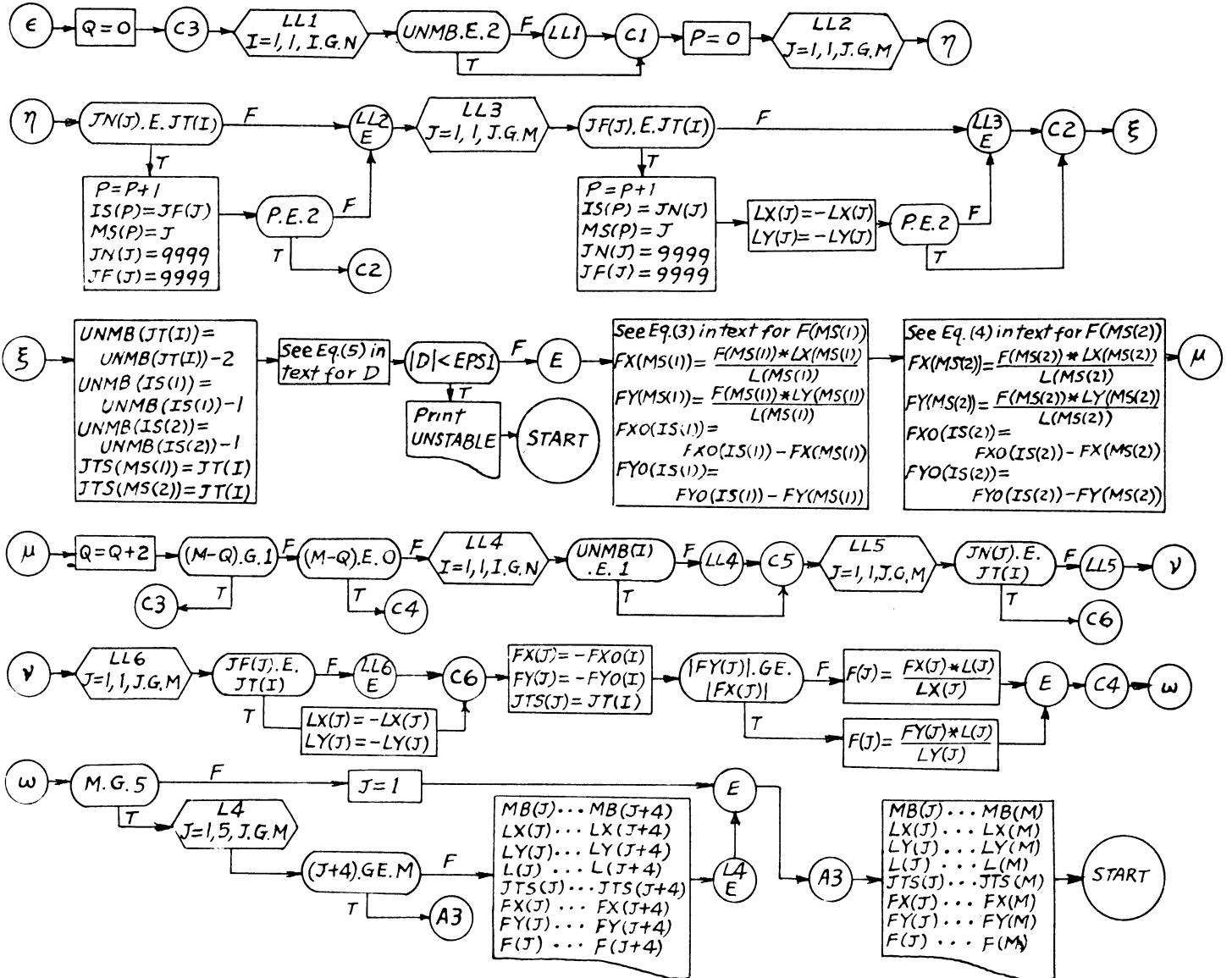
The counter Q registers the number of members already solved. If only one member is left to be solved, $(M-Q) = 1$, the joint with UNMB=1 is searched out by the loop LL4. The member to be solved is searched out by the loops LL5 and LL6. The force in the member is found by the statement starting from the statement labeled C6 to the statement END OF CONDITIONAL just before the statement labeled C4.

The rest of the statements are for printing out the answers.

Flow Diagram



Flow Diagram, Continued



MAD Program

```

KUANG-HAN CHU          D028N          001  003  000
KUANG-HAN CHU          D028N          001  003  000
$COMPILE MAD, EXECUTE
R  A MAD PROGRAM FOR TRUSS ANALYSIS
DIMENSION JT(50),FXO(50),FYO(50),UNMB(50),JX(50),JY(50)
DIMENSION MB(100),JN(100),JF(100),LX(100),LY(100),L(100)
DIMENSION JTS(100),FX(100),FY(100),F(100),IS(2),MS(2),FR(3)
INTEGER JT,UNMB,JN,JF,MB,JTS,IS,MS,P,Q,I,J,M,N,JRO,JR1,R1
START
READ FORMAT DATA1,N,M,EPS1
VECTOR VALUES DATA1=$2I4,F8.5*$
READ FORMAT DATA2,JT(1)....JT(N)
VECTOR VALUES DATA2=$ (18I4)*$
READ FORMAT DATA3,JX(1)....JX(N)
VECTOR VALUES DATA3=$ (8F9.3)*$
READ FORMAT DATA4,JY(1)....JY(N)
VECTOR VALUES DATA4=$ (8F9.3)*$
READ FORMAT DATA5,FXO(1)....FXO(N)
VECTOR VALUES DATA5=$ (8F9.2)*$
    
```

A MAD Program for Truss Analysis

MAD Program, Continued

```

READ FORMAT DATA6,FYO(1)...FYO(N)
VECTOR VALUES DATA6=$(8F9.2)*$
READ FORMAT DATA7,UNMB(1)...UNMB(N)
VECTOR VALUES DATA7=$(18I4)*$
READ FORMAT DATA11,MB(1)...MB(M)
VECTOR VALUES DATA11=$(18I4)*$
READ FORMAT DATA12,JN(1)...JN(M)
VECTOR VALUES DATA12=$(18I4)*$
READ FORMAT DATA13,JF(1)...JF(M)
VECTOR VALUES DATA13=$(18I4)*$
PRINT FORMAT ECHO1,N,M,EPS1
VECTOR VALUES ECHO1=$1H1,S7,16HNO. OF JOINTS = ,I2,S4,
1 17HNO. OF MEMBERS = ,I3,S5,7HEPS1 = ,F8.5/*$
WHENEVER N.G.5
THROUGH L1,FOR I=1, 5,I.G.N
WHENEVER (I+4).GE.N
TRANSFER TO A1
OTHERWISE
PRINT FORMAT ECHO2,JT(I)...JT(I+4)
VECTOR VALUES ECHO2=$1H0,S7,2HJT,S2, 5I10/*$
PRINT FORMAT ECHO3,JX(I)...JX(I+4)
VECTOR VALUES ECHO3=$S8,2HJX,S5, 5F10.3/*$
PRINT FORMAT ECHO4,JY(I)...JY(I+4)
VECTOR VALUES ECHO4=$S8,2HJY,S5, 5F10.3/*$
PRINT FORMAT ECHO5,FXO(I)...FXO(I+4)
VECTOR VALUES ECHO5=$S8,3HFXO,S4, 5F10.2/*$
PRINT FORMAT ECHO6,FYO(I)...FYO(I+4)
VECTOR VALUES ECHO6=$S8,3HFYO,S4, 5F10.2/*$
PRINT FORMAT ECHO7,UNMB(I)...UNMB(I+4)
VECTOR VALUES ECHO7=$S8,4HUNMB, 5I10/*$
L1 END OF CONDITIONAL
OTHERWISE
I=1
END OF CONDITIONAL
A1 PRINT FORMAT EECHO2, JT(I)...JT(N)
VECTOR VALUES EECHO2=$S8,2HJT,S2, 5I10/*$
PRINT FORMAT EECHO3,JX(I)...JX(N)
VECTOR VALUES EECHO3=$S8,2HJX,S5, 5F10.3/*$
PRINT FORMAT EECHO4,JY(I)...JY(N)
VECTOR VALUES EECHO4=$S8,2HJY,S5, 5F10.3/*$
PRINT FORMAT EECHO5,FXO(I)...FXO(N)
VECTOR VALUES EECHO5=$S8,3HFXO,S4, 5F10.2/*$
PRINT FORMAT EECHO6,FYO(I)...FYO(N)
VECTOR VALUES EECHO6=$S8,3HFYO,S4, 5F10.2/*$
PRINT FORMAT EECHO7,UNMB(I)...UNMB(N)
VECTOR VALUES EECHO7=$S8,4HUNMB, 5I10/*$
WHENEVER M.G.5
THROUGH L2,FOR J=1, 5,J.G.M
WHENEVER (J+4).GE.M
TRANSFER TO A2
OTHERWISE
PRINT FORMAT ECHO11,MB(J)...MB(J+4)
VECTOR VALUES ECHO11=$1H0,S7,2HMB,S2, 5I10/*$
PRINT FORMAT ECHO12,JN(J)...JN(J+4)
VECTOR VALUES ECHO12=$S8,2HJN,S2, 5I10/*$
PRINT FORMAT ECHO13,JF(J)...JF(J+4)
VECTOR VALUES ECHO13=$S8,2HJF,S2, 5I10/*$
L2 END OF CONDITIONAL
OTHERWISE
J=1
END OF CONDITIONAL
A2 PRINT FORMAT EECO11,MB(J)...MB(M)
VECTOR VALUES EECO11=$1H0,S7,2HMB,S2, 5I10/*$
PRINT FORMAT EECO12,JN(J)...JN(M)
VECTOR VALUES EECO12=$S8,2HJN,S2, 5I10/*$
PRINT FORMAT EECO13,JF(J)...JF(M)
VECTOR VALUES EECO13=$S8,2HJF,S2, 5I10/*$
READ FORMAT RO,JRO
VECTOR VALUES RO=$I4*$
READ FORMAT RA,R1,JR1,LXR1,LYR1,LR1
VECTOR VALUES RA=$I4,S2,I4,3F10.5*$
PRINT FORMAT RRO,JRO
VECTOR VALUES RRO=$S8,4HJRO ,I3/*$

