PROCEEDINGS

Computation Seminar

ли с изт 1951

Computation Seminar

august 1951

×

EDITED BY IBM APPLIED SCIENCE DEPARTMENT

CUTHBERT C. HURD, Director

INTERNATIONAL BUSINESS MACHINES CORPORATION

NEW YORK + NEW YORK

Copyright 1951 International Business Machines Corporation 590 Madison Avenue, New York 22, N. Y. Form 22-8705-0

UNITED

S T A T E S

O F

AMERICA

PRINTED

I N

т н е

FOREWORD

COMPUTATION SEMINAR, sponsored by the International Business **L** Machines Corporation, was held in the IBM Department of Education, Endicott, New York, from August 13 to August 17, 1951. Participating in this Seminar were ninety research engineers and scientists representing computing facilities which employ IBM Card-Programmed Electronic Calculators. The discussion centered on the mathematical and computational aspects of a variety of important problems which have been solved on the Card-Programmed Electronic Calculator. The formal papers of the Seminar and a digest of the ensuing discussion are published in this volume. In addition, informal papers were presented at several valuable supplementary sessions. Dr. W. J. Eckert presided at a session on Training of Personnel for Computing Laboratories at which Mr. Murray Lesser, Mr. Walter Ramshaw and Professor Frank Verzuh led the discussion. Mr. P. M. Thompson presided at a session on the Organization of a Computing Installation at which Mr. W. D. Bell, Dr. H. R. J. Grosch, and Mr. J. D. Madden gave short papers. Mr. E. B. Gardner presided at a session in which there was widespread discussion of the subject of Data Reduction. More generally, all participants in the Seminar contributed generously in discussions. The International Business Machines Corporation wishes to express its appreciation to all who participated in this Seminar.

CONTENTS

Application of the IBM Card-Programmed Electronic Calculator to Engineering Procedures at The Glenn L. Martin Company	—WARREN B. KOCH
Reduction of Six-Component Wind Tunnel Data Using the IBM Card-Programmed Electronic Calculator, Model II	—Murray L. Lesser
IBM Card-Programmed Electronic Calculator Operations Using a Type 402-417BB and 604-2	—Harley E. Tillitt Martha Kenyon Bruce Oldfield
The Combomat	—John D. Madden 33
The IBM Type 604 Electronic Calculating Punch As a Miniature Card-Programmed Electronic Calculator	—Paul T. Nims
General-Purpose Floating Point Control Panels for the IBM Card-Programmed Electronic Calculator	—Norman A. Patton Kathleen Berger L. Richard Turner 48
Catapult Takeoff Analysis	—John R. Lowe
Computation of Loan Amortization Schedules on the IBM Card-Programmed Electronic Calculator	—Charles H. Gushee 72
Techniques for Handling Graphical Data on the IBM Card-Programmed Electronic Calculator	William D. Bell 77
Calculation of the Flow Properties in an Arbitrary Two-Dimensional Cascade	—John T. Horner 81
Automatic Calculation of the Roots of Complex Polynomial Equations Using the IBM Card-Programmed Electronic Calculator	—John Gallishaw, Jr 87
A Recursion Relation for Computing Least Square Polynomials over Moving-Arcs	—George R. Trimble, Jr 93
Numerical Solution of Second-Order Non-Linear Simultaneous Differential Equations	—Henry S. Wolanski 98
Matrix Inversion and Solution of Simultaneous Linear Algebraic Equations with the IBM Type 604 Electronic Calculating Punch	—George W. Petrie, III 105
The Determination of Eigenvectors and Eigenvalues of Symmetric Matrices	—Everett C. Yowell 112
An Application of the IBM Card-Programmed Electronic Calculator to Analysis of Airplane Maneuvering Horizontal Tail Loads	—Logan T. Waterman 120
Fifth-Order Aberration in an Optical System	
Theory of Elastic Vibrations of Helicopter Fuselages	—Peter F. Leone
Machine Procedure for Computation of Elastic Vibrations of Helicopter Fuselages	—William P. Heising

PARTICIPANTS

- ANDERSON, RUTH K., Mathematician National Bureau of Standards Washington, D. C.
- ARNOLD, KENNETH J., Assistant Professor of Mathematics University of Wisconsin Madison, Wisconsin
- AROIAN, LEO A., Chief, Computation Group Research and Development Laboratories Hughes Aircraft Company, Culver City, California
- BAUGH, HAROLD W., *Electronic Scientist* Computer Branch, U. S. Naval Air Missile Test Center Point Mugu, California
- BELL, WILLIAM D., Vice-President Telecomputing Corporation Burbank, California
- BERKOWITZ, RAYMOND S., Associate Moore School of Electrical Engineering University of Pennsylvania, Philadelphia, Pennsylvania
- BRINKLEY, STUART R., JR., Head Mathematical and Theoretical Physics Section U. S. Bureau of Mines, Pittsburgh, Pennsylvania
- BROOKS, JACK E., *Principal Stress Analyst* Republic Aviation Corporation Farmingdale, New York
- BROOKSHIRE, JACK W., *Tabulating Research Specialist* North American Aviation, Incorporated Los Angeles, California
- BROWN, DONALD M., Supervisor Mathematical Services Group, Willow Run Research Center University of Michigan, Ann Arbor, Michigan
- CARTER, DAVID S., *Research Associate* Forrestal Research Center, Princeton University Princeton, New Jersey
- CHEYDLEUR, BENJAMIN F., *Chief* Applied Mathematics Laboratory U. S. Naval Ordnance Laboratory, Silver Spring, Maryland
- CONANT, GEORGE H., JR., Mathematician The Perkin-Elmer Corporation Cambridge, Massachusetts
- COZZONE, FRANK P., Manager Mathematical Analysis Department Lockheed Aircraft Corporation, Burbank, California
- DEUTSCH, MURRAY L., *Physicist* Research and Development Laboratory Socony-Vacuum Oil Company, Paulsboro, New Jersey
- DEVRIES, JOHN A., Senior Stress Analyst Propeller Division, Curtiss-Wright Corporation Caldwell, New Jersey
- DUGGAN, JOHN M., Mathematician A. V. Roe, Canada, Limited Malton, Ontario

- ECKERT, WALLACE J., *Director* Department of Pure Science, Watson Scientific Computing Laboratory IBM Corporation, New York, New York
- EVANS, GEORGE W., II, Associate Mathematician Argonne National Laboratory Chicago, Illinois
- FEIGENBAUM, DAVID, Associate Research Engineer Cornell Aeronautical Laboratory Buffalo, New York
- FERBER, BENJAMIN, Senior Research Engineer Consolidated Vultee Aircraft Corporation San Diego, California
- FUHRMEISTER, PAUL F., Aeronautical Research Scientist National Advisory Committee for Aeronautics Langley Field, Virginia
- GALLISHAW, JOHN, JR., *Tabulating Methods Specialist* Chance Vought Aircraft Dallas, Texas
- GARDNER, EARL B., Chief Automatic Computing Unit, Data Reduction Branch Holloman Air Force Base, New Mexico
- GROSCH, H. R. J., *Head* Technical Computing Bureau, IBM Corporation Washington, D. C.
- GROSH, LOUIS E., JR., Associate Statistical Laboratory, Purdue University West Lafayette, Indiana
- GROSS, GEORGE L., *Research Engineer* Grumman Aircraft Engineering Corporation Bethpage, Long Island, New York
- GUSHEE, CHARLES H., President Financial Publishing Company Boston, Massachusetts
- HAMMING, RICHARD W., Research Mathematician Bell Telephone Laboratories Murray Hill, New Jersey
- HARP, WILLIAM M., Section Head Refining Technical Service Division Humble Oil and Refining Company, Baytown, Texas
- HEISING, WILLIAM P., Mathematician Technical Computing Bureau, IBM Corporation New York, New York
- HERGET, PAUL, Director University of Cincinnati Observatory Cincinnati, Ohio
- HILL, W. HENRY, *Coordinator of Naval Aviation Statistics* Office of Chief of Naval Operations Washington, D. C.
- HOELZER, HELMUT, Chief, Computation Center Ordnance Guided Missile Center Redstone Arsenal, Huntsville, Alabama

- HORNER, JOHN T., Senior Project Engineer Allison Division, General Motors Corporation Indianapolis, Indiana
- HURD, CUTHBERT C., Director Applied Science Department, IBM Corporation New York, New York
- JACKSON, IRWIN E., JR., Marine Engineer Bureau of Ships, U. S. Navy Department Washington, D. C.
- JACOBSON, ARVID W., Associate Professor of Mathematics Wayne University Detroit, Michigan
- KELLY, ROBERT G., Mathematician Applied Physics Laboratory, Johns Hopkins University Silver Spring, Maryland
- KENNEDY, ERNEST C., Senior Research Engineer Ordnance Aerophysics Laboratory, Consolidated Vultee Aircraft Corporation, Daingerfield, Texas
- KOCH, WARREN B., Senior Mathematician The Glenn L. Martin Company Baltimore, Maryland
- LEONE, PETER F. Piasecki Helicopter Corporation Morton, Pennsylvania
- LESSER, MURRAY L., Methods-Coordination Engineer Theoretical Aerodynamics Department Northrop Aircraft, Inc., Hawthorne, California
- LOWE, JOHN R., Computing Engineer Douglas Aircraft Company, Inc. Santa Monica, California
- MACINTYRE, NEIL W., Administrative Assistant Mutual Life Insurance Company of New York New York, New York
- MADDEN, JOHN D., Assistant Mathematician The RAND Corporation Santa Monica, California
- MAGINNISS, FRANK J., Application Engineer Analytical Division, General Electric Company Schenectady, New York
- MCPHERSON, JOHN C., Vice-President IBM Corporation New York, New York
- NIMS, PAUL T., Staff Engineer—Research Engineering Division, Chrysler Corporation Detroit, Michigan
- PATTON, NORMAN A., Aeronautical Research Scientist Lewis Flight Propulsion Laboratory National Advisory Committee for Aeronautics, Cleveland, Ohio
- PETRIE, GEORGE W., III, Mathematician Applied Science Department, IBM Corporation Washington, D. C.

- PORTER, RANDALL E., *Research Engineer* Boeing Airplane Company Seattle, Washington
- RACHFORD, HENRY H., JR., Research Engineer Humble Oil and Refining Company Houston, Texas
- RAMSHAW, WALTER A., Analytical Engineer United Aircraft Corporation East Hartford, Connecticut
- RIDGWAY, ANDRESS O., Mathematician U. S. Navy Hydrographic Office Washington, D. C.
- ROSE, ARTHUR, Associate Professor of Chemical Engineering The Pennsylvania State College State College, Pennsylvania
- SCHLIESER, WALTER C., JR., Computing Supervisor Douglas Aircraft Company, Inc. El Segundo, California
- SCHUTZBERGER, HENRY, Supervisor Test Data Division, Sandia Corporation Albuquerque, New Mexico
- SELLS, BERT E., Turbine Engineer General Electric Company Boston, Massachusetts
- SHELDON, JOHN W., Head Technical Computing Bureau, IBM Corporation New York, New York
- SHIPMAN, JEROME S., Mathematician Laboratory for Electronics, Incorporated Boston, Massachusetts
- SMITH, CHARLES V. L., *Head* Computer Branch, Office of Naval Research Washington, D. C.
- SMITH, ROBERT W., JR., Mathematician U. S. Bureau of Mines Pittsburgh, Pennsylvania
- SOWERS, NELSON E., Mathematical Data Analyst Army Field Forces Board 4 Fort Bliss, Texas
- THOMAS, L. H., Senior Staff Member Watson Scientific Computing Laboratory, IBM Corporation New York, New York
- THOMPSON, PHILIP M., *Head* Computing Laboratory, Hanford Works, General Electric Company Richland, Washington
- TILLITT, HARLEY E. U. S. Naval Ordnance Test Station, Inyokern China Lake, California
- TRIMBLE, GEORGE R., JR., Mathematician Ordnance Ballistic Research Laboratory Aberdeen Proving Ground, Maryland

- VERZUH, FRANK M., Director of Statistical Services Massachusetts Institute of Technology Cambridge, Massachusetts
- VIALL, JOHN, Supervisor IBM Computing Installation, Douglas Aircraft Company, Inc. Long Beach, California
- VON HOLDT, RICHARD, Staff Member University of California; Los Alamos Scientific Laboratory Los Alamos, New Mexico
- WADDELL, WILLIAM L., Group Engineer Computation Analysis, Northrop Aircraft, Inc. Hawthorne, California
- WALKER, JOHN H., Mathematician U. S. Naval Proving Ground Dahlgren, Virginia
- WATERMAN, LOGAN T., Chief of Flutter and Vibrations Fairchild Aircraft Division Fairchild Engine and Airplane Corporation, Hagerstown, Maryland
- WELMERS, EVERETT T., Chief of Dynamics Bell Aircraft Corporation, Niagara Falls, New York

WEST, GEORGE P., Aeronautical Research Scientist Ames Aeronautical Laboratory National Advisory Committee for Aeronautics Moffett Field, California

- WHITE, J. HUNTER, JR., Mathematician Applied Science Department, IBM Corporation New York, New York
- WHITE, LAWRENCE L., Mathematician IBM Unit, U. S. Air Force Flight Research Laboratory Dayton, Ohio
- WITT, EDWARD C., Mathematician K-25 Plant, Carbide and Carbon Chemicals Corporation Oak Ridge, Tennessee
- WOLANSKI, HENRY S., Aerophysics Engineer Consolidated Vultee Aircraft Corporation Fort Worth, Texas
- YOWELL, EVERETT C., Mathematician Institute for Numerical Analysis National Bureau of Standards Los Angeles, California

Application of the IBM Card-Programmed Electronic Calculator to Engineering Procedures at The Glenn L. Martin Company

WARREN B. KOCH

The Glenn L. Martin Company

THE PURPOSE of this paper is to describe the most important engineering problems that have been adapted for processing on the IBM Card-Programmed Electronic Calculator at The Glenn L. Martin Company. However, before starting upon any such discourse, it is advisable to consider first the tools of operation—in this case, the type of control panels that are in use.

It has been found necessary to design only two control panels for the CPC, the operations available from one or the other proving adequate to cover all needs up to the present time. They are a floating decimal type with sevendigit capacity and a fixed decimal with six-digit capacity. Of course, it would always be advantageous to program jobs on a floating decimal setup, but because of the additional complications involved in wiring such a control panel, fewer operations are available than on a fixed decimal design.

The operations programmed on the floating decimal control panel include only addition, subtraction, multiplication, division, and square root, while the fixed decimal control panel allows for all these operations in addition to that of substitution of an argument into a polynomial of at most the fourth degree. This latter operation works in conjunction with a selection which makes it possible to select any one of a number of polynomials, depending upon the magnitude of the argument used. This type of operation is particularly valuable when it is necessary to approximate graphical data with polynomials, and different equations must be used over certain ranges of the curve.

FLOATING DECIMAL JOBS

Simultaneous Linear Algebraic Equations

We shall now consider the more or less routine jobs that have been adapted for the floating decimal control panel. The first of these is the solution of simultaneous linear algebraic equations. The method used is essentially the GaussJordan method resulting in a unit diagonal matrix. Since the floating decimal setup is used, no attempt has been made to select the largest terms as divisors; rather, these operator coefficients are selected down the main diagonal. The solution is substituted back into the original equations and, when necessary, the errors thus resulting are operated upon to obtain corrections to the original results.

Flutter Analysis

The only type of flutter analysis which has been thus far investigated is that of determining the critical flutter speed of any aircraft; that is, the speed at which the damping of the structure is attained. This job is laborious in that it involves the expansion of a number of determinants of order equal to the number of degrees of freedom for which the aircraft is designed. The Dynamics Department has limited the size of these determinants to fourth order through various simplifying assumptions, but a setup is now being evolved which will handle sixth order determinants on the CPC. The result of an expansion of one of these determinants is a polynomial with complex coefficients of degree equal to the order of the determinant.

A method has been devised of applying the CPC to the solution of any polynomial with coefficients either properly complex or real. Because of the limitations of storage, the system has been designed for polynomials of degree no more than eight. The mathematical technique involved is Newton's method. Up to the present time, no difficulty has been realized in obtaining any roots—in fact, the number of iterations required to obtain six-digit accuracy in the root usually has been less than twenty.

Vibration Frequency Analysis

A method has been developed in the Dynamics Department for determining the natural vibration frequencies of a beam for either bending vibrations alone or coupled bending-torsion vibrations. The calculation is initiated by first dividing the beam into sections with discrete masses concentrated at each station and then describing these stations by series of matrices.^a

The matrices are written with the frequency factor, K, as a variable. Then, by simply multiplying the proper coefficients by any value of K, the problem may be processed for any particular frequency. The routine is simply one of trial and error, where a value of K is substituted, and the associated matrices of successive beam stations are continually premultiplied until a final matrix equation is obtained relating the boundary conditions on both ends of the vibrating beam. The imbalance of any chosen end condition is plotted against K and the process repeated for a new value of Kuntil a K is found which gives no imbalance. The K resulting in this balance condition determines the vibrating frequency desired. The machine calculations begin after the matrices have been written and cover the substitution of the K values, the matrix multiplication, and the determination of the imbalance between end conditions.

Transfer Functions

Quite extensive use of the CPC has been made in the formulation and evaluation of transfer functions in the investigation of dynamic stability of guided missiles. The transfer function of any component is essentially its equation of motion expressed in differential operator form as the ratio of the input to the output. The closed loop stability characteristics can be determined from the so-called Nyquist diagram obtained from the evaluation of the transfer function. For a sinusoidal input, the phase and amplitude of the output are plotted in the complex plane as a function of frequency. The stability is determined as a function of the encirclements of the -1 point on the real axis.

For the airframe itself, any transfer function (such as a function of elevator, roll as a function of aileron, etc.) is completely determined by the flight condition and aerodynamic properties as determined in a wind tunnel. A CPC procedure has been set up to accept this type of data in order to formulate any transfer function describing the missile dynamics. The transfer functions of other components (servos, amplifiers, valves, etc.) are determined by either analysis or test.

Given the transfer functions of *n* consecutive components in a control system—call them A_i , where $i = 1, \ldots, n$ another procedure on the CPC obtains the response as a function of frequency of not only each component A_i but also of the products $A_1 \times A_2, A_1 \times A_2 \times A_3, \ldots, A_1 \times A_2 \times \ldots \times A_n$.

FIXED DECIMAL JOBS

Simultaneous Ordinary Differential Equations

The most important use of the fixed decimal control panel, and in fact of the CPC itself, has been in the solution of simultaneous ordinary differential equations. This problem has become one of frequent occurrence with the entrance into the guided missile field and its attendant problems of automatic control and trajectory computations. Of course, this problem occurs within other problems but, by far, the preponderance of work done on the CPC at The Glenn L. Martin Company has been concerned with trajectory calculations.

A stepwise method is used to solve the equations with values of the dependent variables at the beginning of each step determined by quadratic extrapolation from previous information. These extrapolated values are then improved upon later by a single iteration and the difference between extrapolated and iterated values compared. This error is used to determine the maximum allowable interval into which the independent variable may be divided and also serves as a check on the machine calculations.

There is almost no limit on the number of equations (or dependent variables) that may be handled simultaneously, and there is no problem in taking care of complicated nonlinearities in the coefficients.

Recently, a trajectory has been calculated at The Glenn L. Martin Company in which three distinct rectangular coordinate systems, each in three dimensions, were handled simultaneously. It was necessary to write a set of 76 equations to describe the system completely. Of these, about one-third were differential equations, and the rest were either simply definition or angular resolution equations. Of course, it was impossible to store all values of the dependent variables; so it became necessary to summary punch many of them and reload them into the machine at various times in the following interval. This was done by prepunching decks of summary cards with operation instructions and then merely running the deck of program cards through the 402 once, replacing the pertinent cards with newly punched summary cards and continuing this process until a complete trajectory was calculated. Under this system, it was possible to go through one complete cycle of processing about 400 cards through the accounting machine, reloading the new summary cards and starting to process again in five minutes. A complete solution required approximately 80 machine hours and described 13 seconds of flight time. Manually, this job would have taken some 3,000 hours.

Analysis of a Cabin Conditioning System

During the course of a recent investigation of the performance of a cabin air conditioning and pressurization system, an analytical method for studying this problem was

^aThe method of developing these matrices has been fully described in an article in the October, 1947, issue of the *Journal of the Aeronautical Sciences* by W. P. Targoff entitled "The Associated Matrices of Bending and Coupled Bending-Torsion Vibrations."

devised. This method, although applied to just this one system, should have extended application, and it can be tied in exceedingly well with CPC methods. This is so because the system is basically one of trial and error, where each trial involves a lengthy calculation on a small amount of data.