```

MAD Program, Continued

```

PRINT FORMAT RR1,JR1,LXR1,LYR1,LR1
VECTOR VALUES RR1=$S8,4HJR1 ,I3/S8,4HLXR1,S3,F10.3,
1 S5,5HLYR1 ,F10.3,S5,5HLR1 ,F10.3/*$
THROUGH L3,FOR J=1,1,J.G.M
LX(J)=JX(JF(J))-JX(JN(J))
LY(J)=JY(JF(J))-JY(JN(J))
L3 L(J)=SQRT.(LX(J)*LX(J)+LY(J)*LY(J))
SFX=0
SFY=0
SMO=0
THROUGH LN1,FOR I=1,1,I.G.N
SFX=SFX+FXO(I)
SFY=SFY+FYO(I)
LN1 SMO=SMO+(JX(I)-JX(JRO))*FYO(I)-(JY(I)-JY(JRO))*FXO(I)
FR(R1)=SMO*LR1/((JY(JR1)-JY(JRO))*LXR1
1 -(JX(JR1)-JX(JRO))*LYR1)
FXR1=FR(R1)*LXR1/LR1
FYR1=FR(R1)*LYR1/LR1
FXO(JR1)=FXO(JR1)+FXR1
FYO(JR1)=FYO(JR1)+FYR1
FXRO=-(SFX+FXR1)
FYRO=-(SFY+FYR1)
FXO(JRO)=FXO(JRO)+FXRO
FYO(JRO)=FYO(JRO)+FYRO
PRINT FORMAT ANS4,R1,FR(R1),FXR1,FYR1,FXRO,FYRO
VECTOR VALUES ANS4=$1H0,S7,3HFR(,I2,1H),S1,F10.2,S5,4HFXR1,
1 S1,F10.2,S5,4HFYR1,S1,F10.2/S8,4HFXRO,S3,F10.2,S5,4HFYRO,S1,
2F10.2/*$
PRINT FORMAT ANS5,JRO,FXO(JRO),JRO,FYO(JRO),
1 JR1,FXO(JR1),JR1,FYO(JR1)
VECTOR VALUES ANS5=$S8,4HFXO(,I2,1H),F10.2,S3,4HFYO(,I2,1H),
1F10.2/S8,4HFXO(,I2,1H),F10.2,S3,4HFYO(,I2,1H),F10.2*$
Q=0
C3 THROUGH LL1,FOR I=1,1,I.G.N
LL1 WHENEVER UNMB(I).E.2,TRANSFER TO C1
C1 P=0
THROUGH LL2,FOR J=1,1,J.G.M
WHENEVER JN(J).E.JT(I)
P=P+1
IS(P)=JF(J)
MS(P)=J
JN(J)=9999
JF(J)=9999
WHENEVER P.E.2,TRANSFER TO C2
LL2 END OF CONDITIONAL
THROUGH LL3,FOR J=1,1,J.G.M
WHENEVER JF(J).E.JT(I)
P=P+1
IS(P)=JN(J)
MS(P)=J
JN(J)=9999
JF(J)=9999
LX(J)=-LX(J)
LY(J)=-LY(J)
WHENEVER P.E.2,TRANSFER TO C2
LL3 END OF CONDITIONAL
C2 UNMB(JT(I))=UNMB(JT(I))-2
UNMB(IS(1))=UNMB(IS(1))-1
UNMB(IS(2))=UNMB(IS(2))-1
JTS(MS(1))=JT(I)
JTS(MS(2))=JT(I)
D=LX(MS(1))*LY(MS(2))/(L(MS(1))*L(MS(2)))
1 -LY(MS(1))*LX(MS(2))/(L(MS(1))*L(MS(2)))
WHENEVER .ABS.(D).L.EPS1
PRINT FORMAT UNSTBL
VECTOR VALUES UNSTBL=$S8,22HTHIS TRUSS IS UNSTABLE*$
TRANSFER TO START
END OF CONDITIONAL

```

MAD Program, Continued

```

F(MS(1))=(-FXO(JT(I))*LY(MS(2))/L(MS(2)))
1 +FYO(JT(I))*LX(MS(2))/L(MS(2))/D
FX(MS(1))=F(MS(1))*LX(MS(1))/L(MS(1))
FY(MS(1))=F(MS(1))*LY(MS(1))/L(MS(1))
FXO(IS(1))=FXO(IS(1))-FX(MS(1))
FYO(IS(1))=FYO(IS(1))-FY(MS(1))
F(MS(2))=(FXO(JT(I))*LY(MS(1))/L(MS(1)))
1 -FYO(JT(I))*LX(MS(1))/L(MS(1))/D
FX(MS(2))=F(MS(2))*LX(MS(2))/L(MS(2))
FY(MS(2))=F(MS(2))*LY(MS(2))/L(MS(2))
FXO(IS(2))=FXO(IS(2))-FX(MS(2))
FYO(IS(2))=FYO(IS(2))-FY(MS(2))
Q=Q+2
WHENEVER (M-Q).G.1,TRANSFER TO C3
WHENEVER (M-Q).E.0,TRANSFER TO C4
THROUGH LL4,FOR I=1,1,I.G.N
LL4  WHENEVER UNMB(I).E.1,TRANSFER TO C5
C5  THROUGH LL5,FOR J=1,1,J.G.M
LL5  WHENEVER JN(J).E.JT(I),TRANSFER TO C6
THROUGH LL6,FOR J=1,1,J.G.M
WHENEVER JF(J).E.JT(I)
LX(J)=-LX(J)
LY(J)=-LY(J)
TRANSFER TO C6
LL6  END OF CONDITIONAL
C6  FX(J)=-FXO(I)
FY(J)=-FYO(I)
JTS(J)=JT(I)
WHENEVER .ABS.(FY(J)).GE..ABS.(FX(J))
F(J)=FY(J)*L(J)/LY(J)
OTHERWISE
F(J)=FX(J)*L(J)/LX(J)
END OF CONDITIONAL
C4  WHENEVER M.G.5
THROUGH L4,FOR J=1, 5,J.G.M
WHENEVER (J+4).GE.M
TRANSFER TO A3
OTHERWISE
PRINT FORMAT ECHO11,MB(J)...MB(J+4)
PRINT FORMAT ANS1,LX(J)...LX(J+4)
VECTOR VALUES ANS1=$S8,2HLX,S5, 5F10.3/*$
PRINT FORMAT ANS2,LY(J)...LY(J+4)
VECTOR VALUES ANS2=$S8,2HLY,S5, 5F10.3/*$
PRINT FORMAT ANS3,L(J)...L(J+4)
VECTOR VALUES ANS3=$S8,1HL,S6, 5F10.3/*$
PRINT FORMAT ANS11,JTS(J)...JTS(J+4)
VECTOR VALUES ANS11=$S8,3HJTS,S1, 5I10/*$
PRINT FORMAT ANS12,FX(J)...FX(J+4)
VECTOR VALUES ANS12=$S8,2HFX,S5, 5F10.2/*$
PRINT FORMAT ANS13,FY(J)...FY(J+4)
VECTOR VALUES ANS13=$S8,2HFY,S5, 5F10.2/*$
PRINT FORMAT ANS14,F(J)...F(J+4)
VECTOR VALUES ANS14=$S8,1HF,S6, 5F10.2/*$
L4  END OF CONDITIONAL
OTHERWISE
J=1
END OF CONDITIONAL
A3  PRINT FORMAT EECO11,MB(J)...MB(M)
PRINT FORMAT AANS1,LX(J)...LX(M)
VECTOR VALUES AANS1=$S8,2HLX,S5, 5F10.3/*$
PRINT FORMAT AANS2,LY(J)...LY(M)
VECTOR VALUES AANS2=$S8,2HLY,S5, 5F10.3/*$
PRINT FORMAT AANS3,L(J)...L(M)
VECTOR VALUES AANS3=$S8,1HL,S6, 5F10.3*$
PRINT FORMAT AANS11,JTS(J)...JTS(M)
VECTOR VALUES AANS11=$S8,3HJTS,S1, 5I10/*$
PRINT FORMAT AANS12,FX(J)...FX(M)
VECTOR VALUES AANS12=$S8,2HFX,S5, 5F10.2/*$
PRINT FORMAT AANS13,FY(J)...FY(M)
VECTOR VALUES AANS13=$S8,2HFY,S5, 5F10.2/*$
PRINT FORMAT AANS14,F(J)...F(M)
VECTOR VALUES AANS14=$S8,1HF,S6, 5F10.2*$
TRANSFER TO START
END OF PROGRAM

```

Example Problem No. 82

Data

```

SDATA
  9 15 0.001
  1 2 3 4 5 6 7 8 9
    0. 30. 60. 90. 120. 120. 90. 60.
    30. 0. 0. 0. 0. 60. 40. 40.
    40. 0. 0. 0. 0. 0. 0. 0.
    0. 0. 0. 0. 0. 0. 0. 0.
   -10. -10. -10. -10. 0. 0. 0. 0.
    0.
  2 3 5 3 3 2 5 3 4
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  1 2 3 4 5 6 7 8 9 2 3 3 3 4 5
  2 3 4 5 6 7 8 9 1 9 9 8 7 7 7
  5
  1 6 1.0 0.0 1.0
  
```

Computer Output

```

NO. OF JOINTS = 9      NO. OF MEMBERS = 15      EPS1 = 0.00100

JT      1      2      3      4      5
JX      0.000  30.000  60.000  90.000  120.000
JY      0.000  0.000  0.000  0.000  0.000
FXO     0.00  0.00  0.00  0.00  0.00
FYO    -10.00 -10.00 -10.00 -10.00  0.00
UNMB    2      3      5      3      3
JT      6      7      8      9
JX     120.000  90.000  60.000  30.000
JY     60.000  40.000  40.000  40.000
FXO     0.00  0.00  0.00  0.00
FYO     0.00  0.00  0.00  0.00
UNMB    2      5      3      4

MB      1      2      3      4      5
JN      1      2      3      4      5
JF      2      3      4      5      6

MB      6      7      8      9      10
JN      6      7      8      9      2
JF      7      8      9      1      9

MB      11     12     13     14     15
JN      3      3      3      4      5
JF      9      8      7      7      7
JRO    5
JR1    6
LXR1   1.000  LYR1   0.000  LR1    1.000

FR( 1)  50.00  FXR1   50.00  FYR1   0.00
FXRO   -50.00  FYRO   40.00
FXO( 5) -50.00  FYO( 5) 40.00
FXO( 6)  50.00  FYO( 6)  0.00

MB      1      2      3      4      5
LX     30.000  30.000 -30.000 -30.000 -0.000
LY      0.000  0.000 -0.000 -0.000 -60.000
L      30.000  30.000  30.000  30.000  60.000
JTS    1      2      4      5      6
FX     -7.50  -7.50  45.00  45.00  0.00
FY     -0.00  -0.00  0.00  0.00  33.33
F      -7.50  -7.50 -45.00 -45.00 -33.33

MB      6      7      8      9      10
LX    -30.000 -30.000 -30.000  30.000  0.000
LY    -20.000  0.000  0.000  40.000  40.000
L     36.056  30.000  30.000  50.000  40.000
JTS    6      7      8      1      2
FX    -50.00  -22.50 -22.50  7.50  0.00
FY    -33.33  0.00  0.00  10.00  10.00
F      60.09  22.50  22.50  12.50  10.00

MB      11     12     13     14     15
LX    -30.000  0.000 -30.000  0.000 -30.000
LY     40.000  40.000 -40.000  40.000  40.000
L     50.000  40.000  50.000  40.000  50.000
JTS    3      3      7      4      5
FX     15.00  0.00  -22.50  0.00  5.00
FY    -20.00  0.00  -30.00  10.00 -6.67
F     -25.00  0.00  37.50  10.00 -8.33
  
```

Critique

In the writer's opinion, the program is well suited for instructive purposes. The problem can be separated into three parts. In the first part, the students can be asked to write a program (or draw a flow diagram) for computing the lengths and components of the length vector of the members. In the second part, the students can be asked to write a program for computing the reactions and modifying the external forces at the joints where the reactions are applied. There are also possibilities of writing the program for computing length or reactions or both as an external function.* The students may be asked to write a program for computing three reactions along three given lines of action and applied at three different joints of the truss.* This program, if it is written as an external function, may be applicable to trusses which can be solved only by method of sections.*

Then the students can be asked to write a program for computing two unknown forces in members meeting at a joint using the method of joints and modifying the external forces applying at the joint at the other ends of the members. The process of searching for the joint to be solved and updating the various items involved is a rather complicated procedure. Detailed instructions as given in the text seem to be necessary. However, this complicated procedure is very instructive since it dramatically illustrates the computer's ability to handle problems which are essentially logical rather than computational. Often a human being is scarcely aware of the logical detail which is involved where he solves a problem manually.

* Due to lack of time, these problems have not been worked out by the writer.

Example Problem No. 83
STIFFNESS FACTORS FOR A FLAT SLAB COLUMN

by

Robert B. Harris
Department of Civil Engineering
The University of Michigan

Course: Advanced Design of Structures Credit hours: 3 Level: Graduate

Statement of Problem

Estimate the slab thickness, drop panel thickness, and the column diameter for a typical panel in your flat slab structure. Use the ACI methods for this approximation.

Using these results along with the story height, study the effect of column capital size upon the stiffness coefficients of the column and the carry-over factors.

Program your solution for the digital computer using the MAD language.

Solution

Stiffness coefficients for members with variable cross sections may be found by expressing the relationship between unit end moments applied to a member and the rotations of the end tangents. If the ends are defined by a and b respectively, and any end moments M_{ab} and M_{ba} are applied simultaneously, the rotation of the end tangents θ_a and θ_b must be, according to reference 1:

$$\theta_a = M_{ab}\beta_{a1} - M_{ba}\beta_{a2}$$

$$\theta_b = -M_{ab}\beta_{b1} + M_{ba}\beta_{b2}$$

β_{a1} and β_{b1} are the angle changes due to a unit moment applied to end a and β_{a2} and β_{b2} are the angle changes due to a unit moment applied at end b. By the reciprocal theorem, angle β_{b1} is equal to β_{a2} .

Solving the above equations for M_{ab} and M_{ba} :

$$M_{ab} = \frac{\beta_{b2}}{A} \theta_a + \frac{\beta_{b1}}{A} \theta_b$$

$$M_{ba} = \frac{\beta_{b1}}{A} \theta_a + \frac{\beta_{a1}}{A} \theta_b$$

Stiffness Factors for a Flat Slab Column

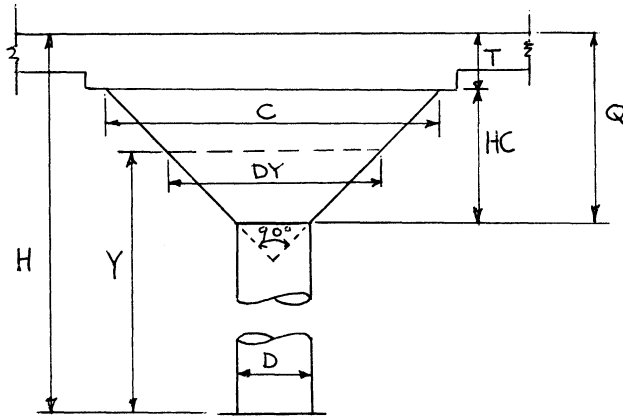
in which

$$A = \beta_{a1}\beta_{b2} - (\beta_{b1})^2$$

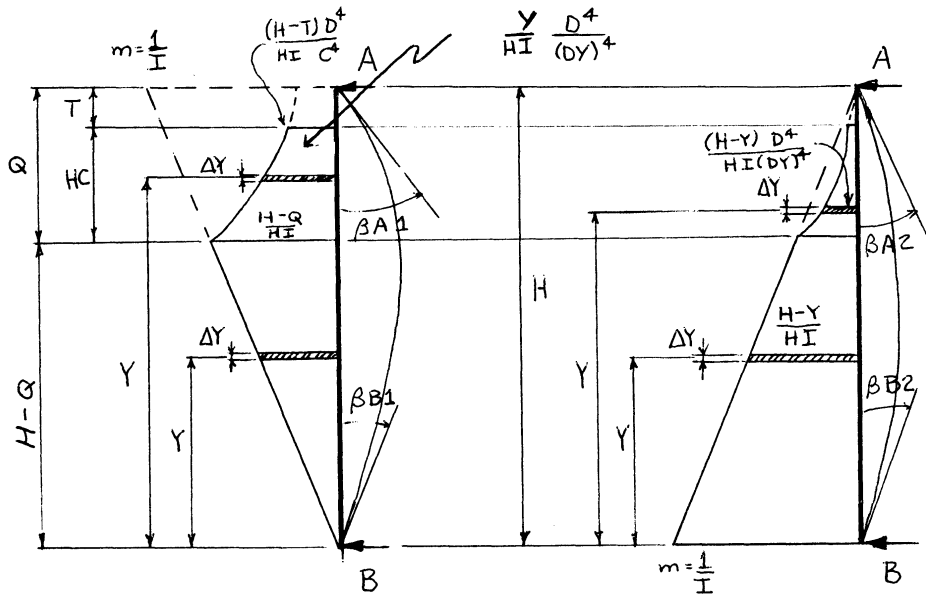
If I is the minimum moment of inertia then $K = \frac{EI}{L}$ and:

$$c_1 = \frac{\beta_{a1}}{KA} ; \quad c_2 = \frac{\beta_{b1}}{KA} ; \quad c_3 = \frac{\beta_{b2}}{KA}$$

The following sketch defines lengths used.



The following sketch shows the conjugate beams loaded with the appropriate $\frac{M}{EI}$ diagrams from which the values of β_{a1} , β_{b1} , and β_{b2} can be found.