The system contains essentially a primary compressor, secondary compressor turbine unit, two heat exchangers, a water separator, two combustion heaters and fans, and various ducts, valves, and control mechanisms. The primary problem is the establishment of criteria for determining an unique set of operating points for the various components. The only parameter known is the primary compressor impeller speed. All other compressor variables are unknown; furthermore, the performances of the secondary compressor and turbine are dependent on each other, and the performances of this entire unit and the primary compressor unit are dependent on each other. Therefore, the performance of the entire system cannot be predicted by any straightforward method of calculation, and a trial-and-error system must be adopted.

The method most frequently used in this type of problem is to assume several values of each of the independent variables, and calculate the performance until the values chosen satisfy the equilibrium criteria. However, because of the wide variations of the conditions of flight, and because of the customary use of various automatic limiting devices which arbitrarily change the functional configuration, it is felt that this simple trial-and-error method is unsatisfactory. Rather, several values of two arbitrarily chosen variables are assumed in a systematic relationship so that existing trends become apparent. In addition, if the various assumed operating points bracket (or nearly bracket) the actual operating point, this point can be established with acceptable accuracy.

The CPC is particularly suited to this procedure, since a few points may be computed rapidly and then any desired re-trials may be run by simply changing a few load cards.

Wind Tunnel Data Reduction

Until recently, all wind tunnel data reductions have been calculated on the IBM Type 604 Electronic Calculating Punch. It has been found that the CPC can be adequately adapted to this calculation with a resultant time saving of 60% over the 604.

All computations, with the exception of the correction of forces and moments for residual balance readings, are performed on the CPC. This initial correction is obtained more conveniently on the 604. The remaining calculations include:

- 1. Converting the force and moment data to coefficient form.
- 2. Correcting these coefficients for:

- a. Tare and interference, which is the influence exerted on the model by the supports and support fairings.
- b. Alignment, or inclination of the wind stream to the balance system.
- c. Effect of the constraint of the wind tunnel walls.
- d. Buoyancy, which is the effect of fore and aft pressure gradients in the tunnel.

All of the data required to apply these corrections are determined by exacting wind tunnel testing, and the corrections applied in equation form to the aerodynamic coefficients.

These corrected coefficients lie in a wind axis coordinate system with the origin at the point of attachment between model and support. The coefficients are finally transferred to two other coordinate systems: (1) the wind axes with origin at the center of gravity of the model, and (2) the stability axes.

Aeroelastic Influences on Control Surface Effectiveness

The aeroelastic problem is concerned with the loss of effectiveness of control surfaces, including ailerons, flaps and elevators, because of the elastic deformations of wing fuselage and tail. The deformations are brought about by both the twisting and bending of these various parts, which are due to angle of attack and deflection of the control surfaces.

Twisting is brought about by the fact that the chordwise centers of pressure of loads which are due to control surface deflections and angle of attack are not located at the elastic axis, thus producing powerful torques which tend to twist the wing or tail surface and change the effective angle of attack. This twisting, in turn, produces loads which further influence the elastic deformation of the wing or tail.

The problem is one of iteration since, to begin with, the final spanwise twist distribution cannot be predicted accurately. A twist distribution must be assumed, the torques which are due to this twist and basic loads must be computed, and a new twist distribution is determined from these torques. With this new twist distribution, the problem is started anew, and the same steps are carried out to find a third twist distribution, and so forth, until the solutions converge to a final twist distribution.

Once the nature of the twist curve is determined, the problem becomes simple and straightforward. Now the load distributions, caused both by the basic load moments and the moments resulting from the twist distribution loads, are computed accurately by the Weissinger method.

This entire computation procedure is programmed by approximately two thousand cards, of which half are used as many times as necessary (usually three) in the iteration procedure. There are only about two hundred cards containing data that must be changed for different configurations, and these are specially coded so that they may be changed readily between jobs. This system, which takes about four hours per configuration, has replaced a procedure that usually required three weeks to complete by handcomputing methods.

DISCUSSION

Mr. Von Holdt: At Los Alamos we punch out eight different numbers and put the instructions in 1 to 10 automatically. They are in the instruction deck and go into the summary punched card; so it is not necessary to have prepunched cards.

Dr. Brown. When you mentioned the 80 hours of running time on the machine, Mr. Koch, how did that compare with the number of hours it would have taken by hand operations?

Mr. Koch: It would have taken 3,000 hours.

Mr. Lowe: If I understood you correctly, you extrapolate in your trajectory programming with some quadratic extrapolations involving a prior point. Is that correct, or do you use two prior points?

Mr. Koch: We use two prior points.

Dr. Yowell: I am very much interested in some of the systems you say you have handled with more than eight differential equations at a time, where you have had to punch out the results from one step to the next and put them back in again. Has it been necessary to punch more than two cards with the same values on them?

Mr. Koch: Yes, it has. Occasionally we have had to punch the same value in a number of different cards to be used at different places in the calculation.

Mr. Lesser: With regard to summary punching multiple cards from the same set of data, the technique that we used was to bring out the 12-impulse from the digit-emitter in the summary punch back through the summary punch X control wires to the accounting machine through a latch selector on the accounting machine, and then back through a second summary punch X control wire to pick up punch selectors. On additional summary punch cycles, the punch selectors caused the data to be gang punched from the lower brushes back to the punch magnets as many times as we desired.

Also I was wondering if you have done any investigating so that, instead of having to predict the next stop, you use a much simpler form of numerical analysis and just take your intervals a little closer together.

Mr. Koch: No matter how small the intervals are made, we always have to predict.

Dr. Grosch: I would like to suggest the possibility of increasing rather than decreasing the interval, for this reason: The steps of Mr. Koch's procedure take him about five minutes; so this looks like something of the order of 1,000 steps he has to go through in his 80 hours. But suppose we could decrease the number of steps tenfold by taking the interval ten times as large, at the expense of having to store more data for the extrapolation. For instance, if you have nine space intervals you might have to store four or five extra orders of differences for each one of those nine, i.e., 36; and that might mean punching out twenty or thirty extra summary cards, which will take up ten or twenty per cent extra time per step. But if you have reduced the number of steps tenfold, you still have a gain of eightfold in your time of running, and cut down from 80 hours to ten hours.

It is true that the programming will be more complex, and it is also true that every time you get a discontinuity you are going to have more trouble in getting past. But I think you are more likely to get more results out of your machine by increasing the interval than by decreasing it.

Dr. Thomas: I would like to warn people that while what Dr. Grosch has said is likely to be very true for computations which are carried to great accuracy, like astronomical computations, it is not at all likely to be true for calculations which do not have to be carried to such great accuracy, like trajectory problems. If you are only going to something like a tenth of a per cent to start with, then the optimum interval for simple formulas is not going to be very greatly increased by using more complicated formulas.

If, however, you wanted to carry the same calculation to twice as many decimal places, you would probably do very much better by using more complicated formulas than by using shorter intervals. But for calculations which are only of relatively low accuracy, from the astronomical point of view, the very complicated formulas with many differences really do not pay.

Reduction of Six-Component Wind Tunnel Data Using the IBM Card-Programmed Electronic Calculator, Model II*

MURRAY L. LESSER

Northrop Aircraft, Inc.

×

TEST DATA taken during a wind tunnel program on a scale model of a Northrop airplane, conducted in the tenfoot pressure tunnel at Wright-Patterson Air Force Base, were reduced to coefficient form using the IBM Card-Programmed Electronic Calculator, Model II. Six-component test data^a for each test point were key-punched into a single IBM card (average key punch time was approximately 75 test points per hour). The data cards were machine collated into a prepared program deck, and the results were computed and printed in coefficient form for both stability and wind axes (simultaneously) in one pass through the calculator at a rate of three test points per minute. Because of the nature of the balance system and the small size of the model relative to the tunnel, corrections were not required for tares or wall effects.

The programming and wiring of the CPC for this computation are discussed in more than usual detail as an illustrative example of the flexibility available in the CPC Model II.^b This flexibility allows parallel-serial operation of the calculator where a small number of digits will carry the required information. With few exceptions, the computations required are elementary, and no originality is claimed for the manner in which they are accomplished.

It should be pointed out that the machine time for an equivalent computation on the CPC Model I would have been approximately twice as long. Also, it would have been necessary to split the key-punched data for one test point into two cards, thus increasing the possibility of error through operator mishandling as well as increasing the key-punch time.

I. DATA REDUCTION FORMULAS

The test data are recorded, from a strain gage balance system located in the sting mount, in the form of readings on a Brown self-balancing potentiometer. In order to eliminate tares and effects of battery voltage variation, a "zero" reading for each strain gage is subtracted from the particular reading before the point is recorded on the key-punch form (Figure 1). The recorded values, as entered on the form, are the net strain gage readings divided by 100. The six-component data are recorded from the balance system in the form of forward and rear normal force gage readings, n_1 and n_2 , respectively; chord force reading, c; forward and rear side force readings, s_1 and s_2 ; and a rolling moment reading, r. The associated loads are obtained from the following calibration data :

$$N_{1} = 316.0 n_{1} \text{ (lb.)}$$

$$N_{2} = 80.0 n_{2} \text{ (lb.)}$$

$$C' = 31.7 c \text{ (lb.)}$$

$$S_{1} = 44.2 s_{1} \text{ (lb.)} \text{ (1)}$$

$$S_{2} = 22.9 s_{2} \text{ (lb.)}$$

$$R = 446.0 r \text{ (in.-lb.)}$$

In addition to the chord force determined directly from the reading of c, it was found during the calibration tests that there existed interaction effects on c because of the other loads. Hence, the net chord force, in pounds, is given by the following expression:

$$C = C' + 0.0116 N_1 + 0.0050 N_2 + 0.011 S_1 + 0.022 R + 0.888 \psi.$$
(2)

The value of ψ used above is the preset value, ψ_u , corrected for deflection in the mounting system due to airloads. Calibration tests provided the following correction

^aData key-punched into the card consisted of the six wind tunnel strain gage readings, the uncorrected angles of attack and yaw, the tunnel dynamic pressure, and an identifying run number.

^bIt should be noted that the discussion on machine operation (Section II) assumes a speaking knowledge of IBM computing techniques on the part of the reader; in particular, familiarity with the concepts of the CPC Model I is implied.

^{*}The terms Model I and Model II are used in this paper as simplified CPC designations for the Type 402-417BB, 604-3, and the Type 402-417AA, 604-2, respectively; the terms Model I and Model II do not refer to actual machine types.—*Editor's note*.

		<u>IC</u>)- FOOT	WIND TUN	NEL TEST	DATA			
Ę	6 1016	2021	25	26 30	31 353	6 404	45 46	5051	5
q	RUN NO.	d	¥.	n,	ng	С	S ₁ S	<u>2</u> r	
									T
	<u> t</u>								t
<u> </u>	┟╾┽╌┽╌┽╌┽╌					+ + + + +	+ + - + - +		+
<u> </u>						+ + + + + + + + + + + + + + + + + + + +			+-
								·····	
			L						
									Ļ
									1
									L
		· · · · · · · · · · · · · · · · · · ·			┋╾╵╌╹╌┟				1-
		• • • • • • • • •							+-
u i.	<u> </u>					÷			+
<u> </u>									
					<u>н</u> , Ц				
· · · · · · · ·	<u>↓ ↓ ↓ ↓ ↓ ↓ ↓</u>					<u>+</u>			t
	┟──┴──┴──┼──	┈╪╌┶╼┢╧┝╌┥			<u>t-</u>				t
	╞╴╺╴╍╍╍╌┟╌	┷╪╼╍┰╌┨╾┙			- <u> </u>]-				+
									+
						÷			
	<u> </u>								t
<u> </u>	<u>↓</u>								
<u> </u>	<u>Letter territorial de la constante de la cons</u>	<u>. </u>		└┈┊╶╅═╋╼┿┷╘					- مار د

FIGURE 1. TEST DATA FORM

expressions for the angle of attack, α , and the angle of yaw, ψ , in degrees :

$$\alpha = \alpha_u + 0.0016 N_2 + f(N_1)$$

$$f(N_1) = 0.0022 N_1 \text{ (for } N_1 > 0)$$

$$= 0.0032 N_1 \text{ (for } N_1 < 0)$$
(3)

$$\psi = \psi_u + 0.0028 S_1 + 0.0020 S_2.$$

From the geometry of the model and balance system, and the conventional definitions of the aerodynamic forces and moments, the following expressions are derived :

A. Stability Axes

		(4)
Lift =	$L_s = (N_1 + N_2)\cos \theta$	$\alpha - C \sin \alpha$ (lb.)
Drag 🚃	$D_s = C \cos \alpha + (N_1 - C)$	$+N_2$) sin α (lb.)
Side force \equiv	$Y_s = S_1 + S_2$	(1b.)
Pitching moment \equiv	$M_s = -0.169 N_1 - 9$	$9.581 N_2$ (inlb.)
Yawing moment \equiv	$n_s = (-1.134 S_1$	$\frac{10.546 S_2) \cos \alpha}{R \sin \alpha \text{ (inlb.)}}$
Rolling moment \equiv	$l_s = (-1.134 S_1 - +$	10.546 S_2) sin α $R \cos \alpha$ (inlb.)

B. Wind Axes

Lift =
$$L_w = L_s$$

Drag = $D_w = D_s \cos \psi + Y_s \sin \psi$
Side force = $Y_w = Y_s \cos \psi - D_s \sin \psi$ (5)
Pitching moment = $M_w = M_s \cos \psi + l_s \sin \psi$
Yawing moment = $n_w = n_s$
Rolling moment = $l_w = l_s \cos \psi - M_s \sin \psi$

The aerodynamic coefficients are determined from the airloads in the conventional manner.

$$C_{L} = L/qS_{w}$$

$$C_{D} = D/qS_{w}$$

$$C_{Y} = Y/qS_{w}$$

$$C_{M} = M/qS_{w}\overline{c}$$

$$C_{n} = n/qS_{w}b$$

$$C_{l} = l/qS_{w}b$$
(6)

where

 $q = Dynamic pressure (lb./ft.^2)$ $S_{w} =$ Model wing area (ft.²) \bar{c} = Model mean aerodynamic chord (in.)

b = Model span (in.).

All numerical values are carried in five-digit (plus sign) counters. The lift force and all moments are computed to one decimal place, the chord and side forces to two decimal places, and the angles of attack and yaw to three decimal places. The final coefficients are presented to four decimal places (a maximum of five significant figures in all cases).

II. COMPUTATION PLANNING CHART

Detailed planning of the computation was carried out directly on the program-deck key-punch form in the normal CPC manner. This planning sheet is shown in Figure 2. However, before such a planning sheet could be prepared it was necessary to establish the desired characteristics of the particular computing machine to be used for the problem. By this is meant that through proper design of the control panel wiring, a card-programmed calculator having the particular arithmetic and transfer properties best suited to the problem at hand was constructed from the available units in the CPC Model II. The process of design for this problem was somewhat as follows.^e

Since all of the numerical values to be computed would be limited to five significant figures, the counters of the IBM Type 417 Accounting Machine, Model AA, could be divided into 12 six-digit counter groups, each having a capacity of five numerical digits plus sign. (Note: Four counter groups, counters 8A, 8B, 8C, and 8D, each have a 19

capacity of seven digits plus sign. The unused digits are not wired to the entry or exit chains.) The use of 12 counter groups was also suggested by the fact that there are twelve items (exclusive of run number) to compute and print for each point: α , ψ , C_L , C_n , and two values each of C_D , C_Y , C_M , and C_{l} . Also, twelve is the maximum number of counter groups that can be controlled by punchings in a single card column, by numbering the counter groups from 0 to 9 and X and R.^d Thus, control of a given counter group for any function desired (read-out, read-in, reset)^e could be effected by splitting time pulses from punchings in a single card column, for each function, through a coding selector (for the digits) and a pair of co-selectors picked up by the split column controls (for the X and R impulses). Schematic diagrams of the various counter control circuits are shown in Figures 3A and 3B, pages 21 and 22.

Inspection of equations 2, 3, 4 and 5 indicate that most of the arithmetic operations involved may be put into the forms

$$F_1 = a_1 x + b_1 y F_2 = a_2 x - b_2 y,$$
(7)

where the *a*'s and *b*'s are either constants or trigonometric functions of variables, and the x's and y's are the variables involved in the equations. Consequently, the entry selector chains, i.e., the relay paths or channels over which information is transmitted from specified 417AA counter groups to the 604 Electronic Calculator (the arithmetic unit of the CPC), were designed with these two functions in mind. The two built-in channels, A and B, are used only for entry of constants from cards in the program deck, or of factors from the storage register (where the needed sines and cosines are stored as computed). Channels D and E are made up from selector chains to carry information from the 417AA counters. Card read-in (positive numbers only) is also available on channel D. The "normal" control procedure for reading information from the storage registers (16 registers of ten digits plus sign) over channels A and B is modified to allow choice of either the five low-order digits (with sign) or the five high-order digits (positive numbers

(Continued on page 23)

eSchematic wiring diagrams of the Type 417 circuits are shown in Figures 3A and 3B, pages 21 and 22. Definitions of the notation used are given in Figure 3B.

^dOver-punches referred to as X and R are 11 and 12 punches, re-

spectively. eAs wired for this problem, numbers stored in 417AA counters are left unchanged when they are read out to the IBM Type 604 Electronic Calculator. Numbers read in to the counter from the 604 are added algebraically to the previous value standing in the counter. Resetting the counter destroys the number, and leaves the counter standing at zero. The control panel wiring is such that only one of these three functions may be performed on a given counter group during a single card cycle. However, because the read-out instructions are obtained from upper brushes and reset instructions from lower brushes, read-out and reset may be programmed on a single card, the two events occurring on consecutive card cycles. It is not possible (as wired here) to read-out and reset on one card cycle and have the same counter group available for a read-in on the following cycle, i.e., the results of a 604 operation cannot be immediately stored in a 417AA counter group from which one of the factors used was obtained. This restriction is recognized in programming.

Channel A	Channel B	Channel D	А	B D	E C	CT	ц. ц.	at t	I.	Card	Pomostra
Card R.I.	Card R.I.	Card R.I.	Inst.	Inst. Inst.	Inst. Inst.	Inst. Ug	ope ope	Pri	Ъ В Ч	No.	Keijjarks
1 2 3 4 5	6 7 8 9 10	11 12 13 14 15	56 57	58 59 60	61 62 63	64 65	66 6	7 68	69	77 78	\leftarrow Card Column No.
		$+\cdot+\frac{S_w}{+}++$	00	0 0	1 7		$\begin{bmatrix} 3 \\ 7 \end{bmatrix} 1$	L	x	0 1	Load qS_w & Data (1)
					1 8			5		02	Load Run No.
	3.160			0 0	3	93	3 1			0 3	N ₁
	.8 0 8			0 0	4	0 4	3 1			04	N ₂
	.3 1 7 0			0 0	5	X 5	3 1			05	C'
	•4 4 2 0			0 0	6	R 6	3 1			06	<i>S</i> ₁
	.2 2 9 0			0 0	7	3 7	3 1			07	S ₂
	4.4 6 0			0 0	8	4 8	3 1			0 8	R
	0.0016			0 0	0	1	3 3			09	Δα1
0.0022	0.0032		0 0	0 0	9	1	2 3			1 0	$\Delta \alpha_2 = [F(N_1)]$
0.0028	0.0020		0 0	0 0 R	3	2	3 2	8		1 1	Δψ
	1 7 4 5 3	1.		0 0	1 2 7		$1 \begin{bmatrix} X \\ \xi \end{bmatrix}$	5		1 2	$\cos \alpha$, $\sin \alpha$
	1 7 4 5 3	1.		0 0	2 2 8		$1 \frac{X}{5}$	5		1 3	$\cos \psi$, $\sin \psi$
.0 1 1 6 0	.0 0 5 0 0		00	0 0 9	0	x	3 1			1 4	ΔC_1
.0 1 1 0	.0 2 2 0		0 0	0 0 R	3	x	3 1			1 5	ΔC_2
2 2 0 0 0	.8 8 8 0		0 0	0 0 4	2	x	3 0			16	ΔC3
1.	1.		0 0	0 0 9	0	5	3 8	5		1 7	$N_1 + N_2$
$\cdot 1 6 9 \overset{X}{0}$	9.5 8 1 X 0		0 0	0 0 9	0	6 9	3 1			1 8	M _s
				2 7	5	0	3 2	2		1 9	$(N_1 + N_2) \sin \alpha$
			2 X 7	x		0	3 1			2 0	$C \cos \alpha$ (D_s)
				2 X 7	5	8 5	5 1	L		2 1	$(N_1 + N_2) \cos \alpha$
			2 7	x		8 X	5 0)		2 2	$-C\sin\alpha$ (L _s)
1.	1.		0 0	0 0 R	3	x	3 5	5		2 3	$S_1 + S_2 (Y_s)$
	$1 0 .5 4 \frac{x}{6}$		0 0	0 0 R	3	9 R 3	3 1	L		2 4	$-(1.134S_1+10.546S_2)$
			28	$\begin{array}{ccc} 2 & X \\ 8 & 0 \end{array}$	X 1 1		5 1			25	Yw
-				1 7 	8	3 8	4 5	5		2 6	<i>C</i> _L
			2 7	$2 \frac{5}{7} 4$	9	R	5 1			2 7	<i>n</i>
			$2 \frac{1}{7}$	2 7 4	9	8 9	3 1			2 8	l _s
		$\frac{1}{1}$	1 7		1 6		3 2	·		2 9	qS _w c
		+ + +	1 7		1 5		3 3			3 0	$qS_w b$
			2 8	2 8 6	8 1 2	ļ	3 1	L		3 1	M _w
			2 8	$2 \frac{1}{8} 6$	8 1 3		5 1	·		3 2	l _w
			2 8	2 8 0	X 1 4		3 1			3 3	
				1 7	0	4 0	4 4			3 4	C _{Ds}
				1 6	6	5 6	4 4	E		3 5	C _{Ms}
				1 7	X	6 X	4 4	-		3 6	C _{Ys}
				1 5	8	7 8	4 4			3 7	C _{ls}
				1 5	R	8 R	4 4			3 8	Cn.
			1 3	1 5		9	6 4			3 9	C _{Iw}
			1 2				6 4	t		4 U	C Mw
			1 4	1 7			0 4	·		4 1	C _{Dw}
			1 1	1 /	<u>├</u>	ĸ		*		4 2	CYw Drint
			X		<u> </u>		├	X		4 3	Frint
		1	1 8			1 1	I I	1	[.	0 0	Compare Kun No.