Example Problem No. 83

Assuming circular elements in the capital, the moment of inertia varies as the fourth power of the diameter; therefore

$$\frac{I(H-Q)}{I(H-T)} \approx \frac{D^4}{C^4} \quad \text{and} \quad \frac{I(H-Q)}{I(Y)} \approx \frac{D^4}{(DY)^4}$$

In order to control the accuracy of the solution a variable was introduced called DEL which represents the size of increment ΔY . This was assigned the value of 0.2 in the sample program. This was particularly necessary since rectilinear integration was used.

The following summation equations are the result of the application of the conjugate beam method and the letters appearing in brackets below the various terms are the letters used in the program for that quantity.

$$\begin{aligned}
 H^2EI\theta A1 &= \underbrace{\sum_{n=\frac{\Delta Y}{2}}^{n=H-Q}}_{A1} Y_n^2 \Delta Y + D^4 \underbrace{\sum_{n=(H-Q+\frac{\Delta Y}{2})}^{n=H-T}}_A \frac{Y_n^2}{(DY)^4} \Delta Y && B \\
 H^2EI\theta B1 &= \underbrace{\sum_{n=\frac{\Delta Y}{2}}^{n=H-Q}}_{B1} (H - Y_n) Y_n \Delta Y + D^4 \underbrace{\sum_{n=(H-Q+\frac{\Delta Y}{2})}^{n=H-T}}_G \frac{(H - Y_n) Y_n}{(DY)^4} \Delta Y && K \\
 H^2EI\theta B2 &= \underbrace{\sum_{n=\frac{\Delta Y}{2}}^{n=H-Q}}_{B2} (H - Y_n)^2 \Delta Y + D^4 \underbrace{\sum_{n=(H-Q+\frac{\Delta Y}{2})}^{n=H-T}}_E \frac{(H - Y_n)^2}{(DY)^4} \Delta Y && F
 \end{aligned}$$

Stiffness Factors for a Flat Slab Column

Letting $X = H^2EI$, then $\beta A1 = \frac{A1}{X}$

$$\beta B1 = \frac{B1}{X}$$

$$\beta B2 = \frac{B2}{X}$$

$$A = \beta A1\beta B2 - \beta B1^2 = \frac{[(A1)(B2) - B1^2]}{X^2} = \frac{U}{X^2}$$

where $U = (A1)(B2) - B1^2$

Coefficient at bottom, $C_1 = \frac{(A1)X^2}{XKU} = \frac{X}{K} \cdot \frac{A1}{U}$

$$\frac{X}{K} = \frac{H^2EI}{\frac{EI}{H}} = H^3$$

Hence, $C_1 = \frac{H^3A1}{U}$

Coefficient at top, $C_3 = \frac{H^3B2}{U}$

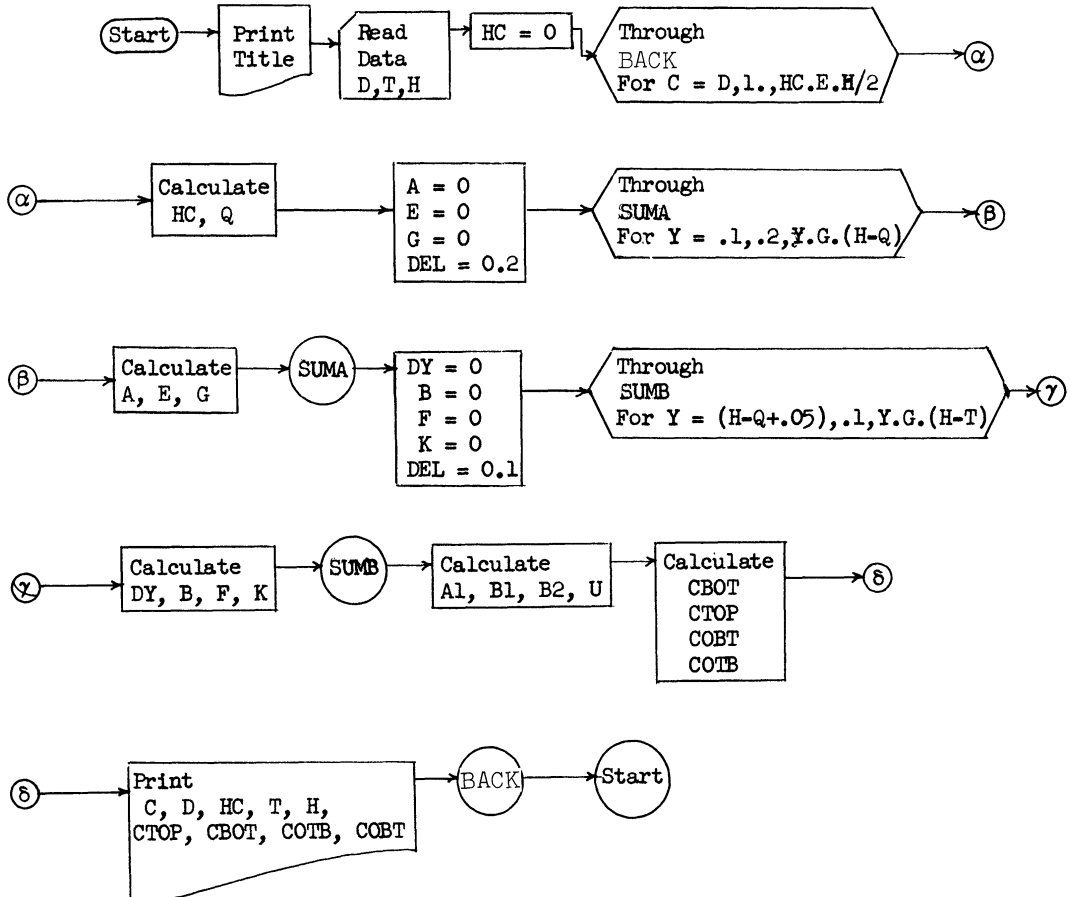
For the carry-over factors the coefficient C_2 is needed.

$$C_2 = \frac{H^3B1}{U}$$

Carry over - top to bottom = $\frac{B1}{B2}$

Carry over - bottom to top = $\frac{B1}{A1}$

Flow Diagram



MAD Program and Data

```

R. B. HARRIS          S164E          003   020   000
R. B. HARRIS          S164E          003   020   000
$COMPILE MAD, EXECUTE
R
R PROGRAM FOR STIFFNESS COEFFICIENTS FOR A COLUMN WITH 90 DE
1GREE CAPITALS
R
START PRINT FORMAT TITLE
R
R HEADINGS
R
VECTOR VALUES TITLE=$1H1,S23,26HCOLUMN STIFFNESS STUDY
1 //S26,19H90 DEGREE CAPITAL ///S5,3HCAP,S3,3HCOL,S4,3HCAP,
2S3,4HSLAB,S5,3HCOL,S4,3HTOP,S6,3HBOT,S5,5HCARRY,S4,5HCARRY/
3S5,4HDIAM,S2,4HDIAM,S2,5HDEPTH,S2,5HTHICK,S2,6HLENGTH,S2,
45HCOEFF,S4,5HCOEFF,S3,7HTOP-BOT,S2,7HBOT-TOP*$
R
R INPUT
R
READ FORMAT DATA, D, T, H
VECTOR VALUES DATA = $3F10.3*$
HC = 0
THROUGH BACK, FOR C = D,1.,HC.E.H/2
HC = C/2 - D/2
Q = T + HC
R
R INTEGRATING STRAIGHT LINE PORTION
R
A = 0.0
E = 0.0
G = 0.0
DEL = 0.2
THROUGH SUMA, FOR Y = .1,.2,Y.G.(H-Q)
A = A + (Y.P.2)*DEL
E = E + ((H-Y).P.2)*DEL
SUMA G = G + ((H-Y)*Y)*DEL
R
R INTEGRATING CURVED PORTION
R
DY = 0.0
B = 0.0
F = 0.0
K = 0.0
DEL = 0.1
THROUGH SUMB, FOR Y = (H-Q+.05),.1,Y.G.(H-T)
DY = DY + D + 2*(C-D)*(Q-H+Y)/HC
B = B + (D.P.4)*DEL*Y.P.2/DY.P.4
SUMB F = F + (D.P.4)*DEL*(H-Y).P.2/DY.P.4
K = K + (D.P.4)*DEL*(H-Y)*Y/DY.P.4
R
R STIFFNESS FACTORS AND OUTPUT
R
A1 = A+B
B1 = G+K
B2 = E+F
U = A1*B2-B1.P.2
CBOT = A1*H.P.3/U
CTOP = B2*H.P.3/U
COBT = B1/A1
COTB = B1/B2
BACK PRINT FORMAT COEFF, C, D, HC, T,H, CTOP,CBOT,COTB,COBT
VECTOR VALUES COEFF = $F8.1,4F7.1,4F9.4*$
TRANSFER TO START
END OF PROGRAM
$DATA
1.0      0.000      10.0
1.0      1.000      10.0

```

Stiffness Factors for a Flat Slab Column

Computer Output

COLUMN STIFFNESS STUDY								
90 DEGREE CAPITAL								
CAP DIAM	COL DIAM	CAP DEPTH	SLAB THICK	COL LENGTH	TOP COEFF	BOT COEFF	CARRY TOP-BOT	CARRY BOT-TOP
1.0	1.0	.0	.0	10.0	4.0012	4.0012	.5002	.5002
2.0	1.0	.5	.0	10.0	4.6159	4.1473	.4985	.5548
3.0	1.0	1.0	.0	10.0	5.9587	4.4210	.4878	.6575
4.0	1.0	1.5	.0	10.0	7.1344	4.6274	.4765	.7346
5.0	1.0	2.0	.0	10.0	9.4565	4.9705	.4538	.8634
6.0	1.0	2.5	.0	10.0	11.5429	5.2332	.4359	.9616
7.0	1.0	3.0	.0	10.0	15.7811	5.6761	.4056	1.1278
8.0	1.0	3.5	.0	10.0	19.7202	6.0218	.3838	1.2568
9.0	1.0	4.0	.0	10.0	28.0289	6.6153	.3492	1.4796
10.0	1.0	4.5	.0	10.0	36.1126	7.0907	.3253	1.6568
11.0	1.0	5.0	.0	10.0	54.0806	7.9271	.2889	1.9707

COLUMN STIFFNESS STUDY								
90 DEGREE CAPITAL								
CAP DIAM	COL DIAM	CAP DEPTH	SLAB THICK	COL LENGTH	TOP COEFF	BOT COEFF	CARRY TOP-BOT	CARRY BOT-TOP
1.0	1.0	.0	1.0	10.0	6.0930	4.4461	.4866	.6669
2.0	1.0	.5	1.0	10.0	7.1344	4.6274	.4765	.7346
3.0	1.0	1.0	1.0	10.0	9.4565	4.9705	.4538	.8634
4.0	1.0	1.5	1.0	10.0	11.5429	5.2332	.4359	.9616
5.0	1.0	2.0	1.0	10.0	15.7811	5.6761	.4056	1.1278
6.0	1.0	2.5	1.0	10.0	19.7202	6.0218	.3838	1.2568
7.0	1.0	3.0	1.0	10.0	28.0289	6.6153	.3492	1.4796
8.0	1.0	3.5	1.0	10.0	36.1126	7.0907	.3253	1.6568
9.0	1.0	4.0	1.0	10.0	54.0806	7.9271	.2889	1.9707
10.0	1.0	4.5	1.0	10.0	72.7273	8.6220	.2643	2.2293
11.0	1.0	5.0	1.0	10.0	117.4918	9.8889	.2276	2.7045

Discussion of Results

Capital diameter, column diameter, capital depth, slab thickness, and column length must all be put in using the same units. In the sample these are all in feet and are printed out with the results for checking. The program limits the capital depth to one-half the column length and increments by one-half a length unit.

Coefficients obtained give an accuracy of four significant figures. This accuracy may be improved by making the integration increments smaller in the SUMA and SUMB iterations.

Critique

This problem was selected as representative of the use of the computer in making repetitive computations which would take considerable time to do longhand. In a sense, the problem is synthetic. If, however, such information were available with the detail obtained, it would materially help in preliminary design and checking of flat slab construction.

The problem was designed to serve as an introduction to the use of a more complicated program prepared by the instructor, the use of which would allow the student to make adjustments in columns, slabs, drop panels, and capitals, and exercise the broadest judgement in design.

As an introduction, the student reaction was quite good and little difficulty was experienced in programming. About one week and a half (3 laboratory periods) were used in this activity, but actual results were not turned in for some time after this since mechanical delays with the Computing Center prolonged the submission date.

This problem served the purpose for which it was intended and in addition firmed up in the students' minds the effectiveness of the column capital in flat slab construction.

Reference

1. Maugh, L. C. Statically Indeterminate Structures, page 133.

Example Problem No. 84

VIBRATION OF BEAMS ON SPRING SUPPORTS

by

Shan S. Kuo

Department of Civil Engineering

Yale University

Course: Structural Dynamics Credit Hours: 2 Level: Senior, Graduate

Statement of the Problem

Write a computer program to determine the natural frequencies and the modal shapes for a beam on spring supports as shown in Figure 1. The beam is idealized

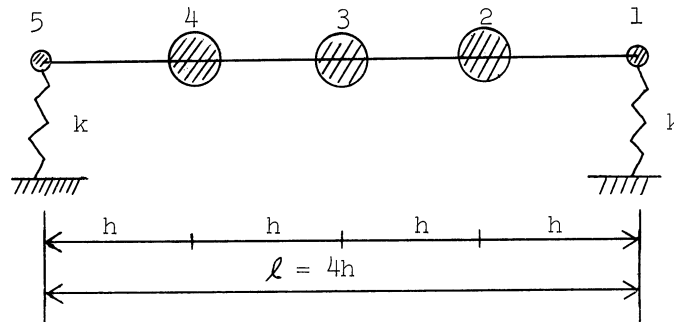


Figure 1. Beam on spring supports

as a multi-mass system. Each of the two end masses is taken as one half of an interior mass, and the sum of all masses is equal to the total mass of beam m . In order to study the effect of spring constant k , we take ω_R / ω_H as a parameter where

$$\omega_R = \sqrt{2k/m} = \text{"rigid body" frequency}$$

$$\omega_H = \pi^2 \sqrt{EI/ml^3} = \text{fundamental frequency of simply supported beam}$$

The program should be so written that any number of lumped masses can be handled.

Solution

We are concerned with the following matrix equation:

$$\frac{1}{\omega^2} X = FX$$

where

F = flexibility matrix due to the spring action and the beam action

M = mass matrix, a diagonal matrix

X = displacement of lumped mass

ω = frequency

It should be noted that the matrix FM is not symmetrical; therefore, Jacobi's method cannot be applied to this case directly. The following is a general procedure for solving this problem using a digital computer.

- (a) Formulate the flexibility matrix F. It is considered here as the sum of three action matrices, one from the beam action and the other two from the spring action. In connection with the beam action matrix, we take

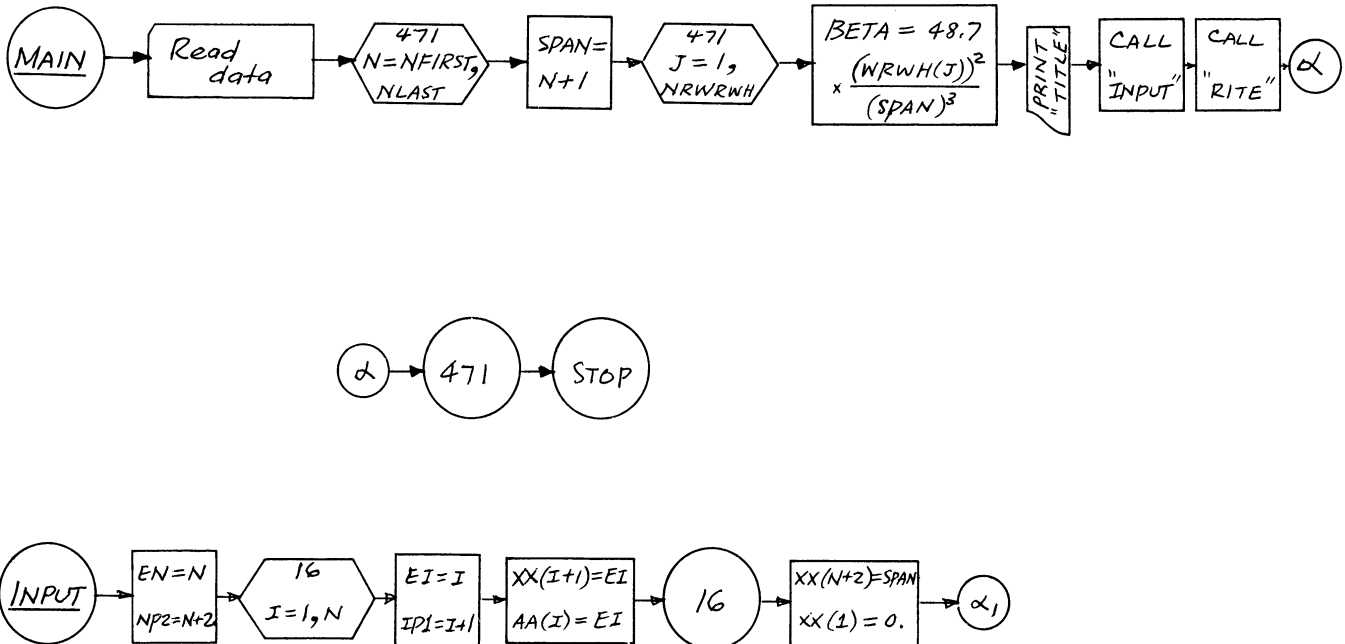
$$\beta = \frac{kh}{EI} = \frac{\pi^4}{2(n+1)^3} \left(\frac{\omega R}{\omega H}\right)^2$$

where n = number of interior masses, or number of total masses - 2.

- (b) Formulate the mass matrix M.
 (c) Formulate the matrix $M^{1/2} FM^{1/2}$
 (d) Make use of the Jacobi's method to find the eigenvalues and the eigenvectors of the matrix found in the step c.
 (e) The natural frequencies are then equal to (eigenvalues)^{-1/2}
 (f) The true eigenvectors are equal to $M^{-1/2}$ times the computed eigenvectors.