Note 1. The information on program-deck card 1 is gang punched into all the data cards key punched from Figure 1. These cards are collated into the series of program decks (using X-69), and become, in turn, card 1 of each program deck.

FIGURE 2. PROGRAM-DECK KEY PUNCH FORM





NOTE 2: CHANNEL D CHAIN. Since FS 1, 3 requires six digits for complementing negative counter readings, take sixth place from C.G. sixth-order position. Carry this digit in unused side of PS chain (sign chain). Note, also, there is no provision for negative numbers on channel D card read-in.

NOTE 3: Data entry sign control pilot selectors picked up by U. B. -X. P. U. X in low order of field denotes negative value.

FIGURE 3A. SCHEMATIC OF TYPE 417 CONTROL PANEL WIRING DIAGRAM



FIGURE 3B. SCHEMATIC OF TYPE 417 CONTROL PANEL WIRING DIAGRAM

only) under control of X punches over the channels A and B instruction columns. The reason for this will be discussed below. Accounting machine wiring is such that all read-in stations in the 604 reset on every card cycle if no new information is entered.

Read-out from the 604 to the 417AA or the storage register is either from the electronic counter or from general storages 2, 4, and 4-sign to the 417AA channel C shift entry, under control of an X punch in the card column controlling the shift. The primary purpose of this arrangement is so that both the sine and cosine of an angle can be computed simultaneously in the 604, the sine being computed in general storage 4 (with sign) and the cosine in general storage 2 (always positive, as all angles are in the first or fourth quadrant). Thus, both the sine and cosine are computed and stored in a single storage register on one card feed. These two sources are wired into the channel C shift entry, through co-selectors, such that a column shift of five would match low-order digits at the channel C shift exit. Successive low-order digits may be dropped off (and the corresponding high-order digits picked up) by calling for shifts of 4, 3, 2, 1, or 0. Numbers at the channel C shift exit are available directly to any one of the 16 storage registers under digit control from the cards to channel C control (read-in of a new number to a storage register destroys the number previously stored there). These numbers are also made available to the various counter groups through the channel C_t selector chain, along with an "add" instruction to the counter group receiving the number. As the 604 reads out true numbers plus sign, the counter reversal controls were wired to the channel C exit sign through the same chain. As shown, the field selector was used for the major portion of this chain. There are no provisions for algebraically subtracting the results of a 604 computation from a counter group.

Provision is made for direct read-in of the test data to eight counter groups $(\alpha_u, \psi_u, n_1, n_2, c, s_1, s_2, \text{ and } r)$ from the key-punched data card under control of an X in card column 69. Negative values are indicated by an X punched over the low-order digit of the specific number; the X picks up a pilot selector, which causes "counter minus" instead of "counter plus" to be pulsed for that counter group. The run number is printed directly from the card, and is also stored in the 604 for later transfer to a storage register. The dynamic pressure, q, is multiplied by the wing area, S_w , during the data read-in cycle, and the product is stored in a storage register.

The 604 operations are as follows. The operation numbers are those called out by card control and refer to the calculate selectors used. The detailed planning sheet for the 604 is shown in Figure 4, page 24. Operation 1: $\sin x$, $\cos x$

Noting that

and

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots,$$

$$\sin x + \cos x = 1 + x - \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} - \dots$$

 $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$

or
$$\sin x + \cos x = \sum_{n=0}^{\infty} Y_n$$

where

and
$$f(-1) = -1$$
 for a sin x term

 $Y_0 = 1$

= +1 for a cos x term

 $Y_n = Y_{n-1}\left(\frac{x}{n}\right) \left[f(-1)\right]$

Thus, both sin x and cos x may be computed simultaneously by alternating the sign of the multiplication and the location of the summed quantity on each sweep of the 604, continuing until Y_n tests out to 0. In this process, use is made of the fact that the 604 group suppress triggers act as flip-flops; i.e., if both sides are pulsed simultaneously, the group suppress will take on the condition (on or off) opposite to that previously held. It should be noted that a "one" must be read into channel D (initial value of n) and "17453" into channel B (to convert degrees to radians) from the card in order to compute sin x and cos x.

Operation 2: $f(N_1)$

0.0022 is read into channel A and 0.0032 into channel B. This operation discriminates on the sign of the number read into channel E, and uses the proper multiplying coefficient.

(see equation 3)

Operation 3: BE + AD

This is the F_1 of equation 7. Note that there is no round off, as all digits appear in the 604 counter. Unnecessary decimal places are dropped off in the channel C shift. Note also that D + E may be formed by reading "ones" into channels A and B from the card, and that AD may be computed by leaving either B or E blank.

Operation 4: E/B (B > E/10)

This is used to compute the stability-axes coefficients. The various denominators $(qS_w, qS_w\bar{c}, \text{ and } qS_wb)$ are computed and stored in appropriate storage registers, available through channel B.

Operation 5: BE - AD

This is F_2 of equation 7. E - D may be formed from this. It should be noted that this is the only operation in

	z				FACTOR	STORA	GE				GENERAL	STORA	GE
	onic O	Sel.	ram ber	6-4	ASSIG	NMENT		MULT.	COUNTER	6-4	ASSIG		1
REMARKS	flectr	Supp	Prog	8-6	A	8-6	В	QUOT.	COUNTER	8-6		0-0	
	Sur			1 -	2 -	3 -	4 -	E	-	- 1 -	2 -	- 3 -	4 -
	-	1	R	0.00		0.01	17453	ANGLE					
			1						RR				
	GSI	1	2				RO		MPLY				
	651	, ,	3						1/2 ADJ IN 4				
	651	, , ,	4		RI				RO FR 5				
	63		5						1/2 ADJ IN 5				
	65		6						RR FR 6				RI
EMIT I	65		7								RI		
			8		RO		RI						
<u> </u>	031	+	0			RO			RI				
		1,	3						RI				
EMITI		+	10			RI			RR			+	
		+/;	11		RO				RI	+			
		+'	12			RO	┽┵┹┹┾	╫ ┍╻╹╹╹╹	╋ <u>┵╹╵╹</u> ╋┙┙╵ ╵ ╧				
	-		13		┼┹┹┶	+11	┿┺┹┹┿	╢╵╵┵╹┊	RR			+	
	·····	/	14		+	++++	PO		MPL-	╋┸┶╋	┟┵┵┵┾		
GSPU 3	GSZ	2 /	15	+		+++	RO	╫┶┶┵┊	MPL	+		+	╞╼┸╌╹╼┹╼╄╌
	65:	<u> </u>	16		++++++	+	+		16 ADJ IN 5	╉┹┹┇		+++++	
		1	17	<u> </u>	+	+ $+$ $+$ $+$ $+$ $+$ $+$ $+$ $+$ $+$ $+$ $+$ $+$		$\frac{1}{1}$		╉┹┺╋	+	+++	
		+/	18	\downarrow	+	+		$\left\ \cdot \cdot \cdot \cdot \cdot \cdot \right\ $		+	$\frac{1}{1}$		
		+	19	<u> </u>	$+\cdots$			+			μ	+-+	┢╍╘╺╘╺╘
		/	20		1	\downarrow		∦ii⊥i∔		+		╉╍╍╪	<u>↓</u> ↓↓↓↓
GS DO 3		1	21					\parallel				<u>i</u> i i	<u> </u>
	GSZ	2 1	22	ļuļ		ļui					PO	++++	
	GS	2 /	23			\downarrow	ļ	\parallel				4+	ļu
		1	24		4	<u></u>	+	Huui	R1 70 2			$+ \dots +$	RO
0000 007		1	25			\downarrow	+	Huui		4	<u> </u> 	<u> </u>	
GS PUZ, GS DO 2	0	1	26				fini	<u> </u>					ļ.
		2	R		+ FACTOR	111	-FACTOR	VALUE					Luui
		$^{2}4$	28	ļui		ļuļ	<u> </u>	RO	RI TO 5		ļ		L.
		2	29	Lui					BT, RESET			+	turi.
·	+	2	30		RI		RO						
·····		2	31		RO				MPLY				
			32										
		35	33				RO		MPLY				hui
		35	34			RO		RI					
		3	35		RO			ll i i i	MPLY				
		5	.36		RO				MPLY -				
		6	37		RO				RI 705				Luui
		46	38	111			RO		÷				
		46	30						RR				
		46	40	111				RO	RI				
			41										
		7	42				RO				<u> </u>		RI
L	i			<u> </u>				<u>, , , , , , , , , , , , , , , , , , , </u>	╺╋╼╹╶╫╼╉╴┇╶┚╼╋╼╿╶┠╶╇╼╢╍┠╶╇╼┆	┶╌╄┯╼┨═╌┸╼	┵┛┺╼╋┻┻╌╩		┙╺╋╌┛╺┠╶┠╶┠╸
[40						1				T
<u> </u>		+											

FIGURE 4. TYPE 604 PLANNING CHART

which a choice of a particular counter group to channel D is necessary. Thus, channel D is not wired to all counter groups, but only to those requiring this operation.

Operation 6: A/B (B>A/10)

This is used to compute the wind-axes coefficients, as the wind-axes airloads and moments are stored in registers and are available only to channels A and B.

Operation 7: $B \rightarrow$ General Storage 4

This is used in the initial data read-in to hold the run number in the 604 for transfer to a storage register on the next card cycle. The denominator term, qS_w , is also computed on this card cycle by picking up both calculate selectors 3 and 7 (q in A, run number in B, S_w in D, and nothing in E).

It should be noted that the 417AA counters are not converted prior to resetting except during checking runs (checking runs are made by listing all cycles through use of setup change switch number one). However, all counters are wired for conversion at print-out time. The manner in which this is accomplished is shown in Figures 3A and 3B.

The only other 417AA control of interest is the spacing control between runs. Prior to the calculation of each test point, the run number on the data card is compared to that stored from the previous point. If they are different, an automatic total cycle is initiated, which causes the tape to skip four spaces before starting the new calculation. The only purpose of this is to separate the different runs for ease in reading.

There is also a sequence control to stop the machine if cards in the program deck should be out of proper sequence. The wiring of this is obvious from the schematic diagram (Figure 3), and the fact that each card carries its own number in columns 77-78, and the number of the preceding card in columns 79-80. Cards are numbered from 0100 (columns 77-80) to 0041, 0100 being the data card, and the remaining 41 cards being the associated program deck.

Card-column assignments for factor read-in and the various control operations are shown in Table I.

					Тлві	ζeΙ	
			ĊA	ard (Column	Assi	GNMENTS
Column Numbers			Ι	tem			Value on Data Card
1-5	Channe	el A r	ead-ir	1 (X-	-5, negative	e)	\dots Dynamic pressure, q
6-10	"	В	"	(X-	-10, ")	Run number
11-15	"	D	"				\ldots Wing area, S_w
16-20	Counte	er grou	ıp 1 r	ead-i	n (X-20, r	negativ	ve) $\ldots a_u$
21-25	"	"	2	"	(X-25,	") ψ_u
26-30	**	"	3	"	(X-30,	") $\ldots \ldots n_1$
31-35	"	"	4	"	(X-35,	") $\ldots \ldots n_2$
36-40	"	"	5	"	(X-40,	") <i>c</i>
41-45	"	"	6	"	(X-45,	") $\ldots \ldots \ldots s_1$
46-50	"	"	7	"	(X-50,	"	$) \ldots \ldots \ldots \ldots S_2$
51-55 Column		"	8	"	(X-55,	") <i>r</i>
51-55 Column Numbers	" Channe	" -1 A #	8 I	" tem	(X-55,	")r Value on Program Card
51-55 Column Numbers 1-5 6-10	" Channe	el A r	8 I read-ir "	tem	(X-55, -5, negative	" e))r Value on Program Card Constant
51-55 Column Numbers 1-5 6-10 11-15	" Channe "	el A r B D	8 <u>I</u> read-ir "	" tem 1 (X- (X-	(X-55, -5, negative -10, "	" e)))r Value on Program Card Constant "
51-55 Column Numbers 1-5 6-10 11-15 56-57	Channe "	el A r B D A in	8 <u>I</u> read-ir "	" tem (X- (X- 	(X-55, -5, negative -10, "	" e)))r Value on Program Card Constant " Choose card or storage registe
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59	Channe " "	" el A r B D A in B	8 read-ir " nstruc	" tem (X- (X- 	(X-55, -5, negative -10, "	" e)))r Value on Program Card Constant " " Choose card or storage register """""""""
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59 60	Channe " " "	" el A r B D A in B D	8 read-ir " mstruc "	" tem (X- (X- tions	(X-55, 5, negative -10, "	" e)))r Value on Program Card Constant " " Choose card or storage registe """"""""
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59 60 61	" Channe " " "	" el A r B D A in B D E	8 read-ir " nstruc "	" tem (X- (X- 	(X-55, 5, negative -10, "	" e)))r Value on Program Card Constant " " Choose card or storage registe """""""""
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59 60 61 62-63	Channa " " " "	" el A r B D A ii B D E C to	8 <u>I</u> read-ir " mstruc " " "	" tem (X- tions	(X-55, 5, negative -10, "	" e)))r Value on Program Card Constant " " Choose card or storage registe """""""""" Choose card (blank) or counte Choose counter group Choose storage register
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59 60 61 62-63 64	Channe " " " " "	" el A r B D A in B D E C to C to	8 <u>I</u> read-ir " nstruc " o stora to cou	<i>tem</i> (X- (X- tions age in	(X-55, 5, negative -10, " 	" e)))r Value on Program Card Constant " " Choose card or storage registe """""""""" Choose card (blank) or counte Choose counter group Choose storage register Choose counter group
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59 60 61 62-63 64 65	Channo " " " " " " " " "	" el A r B D A in B D E C to C _t 1	8 Iread-ir " unstruc " " o stora to cou	tem (X- (X- tions age in unter up (0	(X-55, -5, negative -10, " 	" e)))r Value on Program Card Constant " " " " " " Choose card or storage register " " Choose card (blank) or counter Choose counter group Choose storage register Choose counter group Choose counter group Choose counter group Choose counter group Choose counter group Choose counter group
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59 60 61 62-63 64 65 66	" Channu " " " " " " Reset o Operat	" el A r B D A in B D E C to C _t 1 counte	8 <u>I</u> read-ir " mstruc " o stora to coust er grov	" tem (X- (X- tions age in inter up (0	(X-55, -5, negative -10, " 	" e)))r Value on Program Card Constant " " " " Choose card or storage registe " " " " " " " Choose card (blank) or counte Choose counter group Choose counter group Choose storage register Choose counter group Choose counter group Choose counter group Choose counter group Choose counter group Choose 604 operation
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59 60 61 62-63 64 65 66 67	" Channu " " " " " " " " " " " " " " " " " "	" el A r B D A in B C to C to C to counte	8 <u>I</u> ead-ir " " o stora to cou er gro	" tem (X- (X- tions age in inter up (o	(X-55, -5, negative -10, " 	" e)))r Value on Program Card Constant " " " " " " Choose card or storage register " " Choose card (blank) or counter Choose counter group Choose storage register Choose counter group Choose 604 operation Channel C shift control
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59 60 61 62-63 64 65 66 67 68	" Channu " " " " " " " " " " " " " " " " " "	" el A r B D A in B D E C to C _t 1 counte	8 <u>I</u> read-ir " " mstruc " " o stora to cou er gro … …	" tem (X- (X- tions age in inter up (o	(X-55, -5, negative -10, " 	" e)))r Value on Program Card Constant " " " " " Choose card or storage registe " " Choose card (blank) or counte Choose counter group Choose counter group Choose storage register Choose counter group Choose 604 operation Channel C shift control X prints out counters
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59 60 61 62-63 64 65 66 67 68 69	" Channe " " " " " " " Reset o Operat Shift Print Data r	" el A r B D A in B D E C to C _t 1 counte tion ead-in	8 <u>I</u> read-ir " " mstruc " " o stora to coutor groo 	" tem (X- (X tions age in nuter up (o	(X-55, -5, negative -10, " 	" e)))r Value on Program Card Constant " " " " " Choose card or storage registe " " " " " " Choose card (blank) or counte Choose counter group Choose counter group Choose storage register Choose counter group Choose counter group Choose 604 operation Channel C shift control X prints out counters X loads data in all counters
51-55 <i>Column</i> <i>Numbers</i> 1-5 6-10 11-15 56-57 58-59 60 61 62-63 64 65 66 67 68 69 70	" Channe " " " " " " Reset of Operat Shift Print Data r Clear	" el A r B D A in B D E C to C _t 1 Counte tion ead-in all co	8 <i>I</i> read-ir " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "	" tem (X- (X tions age in inter up (0 grou	(X-55, -5, negative -10, " 	" e)))r Value on Program Card Constant " " " " Choose card or storage registe " " " " " " Choose card (blank) or counte Choose counter group Choose storage register Choose storage register Choose storage register Choose counter group Choose counter group Choose 604 operation Channel C shift control X prints out counters X loads data in all counters (not used)

Appendix

The procedure used to design the control panel wiring described in the body of the paper may be extended to apply to any problem within the selector capacity of the machine. In a similar manner, general-purpose panels may also be built. The distinguishing feature of a general-purpose panel is that the numbers in any 417AA counter group are available, under card control, to any of the read-in stations being used in the 604. There are a variety of such panels in use at Northrop, varying in the manner in which numbers are presented to the type bars, the use of net-balance conversion, whether or not the 417AA counter "read out and reset" feature (an 80 code on the CPC Model I) is wired, the manner, if any, in which numbers on cards may be read directly into 417AA counters, and in the number of counter groups available and the number of digits in each. Use of one of these general-purpose panels, together with a general-purpose 604 panel, allows card programming of the simple, infrequent problem in a manner similar to that used for the Model I. Problems worked on such panels include solutions of sets of simultaneous differential equations, inversion of low-order matrices (within the storage capacity of the machine), calculation of the matrix for the spanwise airload distribution on a swept wing, etc.

However, if a problem is very complex or if the problem is to be repeated time after time, it is usually found that the additional planning time involved in a special set of panels is more than paid for by the decreased running time. As an example, a more sophisticated version of the Monte-Carlo problem described by W. W. Woodbury at the November, 1949, Seminar was run using a single card per process. In this problem, on each card two types of "scattering" were considered, two sets of "capture" probabilities were compared, two different mean free paths were considered, and tests were run on the geometrical limits on the problem. A choice was made of each of the alternatives as a function of the location of the particle within the space considered by the problem. The necessary tests were about evenly divided between the 417AA and the 604. Almost every selector in the machine was used. This problem was literally impossible to perform on the Model I because of card handling difficulties, lack of counter flexibility, and elapsed time limitations.