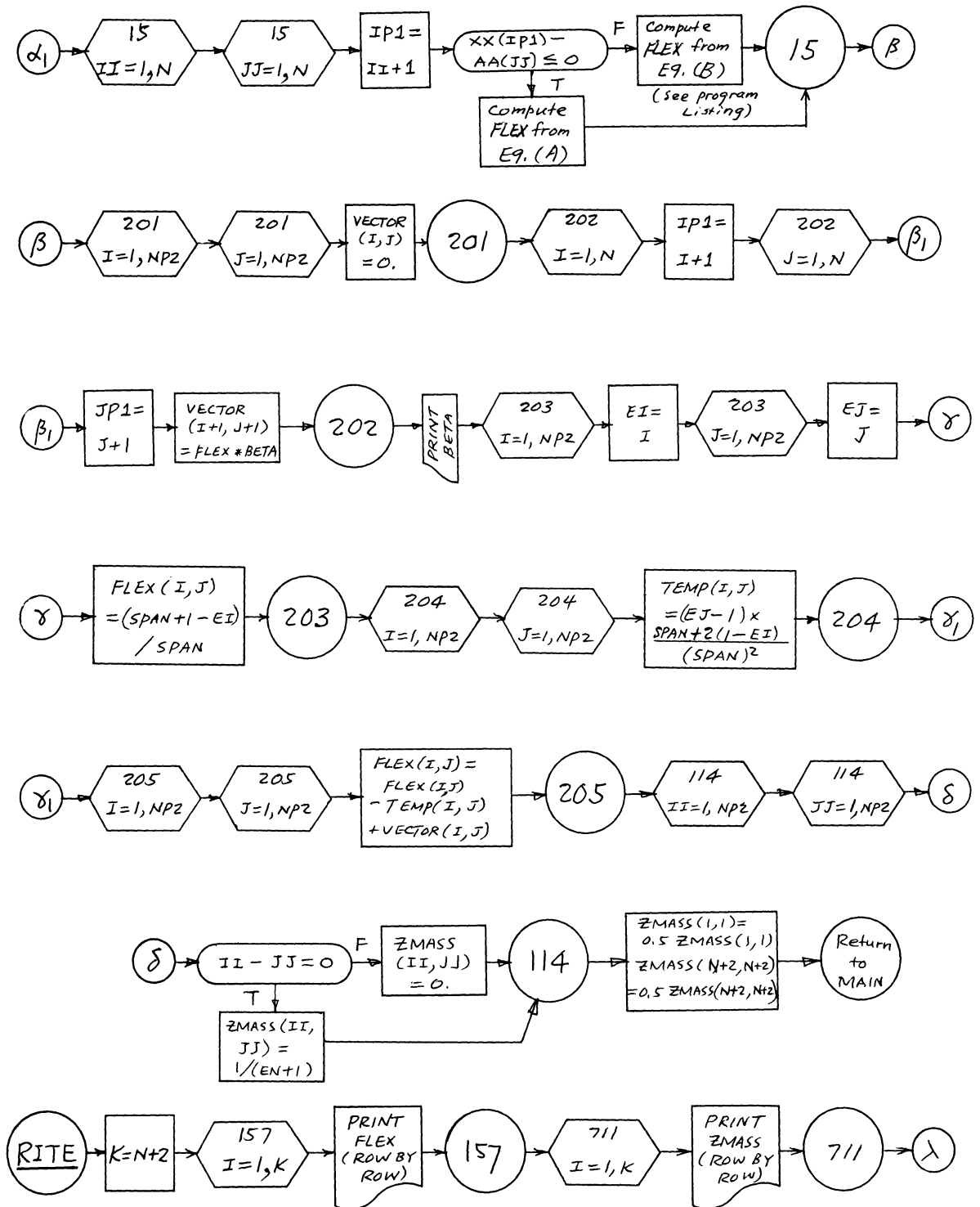
The steps (a) and (b) are incorporated in the subroutine "INPUT", whereas the steps (c) through (f) are carried out by the subroutine "RITE".

Flow Diagram

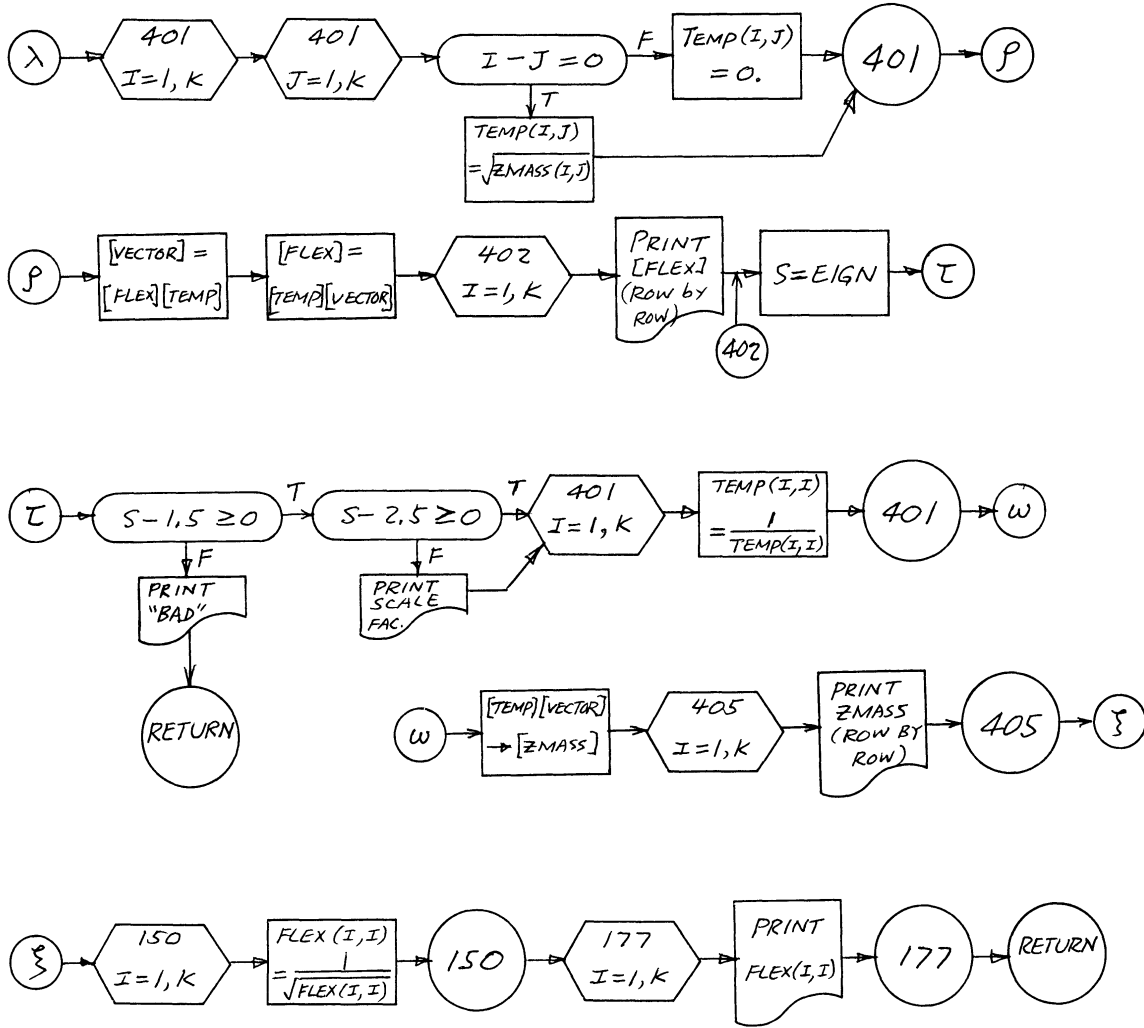


Vibration of Beams on Spring Supports

Flow Diagram, Continued



Flow Diagram, Continued



FORTRAN Programs and Data

```

SHAN S. KUO (YALE UNIV)      D003N  9          004  027  200      SPRING 6
SHAN S. KUO (YALE UNIV)      D003N  9          004  027  200      SPRING 6
$COMPILE FORTRAN, EXECUTE
C
C  VIBRATION OF BEAMS ON SPRING SUPPORTS  SHAN S. KUO  7/29/61
C
  DIMENSION FLEX(5,5),ZMASS(5,5),TEMP(5,5),VECTOR(5,5),
  1 XX(5),AA(5),WRWH(5)
  COMMON N,SPAN,BETA
C
C  NFIRST IS THE FIRST N AND NLAST IS THE LAST N
C
  READ INPUT TAPE 7,12, NFIRST, NLAST, NRWRWH
  12 FORMAT(3I2)
  READ INPUT TAPE 7,14, (WRWH(I1),I1=1,NRWRWH)
  14 FORMAT(6F6.3)
  DO 471 N=NFIRST, NLAST
    SPAN=N+1
    DO 471 J=1, NRWRWH
      BETA=48.70455*WRWH(J)*WRWH(J)/(SPAN*SPAN*SPAN)
      WRITE OUTPUT TAPE 6,223,WRWH(J)
      WRITE OUTPUT TAPE 5,223,WRWH(J)
  223 FORMAT(67H1VIBRATION OF BEAM ON SPRING SUPPORTS. A PROBLEM OF SHOC
  1K ISOLATION/64HORIGID BODY FREQUENCY / FUNDAMENTAL FREQ. OF HINGED
  2-HINGED BEAM=F5.2)
      CALL INPUT(FLEX,ZMASS,VECTOR,TEMP,XX,AA)
      CALL RITE (FLEX,ZMASS,VECTOR,TEMP)
  471 CONTINUE
  