Probably the most spectacular examples of elapsed time savings due to the use of special panels on the Model II are in the field of structural calculations. A typical ratio for most routine calculations, such as wing and fuselage section properties, show an elapsed time savings of about 15 to 1 over the equivalent computations using the maximum effective number (usually about four) of operators at desk calculators. The ratio over the same problem on the Model I is about 3 to 1 in favor of the Model II, including considerably less card handling with its associated possibility of accidental error. The elapsed time savings have allowed us to compute structural analyses, rather than to use "best guesses" for early preliminary design purposes.

DISCUSSION

Mr. Bell: We have had some experience in doing windtunnel work, and have done several things to help remedy the problem of getting enough numbers in the machine. By spread loading, seven numbers can be put in the seven counters. Also, two more numbers can be read into channels A and B, and hence into the 604. The two numbers in the 604 can then be added and their sum stored. On the following card cycle one of the numbers is wiped out by reading nothing in and storing the one number. Then, a little later, that one number can be subtracted from the sum, thus giving the remainder.

Mr. R. W. Smith: We have the Model I CPC, and we find by using a digit selector as a shift unit we can load in 13 numbers on one card, loading two five-digit numbers into a counter at a time. To avoid difficulty with negative numbers, we add a constant to the numbers so that they will all be positive.

Mr. Lowe: I wonder if Mr. Lesser has made an analysis as to just how much time was saved in this wind-tunnel problem by wiring this Model II control panel versus using, let us say, the simplest sort of general-purpose setup.

Mr. Lesser: Actually it is about a two-to-one saving.

Dr. Grosch: How many cards are in a program deck?

Mr. Lesser: Forty-one, including two blank cards that I found necessary afterward.

IBM Card-Programmed Electronic Calculator Operations Using a Type 402-417BB and 604-2

MARTHA KENYON

BRUCE OLDFIELD

HARLEY E. TILLITT

U.S. Naval Ordnance Test Station, Inyokern

×

DURING the past few years, the art of technical computation using punched card methods has advanced very rapidly, and the fact that the International Business Machines Corporation has sponsored five Seminars on Computation in the last three years is evidence of this advancement. Further evidence no doubt is in the minds of many of the members of this 1951 Seminar, devoted to the IBM Card-Programmed Electronic Calculator, who were in 1948 performing calculations with the IBM Type 601 Electric Multiplier. In the space of three years most IBM installations have made several changes in equipment, and are very pleased with the rapid progress in the development of electronic computing equipment.

In rather rapid succession the IBM Type 602 Calculating Punch, the Type 603 Electronic Multiplier, the Type 602-A Calculating Punch, the Type 604 Electronic Calculating Punch and, finally, the Card-Programmed Electronic Calculator were each the "latest thing" in IBM computing machinery. In a comparatively short time after the CPC was introduced, increased storage was available, and now improved machines known as the Model AA 402-417 and Model 2 604 are being manufactured.

In December, 1950, a CPC using a Model 2 604 was put into operation at the Naval Ordnance Test Station, Inyokern, California, and has been found to be superior to the standard CPC for many problems. It is the purpose of this presentation to describe the special features of the Model 2 604 and to demonstrate how these have been employed in two computational problems. The first is a data reduction method for which the required computational time on the standard CPC was reduced by one-half, using the Model 2. The second is the solution of certain differential equations, the numerical methods for which were found to be better suited to the Model 2 than the standard 604 (Model 3).^a

DATA REDUCTION PROBLEM

As one might expect at an establishment where rockets and guided missiles are tested, efficient and rapid data reduction procedures are very important aspects of the development program. One particular reduction procedure for which the Model 2 604 was best suited was that of determining the space coordinates of a missile in flight, making use of elevation and azimuth angles as recorded by cinetheodolites.

The particular mathematical procedure was such that the CPC method which appeared to be the most efficient in terms of computation time per point made it necessary to break the procedure into more than one part. In addition, the 604 control panels as used in each CPC run were so laden with wires that it was time-consuming for checking purposes. Although fully wired control panels are certainly nothing new to CPC and 604 users, it may be interesting to note that one of the 604 control panels originally used in this problem contained approximately 330 feet of wiring.

In an effort to improve the CPC procedure for the data reduction and other problems, modifications of the basic 604 were undertaken. The modifications were based on operational experience gained at Inyokern, as well as at other CPC installations. The Model 2 604, as the advanced version has been designated, differs from the basic CPC 604 in the following ways:

- 1. Three program levels.
- 2. Program pickup.
- 3. Additional calculate selectors.
- 4. Program repeat selectors.
- 5. Repeat delay.
- 6. Program expanders.
- 7. Program source filters.
- 8. Program suppression filters.
- 9. Channel C shift.

Before indicating how the special Model 2 features were used in the data reduction work previously mentioned, it should be emphasized that this discussion is not intended to evaluate the mathematics of the method but to demonstrate how a particular procedure was actually fitted to the machine. Several mathematical approaches to the problem are possible, and different ones have been used at the Naval Station, depending in part upon the available facilities. For

^aThe first problem to be discussed was the responsibility of Mr. Bruce Oldfield, the second was that of Miss Martha Kenyon.

example, the method used with a 602 was different from that using a 604. Likewise, the CPC permits the use of a different mathematical approach than the 604 alone because of the possibility of sequencing calculations.

The method now in use^b is a multi-station solution where, in our work, elevation and azimuth angles from three cinetheodolites located along the range line are given for different times during the missile's flight. Each of the three cinetheodolites, therefore, establishes a line in space, and the missile is at their intersection. Actually, two lines are sufficient to locate the point in space, but an intersection is rare in practice because the data are not exact. There are several reasons for errors in data, including those associated with the many components of the system, the misalignments of the theodolites and the measurement of the film. The CPC solution corrects the angles for certain errors of the type just mentioned and makes use of a method of least squares to locate the most likely point in the region where the three lines nearly intersect.

As a check on the machine computation, on certain constants associated with the station coordinates, as well as on the original data, residual angles are determined. From the point just located, the problem is done in what might be called the reverse order, and the elevation and azimuth angles are computed for each of the three stations which would be required to determine that point. The residuals are the differences between these recomputed angles and the corrected original angles.

The present method, which is designed for a computer capable of considerable sequencing, such as the CPC, makes use of certain weighting factors in the computation. The cinetheodolite stations are widely separated from each other along the range line; hence, the three missile-to-station distances are different. Since for any one point in space the data from far stations does not have the same importance as that from near stations, weighting factors, which are functions of the missile-to-station distances, are determined from the space coordinates of the missile at the last computed point and are introduced into the computational procedure.

^bThis method was submitted to the IBM Section by Dr. John Titus of the Station's Assessment Division.

The computation cycle accomplishes four steps:

- 1. Applies corrections to original elevation and azimuth angles.
- 2. Determines the space coordinates of the missile.
- 3. Computes residual angles.
- 4. Stores results as required to weight data for the next point.

The equations used in the first three steps of this reduction process are shown in Figures 1, 2, and 3, respectively.

As previously mentioned, using the original equipment, the most rapid and efficient method developed at the Naval Station required four runs on the CPC. The first CPC run computed certain corrections and converted the original angles to their first quadrant equivalents. On the second CPC run, a card for each point which contained the sines for each of the six angles as recorded at the three cinetheodolite stations was computed and punched out on the Type 521 Punch Unit. The third CPC run computed the cosines. These sine and cosine cards were then placed in an instruction deck, and computations as described in Figures 2 and 3 were completed. The corrections which were made to the angles in this original procedure were not so complete as those now employed.

Although this original CPC procedure was superior to that used previously, which required nearly twenty 604 control panels or about forty 602 panels, it was preferable to make only one run since such a procedure not only saves time but decreases the chances for errors. By using the greater capacity of the Model 2, only a single run is necessary. Figure 4 shows the operations which were wired in the 604 control panel.

The total machine time for the Model 2 procedure is approximately 2.5 minutes, and uses a 295-card instruction deck which includes three data cards. This time is much better than that obtained with the former CPC method using the Model 3 604, which required about 5.5 minutes per point and, as previously mentioned, did not compute all of the corrections currently used. A breakdown of the problem in terms of groups of instruction cards is shown in Figure 5.

$$A = A' + d \sin A' \tan E' + \theta$$

$$E = E' + d \cos A' - \frac{X^2 \tan E'}{114.6} - 2.22(10^{-7}) \sqrt{(x - X_i)^2 + (z - Z_i)^2} (1 - .0106 E')$$

where

$$A' = S_0 + \Delta SC_1 + [(X_p - X_0)C_3 + C] \sec E' \left[1 + \frac{(Y_p - Y_0)C_4 \tan E'}{57.296} \right] + e_0 \cos (S_0 + \phi) + O_s$$

and

$$E' = E_0 + \Delta EC_2 + (Y_p - Y_0)C_4 + O_E$$



$$\begin{array}{rcl} A_{11}X + A_{12}Y + A_{13}Z = C_{1} \\ A_{21}X + A_{22}Y + A_{23}Z = C_{2} \\ A_{31}X + A_{32}Y + A_{33}Z = C_{3} \end{array}$$
where
$$\begin{array}{rcl} A_{11} &= \sum_{i=1}^{3} \frac{1 - a_{i}^{2}}{p_{i}^{2}}, & A_{21} = A_{12} = -\sum_{i=1}^{3} \frac{a_{i}b_{i}}{p_{i}^{2}} \\ A_{22} &= \sum_{i=1}^{3} \frac{1 - b_{i}^{2}}{p_{i}^{2}}, & A_{31} = A_{13} = -\sum_{i=1}^{3} \frac{a_{i}c_{i}}{p_{i}^{2}} \\ A_{32} &= \sum_{i=1}^{3} \frac{1 - c_{i}^{2}}{p_{i}^{2}}, & A_{31} = A_{13} = -\sum_{i=1}^{3} \frac{b_{i}c_{i}}{p_{i}^{2}} \\ A_{33} &= \sum_{i=1}^{3} \frac{1 - c_{i}^{2}}{p_{i}^{2}}, & A_{32} = A_{23} = -\sum_{i=1}^{3} \frac{b_{i}c_{i}}{p_{i}^{2}} \\ A_{33} &= \sum_{i=1}^{3} \frac{1 - c_{i}^{2}}{p_{i}^{2}}, & A_{32} = A_{23} = -\sum_{i=1}^{3} \frac{b_{i}c_{i}}{p_{i}^{2}} \\ a_{i} &= -\cos E_{i} \cos A_{i} \\ b_{i} &= \sin E_{i} \\ c_{i} &= -\cos E_{i} \sin A_{i} \\ i \text{ takes the values 1, 2, and 3.} \end{array}$$

$$C_{1} &= \sum_{i=1}^{3} X_{i} \left(\frac{1 - a_{i}^{2}}{p_{i}^{2}}\right) - \sum_{i=1}^{3} Y_{i} - \frac{a_{i}b_{i}}{p_{i}^{4}} - \sum_{i=1}^{3} Z_{i} - \frac{a_{i}c_{i}}{p_{i}^{2}}, & p_{1}^{2} = (x_{j-1} - X_{1})^{2} + (y_{j-1} - Y_{1})^{2} + (z_{j-1} - Z_{1})^{2} \\ C_{2} &= -\sum_{i=1}^{3} X_{i} - \frac{a_{i}b_{i}}{p_{i}^{2}} + \sum_{i=1}^{3} Z_{i} \left(\frac{b_{i}c_{i}}{p_{i}^{2}}\right), & p_{2}^{2} = (x_{j-1} - X_{3})^{2} + (y_{j-1} - Y_{2})^{2} + (z_{j-1} - Z_{2})^{2} \\ C_{3} &= -\sum_{i=1}^{3} X_{i} - \frac{a_{i}c_{i}}{p_{i}^{2}} - \sum_{i=1}^{3} Y_{i} - \frac{b_{i}c_{i}}{p_{i}^{2}} + \sum_{i=1}^{3} Z_{i} \left(\frac{1 - c_{i}^{2}}{p_{i}^{2}}\right), & p_{3}^{2} = (x_{j-1} - X_{3})^{2} + (y_{j-1} - Y_{3})^{2} + (z_{j-1} - Z_{3})^{2} \\ \end{array}$$

Figure 2. Equations for Determination of Coordinates

$$RA_{i} \cong 3432 \frac{\tan A_{i}^{(m)} - \tan A_{i}^{(o)}}{1 + \tan A_{i}^{(m)} \tan A_{i}^{(o)}}$$
where

$$RE_{i} \cong 3432 \frac{\tan E_{i}^{(m)} - \tan E_{i}^{(o)}}{1 + \tan E_{i}^{(m)} \tan E_{i}^{(o)}}$$
(m) means "measured angle"

$$\tan A_{i}^{(m)} = \frac{c_{i}}{a_{i}}$$

$$\tan A_{i}^{(o)} = \frac{z - Z_{i}}{x - X_{i}}$$

$$\tan E_{i}^{(m)} = \frac{b_{i}}{\sqrt{a_{i}^{2} + c_{i}^{2}}}$$

$$\tan E_{i}^{(o)} = \frac{(y - Y_{i})}{\sqrt{(x - X_{i})^{2} + (z - Z_{i})^{2}}}$$



	P	rogra	m
	Operation	Level	Remarks*
1.	A + B = C	A	
2.	A - B = C	А	
3.	-A + B = C	А	
4.	$A \cdot B = C$	А	
5.	$-A \cdot B = C$	А	
б.	$(A_1 \cdot B_1) \pm (A_2 \cdot B_2) \pm (A_3 \cdot B_3) = C$	A	Uses results of previous step(s).
7.	Transfer:	А	May drop high-order digit if
	Result in C to A		desired.
	Result in C to B		
	Result in C to 604 storage		
8.	$1 - B^2 = C$	А	
9.	A/B = C	В	
0.	$\sqrt{A} = C$	В	Final answer in B if needed.
1.	Reduction of angles to first quadrant	В	Program level change to compute sine or cosine.
2.	Convert degrees to radians, keep correct sign for sine or cosine and compute function	C	Functions to 7 decimals.

*All operations are 10×10 except computation of sine and cosine.

FIGURE 4. OPERATIONS WIRED IN 604 PANEL

Group	Card Numbers	Operation
A	1- 40	Make corrections. Compute sine and cosine for angles at first station.
В	41-80	Same as Group A but for second station.
С	81-120	Same as Group A but for third station.
D	121-126	Compute direction cosines.
Ę	127-129	Compute tangents of measured eleva- tion angles to be used in computation of residuals.
\mathbf{F}	130-139	Compute weighting factors, p_i^2 .
G	140-206	Compute coefficients of matrix.
Н	207-210	Compute tangents of measured azi- muth angles to be used in computation of residuals.
Ι	211-237	Solve matrix.
J	238-287	Complete computation of residuals.
K	288-295	Print results and store quantities as required for next point.

FIGURE 5. CARD-PROGRAMMED OPERATIONS

DIFFERENTIAL EQUATIONS PROBLEM

Whereas the work just presented is a routine data reduction operation, the second problem is of a theoretical nature in which it was required to solve some differential equations by numerical means in order to determine the desired geometry for certain components of an ordnance item.

The sequential nature, the accuracy requirements and the comparative simplicity of each particular step suggested that it was a very good problem for the CPC. The main difficulty was associated with the number and length of operations involved. The problem was considered for a CPC with a standard 604 but was found to be much more easily set up for the Model 2. The operations which were required on the 604 are shown in Figure 6.

The 180 program steps were treated as three separate sets of 60, and there was no attempt to go through all 180 at once, but instead to pick up the appropriate program level. Operations which were used together were grouped together. However, it was not necessary to have consecutive operations on the same program level, provided the same program steps were not required on both levels. The usual wiring was to plug the program pickup source into the common of the controlling calculate selector and the transferred side connected to the program pickup.

Much use was made of the program suppression filters. Programs were frequently suppressed on plus, minus, zero or non-zero balance or on pairs of these. Some programs were also suppressed without balance test through the normal points of calculate selectors. To prevent back circuits, the suppress on minus, for example, was wired to one "in" hub of a program suppress filter and suppress on zero to the other "in" hub. The "out" hub was then wired to program step suppress to control suppression of that particular step on either of the above conditions, without affecting any other step wired to those suppress sources.

1.	Addition.		
2.	Subtraction.		
3.	$8 \cdot 8$ multiplication.		
4.	$8 \cdot 8$ division.		
5.	\sqrt{x} to 5 decimal places with the	requirement that $\sqrt{-x}$	= 0.
6.	Transfer C to A.		
7.	Transfer C to B.		
8.	½ adjust.		
9.	a^x for $0.1 < a < 1.8$ with x havin	g one of three values:	
	1.25, 1.60, 1.80.		
	$a^{x} = 1 + x \log a + \frac{(x \log a)^{2}}{24}$	$+\frac{(x\log a)^3}{21}+\cdots$	
	where:	31	
	$\log a = (a-1) - \frac{1}{2}(a-1)^2 + \frac{1}{2}$	$\frac{1}{3}(a-1)^3 - \dots$	2 > a > 0.
10.	$ \begin{array}{l} A = 10.29 \text{ B} \\ A = 500.00 \end{array} $ whichever is	s larger.	
11.	$C = [10 \ (x^{-1.6} - x^{-1.8})]^{\frac{1}{2}}$	if $x \leq b$	
	C = .6581	if $x > b$.	
12.	$D = .7225 \left[1 + (x-1)^3 \right]$	if $0 < x < 1$	
	$D = \frac{.7225}{1+2 \ (x-1)^2}$	if $1 < x < \infty$.	
13.	Test to check agreement betwee	n the first and second a	pproxima-
	tions for a variable, with non-a	greement to cause the a	accounting
	machine to stop, allowing an ad	lditional deck of iteratio	on cards to
	be inserted.		

Figure 6. Operations Wired in 604 Control, Panel,

A feature of the machine mentioned previously, but not used in the first problem described, is the channel C shift. This feature is essentially a single-position, five-level "field selector." The level to which this selector transfers is dependent on the digit punched in the channel C shift control column of the instruction card read in the accounting machine. For this particular problem each of the channel shift hubs was wired to the appropriate position of the usual 604 "shift unit, and by this means the instruction card could be made to control a shift for transfer from C to A or C to B within the 604.

The initial values for any point were determined during the previous point's calculation. If desired, the intermediate results for any point, which were ordinarily stored to be used for the next point's calculation, could be punched out on the 521 by inserting a special deck at the end of the regular deck. This was a very useful part of the initial planning, for in cases of machine error it was not difficult to return to some previously computed point and begin again. The frequency of punch-out depended upon the expected machine error rate at the time.

DISCUSSION

Mr. Lesser: How many cards did you end up with? Mr. Tillitt: One hundred eighty.

Dr. Yowell: I notice Mr. Tillitt says he has obtained a reduction in time of a factor of two with his half of the machine. Mr. Lesser reports the same. Does that mean that the rest of us can expect a reduction of a factor of four with a full machine?

Dr. Hurd: I think we need more experience to answer that question.

The Combomat

JOHN D. MADDEN

The RAND Corporation

涿

THE COMBOMAT is a system installed at RAND whereby customers with computation jobs which fit into a certain loosely-defined class may program the calculations directly for the IBM Card-Programmed Electronic Calculator. That is, any person in the company who has a job in this category may code it for one of the CPC setups and submit it directly to the IBM Section for processing, bypassing the procedure preparation which normally takes place within the Numerical Analysis Department. This class of jobs considered appropriate for the Combomat will be discussed later.

Briefly, the process works as follows:

- 1. The customer marks mark-sense cards to accomplish the calculation he requires. These cards are printed for Combomat use.
- 2. He submits these, along with a form which describes the order in which the cards are to be used, to the IBM Section.
- 3. A machine operator who is familiar with Combomat work "senses" the cards and expands the deck to a convenient size.
- 4. The cards are run through the CPC and the printed results delivered to the customer along with his original cards.

In theory, the customer obtains all the information required concerning machine operation and coding from a 14-page manual prepared by the Numerical Analysis Department. However, in practice a consultation is usually necessary before a customer's first job is in shape for IBM processing. Quite often the single briefing suffices, and the ideal arrangement is reached where the customer submits the cards directly to the IBM Section and no analysis time is required by the Numerical Analysis Department.