```

```

$COMPILE FORTRAN
C
C INPUT MATRICES SHAN S. KUO 7/29/61 8/5/61
C
SUBROUTINE INPUT (FLEX,ZMASS,VECTOR,TEMP,XX,AA)
DIMENSION FLEX(5,5),ZMASS(5,5),VECTOR(5,5),XX(5),AA(5),TEMP(5,5)
COMMON N,SPAN,BETA
EN=N
NM1=N-1
NP1=N+1
NP2=N+2
N2=N*2
C
C FORM THE INFLUENCE MATRIX, II AND JJ FOR MASS NUMBERING
C
DO 16 I=1,N
EI=I
IP1=I+1
XX(IP1)=EI
16 AA(I)=EI
XX(NP2)=SPAN
XX(1)=0.
DO 15 II=1,N
DO 15 JJ=1,N
IP1=II+1
IF(XX(IP1)-AA(JJ)) 17,17,18
17 BB=SPAN-AA(JJ)
FLEX(II,JJ)=(BB/(6.*SPAN))*(XX(IP1)*XX(IP1)*XX(IP1)
1 -(SPAN*SPAN-BB*BB)*XX(IP1))*(-1.)
GO TO 15
18 BB=SPAN-AA(JJ)
FLEX(II,JJ)=-((BB/(6.*SPAN))*(XX(IP1)*XX(IP1)*XX(IP1)
1 -(SPAN*SPAN-BB*BB)*XX(IP1)
2 -(SPAN*(XX(IP1)-AA(JJ))*(XX(IP1)-AA(JJ))*(XX(IP1)-AA(JJ))/BB))
15 CONTINUE
C
C SHIFT FLEX MATRIX DUE TO BEAM ACTION. BORROW VECTOR
C
DO 201 I=1,NP2
DO 201 J=1,NP2
201 VECTOR(I,J)=0.
DO 202 I=1,N
IP1=I+1
DO 202 J=1,N
JP1=J+1
202 VECTOR(IP1,JP1)=(FLEX(I,J))*BETA
WRITE OUTPUT TAPE 6, 777, BETA
WRITE OUTPUT TAPE 5, 777, BETA
777 FORMAT(7HOBETA =F6.3)
C
C FLEX IS NOW ONE OF THE 3 MATRICES TO FORM THE FINAL MATRIX FLEX
C
DO 203 I=1,NP2
EI=I
DO 203 J=1,NP2
203 FLEX(I,J)=(SPAN+1.-EI)/SPAN
C
C BORROW TEMP, ONE OF THE 3 MATRICES TO FORM THE FINAL FLEX MATRIX
C
DO 204 I=1,NP2
EI=I
DO 204 J=1,NP2
EJ=J
204 TEMP (I,J)=(EJ-1.)*(SPAN+2.*(1.-EI))/(SPAN*SPAN)
C
C HAPPY FINAL FLEX
C
DO 205 I=1,NP2
DO 205 J=1,NP2
205 FLEX(I,J)=FLEX(I,J)-TEMP(I,J)+VECTOR(I,J)

```

FORTRAN Programs and Data, Continued

```

C
C   FORM MASS MATRIX
C
      DO 114 II=1, NP2
      DO 114 JJ=1, NP2
      IF (II-JJ) 112, 111, 112
111  ZMASS(II, JJ) = 1. / (EN + 1.)
      GO TO 114
112  ZMASS(II, JJ) = 0.
114  CONTINUE
      ZMASS(1, 1) = 0.5 * ZMASS(1, 1)
      ZMASS(NP2, NP2) = 0.5 * ZMASS(NP2, NP2)
      RETURN

$COMPILE FORTRAN
C
C   WRITE OUTPUT SHAN S. KUO 7/29/61
C
      SUBROUTINE RITE(FLEX, ZMASS, VECTOR, TEMP)
      DIMENSION FLEX(5, 5), ZMASS(5, 5), TEMP(5, 5), VECTOR(5, 5)
      COMMON N, SPAN, BETA
      K = N + 2
      DO 157 I = 1, K
      WRITE OUTPUT TAPE 5, 158, I, (FLEX(I, J), J = 1, K)
157  WRITE OUTPUT TAPE 6, 158, I, (FLEX(I, J), J = 1, K)
158  FORMAT(23H FLEXIBILITY MATRIX ROWI3/(1H 5F10.5))
      DO 701 I = 1, K
      WRITE OUTPUT TAPE 5, 711, I, (ZMASS(I, J), J = 1, K)
701  WRITE OUTPUT TAPE 6, 711, I, (ZMASS(I, J), J = 1, K)
711  FORMAT(16H MASS MATRIX ROWI3/(1H 5F9.5))
C
C   FORM TEMP MATRIX = SQ. RT. OF MASS MATRIX
C
      DO 401 I = 1, K
      DO 401 J = 1, K
      IF (I-J) 412, 411, 412
C   IN THE UNIV. OF MICHIGAN USE SQRT NOT SQRTF
411  TEMP(I, J) = SQRT (ZMASS(I, I))
      GO TO 401
412  TEMP(I, J) = 0.
401  CONTINUE
C
C   FORM H MATRIX = TEMP TIMES FLEX TIMES TEMP
C   STORE FLEX TIMES TEMP INTO VECTOR ( TEMPORARY USE)
C
      CALL MATMPY(FLEX, K, K, TEMP, K, VECTOR)
      CALL MATMPY(TEMP, K, K, VECTOR, K, FLEX)
      DO 402 I = 1, K
      WRITE OUTPUT TAPE 5, 403, I, (FLEX(I, J), J = 1, K)
402  WRITE OUTPUT TAPE 6, 403, I, (FLEX(I, J), J = 1, K)
403  FORMAT(13H H MATRIX ROWI3/(1H 5F10.5))
C
C   EIGN IS A SUBROUTINE AVAILABLE IN THE UNIVERSITY OF MICHIGAN C.C.
C
      S = EIGN(FLEX, K, 1, VECTOR(1, 1), F)
      IF (S - 1.5) 13, 13, 14
13  WRITE OUTPUT TAPE 6, 12, S
12  FORMAT(15HOBAD RETURN S = F10.6)
      GO TO 777
14  IF (S - 2.5) 15, 15, 16
15  WRITE OUTPUT TAPE 6, 11, F
11  FORMAT(15HOSCALE FACTOR = F10.6)
C
C   GET TRUE VECTORS
C
16  DO 404 I = 1, K
404  TEMP(I, I) = 1. / (TEMP(I, I))
C
C   USE MASS AS A TEMPORARY STORAGE
C
      CALL MATMPY(TEMP, K, K, VECTOR, K, ZMASS)
      DO 405 I = 1, K
      WRITE OUTPUT TAPE 5, 406, I, (ZMASS(I, J), J = 1, K)
405  WRITE OUTPUT TAPE 6, 406, I, (ZMASS(I, J), J = 1, K)
406  FORMAT(16H TRUE VECTOR ROWI3/(1H 5F10.5))

```

FORTTRAN Programs and Data, Continued

```

C
C TRANSFORM THE EIGENVALUES TO FREQUENCIES
C
      DO 150 I=1,K
150 FLEX(I,I)=SQRT (1./FLEX(I,I))
      DO 177 I=1,K
        WRITE OUTPUT TAPE 5,178,I,FLEX(I,I)
177 WRITE OUTPUT TAPE 6,178,I,FLEX(I,I)
178 FORMAT(28H FREQUENCY IN SQ. RT. OF K/MI6/(1H F30.6))
177 RETURN
C
C MATRIX MULTIPLICATION
C
      SUBROUTINE MATMPY(A,N,M,B,L,C)
      DIMENSION A(5,5),B(5,5),C(5,5)
      DO 75 I=1,N
      DO 75 J=1,L
      C(I,J)=0.
      DO 75 K=1,M
75 C(I,J)=C(I,J)+A(I,K)*B(K,J)
      RETURN

$ DATA
030302
0.5 4.
    
```

Computer Output

```

VIBRATION OF BEAM ON SPRING SUPPORTS. A PROBLEM OF SHOCK ISOLATION
RIGID BODY FREQUENCY / FUNDAMENTAL FREQ. OF HINGED-HINGED BEAM= 0.50

BETA = 0.190
FLEXIBILITY MATRIX ROW 1
  1.00000  0.75000  0.50000  0.25000  0.
FLEXIBILITY MATRIX ROW 2
  0.75000  0.76769  0.67440  0.48598  0.25000
FLEXIBILITY MATRIX ROW 3
  0.50000  0.67440  0.75367  0.67440  0.50000
FLEXIBILITY MATRIX ROW 4
  0.25000  0.48598  0.67440  0.76769  0.75000
FLEXIBILITY MATRIX ROW 5
  0.         0.25000  0.50000  0.75000  1.00000
MASS MATRIX ROW 1
  0.12500  0.         0.         0.         0.
MASS MATRIX ROW 2
  0.         0.25000  0.         0.         0.
MASS MATRIX ROW 3
  0.         0.         0.25000  0.         0.
MASS MATRIX ROW 4
  0.         0.         0.         0.25000  0.
MASS MATRIX ROW 5
  0.         0.         0.         0.         0.12500
H MATRIX ROW 1
  0.12500  0.13258  0.08839  0.04419  0.
H MATRIX ROW 2
  0.13258  0.19192  0.16860  0.12150  0.04419
H MATRIX ROW 3
  0.08839  0.16860  0.18842  0.16860  0.08839
H MATRIX ROW 4
  0.04419  0.12150  0.16860  0.19192  0.13258
H MATRIX ROW 5
  0.         0.04419  0.08839  0.13258  0.12500
TRUE VECTOR ROW 1
  1.60933  -1.18746  -1.71950  0.83396  0.58978
TRUE VECTOR ROW 2
  0.83966  1.13797  0.16622  1.02312  -0.96208
TRUE VECTOR ROW 3
  -0.00000  -0.00000  0.99402  1.10045  1.34199
TRUE VECTOR ROW 4
  -0.83966  -1.13797  0.16622  1.02312  -0.96208
TRUE VECTOR ROW 5
  -1.60933  1.18746  -1.71950  0.83396  0.58978
    
```