The derivation of the name Combomat is somewhat interesting. Several years ago when Northrop Aircraft had the prototype machine, it was sometimes called the "Combo." The "mat" suffix comes from words like "laundromat." Instead of paying for both the use of laundry machinery and the services of people who work in the laundry, in a "laundromat" one pays only for the use of the washing machine and the clothes drier. In using the Combomat, rather than employing the services of an analyst plus the "Combo," one buys only the use of the CPC plus a minimum of operator time. Thus, a Combomat system makes possible the handling of a tremendous number of machine jobs without putting any additional strain on an analysis staff.

At RAND there is a particular need for this kind of service. Not many of the various research divisions have a relatively steady hand-computing work load throughout the year. The load is apt to be light much of the time but extremely heavy at several times during the year. Hence, few of the divisions can afford to keep a large staff of hand computers. The Combomat provides a solution to this problem. It is believed that the Combomat is, in any case, much more economical than a central pool of hand computers because the CPC's are installed at RAND, and the regular continuing machine work suffers very little by interrupting it to do Combomat jobs. Every effort is made to get Combomat work done in 24 hours. Generally, jobs which are submitted in the afternoon are processed and out by the next morning. This is made possible by the fact that the RAND IBM Section works three shifts and has two CPC's with two identical Combomat setups.

The setup currently in use employs an eight-place floating decimal, where the power of ten is stored beside the number; that is, the actual power of ten plus fifty is stored beside the number. The power of ten is translated fifty units to make certain that any algebraic sign associated with the number and its power of ten represents the sign of the number. This limits the true powers of ten to the range -50 to +49.

For Combomat work the CPC is represented as having only sixteen words of storage. The manual does not mention the accounting machine storage. This omission is made for two reasons. First, the instructions for use of the system would be somewhat more complicated if an explanation of the two kinds of storage were required. Also, some storage positions must be available to take care of sub-routines which the customer may need.

Operations available to the Combomat customer are multiplication, division, addition, and transfer. The last operation is used for shuffling numbers from one storage to another or from a card to storage. Allowance is made for reading the answer from one card into channel A, channel B, or both for the succeeding card. The setup actually has several other operations, but it was deemed wise to restrict the operations allowed in Combomat work in the interest of simplicity. In addition to the four Combomat operations, some additional functions of the setup are:

- Operations which compute (AB + K), (KB + A), (A/B + K), and (K/B + A), where A and B refer to the conventional channels, and K must be read from the card.
- 2. Two special operations which are used in obtaining common logs and antilogs, respectively. One of them puts the power of ten associated with a number on the number part of channel C, and the other takes a number out of scientific notation separating the whole number and decimal parts.
- 3. Provision for switching from the ordinary control field to an alternate control field on the basis of a negative balance obtained in the 604. Some means of making a decision is the operation requested most frequently by customers. With this current method of decision-making, a seven-column alternate control field is required to make use of the decision. This is not possible with the system of submitting data now in use.

By far the knottiest problem associated with installing a Combomat system is that of writing a usable manual to be distributed to prospective customers. It is extremely difficult for one familiar with the machines to write a manual using language which will be understandable to the layman; and yet, unless the writer understands the functions of the setup, he is not really qualified to do the job. The current manual is a third attempt and is still a "preliminary draft." The first draft was tried on some selected members of the Numerical Analysis Department. They prepared a lengthy list of suggested additions and criticisms, all of which were then incorporated into the second version. This one received a more general distribution throughout the company, and a second list of additions and criticisms was obtained. This second list was screened carefully, and all of the changes considered desirable were made. This gave the present manual.

Several questions arise in installing the system. Perhaps the first is how the raw data are to be submitted. Three methods of presentation were considered: mark-sense cards, cards key punched by the customer on punches spotted in the various departments, and manuscripts prepared by the customer for key punching by IBM Section card punch operators. Each of the methods has its advantages and disadvantages. The mark-sense card system was chosen because it was felt that this would require a minimum of effort on the part of the IBM Section and disrupt least the flow of regular work. One disadvantage connected with using mark-sense cards is that on the first job submitted by a particular customer difficulty with the "sensing" is often encountered-his marks are either too light or he fails to put some of his marks in the ovals provided. This is a problem which resolves itself as the customer submits more and more jobs. Also, mark-sense cards effectively add a restriction of using only 27 columns. It was found necessary to restrict reading in from the card to only one channel because of this. Requiring one and only one mark per marksense column, so that blank column and double punch detection may be used, further restricts the system. The idea of spotting card punches around the RAND building was eliminated because it was suspected that too much time would be consumed in instructing people in the use of the punch and in removing cards jammed in the machine by inexperienced operators. Further, it was felt that customers would make more errors key punching than card marking. The manuscript approach was discarded because it did not seem in keeping with the aim of placing as little of the burden of the work on the IBM Section as was absolutely necessary.

Another question is how the answers are to be presented. Detail printing is a natural product of CPC computation so that it is certainly one "painless" method of returning results. Selective listing may prove more useful than listing all cards, however. This has been handled informally through notes on the expansion form. Another possible way of transmitting answers is by summary punching. It was decided to install the system without mentioning this possibility and to see if there was much demand for it. To date the demand has been slight.

Another problem which arises is that of devising a surefire, completely mechanical system whereby the user can communicate to the operator the order in which the cards are to be used. At the outset this sounds like a trivial problem, since one might argue that a sequence number will suffice. In practice, however, to minimize card marking some of the sub-decks are used over and over again, while some are used less frequently, and some may be used only once. This makes it imperative to have a communication scheme which is completely flexible. It was found that the section of the manual which explains this scheme, the expansion form section, was the most difficult to write. The word "expansion" is used because the form gives the operator a clue on how many copies of which sub-decks to prepare in order to have a deck of convenient size for running the job. This form actually constitutes a translation of the flow diagram which shows the structure of the calculation. Concentration on making this section clear "mushroomed" it into the longest section in the manual. Further, more comments of "unclear" have been received from users regarding this than regarding any other section. It is hoped that this is due to the content of this portion of the paper rather than to how it is written.

A final consideration in installing the system is to what extent sub-routines are to be used. By sub-routines are meant decks already prepared to compute particular functions. The use of these decks is encouraged. In fact, almost every Combomat job processed uses at least one of the subroutines. The library of such decks was started with a few very basic ones, and some have been added as the need has arisen.^a Sub-routines currently available to Combomat customers include $\log_{10}x$, $\log_e x$, 10^x , e^x , \sqrt{x} , $\sqrt[3]{x}$, $\sin x$ (x in degrees), $\sin x$ (x in radians), $\cos x$ (x in degrees), $\cos x$ (x in radians), $\sin^{-1}x$ (degrees), $\sin^{-1}x$ (radians), $\cos^{-1}x$ (degrees), $\cos^{-1}x$ (radians), $\tan^{-1}x$ (degrees) and $\tan^{-1}x$ (radians).

A few random observations about the system follow.

It is emphasized that the name, Combomat, refers to a system rather than to a particular machine setup—that the setup used may be as simple or as complex as necessary to meet the needs of the installation using the system. One group may feel that a floating decimal with its coding simplifications is necessary, while another may feel that it is willing to substitute log and exponential operations for the floating decimal features. A Combomat system may be used equally well with either setup.

Just as some jobs are better suited for a parallel type of calculation rather than a serial type, some jobs are better suited for Combomat work than others. The ratio of the amount of input required to the amount of arithmetic required is a good index of the suitability of the job for Combomat computation. If this ratio is small, the job probably may be done efficiently using a Combomat system. Of course, this is a fairly good index for judging all CPC jobs, but its value is even more critical in Combomat work, because only one number enters the machine at a time. Pilot studies for large machine computations, iterations, and some kinds of small parameter studies are best suited for the Combomat. As far as volume of work is concerned, it is felt that jobs which fit in the large hand computation or small machine computation categories go most efficiently in the Combomat system.

It has been observed that on some Combomat jobs, especially the smaller ones, it pays to do the job "stupidly." That is, it pays in elapsed time with negligible loss in machine time to use the first workable program which occurs to the coder. He may well waste considerably more time than the job warrants in searching for an optimum program. This is emphasized in the manual.

It is felt that the system would be very worthwhile even if jobs were accepted from members of the immediate department only. Many jobs are received which in the past have been considered border-line cases where the calculations may be done about equally efficiently on either desk calculator or IBM machinery. At present, more and more of these jobs are finding their way to the CPC within the framework of the Combomat. This occurs because the analyst may do a little rapid coding and "wash his hands" of the job until he is presented with the results.

In April, 191 hours of CPC time were logged on Combomat jobs exclusive of the time logged on work for members of the Numerical Analysis Department. Note that this is almost eight full 24-hour days. In May, only 65 hours of such time were logged. The 191 hours were due largely to only two contributors, while the 65 hours represented work from more customers.

An example of a small parameter study which was done efficiently with the Combomat is the following. It was required to compute

$$F = P_1 \left[1 - (1 - C_1 F_1 F_2)^{P_2} \right],$$

where $F_1 = e^{\frac{-P_3 F_3}{C_2}}$,
 $F_2 = e^{\frac{-P_1 P_4}{C_3 F_1}}$,

and where F_3 is a polynomial function of P_1 . Here the *C*'s indicate constants, the *F*'s indicate computed functions, and the *P*'s indicate parameters. The routine involved evaluating F_3 , then F_1 , then F_2 , and finally *F*. This was done for 3,660 parameter combinations. The job as coded by the customer took about thirty machine hours. A similar job done by hand computers in the user's division took 280 man hours. This job was longer than what is believed to be the optimum length for a Combomat job, although it does show a very favorable ratio of Combomat time to hand-computing time.

Another example of an efficient Combomat job is the following. The customer required calculation of

$$\phi_L = \sum_{j=L+1}^{\infty} A_j \rho^{j-L-1}$$

and $\phi_L^* = \sum_{j=L+1}^{\infty} j A_j \rho^{j-L-1}$,

where $A_j = (2\eta A_{j-1} - A_{j-2})(j+L)^{-1}(j-L-1)^{-1}$, $A_L = 0$, and $A_{L+1} = 1$.

Here 16 combinations of η , ρ , and L were used. The customer set up the problem so that the inputs were read in, and then a computation deck was run over and over again, each time computing terms for each of the summations. This process was continued until a final term was less than a specified value. Summations for each parameter set took approximately ten terms to reach the desired accuracy. One hour of CPC time was logged on this job, and it is believed

^aMr. Cecil Hastings, Jr., of The RAND Corporation has been of invaluable aid in developing efficient approximations for needed sub-routines.

that it would have taken much longer doing the work on desk calculators. The fact that this calculation was of a recursive nature made it extremely easy for the customer to code as well.

DISCUSSION

Dr. Hurd: This paper represents a great contribution to the art of automatic computing.

Mr. Tillitt: You remarked about running out of space in the mark-sensed card. You might consider turning it over and getting 27 more spaces.

Mr. Madden: That is a good point which we haven't considered.

Dr. Berkowitz: What is the cost to the research people in this setup? In other words, how does the time they spend writing up a program compare with the time that they would spend giving a problem to the analysis group to program? The customer has two choices: he can either code it himself or come to you and let you code it. It is going to take him some time to explain the problem to you. Is this time of the same order of magnitude as the time it would take him to encode the problem for himself?

Mr. Madden: I think that is likely.

Dr. Hurd: Is there anybody else who is using similar ideas and would like to discuss them at this time?

Mr. Cheydleur: We are not using mark-sensed cards, but we are using more and more systems that are flexible and that enable our customers to do more of their own coding.

Professor Verzuh: At MIT "A Guide for Coding" has been prepared and distributed to allow users to prepare their own sets of instructions for the Whirlwind. The machine is arranged in such a fashion that codes can be printed out. This has facilitated detecting errors in coding.

Dr. Hurd: Thank you. As you know, it is standard operating procedure to print operation codes and intermediate results on the CPC while the calculation is in progress.

Mr. Lowe: It has been our practice to select out those of our accounting machine operators in different divisions who evidence particular interest in the computing work. We then bring these people into our computing group and have them work with us for a few days to a week. In that period they actually come to understand quite a bit about the way we normally code for the machine and the use of the machine; they go back and code their problems for us from then on. However, under these circumstances we always reserve the right to edit, audit and check over their programs.

Mr. Lesser: It is easy to train a selective few of the customers so they can get the "feel" of the machinery.

The IBM Type 604 Electronic Calculating Punch As a Miniature Card-Programmed Electronic Calculator

PAUL T. NIMS

Chrysler Corporation

X

IN ENGINEERING work there are a great many computations carried out to slide-rule accuracy which require the repeated evaluation of rather simple formulas. If an IBM Type 604 Electronic Calculating Punch is available, there is a possibility of adapting it for this work and making a small sequence-controlled calculator with storage and all the other features of larger machines. With the cooperation of the Statistical Department of the Chrysler Corporation, we, in the Engineering Division, have wired a special set of control panels for their machine to use as an interim calculator until a larger card-programmed machine is delivered. Besides doing useful problems, this machine has served as a training ground for programmers and operators. This training was a necessary preliminary to the use of the Model II Card-Programmed Electronic Calculator, which was recently delivered.

Card Programming

The IBM Type 604 Electronic Calculating Punch is normally used to perform identical simple arithmetic operations on data in a series of cards placed in the hopper. However, the flexibility of the control panel wiring is such that it is possible to operate "card-controlled," that is, to perform a different arithmetic operation on each card, depending on operation codes punched in the card. Such operation permits sequential calculation, i.e., following in order all of the arithmetic steps required for the evaluation of a quantity given by a complicated algebraic formula with given parameters. The sequence can then be repeated for other values of the parameters.

As wired for this purpose, the 604 is capable of receiving (at one time) 7 five-digit numbers. These numbers are entered in storage registers A, B, D, E, F, G, and H but not in the output register C. When an operation code is punched in a card, the factors in A and B are operated upon and the result placed in the output register C as well as in storage register B. The result in B is thus immediately available for the next calculation. The result in C may be punched in

the card or stored in one of registers D, E, F, G, or H but not both at the same time.

On the next card, a new factor may be entered in A, or transferred to A from a storage unit, another operation performed on it and the quantity in B, and the result punched or stored.

Operations

The four fundamental operations are addition, subtraction, multiplication, and division, and they are performed with due regard for the location of the decimal point (to be discussed later) and for the signs of the quantities. In addition, it is possible to shift, i.e., multiply the result of the above operation by 10 without using another card cycle; to look up entries in a table of functions punched on successive cards; to obtain square roots by an iterative process; to stop the machine when unwanted negative answers occur; and to stop the machine if a check quantity is greater than 9 in the last digit. The control panel wiring for all of these functions is given in an appendix and need not be rearranged from problem to problem.

This control panel wiring of the machine is set up on the assumption that all numbers are less than 1. That is, .99999 is the largest number used in a calculation, and powers of ten are cared for by the programming of suitable shift operations. Note that the operating codes give .1(A + B) and $.1(B \div A)$ unless a shift is called for.

Problem-Solving Procedure

The proposer of a problem for solution will have first to reduce his problem to a sequence of the elementary arithmetic and other operations listed above. Presumably, the formula is to be evaluated for a number of values of the parameters, and a trial calculation should be made by hand for one set of values of the parameters. This will give the order of magnitude of the quantities involved and permit keeping track of the decimal point in the programming.
	Card No.	Transfer to A	Operation	Transfer from C	X 10 Punch Suppress Table Look-up		1	Card Entry to A				Card Entry to B			Card Entry to D	Сольд Польто И	Caru tainty to ta		Card Entry to F		Card Entry to G			Card Entry to H		Neg. Bal. Stop Check Stop Punch D, E, F
	1 2	3 4 5	6	789	10 11 12	13	14	15 i	16 1	17 1	18 1	9 20	21	22	23 24 25 26 27	28 29 3	0 31 32	33 34	35 36 37	38 3	9 40	41 42	43 44	45 40	5 47	48 49 50
Average Problem	0 1		1		5	12 0	0	1	2 :	3	2	0 0	0	0												
	0 2		1		5	12 0	0	3	2	1																
	0 3		1		5	12 0	0	4	5 (6																
\sim																										
Last Card to Add	1 2		1		5	12 0	0	3	1 4	4																
Divide by 1.2	1 3		4			12 1	2	0	0 (0																
1/10 of Answer on (13)	1 4		1			12 0	0	0	0 (0																
										T																

FIGURE 1

Programming is worked out on forms like that of Figure 1. The 80 narrow columns represent the 80 columns in the cards to be punched. Each line represents one card, and the programmer writes in the squares the digits he wants punched on the card.

Organization of the Calculator

As stated, the various operations of the machine were obtained by special control panel wiring. The procedure set forth in this paper is for a minimum capacity 604; i.e., 20 program steps. In this section the features of the machine, as wired, are taken up in some detail, in the order in which the card is punched to control them.

Card Number. This punching is not read by the machine but is used to sort and collate the cards in preparation for the machine run. Enough columns could be included to permit two independent numbering schemes on the same deck of cards, which is of use with certain subsequence operations, like square root or table look-up.

Transfer to A. As all operations involve only the numbers in registers A and B, these three columns, 3-5, provide a means of transferring numbers during calculate time from storage registers B through H into register A. Control or "11" punches are used in this fashion:

Register to read into A	CA	rd Colui	MN
	3	4	5
В	11	11	
D	11		
E	11		11
\mathbf{F}		11	
G		11	11
H	11	11	11

These control punches are put in the same card as the punching for the operation to be performed on the data transferred to A. Transfer of data from storage units D through H into B takes one card by itself. On this card the data are transferred to A, 00000 is entered in B, addition is performed, and the result multiplied by 10 to preserve the decimal location. On the next card, another factor can be entered in A and an operation performed.

Operations. A single digit in column 6 controls the basic machine operations. The digit 1 calls for 1/10 (A + B) and 2 for 1/10 (A - B). The reason for the 1/10 factor is to insure that the sum has no more than five digits and hence does not run over the digit capacity of the registers. Normally the sum of two 5-digit numbers might be six digits, but the 1/10 operation drops the last digit. In the

event it is known that the result will not exceed 5 digits, a "5" punch in column 10 will prevent the 1/10 operation and preserve all the digits of the sum.

The digit 3 calls for multiplication $A \times B$ and the result retained is the first five digits. For example, .90000 \times .90000 = .81000.

The digit 4 calls for division 1/10 (B/A). Here again the 1/10 factor is intended to keep the answer within the digit capacity of the registers. For example, 1/10(.90000/.20000) = .45000. However, care must be used in programming division, as too small a denominator will cause too large a quotient to be attempted. For example, 1/10 (.90000/.02000) = 4.50000 and the answer will overflow the B register. In such a case, zeros are punched and stored for an answer, and all subsequent calculations with this data will be in error.

The digit 6 calls for the basic step for the extraction of a square root; that is, it is one step of the iteration process used. The actual operation is:

$$\frac{1}{2}\left(\frac{D}{A}+A\right)$$

and the use of this operation is taken up in a later section headed Examples of *Programming*, part 3. The X 10 operation may not be used with operation 6.

If no code is punched in column 6, the answer storage unit C will be cleared, but the numbers in the other storage units are undisturbed.

Transfer from C. The result of an operation is always stored in B and in C. Normally, on the next card cycle, the number stored in C is punched in columns 75-80, and C is reset to zero. However, if a set of "11" punches is put in columns 7-9, then at punch time, the number in C is transferred to one of the storage units D through H instead of being punched. The code used is the same as that for reading into A from the storage units except that there is no need for transferring from C to B.

X 10. A "5" punch in column 10 will multiply the result of any arithmetic operation by 10. For example, if operate code 2 were used (subtraction) and a "5" punched in column 10, the answer developed in B and C would be A - Bwith no shifting of the decimal. If operate code 3 (multiplication) were punched in column 6 and "5" in column 10, the result developed in B and C would be 10 ($A \times B$). If no operation code is punched, a "5" in column 10 will transfer ten times the number in B into C, without changing the number in B. The larger number in C can be transferred to storage D through H for further use. The digit "5" was used instead of an "11" punch to prevent any possibility of early transfer of calculate selector 5 interfering with the last program step of the calculation for the previous card. Punch Suppress. Quite often it will be desirable not to punch intermediate results in order that the program cards can be used over and over with new data. An "11" punch in column 11 will suppress punching without affecting other operations.

Table Look-up. An "11" punch in column 12 is used on each card of a sequence forming part of a table of trigonometric or other functions. This prepares the machine for the look-up operation which will be described later under Examples of Programming, part 5. When the proper table entry is found, storage units E, F, G, and H are impulsed to reset and to read in the information punched in their fields on the card. If no information is punched, the storage unit is simply reset. On every table card an "11" must be punched in column 11 to protect the card.