Computer Output, Continued

```

FREQUENCY IN SQ. RT. OF K/M      1
                                2.292843
-----
FREQUENCY IN SQ. RT. OF K/M      2
                                13.855160
-----
FREQUENCY IN SQ. RT. OF K/M      3
                                5.915964
-----
FREQUENCY IN SQ. RT. OF K/M      4
                                1.294614
-----
FREQUENCY IN SQ. RT. OF K/M      5
                                24.901564
-----

VIBRATION OF BEAM ON SPRING SUPPORTS. A PROBLEM OF SHOCK ISOLATION
-----
RIGID BODY FREQUENCY / FUNDAMENTAL FREQ. OF HINGED-HINGED BEAM= 4.00
-----
BETA =12.176
-----
FLEXIBILITY MATRIX ROW 1
  1.00000  0.75000  0.50000  0.25000  0.
-----
FLEXIBILITY MATRIX ROW 2
  0.75000  9.75710  11.66146  7.47775  0.25000
-----
FLEXIBILITY MATRIX ROW 3
  0.50000  11.66146  16.73485  11.66146  0.50000
-----
FLEXIBILITY MATRIX ROW 4
  0.25000  7.47775  11.66146  9.75710  0.75000
-----
FLEXIBILITY MATRIX ROW 5
  0.        0.25000  0.50000  0.75000  1.00000
-----
MASS MATRIX ROW 1
  0.12500  0.        0.        0.        0.
-----
MASS MATRIX ROW 2
  0.        0.25000  0.        0.        0.
-----
MASS MATRIX ROW 3
  0.        0.        0.25000  0.        0.
-----
MASS MATRIX ROW 4
  0.        0.        0.        0.25000  0.
-----
MASS MATRIX ROW 5
  0.        0.        0.        0.        0.12500
-----
H MATRIX ROW 1
  0.12500  0.13258  0.08839  0.04419  0.
-----
H MATRIX ROW 2
  0.13258  2.43928  2.91536  1.86944  0.04419
-----
H MATRIX ROW 3
  0.08839  2.91536  4.18371  2.91536  0.08839
-----
H MATRIX ROW 4
  0.04419  1.86944  2.91536  2.43928  0.13258
-----
H MATRIX ROW 5
  0.        0.04419  0.08839  0.13258  0.12500
-----
TRUE VECTOR ROW 1
  1.96434  0.37601  0.05178  1.37979  1.44690
-----
TRUE VECTOR ROW 2
  -0.26588  1.38900  1.00732  0.70010  -0.70368
-----
TRUE VECTOR ROW 3
  -0.00000  -0.00000  1.40283  -1.05636  0.95717
-----
TRUE VECTOR ROW 4
  0.26588  -1.38900  1.00732  0.70010  -0.70368
-----
TRUE VECTOR ROW 5
  -1.96434  -0.37601  0.05178  1.37979  1.44690
-----
FREQUENCY IN SQ. RT. OF K/M      1
                                3.041763
-----
FREQUENCY IN SQ. RT. OF K/M      2
                                1.305481
-----
FREQUENCY IN SQ. RT. OF K/M      3
                                0.345544
-----
FREQUENCY IN SQ. RT. OF K/M      4
                                2.530636
-----
FREQUENCY IN SQ. RT. OF K/M      5
                                3.407844
-----

```

Vibration of Beams on Spring Supports

Discussion of Results

The natural frequencies are equal to the computed values times the square root of k/m . The modal shapes are normalized in the subroutine BIGN according to the following relation:

$$\sum_{i=1}^{n+2} m_{ij} \phi_{ij}^2 = m \quad j = 1, 2, \dots, n+2$$

Figure 2 shows the computed normal modes for two different ω_R/ω_H values.

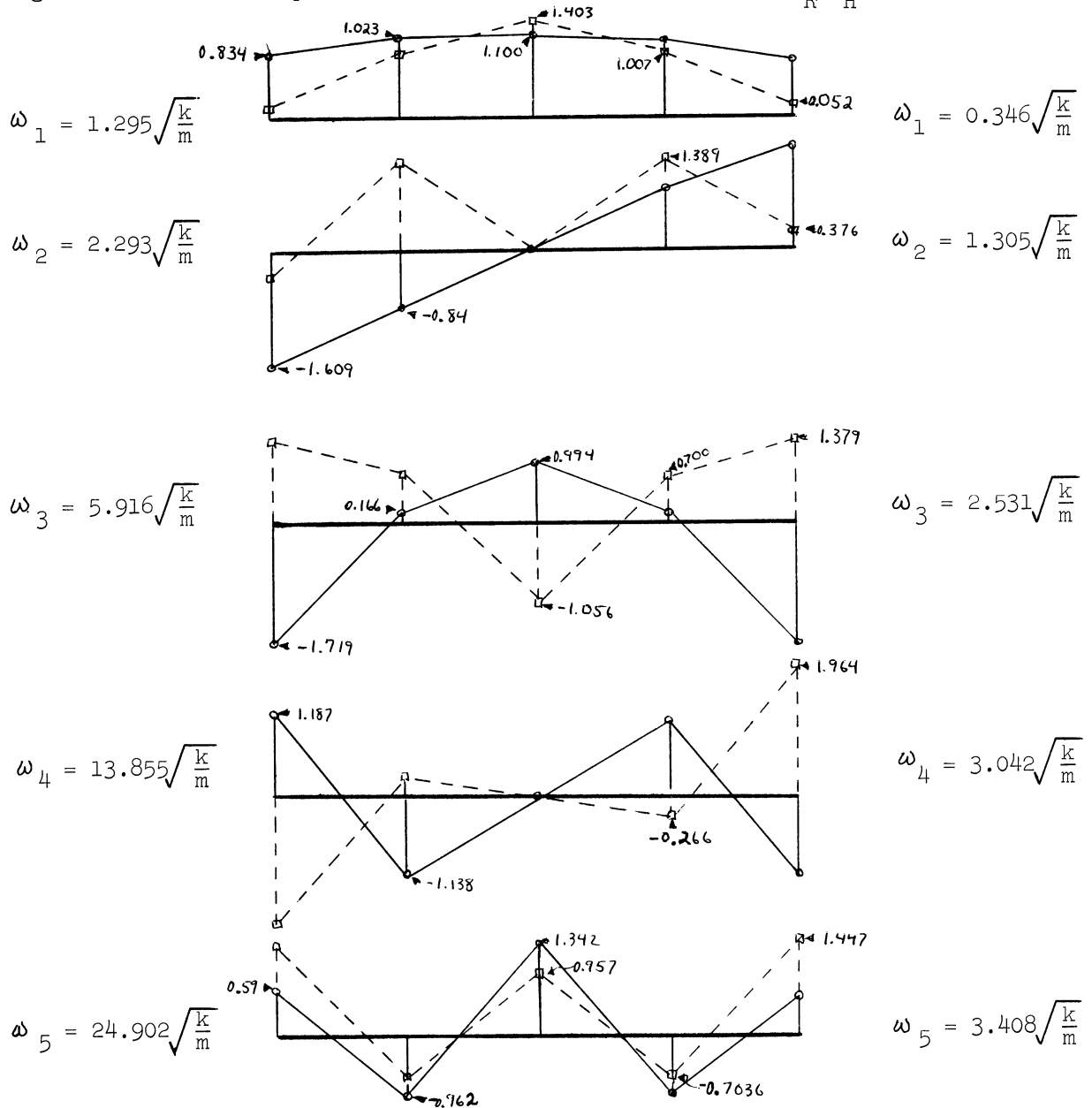


Figure 2. Normal modes and frequencies

————— $\omega_R/\omega_H = 0.5$
 - - - - - " = 4.0

Critique

A problem of this type is of great importance in the design of underground structures to isolate shock loads. Many textbooks on vibration do not mention Jacobi's method. With the advent of high speed digital computers, this method becomes very important. Its algorithm is simple and all the eigenvalues and eigenvectors are computed at the same time. Further extensions of this problem include the computation of bending moment matrix, shear matrix, reaction coefficients, and the moment-time relation for a given shock function.

VII. REFERENCES

1. Organick, E. I., A Computer Primer for the MAD Language, University of Houston, Computing and Data Processing Center, 1961.
2. Organick, E. I., A Computer Primer for the FORTRAN Language, University of Houston, Computing and Data Processing Center, 1961.
3. Project on the Use of Computers in Engineering Education, First Annual Report, The University of Michigan College of Engineering, Ann Arbor, August 26, 1960.
4. Project on the Use of Computers in Engineering Education, Second Annual Report, The University of Michigan College of Engineering, Ann Arbor, December 15, 1961.
5. Westervelt, F. H., Use of Computers in Mechanical Engineering Education, Project on the Use of Computers in Engineering Education, The University of Michigan College of Engineering, Ann Arbor, June 1, 1962.
6. Wilson, R. C., Use of Computers in Industrial Engineering Education, Project on the Use of Computers in Engineering Education, The University of Michigan College of Engineering, Ann Arbor, January 27, 1962.

UNIVERSITY OF MICHIGAN



3 9015 03627 7328