Negative Balance Stop. Every result developed in storage C is examined for sign and, if an "11" punch is in column 48 of the following card, the machine will stop when the quantity is negative. This is useful in checking operations of various types where, for example, it is known that the difference of two quantities, or some other result, cannot be negative.

Check Stop. In certain other checking operations, it is known that the check quantity should be \pm .00009 or less. If an "11" punch is in column 49 of the card, the machine will stop if the result in C is greater in absolute value than .00009. This test is actually made on the second and third digits from the right so that a large error like .01000 or .11000 will not be found.

Punch D, E, F. An "11" punch in column 50 will cause the contents of registers D, E, and F to be punched in the same card in columns 61-75 without clearing these registers. This operation may not be performed if any number is being transferred from C to a storage register on the same card. That is, nothing must be punched in columns 7, 8, and 9 of the same card.

Double Punch Stop. The machine is wired to stop if it senses two numbers punched in the same column in the field 76-79. This is of use when the accuracy of machine calculation is being checked by running the same cards twice.

Card Entry into A and B. Numbers punched in columns 13-17 will be entered in storage A provided a "12" control punch is also punched in column 13. They will be entered as a negative quantity if an "11" control punch is also punched in column 17. Likewise, numbers in columns 18-22 will enter storage B if there is a "12" in column 18, and will be negative if there is an "11" in column 22. These numbers enter the storage units before the calculation or transfer to A, called for on the card, takes place and, hence, will be used in the calculation and transfer. If the "12" code for entry is punched but there are no digits, the storage unit is cleared. A "12" code was used for entry to insure reading the "11" code for negative numbers.

Card Entry into D through H. The entry and sign codes are the same as for A and B, that is, a "12" punch in columns 23, 28, 33, 38 and 43 controls entry, and an "11" punch in columns 27, 32, 37, 42 and 47 makes the quantity negative. However, the quantities are entered into the storage units on the third card cycle and hence may not be transferred to A for use until the third card after the one on which they are punched. Because of this, the second and third cards in a program sequence are often blank except for their card number, and computations are made with the first, fourth, and succeeding cards. It is not possible to read into D through H if the following card has a transfer from C to D through H, as these two operations would occur simultaneously.

Punch Exit from D, E, F, and C. Answers from C are usually punched in columns 76-80, but the contents of registers D, E, and F may be punched in columns 61-75 as mentioned above. These columns are never read and can also be used for identifying data if necessary.

Examples of Programming

1. First, as a simple example, consider the problem of finding the average of 12 three-digit numbers punched in 12 successive cards (see Figure 1). Card number one would carry the first number punched in card entry A (cols. 13-17) and would have zero punched in card entry B (18-22). To make the machine read these numbers, columns 13 and 18 would have, in addition to the digit punch, a "12" or special control punch. To make the machine add, a 1 would be punched in column 6. As the numbers are small, of the form .00369, there is no need to allow the machine to divide the sum by 10 as it usually does. Hence, shift or "X 10" is ordered with a "5" punch in column 10.

The second card carries the next card entry into A, but no entry into B as it now has the first partial sum in it. Succeeding cards continue to enter numbers in A and add them to the amount in B. Card 13 is coded to divide the sum (about .09999) by .12 and to retain 1/10 of this quotient. This is equivalent to dividing by 1.2. As we wanted division by 12, card 14 is coded to add zero to the quotient, and this divides the sum by 10 again, as nothing is punched in column 10. This gives the same result as originally dividing the sum by 12.

2. The greatest use of the machine is in more complicated sequences. These sequences are built up of simple subroutines. Let us consider the evaluation of

$$\frac{Z(X+Y) + VT}{W}$$

for values of the parameters in the neighborhood of Z = .2, X = 300, Y = -20, V = 3, T = -.5, and W = .01.

There are six quantities to enter into the seven registers so that we can enter them all on card one, as shown in Figure 2. This is convenient, as re-evaluation of the formula for other numerical values will involve changing only card one (the entry card) and the last card (the answer card).

As a general rule, the quantities within parentheses should be evaluated first, because the result is then ready for further multiplication or division. Hence, X and Y are entered in registers A and B. To locate the decimal point in a handy place, we write $10^{-3} X = .30000$ and $10^{-3} Y =$ -.02000. Addition can be performed on the next card, and we can form $10^{-3}(X + Y)$ without fear of overflowing the storage registers. Hence, X 10 is punched. Punching is suppressed, as only a final result is needed, and this intermediate result appears in B. A blank card is inserted to give the storage registers time to read in.

On the third card we can multiply by Z; the decimal point needs no adjustment, so that $10^{-3}Z$ (X + Y) is developed in B. This result is stored in E, as the number there previously is no longer needed.

Card 4 is used to bring T into register B by adding it to zero, read from the card, and allowing division by 10 to save trouble later on. Card 5 multiplies V by T and the product is $10^{-3} V T$. Card 6 adds this to $10^{-3} Z(X + Y)$ to give $10^{-4} [Z(X + Y) + V T]$. Card 7 divides by W to give

 $\frac{10^{-4} \left[Z(X+Y) + V T \right]}{(10 W)}$

which is

$$\frac{10^{-5} \left[Z(X+Y) + VT \right]}{W},$$

and the result is punched in columns 76-80. The decimal point is after column 80, i.e., the number punched is a whole number.

3. Another example is programming for square root, as shown in Figure 3. The number whose square root is wanted, N, is usually developed in some previous sequence of calculations so that it will be in storage B normally. We can, on the same card in the development, put it also in storage D for future reference. In what follows, it is assumed that we start with N in D and develop \sqrt{N} in B by successive approximations. N is, of course, of the form .64000 $\times 10^2$ so that when $\sqrt{.64000}$ is found then $\sqrt{10^2} = 10^1$, and there is no decimal trouble.

For the purposes of the example in Figure 3, the number .64000 is loaded in storage D, and the loading card is followed by two blank cards. The fourth card assumes an approximate root of .5 (midway of all probable roots) and makes the first iteration, storing the result in B. Card 5 is the second iteration so that this result is transferred to A in the usual way. Card 6 is the third iteration and is identical with card 5. As many iterations as are needed can be used, but 7 seems to be enough, based on some test runs. The result will be in B.

	Card No.	Transfer to A	Operation	Transfer from C	X 10 Punch Suppress Table Look-up	Card Entry to A		Card Entry to B	Card Entry to D	Card Entry to E	Card Entry to F	Card Entry to G	Card Entry to H	Neg. Bal. Stop Check Stop Punch D, E, F
	1 2	3 4 5	6	789	10 11 12	13 14 15 16 17	7 1	18 19 20 21 22	23 24 25 26 27	28 29 30 31 32	33 34 35 36 37	38 39 40 41 42	43 44 45 46 47	48 49 50
$[Z(X+Y)+VT] \frac{1}{W} (Load)$	0 1				11	$\begin{smallmatrix}12\\3&0&0&0&0\end{smallmatrix}$		$\begin{smallmatrix}2&&&11\\0&2&0&0&0\end{smallmatrix}$		12 2 0 0 0 0	$\begin{smallmatrix}12\\0&3&0&0&0\end{smallmatrix}$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{smallmatrix}12\\1&0&0&0&0\end{smallmatrix}$	
$10^{-3}(X + Y)$ Add	02		1		5 11									
Blank	0 3				11									
$10^{-3} Z(X+Y)$	04	11 11	3	11 11	11									
1/10 T	05	11 11	1		11		1	200000			l			
10 ⁻³ VT	06	11	3		11									
$10^{-4} [Z(X+Y) + VT]$	07	11 11	1		11									
10 ⁻⁵ Answer	08	11 11 11	4		5									
							Ţ							
					,									

Figure 2

	Card No.	Transfer to A	Operation	Transfer from C	X 10 Punch Suppress Table Look-up	Card Entry to A	Card Entry to B	Card Entry to D	Card Entry to E	Card Entry to F	Card Entry to G	Card Entry to H	Neg. Bal. Stop Check Stop Punch D, E, F
	1 2	3 4 5	6	789	10 11 12	13 14 15 16 17	18 19 20 21 22	23 24 25 26 27	28 29 30 31 32	33 34 35 36 37	38 39 40 41 42	43 44 45 46 47	48 49 50
Square Root (Load)	0 1				11			$\begin{smallmatrix}12\\6&4&0&0&0\end{smallmatrix}$					
Blank	0 2				11								
Blank	0 3				11								
(D/.5+.5)1/2	04		6		11	$\begin{smallmatrix}12\\5&0&0&0&0\end{smallmatrix}$							
(D/A+A)1/2	05	11 11	6		11								
	06	11 11	6		11				}				•
" "	07	11 11	6		11								
"	0 8	11 11	6		11								2
Answer in B	09	11 11	6										

FIGURE 3

																												_					
	Cord No.	Calu INO.		Transfer to A		Operation		Transfer from C		X 10	Punch Suppress	Table Look-up			Card Entry to A					Card Entry to B						Card Entry to F					Card Entry to G		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	:	33	34	35	36	37	38	39	40	41	42
Load	0	1									11		12 6	6	9	0	0	12 5	0	0	0	0	-	12 1	0	0	0	0	12 0	0	0	0	0
1/10 <i>l/c</i>	0	2				4					11																						
$1/100 (l/c)^2$	0	3	11	11		3	11	11	11		11																						
1/10 h	0	4		11		1 .				5	11							12 0	0	0	0	0											
1/100 h ²	0	5	11	11		3	11				11																						
1/10 k	0	6		11	11	1				5	11							12 0	0	0	0	0											
1/100 k ²	0	7	11	11		3	11		11		11																						
1/100 hk	0	8		11		3				Γ	11																						
$1/100 (h^2 + hk)$	0	9	11			1	Γ			5	11																						
$1/100 (h^2+hk+k^2)$	1	0	11		11	1				5	11								_														
$10 \times .13333 \times 1/100 \ (h^2 + hk + k^2)$	1	1				3				5	11		12 1	3	3	3	3																
$\frac{1}{100[4/3(h^2+hk+k^2)+(l/c)^2]}$	1	2	11	11	11	1	11			5	11																						
V	1	3				6					11		12 3	0	0	0	0							_									
V	1	4	11	11		6	T				11																						
Transfer $$ to A	1	9	11	11							11														_								
Answer	2	0	Γ			4	Γ											12 2	4	5	5	0											

FIGURE 4.
$$d = \frac{a_0}{[4/3(h^2 + hk + k^2) + (l/c)^2]^{1/2}}$$

4. The program for another actual problem is shown in Figure 4.^a

5. As a final example, we consider the programming for table look-up $(\sin x)$. The table look-up wiring permits subtracting the argument on each table card from an argument stored in A. When the difference is negative, i.e., when the table card next above the wanted argument is sensed, the previous card is read into storage units F, G, and H. This means the card for an argument next below the wanted argument is read. Figure 5 shows the program.

The procedure for quadrant selection is as follows. If the angle is not in the first quadrant, then we must put a plus or minus sign on the result of a table look-up, depending on the quadrant. This can be done with a small preliminary table look-up operation, where the table consists of quadrants and signs. This operation should be performed before looking up the sine or other function. It is assumed that x is in A to start the sequence. The sign is carried along by reading in

.99999 with the appropriate sign to use as a multiplier for the function to be determined later. On card 6 we transfer xto B. On card 7 we form $x_0 - x$, and on card 8 we transfer it to E for future use. On card 9 we transfer the sign to D. In the main portion of the table look-up which follows, three quantities are read to permit second-order interpolation if desired. For this example, we assume the sine read into F, modified first difference into G, and modified second difference into H. The interpolation formula is an adaptation of Stirling's formula.^b

$$\sin x = \sin x_0 - \left(\frac{-x + x_0}{x_1 - x_0}\right) \left[\frac{\delta_{-1/2} + \delta_{+1/2}}{2} - \left(\frac{-x + x_0}{x_1 - x_0}\right)\frac{\delta_0''}{2}\right].$$

Here $x_1 - x_0 = 10$ degrees, so that $\frac{x_1 - x_0}{x_1 - x_0} = \mu$ is obtained simply by subtracting and dividing by 10.

^aThis problem was proposed by the Physics Research group at the Chrysler Corporation and is concerned with some work with the X-ray spectroscope.

^bThis method of interpolation is explained in a paper by Eleanor Krawitz entitled "Punched Card Mathematical Tables on Standard IBM Equipment," published by IBM in Proceedings, Industrial Computation Seminar, September, 1950.

 $\frac{\delta_{-1/2} + \delta_{+1/2}}{2}$ is the average of the adjacent first differences or the modified first difference which is punched in field G. $\delta_0''/2$ is half of the second difference or the modified second difference and is punched in field H. Therefore, our formula is: $\sin x = \sin x_0 - \mu(G - \mu H)$. Card 11 begins evaluation by putting x in B. Card 12 forms $1/10(x_0 - x) = \mu$ with the decimal shifted and puts the result in E. Card 13 forms μ H with another decimal shift, and card 14 forms G - μ H. Card 15 gives μ (G - μ H), card 16 gives the sine, and card 17 gives the proper sign. A similar procedure is used for any other function.

	Card No.	Transfer to A	Operation	Transfer from C	X 10 Punch Suppress Table Look-up	Card Entry to A	Card Entry to B	Card Entry to D	Card Entry to E	Card Entry to F	Card Entry to G	Card Entry to H	Neg. Bal. Stop Check Stop Punch D, E, F
	1 2	3 4 5	6	789	10 11 12	13 14 15 16 17	18 19 20 21 22	23 24 25 26 27	28 29 30 31 32	33 34 35 36 37	38 39 40 41 42	43 44 45 46 47	48 49 50
QUADRANT LOOK-UP	-												
Search	0 1		2		5 11 11		12			0 0 0 0 0			
"	0 2		2		5 11 11		12			0 0 0 0 0			
	0 3		2		5 11 11		12 1 8 0 0 0						
	0 1		2		5 11 11		12 7000						
	0 5		2		5 11 11		12 7 0.0 0			0 0 0 0 0			
Transfer <i>x</i> to B	0 5		1		5 11 11 E 11		12		30000	99999	·		
$+x_0 - x$	0 7	11 11			5 11		0 0 0.0 0						
Shift	0 2			11 11	5 11								
Transfer Sign to D	0 0	11		11	5 11	,	12						
			ŀ										
TABLE LOOK-UP					· · · · · · · · · · · · · · · · · · ·								
Transfer to A	0 1	11 11	2		5 11 11		12			0 0 0 0 0	17365	0 0 0 0 0	
Search	0 2		2		5 11 11		12		10000	17365	1 7 1 0 1	0 0 2 6 4	
<i></i>	0 3		2		5 11 11		12 2 0 0 0 0		2 0 0 0 0	3 4 2 0 2	16317	0 0 5 1 9	
"	0 4		2		5 11 11		$12 \\ 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0$		30000	5 0 0 0 0	1 5 0 3 8	0 0 7 5 9	
"	0 5		2		5 11 11		$\begin{array}{c}12\\4&0&0&0\\\end{array}$		4 0 0 0 0	64279	1 3 3 0 2	0 0 9 7 7	
"	0 6		2		5 11 11		12 5 0 0 0 0		50000	7 6 6 0 4	1 1 1 6 2	0 1 1 6 3	
"	0 7		2		5 11 11		12		60000	86603	0 8 6 8 2	0 1 3 1 6	
'44	0 8		2		5 11 11		$12 \\ 7 0.0 0 0$		70000	9 3 9 6 9	0 5 9 3 9	0 1 4 2 7	
"	0 9		2		5 11 11		$\begin{smallmatrix}12\\8&0.0&0&0\end{smallmatrix}$		80000	98481	0 3 0 1 5	01497	
66	1 0		2		5 11 11		$\begin{smallmatrix}12\\9&0.0&0&0\end{smallmatrix}$		90000	999999	0 0 0 0 0	01518	
Transfer	1 1		1		5 11		$ \begin{smallmatrix} 12\\0&0.0&0&0 \end{smallmatrix} $						
$1/100 (+x_0-x) = \mu/10$	1 2	11 11	2	11 11	5 11								
μH	1 3	11 11 11	3		5 11								
$G - \mu H$	1 4	11 11	2		5 11					1			
$\mu(G-\mu H)$	1 5	11 11	3		5 11								
$\sin x = \sin x_0 - \mu \left(G - \mu H \right)$	1.6	11	2		5 11								
Adjust Sign	17	11	3										

FIGURE 5. TABLE LOOK-UP, SIN x (x in A)

Appendix

Wiring Diagrams

The diagram in Figure 6 gives the read-in wiring, but the selector diagrams are repeated in Figures 7 to 11. The calculate control panel is not diagrammed, but the connections are listed in detail in Table I.

The basic operations are obtained by wiring them all on the calculate panel but suppressing all steps (except step 1) until one or more calculate selectors pick up. This permits a complete assortment of operations from one panel with no setup changes between problems. There are other possible arrangements of the various functions which might save some program steps in Table I; for example, step 5 is identical with step 11.

The various units of electronic storage are assigned to the storage registers mentioned under programming as follows:

F S 2	А
M Q	В
Counter	С
G S 1 & 3	D
G S 2	E
G S 4	\mathbf{F}
F S 4	G
F S 1 & 3	Η

Read-in to A and B (F S 2 and M Q) is at first reading, as are all transfer and operation codes except table look-up. A "12" punch is used to control read-in to permit reading of the "11" punches for sign control. Most of the rest of the control punches (transfer operations, table look-up, stop, etc.) are "11" punches. However, a "5" punch is used for X 10 (column 10) because this picks up calculate selector 5, and a "12" punch might transfer it early enough to affect the previous electronic cycle. The timing chart, Figure 12, explains this.

Read in to D through H (all of the remaining electronic storage units) is at second reading. While this may be slightly inconvenient for spread entry of quantities into storage, it permits the table look-up operation to occur as neatly as it does, the same second-reading brushes being used in table look-up. A "12" punch is normally used to control read-in (through a column split) but storage E, F, G, and H also obtain a card cycle read-in pulse when the negative balance selector picks up and an X is punched in column 21 of the card.

Product Overflow Stop. An X punch in column 49 transfers pilot selector 2 at punch time and makes a product overflow check of the second and third digits. This is of use in checking that the difference of two quantities is nearly zero. For example, it may be used in checking the result of



FIGURE 6. TYPE 604 CONTROL PANEL WIRED FOR CPC OPERATION

a square root operation by squaring the root and subtracting from the original number.

Negative Balance Stop. An X punch in column 48 of the following card will go through a set of negative balance selector contacts, if the number in C is negative at the end of a calculation. This stops the machine and is used to check results which cannot be negative.

Punch Selectors. The punch selector wiring of units 4-8, as shown in Figure 10, is intended to permit punching or transfer to a particular storage unit as shown. The pickup wires for these selectors are wired to pilot selectors 3, 4, and 5, as shown in Figure 9. Read-in pulses are supplied through punch selector 3 from the same digit pulses used to pick up punch selectors 4-5.

Square Root Wiring. As the square root iteration 1/2[(D/A) + A] is compounded of simple arithmetical operations, the regular calculating steps are used by arranging the various operations in the proper order. Table I shows that, after transfer to A and addition, there are in sequence a division, a special set of steps for square root, and a multiplication. Thus, if calculate selectors 3, 4, and 5

Wirth	NC OF FIF	TABLE I
Operation	Step	Wiring
Clear	1Read	l out and reset counter.
Transfer {	2Read stora age 2 3Read 4, as 4Read quot	d out general storage 1 and 3 or general age 2, as selected; read in factor stor- 2. d out general storage 4 or factor storage selected; read in factor storage 2. d out factor storage 1 and 3 or multiplier ient, as selected; read in factor storage 2.
± {	5Read 6Read + or 7Read as set	l out factor storage 2; read in counter +. l out multiplier quotient; read in counter -, as selected. l out and reset counter, out of 2nd or 1st, lected; read in multiplier quotient.
÷	8Read age into 9Read 10Read	l out multiplier quotient (or general stor- l and 3, as selected); read in counter +, 5th or 6th, as selected. l out factor storage 2; divide. l out and reset counter.
$\sqrt{-}$	11Read 12Read +, ii 13Read in fa 14Emit	l out factor storage 2; read in counter +. l out multiplier quotient; read in counter nto 1st. l out and reset counter, out of 2nd; read ctor storage 2. t 5; read in multiplier quotient, into 5th.
×	15Read 161/2 17Read as se	l out factor storage 2; multiply +. adjust, into 5th or 4th, as selected. l out and reset counter, out of 6th or 5th, lected; read in multiplier quotient.
±,×,÷,√ X 10	18Read coun 19Read	1 out multiplier quotient; read in ter $+$. 1 out multiplier quotient; read in ter $+$, into 2nd.
	20Bala	nce test for selector pickup.

are picked up by punch selector 2, all the parts of one square root iteration are performed. The coding is shown in Figure 3.

Calculate Selector Wiring. The wiring of suppress pulses and of read-in pulses is shown in Figures 7 and 8. This wiring is fairly complex because few selectors were available. Note that the logic of switching suppress pulses is the inverse of Shannon's logic of relay and switching circuits.







FIGURE 8. CALCULATE SELECTOR WIRING FOR TRANSFER TO A



FIGURE 9. PUNCH SELECTOR PICKUP WIRING







FIGURE 11. TABLE LOOK-UP WIRING

DISCUSSION

Mr. Ferber: Did you at any time try to use the card itself for memory by ganging and then suppress ganging?

Mr. Nims: So-called dynamic storage? We thought about it, but the chief difficulty was that we were running out of

pilot selectors. We had no way of getting this stored information back into channel A conveniently.

Mr. Ferber: We do that by setting it up to read from different places on the card, either from first reading or second reading, but we also used a temporary memory.



FIGURE 12. TYPE 604 TIMING CHART

When we needed one more memory than we had, we would punch it out and then read it back at the second reading as an additional memory, although we used, of course, the principle of reading it electronically, just as you do.

We also set up the possibility of doing A times B plus or minus one of the electronic memories, using the calculate selectors, which might still be open.

Dr. Herget: Have you considered using Cecil Hastings' series for any of your functions? Would that be less economical than what you do?

Mr. Nims: It might be, because you would need simple instructions for each term in this series, which would also use many cards. The real reason we were interested in table look-up is that we have considerable empirical data that we want to consult, and it could be represented by a table with first or second differences, but it would be quite a chore to get a series to cover the whole range, test data, and so on.

Mr. Feigenbaum: At Cornell Aeronautical Laboratory we did a sort of backward operation. We used the CPC as a 604 because we wanted printed results rather than punched results. We found it was much quicker to feed the numbers through the CPC and print them out right away, rather than punch and then print them later.

Dr. Brinkley: We have occasionally used the generalpurpose 604 control panel that goes with the general-purpose CPC setup for pure 604 operations. If you want to do one or two operations and then punch the result, this will save time.

Mr. Lowe: I understand that at the Watson Laboratory they have set up a 602-A in somewhat the manner that Mr. Nims describes, to form a miniature CPC. This may be of interest to our 602-A users.

General-Purpose Floating Point Control Panels for the IBM Card-Programmed Electronic Calculator

KATHLEEN BERGER L. RICHARD TURNER NORMAN A. PATTON

Lewis Flight Propulsion Laboratory, National Advisory Committee for Aeronautics

)¢{

THE DESIGN of control panels^a for the operation of the IBM Card-Programmed Electronic Calculator on numbers expressed in floating point notation was undertaken for two basic reasons: first, to expedite the solution of certain non-linear algebraic equations, notably those defining gas-phase chemical equilibria; and second, to reduce the programming time for short problems or for problems in which scale factors are impossible or, at least, difficult to predict. In the case of the calculation of chemical equilibria, for example, a single routine may be used for almost the entire range of possible parameters if the calculations are performed in floating point notation. With fixed point computation, however, approximately thirty-five different routines may be required for the carbon-hydrogen-oxygennitrogen system.

There are many significant disadvantages of floating point notation. These are principally the difficulty of interpretation of a listing of the results, the virtual impossibility of carrying out multiple precision operations, the false sense of precision that may be created, and the increase in computing time that appears to be unavoidable. These facts were used as a basis for the initial requirements that the arithmetic operations be as precise as logically possible, that the notation be as natural as possible, that the operation of the calculator be as flexible as possible, and that lost motion be largely eliminated.

After a considerable period of evolution the computer now has the following general facilities:

- 1. Basic arithmetic operations of add, divide, multiply, reverse the sign of operand *B*, and suspend round-off in the command "*A* operation *B* equals *C*." Round-off, when carried out, is almost always exactly correct.
- 2. Three complete independent program fields.
- 3. Four numerical fields with conditional or semi-conditional entry from all fields.

- 4. Automatic extraction of integral roots from second to seventh.
- 5. Eight kinds of conditional operations including transfer of control among the three program fields which may be based either on the sign of the number or on the sign of its exponent symbols (negative exponents correspond to a zero result).
- 6. Zero results are stored as zero, and all improper operations—such as an unintentional division by zero or the formation of a result too large for proper storage —cause a machine stop with an automatic list cycle.

The extraction of integral roots, rather than other functional operations, was chosen as the basic wired subsequence because of the interest in aerodynamics of second, fifth, and seventh roots, the fact that it is not generally possible to predict the length of a routine for root extraction, and partly because it could be done fairly easily.

GENERAL

The notation was chosen for ease of interpretation within the limits of the storage capacity of the CPC. A number is represented by eight numerical digits with the decimal point to the left, and a pair of exponent digits whose value is 50 greater than the power of 10 by which the numerical digits are to be multiplied. The exponent digits are stored to the left of the numerical digits. Unity has the representation (51; 1), and in general, a number is interpreted by moving the decimal point to the right by a number of places equal to the excess of the exponent digits above 50 or by prefixing the numerical digits with a number of zeros equal to the defect of the exponent digits below 50. Zero has the normal notation 00;00000000. Any result for which the exponent digits exceed 99 is treated as infinity, and the computation is stopped.

The 604 control panel executes only the basic arithmetic operations. The choice of the median exponent digit, a part of the calculation of exponents for multiplication and divi-

^aThese control panels were designed at the Lewis Flight Propulsion Laboratory during the summer of 1950 and first placed in operation in early September, 1950. Since then the control panels have been gradually modified to provide increased flexibility and have only recently achieved their present, and possibly final, forms.

sion, and the sensing of the existence of a zero result are partly controlled from the accounting machine control panel.

All other features, including the management of limiting cases, impossible problems, conditional operations, provision for the immediate use of results as factors in calculation, and root extraction are wired in the 402-417 control panel.

THE 604 CONTROL PANEL

Operation Codes. The operation codes of the symbolic operation "A op B = C" are

- 1: Add A and B.
- 2: Divide A by B.
- 3: Multiply A by B.
- 4: Cancel normalization of result with initial zeros.
- 5: Reverse sign of operand B.
- 8: Cancel round-off ($\frac{1}{2}$ adjust).

For completeness:

- X: List.
- 9: Summary punch.
- 6: Equivalent to 1 code.
- Causes division and simultaneously multiplies the result by 10⁵. (The use of codes 6 and 7 is generally not recommended.)

Factor Entry. The numerical portion of operand A enters into factor storage 1 and 2. The numerical portion of operand B enters into factor storage 3 and 4. The exponent portions of A and B enter general storage 1 and 3 which are given an 8-6 assignment but are not coupled on the 604 control panel to any of the five-digit storage units. This wiring causes read-in and subsequent use of the exponent digits as their positive absolute value. For division, 52 is read into the multiplier quotient and for multiplication, -50 is read into the multiplier quotient. The 2 of the number 52 is obtained from the code 2, and the minus of the number -50is obtained from the code 3.

Analysis. The operation of the 604 is divided into two parts: on the first 44 program steps the arithmetic operations are completed; the last 16 programs, which are active on program repeat, are used to shift the result into its normal notation. One program repeat is always required. The wiring of the 604 control panel is quite tight; there are 95 logically different operations which were condensed into 60 through the use of calculate selectors and the use of two 6H6 diode circuits. Details of analysis and wiring of this feature of the 604 control panel and all other control panels are described in the Appendix.

The first two program steps reverse the sign of the numerical digits of operand B if calculate selector 5 is transferred. This step is placed first, in order that subtraction may be converted into an addition, which is symmetrical in

its operands. Program steps 3 to 8 manipulate exponents so that for the process of multiplication or division the trial exponent of the result is computed, and for the process of addition the larger of the two exponents is placed in general storage 1 and the absolute value of their difference in general storage 4. The final calculation of the trial exponent digits for the addition process is deferred to program steps 21 to 23.

The equivalent of five program exits is obtained in program step 6. One exit is expanded into two by the use of diodes to drive both emit 1 and counter plus. Use is made of the fact that the "read units into 6th" is inactive on "counter read-out" in order to couple "in 6" to "general storage 3 read-in."

For addition, the decision as to whether to interchange operands A and B prior to alignment of the decimal points is made by a balance test for step suppression on program step 4. If this interchange is required, it takes place on program steps 9 to 11.

Decimal point alignment is done in three steps. If the difference of the exponents is greater than four this difference is reduced by four, and operand B is shifted four places to the right (program steps 12 to 19). The next part of the alignment is done by multiplying B by an appropriate factor. Unity is successively subtracted from the amended difference of the exponent digits, and unity is emitted into the multiplier quotient in successive positions until a negative balance is obtained (program steps 24 to 36). Finally, if a shift of more than eight places is required, a negative balance does not occur, and operand B is set equal to zero on program step 37. The addition is completed on program steps 38 to 46, with program steps 41 and 43 inactive.

Multiplication and division are carried out by conventional programs interspersed among the addition programs. On program step 46 each of the possible results, as yet unrounded, is similarly placed, with the numerical digits in the electronic counter and with the appropriate exponent digits in general storage 4. The 11th counter position is the overflow place for addition or division, or the 16th place when a full 16-place product is formed in multiplication. The two right-hand positions are the two possible round-off positions. Round-off ($\frac{1}{2}$ adjust) occurs only on the last program repeat.

The overflow position is zero tested, and if the result of the test is zero, the exponent digits are reduced by one and the computed result shifted one place to the left. The exponent digits are reduced, and a balance test for selector pickup is ordered on program step 57. The previous process is repeated until a non-zero is found, which suppresses further program repeat pickup, or until transfer of the negative balance selector, which is wired on the 402-417 control panel to transfer calculate selector 4. (Program repeat is initially picked up on program step 3 but further pickups occur on program step 49 which is wired through the common to the normal points of calculate selector 4.)

When a non-zero digit is found (program step 55), reduction of the exponent digits (program steps 56 to 58) is suppressed, and further pickup of program repeat is also suppressed. On the final 60 program steps the result is rounded, and a final zero test is performed to determine whether round-off caused an overflow. If no overflow occurs, the exponent digits are again reduced by unity. If an overflow does occur, the exponent digits are not reduced. This last is provided to take care of the special case in which the unrounded result consists of eight 9's with a ninth digit greater than or equal to 5. Program step 59 is suppressed on non-zero, because if overflow occurs the correctly rounded result is in factor storage 3-4; otherwise, the result is in factor storage 1-2.

The suppressions are described in detail in the Appendix. The only unusual feature of the suppressions is the use of a pair of diodes to provide for suppression either by group suppressor 4 or by suppress on minus balance.

THE 402-417 CONTROL PANEL

The 402-417 control panel used at the Lewis Flight Propulsion Laboratory of the NACA is wired to correspond to the card diagram shown in Figure 1 and the table of codes shown in Figure 2. Three independent ten-column program fields, card columns 4-13, 54-63, and 64-73, are provided. Four numerical fields (a, b, t and s) are provided in card columns 14-23, 24-33, 34-43, and 44-53, respectively. Card columns 1-3 and 80 are used for card numbering. Card columns 41-47 are wired from second brushes to counter minus for the purpose of clearing the counters of the accounting machine. Card column 48 is wired to the X hub of storage clear. Card column 74 is used for carriage control and card column 75-79 for listing of identification of results with alphabetical or numerical designations on a 402, or numerical codes on a 417 accounting machine.

The three program fields 4-13, 54-63, and 64-73 are identical in function. Their wiring, which involves co-selectors 5 and 6, 9 and 10, and 11 and 12, is shown in the Appendix. Co-selectors 11 and 12 are required to permit subsequencing on special program for the calculation of integral roots.

The program field is subdivided as follows:

Card columns 4 and 5	Channel A address, symbol A
Card column 6	Operation code, symbol op
Card columns 7 and 8	Channel B address, symbol B
Card columns 9 and 10	Channel C address, symbol C
Card column 11	Shift, symbol S
Card column 12	"Normal" column for special
	codes, symbol N
Card column 13	"Transferred" column for
	special codes, symbol T.

The regular channel A, B, and C codes perform in the normal manner as do the operation codes and the shift code. A shift of two has been taken as "normal." The special code columns 12 and 13 are read conditionally. Discussion of their significance is deferred.

															-						_	F	-					12	CL. 2 3	EA 4	R 3	's 7	st			-							T												
CA N	RE D.		P	RO FII	GR. ELI	A M D				נ נ	FAC FIE	CTC LD	OR), a		3		1	FA	CT	OR 9, 1	, ,				TA FII	.BL ELI	.E D,	t		:	SUI	3ST FIE		JTI , s	ON			P F	RO IEL	GR. .D	A.M 50	[P F	RO IEI	GR "D	A.№ 60	ĩ			LIS IDI	ST EN1	r.	
0	חו			B	ן זו ה		5 N 0 0	T	n n	חו	0 1	n n	0	0 0	0	n () N	n i	n n	n	n n	n	0 1	חו	ß	0 0	10	0 0	1.0	0 1	חו	n	n n	n	nn	0	ົ	O P	в n л			N	T	A N N	O P N	в Л Л			5 N	T N	م أد	10	n n		n
	23	4 5	6	7 8	9	10 1	1 12	13	14 19	5 16	17 1	8 19	20	21 2	2 23	24 2	5 26	27 2	8 29	30	31 32	33	34.3	536	37 :	38 39	9 40	41 42	2 43	44 4	546	47	18 49	50 1	51 52	53 5	4 55	56 5	17 58	59 6	50 61	62	63 6	4 65	66	17 M	69	70 7	1 72	73	14	5 76	77 78	a 79 F	
	1 1	1.	1	1 1	1	1	1 1	1	1 1	1	1 1	1 1	1	1 1	1	1 1	1 1	1	1 1	1	1 1	1	1	11	1	1 1	1	1 1	1	1	11	1	1 1	1	1 1	1	1 1	1	1 1	1	1 1	1	1	11	1	1 1	1	1	1	1	1	11	1 1	1	ĩ
Ľ		Ľ	' '				"	11			1	• •			''			1		1		1	1						I	Ľ		1			• •	1	• •	'		!	'l'	11	Ί		11		1		יוי	'	' '			1	'
2 :	2 2	2 2	2 2	2 2	2	2 2	2 2	2	2 2	2 2	2 2	2 2	2	22	2 2	22	2 2	2 2	2 2	2	22	2	2 2	2 2	2	22	2	22	2 2	2 :	2 2	2	22	2	22	2	2 2	2	22	2	2 2	2	2	2 2	2	22	2	2	2 2	2	2	2 2	22	2	2
3 :	33	3 3	3	3 3	3	3 3	3 3	3	33	3	3 3	33	3	33	3	33	3	3 :	33	3	33	3	3 :	3	3	33	3	33	3	3 :	33	3	33	3	33	3	33	3	33	3	3 3	3	3	33	3	33	3	3	3 3	3	3	33	33	3	3
4	14	4 4	4	4 4	4	4 4	4	4	4 4	4	4 4	4 4	4	44	4	4 4	4	4 4	44	4	44	4	4 4	4	4	44	4	44	4	4 4	14	4	44	4	44	4	4 4	4	44	4	4 4	4	4	1 4	4	44	4	4	14	4	4	4	44	4	4
5	55	5 5	5 5	5 5	5	5 5	5 5	5	55	55	5 !	55	5	55	5 5	55	i 5	5 !	55	5	55	5	5 !	5 S	5	55	i 5	55	i 5	5 !	55	5	55	5	55	5	55	5	55	5	5 5	5	5	55	5	55	5	5	5 5	5	5 5	55	55	5	5
6 (66	6 6	6	6 8	6	6 6	6 6	6	6 8	66	6 (66	6	6 6	6 6	6 8	66	6 (66	6	66	6	6 (66	6	66	66	6 6	66	6 (66	6	66	6	66	6	66	6	66	6	6 6	6	6	66	6	66	6	6	66	6	6 6	66	66	6	6
7	7	77	7	77	1	7	7	7	11	17	7	77	7	11	7	77	17	7	7	7	77	7	1	7	7	77	17	11	17	7 :	7	7	77	7	77	7	7	7	77	7	7 7	7	7	77	7	77	7	7	17	7	7	17	77	7	7
8 1	38	8 8	8	88	8	8 8	8 8	8	88	8 8	8 8	88	8	88	8	88	8 8	8	88	8	88	8	8 8	38	8	88	8 8	88	8 8	8 1	38	8	88	8	88	8	88	8	88	8	8 8	8	8	B 8	8	88	8	8	8 8	8	8	38	88	8	8
9 ! 1 :	9 9 2 3	9 9 4 5) 9 6	9 9 7 8	9	9 9 10 1	9 9 1 12	9 13	99 14 1	9 5 16	99	9 9 15 19	9 9 20	99 212	9 2 23	9 9 24 2	9 5 26	9 9 27 2	9 9 28 29	9 30	99 3132	9 33	9 ! 34 3	9 5 36	9 37	99 383	9 40	9 9 41 4	9 2 43	9	9 5 46	9 47	9 9 48 49 ICEN	9 50 SED	99 51 52 FOR L	9 53 SE	9 9 14 55 JNDE	9 56 8 8	9 9 57 58	9 59 (1,7	9 9 50 6 72 4	9 1 62 92	9 9 63 6	9 9 14 65	9 66	99 67 6	9	9 9 70 7	9 1 72	9 73	9 S	9 5 76	99 777	9 3 79 8	9 80

FIGURE 1. CARD DIAGRAM

	Code	Function
Channels A and B (columns 4-5	7n 8n 00	Read out counter group n Read out and reset counter group n Read factor from card
017-0)	$ \begin{cases} X \\ 9 \\ X \end{cases} $	Re-read result of previous calculation
	{8n	Re-read previous result and clear counter group n
Channel C (columns 9-10)	7n 8n 00	Read in counter group <i>n</i> plus Read in counter group <i>n</i> minus Read out and reset electronic counter but do not store
	Nn	result Read in storage unit <i>n</i> , bank N
Operation Codes (column 6	1 2 3	Add A and B Divide A by B Multiply A by B
or 66)	5 8 9	Reverse the sign of element B Suspend round-off Summary punch
0 11/2 1	$\frac{\Lambda}{* X(N)}$	
Conditional Codes (column 12 or 13 or 62 or	*X(N) *X(T) 0	Ignore abnormal notation of factor A Ignore abnormal notation of factor B Cause discrimination to be based on "zero-not zero" result
63 or 72 or 73)	1 2	Start total program for Nth Rooting Substitute number in "s" field for result of previous cal- culation retaining sign of result
	3 5 6	Add number in table field to counter group 7 Read next line of instruction from program field 50 Read next line of instruction from program field 60
	7	Store previous result with sign punched in "s" field

*Read unconditionally.



Additional channel A and B codes have been synthesized by the use of an X punch in card column 4 or 7. This X punch causes the result C of a calculation with its sign to be re-used as the corresponding A or B factor in the next calculation. For convenience in listing, an 8 or 9 is also punched in order to list as Q or R. If the code Q is used and a digit 1 to 7 is punched in the units digit of the channel A or B code fields, the result of calculation may be re-used and the counter group n(n = 1, ..., 7) is simultaneously cleared, subject to the usual prohibition against attempting to read in and read out of a counter at the same time. The counter complement 9's are prevented from entering the exponent registers, general storage 1 or 3, by the wiring in the lower half of pilot selectors 2 and 3. In addition to wiring the factors through selectors it is necessary to wire the signs of channel C to channel A or B as desired. To prevent accidental setup of a false sign when an R or Q code is used, the channel C sign is wired through pilot selector 13 which is transferred only on electronic all-cycles. The sign is also wired through pilot selector 14 so that positive or negative absolute values may be formed conditionally.

The wiring of the 604 control panel requires that the electronic counter and electronic A and B be cleared of all numbers left from the previous calculation. In order to simplify card punching and to provide selective reading of constants from card fields A or B, the electronic counter readout and reset and the storage units read-in and read-out are wired from electronic all-cycles rather than from channels A, B, and C commons.

The correction required for the exponent digits is read into the multiplier quotient unit by emitting a 5 into the second position from the C emitter. The pickup of calculate selector 2 is wired to the units digit to provide 52 for division, and the pickup of calculate selector 3 is wired to multiplier quotient minus to provide the -50 required for exponent correction during multiplication.

One of the major problems in floating point operation is to provide for the correct management of commands or computation that involve zero factors or zero results. The desired treatment of commands with zero factors is as follows:

Operation	A	В	Results
Addition	0	$\neq 0$	$\neq 0$
	$\neq 0$	0	$\neq 0$
	0	0	= 0
Multiplication	0	$\neq 0$	= 0
	$\neq 0$	0	= 0
	0.	0	= 0
Division	0	$\neq 0$	= 0
	$\neq 0$	0	Impossible; stop
	0	0	Impossible; stop.

It is also desired to treat as zero any result for which the exponent digits become negative, and it is necessary that the division of a number in normal notation by a non-zero factor in abnormal notation—that is, by a factor with one or more of its initial numerical digits equal to zero—be classed as an impossible operation. In practice these conditions are realized by treating any number as zero whose first numerical digit is zero. This action can be prevented by suitable coding in the program columns N or T (card columns 12 or 13, etc.).

The existence of a non-zero initial numerical digit is detected by wiring these digit positions to comparing, and picking up pilot selectors 4 and 5 (for A and B, respectively) from the corresponding unequal impulse. Discrimination against zero factors may be prevented by punching XN or XT which causes pickup of pilot selectors 4 or 5, respectively, from second reading. Pilot selector 4 is also picked up on its digit pickup for the operation of root extraction on special program in order to assure a "zero-not zero" discrimination by which special program is terminated.

Pilot selector 6 is transferred on immediate pickup if there is a 2 code (division), and pilot selector 7 is similarly transferred if there is a 3 code (multiplication).

These four pilot selectors cause the following results:

- a. If there is neither a 2 or a 3 code, a 1 (addition) code is formed as a true 1 impulse.
- b. For multiplication or division, if either or both factors are zero—and, for addition, if both factors are zero—calculate selector 4 is picked up by a 0 impulse which lists as 0 on a 402 when wired to the alphamerical list entry. In any of these cases pilot selector 9 is picked up on its X pickup. Pilot selector 9 is coupled to co-selectors 7 and 8 which actually cause the result to be zero in case the "zero result command" was generated

by the attempt to carry out computation with nonzero factors in abnormal notation.

c. In division, if factor B is zero, a "half-after-11" impulse picks up pilot selector 8 through its X pickup. Pilot selector 8 is wired to cause a machine stop when transferred. (This pilot selector is used for all of the various kinds of machine stops which can occur during calculator operation.)

The effective portion of the wiring of pilot selectors 4, 5, 6, 7, 8, 9, 13 required for each of these operations is indicated in the successive parts of Figure 3, pages 53 and 54. The five high-order positions of the electronic counter and the three high-order positions of general storage 4 are wired to comparing inlet to synthesize a product overflow. The unequal impulse hubs are wired to the digit pickup of pilot selector 8 to provide a stop. This feature terminates computation if the exponent digits of a computed result exceed 99 or if digits other than zero are read out of any of the positions so wired. To avoid split wiring, a conditionally programmed stop code (8 digit) is also wired to pilot selector 8 by the use of a position of comparing.

The previous wiring prevents the use of a large number of program repeats in the 604 to detect a zero result which would occur with blank cards and also provides for the transfer of calculate selector 4 when the occurrence of negative exponent digits causes negative balance selector 1 to transfer. For this purpose it was necessary to synthesize an impulse that was available only on idle cycles. This impulse is available at the lower normal point of pilot selector 13 which transfers only on electronic all-cycles. The transfer of the negative balance selector also causes co-selectors 7 and 8 to be transferred, thus storing a zero result.

Conditional Operations and Special Numerical Fields. The special program fields N and T contain eight conditional commands. One of the fields N or T is always read: N for a positive sign, and T for a negative sign. Conditional code 0, if read, causes the choice to be based on the sign of the exponent digits which thus forms a "zero-not zero" discrimination. If no 0 code is read, the choice is based on the sign of the numerical digits of the result. This alternation is effected by wiring 0 from a digit selector to the digit pickup of pilot selector 15 by which the sign exit of the electronic counter or of general storage 4 is wired to the coupling exit of negative balance selector 2. Card column N is wired to the normal point of this selector and card column T to the transferred point. The common is wired to the common of the previously-mentioned digit selector.

Code 1 is used to pick up special program in connection with the *N*th rooting subsequence.

Code 2 causes the value punched in the substitution field, "s" (card columns 44-53), to be stored in place of the computed result with the sign of the computed result, unless code 7 (below) is also used, in which case the sign of the



Pickup wiring for selectors 4, 5, 6 and 7, and generation of a 1 code in the absence of both 2 or 3 codes. Selectors 6 and 7 are both normal.

Figure 3A







Factors A and B zero on addition, multiplication or division. Selectors 4 and 5 are both normal.

FIGURE 3B



Division: Factor A zero. Selectors 4 and 7 are normal. Selectors 5 and 6 are transferred.



Division: Factor B zero. Selectors 4 and 5 are transferred. Selectors 5 and 7 are normal.

FIGURE 3C

"s" field is stored. The reading of the "s" field is effected by wiring a 2 from the digit selector to the digit pickup of pilot selector 9 which in turn transfers co-selectors 7 and 8. It is noted that the numbers punched in the "s" field will replace any zero result. Therefore, care must be exercised in the use of the "s" field. Miscellaneous examples of its use are shown in the Appendix.

Code 3 causes entry of a factor punched in the "t" field (card columns 34-43) into counter group 7. The 3 digit is wired to the digit pickup of spread read-in and also to the pickup of the field selector so that the sign X (card column 34) is read to counter group 7 minus. Since this is a conditional operation, accidental read-in of digits to counter group 7 is prevented as follows: the card field is wired to counter exit, and counter exit suppression is wired through position 4 of the field selector to an all-cycles impulse. When the field selector is transferred to position 3, the allcycle impulse fails to reach the exit suppression hubs. Care must be exercised in the use of 3 codes so that no counter storage is ordered on the card for any program field in which a 3 might be read. As long as there is no 3 code on a card, columns 34 to 40 may be used for any punching desired. This field has been found useful for codes to identify special cards.

Codes 5 and 6 are used to control reading of program instructions from the alternate program fields. If a 5 is read, the instructions on the next cycle will be read from program field 50, so named because its card columns are 50 in excess of the basic field, card columns 4-13. If a 6 is read, the next commands will be read from program field 60. It is noted in passing that since the N and T fields are read at second reading, the channel A and B codes are read from the same program field, while the operation code, channel C code and shift instruction are read from the transfer field.

Code 7 causes the sign of the "s" field to be substituted for the sign of the computed result. This code permits the conditional formation of positive or negative absolute values.

Code 8 causes a stop.

Code 9 causes summary punching.

As previously noted, each of these codes is conditionally read, but may be made unconditionally read by punching the code in both the N and T columns. Each program field has separate N and T columns.

NTH ROOTING

The process of root extraction was chosen as the basic subroutine for several reasons:

- a. In aerodynamics, second, fifth, and seventh roots occur quite frequently.
- b. The number of iterations required for the extraction of a root is difficult to predict.
- c. The calculation of the elementary transcendentals can always be programmed in a sequence of fixed length and if done on a subsequence would require five storage addresses and roughly three times as many cycles as the known fixed routines.

The programs for root extraction are emitted in part from the C emitter through the field selector. The specification of the root to be computed is controlled by modifying the order of pickup of the field selector which is in turn controlled by storing an appropriate code number in storage unit 28. To carry out this operation it is necessary to store the number whose root is desired in storage unit 26 and to store an estimate of the root in storage unit 22. The result appears in storage unit 22. The code number stored in unit 28 serves as the divisor n in the iterative formula below for $\sqrt[n]{a}$. Let X_k denote the *k*th approximation to $\sqrt[n]{a}$, then

$$X_{k+1} = X_k + \left(\frac{a}{X_k^{n-1}} - X_k\right) \cdot \frac{1}{n} \quad .$$

The four low-order digits of the code number also serve to control the field selector. Storage unit 28 is read out over channel B, and the program is arranged so that an immediate pickup R code is provided on every cycle except when the content of 28 is used as a divisor. The low-order positions of channel B are wired to appropriate transfer positions of the field selector and hence through the transferred points of co-selector 12 to the pickup of the field selector. Co-selectors 11 and 12 are transferred on all total cycles.

Field S	Selector						Fiel	ld Selector		Roo	t→	2	3	4	5	6	7
Pos	ition	А	op	В	(2	S	Pickup		1)	4	4	4	4	4	- 4
· · · · · · · · · · · · · · · · · · ·				X					-		5	6	6	6	$\vec{6}$	$\overline{6}$	6
	0	22	2	2			2	5		3)	õ	1	3	3	1	3
	1	X	2	X			2	*1.		4)	57	05	2	1	3	2
	1	20	ა	28 V			Ζ	*D5		6	5	8	7	5	5	õ	$\dot{0}$
	2	26	3	$\frac{1}{2}8$			2	*be		7)	9	8	7	7	5	5
		Х		Х				•		8	{	4	9 4	8	8	8	8
	3	22	3	28			2	*b7		10	Ś			4	4	9	9
	4	2	5	X 2 S	2	2	2	6		11)					4	4
	7	4	5	2 0 X	4	4	4	U									
	5	2	5	28			2	7									
				Х													
	6	$\frac{2}{v}6$	1	28			2	b_8									
	Ż	$\hat{2}^{\Lambda}2$	2	2			2	8									
			8	Х				-									
	8	22	5	2			2	9									
	0	~	Ę	X			2	A									
	<u>,</u>	<u> </u>		<u>_</u>			<i>(</i> .										
= un nun	herical d	ligit (of far	ctor F	3												
= nn nun	nerical d	ligit o	of fac	ctor I	3												
i = ith hun Equiv	nerical d valent C	ligit o Lard J	of fac	ctor I ramm	3 1ing	for	Square	e Root									
i = ith hun Equiv Field	valent C	ligit o	of fac	ctor I ramm	3 1ing	for	Square	e Root		Root	Со	ode l	Vun	nber	•		
Equivies Field Selector Position	valent C	ligit o	of fac	ramm	3 ning	for	Square	e Root		$\frac{\text{Root}}{2}$	Cc 5	ode 1 1;200	Vun 0000	nber	-		
$\frac{Equiv}{Field}$ $\frac{Field}{Position}$	valent C	ligit o ard I op	of fac Prog B	ctor H ramm C	3 ning S 2	for X-	Square Re	e Root sult		$\begin{array}{ c c }\hline Root\\\hline 2\\\hline 3\\4 \end{array}$	Cc 5 5	ode N 1;200 1;300	Vun 0000	1ber 000 001	-		
i = ith num Equiv Field Selector Position 4	A of the second	ligit o ard I op 1 I	of fac	ctor I ramm C	$\frac{1}{3}$	for X _k	Square Re	e Root sult	-	Root 2 3 4 5	Cc 5 5 5	ode 1 1;200 1;300 1;400 1;500	Vun 0000 0000	nber 000 001 023 013	-		
$ \frac{\text{Equiv}}{\text{Field}} $ Field Selector Position 4 6	A of	ligit o ard I op 1 H 2 H	of fac Prog B Z	ctor I ramm	3 ning S 2 2	for X_k $a/2$	Square Re X _k	e Root sult	-	Root 2 3 4 5 6	Cc 5 5 5	ode 1 1;200 1;300 1;400 1;500	Vun 0000 0000 0000 0000 0000	nber 000 001 023 013 021	-		
i = ith hun Field Selector Position 4 6	A of a contract	ligit o ard H op 1 H 2 H 1	of fac Prog: B Z	ramm C	3 ning S 2 2 2	for X_k $a/2$	Square Re X _k	e Root sult	-	Root 2 3 4 5 6 7	Cc 5 5 5 5 5 5	ode 1 1;200 1;300 1;400 1;500 1;600 1;700	Vun 0000 0000 0000 0000 0000	nber 000 001 023 013 021 123			
i = ith hun Equiv Field Selector Position 4 6	A of a constraint of a constra	op 1 H 2 H 5 H	of fac	ramm C	3 ning S 2 2	for X_k a/2 X_k 1/2	Square Re X_k	e Root	-	Root 2 3 4 5 6 7	Cc 55 55 55 55	ode 1 1;200 1;300 1;400 1;500 1;600 1;700	Vum 0000 0000 0000 0000 0000 0001	nber 000 001 023 013 021 123			
i = ith hun Equiv Field Selector Position 4 6 0 5	A of	op 1 F 2 F 5 F 2 2	of factors $\frac{Prog}{B}$	ramm C	3 ning S 2 2 2	for X_k a/2 X_k $\frac{1}{2}$	$\frac{\text{Square}}{X_k}$	e Root sult $\frac{a}{X_k}$	_	Root 2 3 4 5 6 7	Ccc 5 5 5 5 5 5	ode 1 1;200 1;300 1;400 1;500 1;600 1;700	Vun 0000 0000 0000 0000 0000 0000	nber 000 001 023 013 021 123	-		
$i = \pi h$ hun Equiv Field Selector Position 4 6 0 5 7	A of	op 1 F 2 F 1 S F 2 Z 1 Z	of factors $\frac{P - rog}{B}$	ramm C	3 ning S 2 2 2 2	for X_k a/2 $\frac{X_k}{12}$	Square Re X_k z = a/X $X_k = -a/X$	$\frac{a}{X_k}$		Root 2 3 4 5 6 7	Cc 5 5 5 5 5 5	ode 1 1;200 1;300 1;400 1;500 1;600 1;700	Vum 0000 0000 0000 0000 0000 0000	nber 000 001 023 013 021 123	-		
i = ith hun Equiv Field Selector Position 4 6 0 5 7	A of	op 1 F 2 F 1 S F 2 Z 1 S F 2 Z 1 S F 8	bf fac Prog: B C C S S S S S S S S S S S S S	c c	3 ning S 2 2 2 2 2 2	for $ \begin{array}{c} X_k \\ a/2 \\ X_k \\ \frac{1}{2} \\ X_k \end{array} $	$\frac{\text{Re}}{X_k}$ $\frac{-a/X_k}{X_k}$ $\frac{-a}{X_k}$	$\frac{a}{X_{k}}$ $\frac{a}{X_{k}} - X_{k}$	-	Root 2 3 4 5 6 7	Cc 5 5 5 5 5	ode 1 1;200 1;300 1;500 1;500 1;700	V un 0000 0000 0000 0000 0000 0000	nber 000 001 023 013 021 123			
i = ith hun Equiv Field Selector Position 4 6 0 5 7	A of	ligit d ard I 0 1 F 2 F 1 5 F 2 2 1 5 F 2 2 1 5 F 2 2 1 5 F 2 2 1 1 5 F 2 7 1 1 F 2 7 1 F 2 7 1 7	bf fac Prog: B L L L 8 R	c	3 iing S 2 2 2 2 2 2	for X_{k} a/2 X_{k} $\frac{1}{2}$	Square Re X_k $-a/\lambda$ $X_k - a/\lambda$ $+\frac{1}{2}($	e Root sult $\left(\frac{a}{X_k}\right)$ $\left(\frac{a}{X_k}-X_k\right)$	-	Root 2 3 4 5 6 7	Ccc 55555555555555555555555555555555555	ode 1 1;200 1;300 1;400 1;500 1;600	Vum 2000 2000 2000 2000 2000 2000 2000	nber 000 001 023 013 021 123	-		
i = ith hun Equiv Field Selector Position 4 6 0 5 7 8	A of	op 1 F 2 F 1 5 2 2 1 5 5 F 1 5 5 F 1 5 1 5 1 5 1 5 1 5	Dif fac Prog: B L L 8 R	ramm C	3 iing S 2 2 2 2 2 2 2 2	for	Square Re X_k -a/X $X_k - \frac{1}{2}$	e Root sult $\frac{\zeta_k}{\frac{a}{X_k}}$ $\frac{a}{X_k} - X_k$	-	Root 2 3 4 5 6 7	Ccc 55 55 55 55 55 55	ode 1 1;200 1;300 1;400 1;500 1;700	Vun 0000 0000 0000 0000 0000 0000	nber 2000 201 223 211 123	-		
i = ith hun Equiv Field Selector Position 4 6 0 5 7 8	A o 2 6 2 2 R 2 2 2 2 2 2	op 1 F 2 F 1 F 2 Z 1 5 F 2 Z 1 5 F 1 5 F 1	Dr fac <u>Prog</u> <u>B</u> L L L 8 L L 8 L L L L L L L L L L L L L	ctor I ramm C	3 iing S 2 2 2 2 2 2 2 2 2	for $ \begin{array}{c} X_k \\ a/L \\ X_k \\ \frac{1}{2} \\ X_k \\ X_k \end{array} $	Square Re X_k $A = a/X_k$ $A = -a/X_k$ $A = -x_k$	$\frac{a}{X_{k}}$ $\frac{a}{X_{k}}$ $\frac{a}{X_{k}} - X_{k}$	-	Root 2 3 4 5 6 7	Ccc 55 55 55 55 55	ode 1 1;200 1;300 1;400 1;500 1;700	Vum 0000 0000 0000 0000 0000 0000	nber 000 001 023 013 021 123			
i = ith hun Equiv Field Selector Position 4 6 0 5 7 8 9	A of	ligit c ard I op 1 F 2 F 1 5 F 2 2 2 1 5 F 8 1 5 F 1 5 F 1 5 F	Dr fax 2rog: B L L 8 L 8 L 8 L 8 L 1 1 2 2 2 3	ctor I ramm C	3 ning S 2 2 2 2 2 2 2 2 2 2 2 2	for X_k a/2 X_k $\frac{1}{2}$ X_k X_k X_k	Square Re X_k A = a/X $X_k = -a/X$ A = -a/X A = -a/X A = -a/X A = -a/X	$\frac{a}{X_{k}}$ $\frac{a}{X_{k}}$ $\frac{a}{X_{k}} - X_{k}$	-	Root 2 3 4 5 6 7	Ccc 5 5 5 5 5 5 5 5 5 5	ode 1 1;200 1;300 1;400 1;500 1;500	Vun 0000 0000 0000 0000 0000 0000	nber 000 001 023 013 021 123	-		

FIGURE 4. NTH ROOTING

This is necessary to isolate channel A and B from the second reading contact roll and also to permit forming a channel C address with a single 2 digit.

The wiring of the field selector for the extraction of a root is diagrammed in the Appendix. The order of pickup of the field selector and the equivalent card coding produced is shown in Figure 4, page 55. The process is terminated by the occurrence of a zero result order on field selector position 9. To obtain the required zero result in all possible root extractions, it is necessary to suspend round-off on step 8.

The general process of root extraction consists of

- a. computing X_k^{n-1} ,
- b. forming a/X_k^{n-1} and subtracting this quotient from X_k ,
- c. dividing this difference by *n*, using the code number (see Figure 4) as an approximation to *n*,
- d. subtracting this difference from X_k to form X_{k+1} (with round-off suspended),
- e. forming $X_k X_{k+1}$ for a "zero-not zero" test, and finally,
- f. recomputing X_{k+1} by subtracting the above difference from X_k , and storing it at storage address 22 to be used as the next estimate.

The card-programming necessary to begin the *N*th rooting procedure is shown with the examples of programming in the Appendix.

Appendix

The Appendix is devoted to the explanation of wiring diagrams not fully covered in the text and to the presentation of a few examples of programming which serve to illustrate the use of certain of the special functions.

Suppressions. The diagrams of the various types of suppressions (Figure 5) are self-explanatory. Type 6H6 thermionic duodiodes are used.

604 Analysis. The following notes on the 604 analysis give most of the information required to wire the control panel. The details of the analysis and wiring are shown in Figure 6.

1. All three-position storage units are given an 8-6 assignment. Read-in and read-out of factor storage 1-2 and factor storage 3-4 are coupled, respectively.

2. Factor storage 1, 2, 3, and 4, general storage 1 and 3, and the multiplier quotient read-in are coupled to electronic all-cycles.

3. General storage 4 read-out and counter read-out and reset are coupled to electronic all-cycles.

4. On a 3 code, -50 is read into the low-order positions of the multiplier quotient, and on a 2 code, +52 is read in.

5. One of the calculate selectors 1, 2, or 3 must be transferred on every calculate cycle. Calculate selector 6 is coupled to calculate selector 1, and calculate selector 7 is coupled to calculate selector 2.



FIGURE 5. SUPPRESSION WIRING

			Sh	ift			E	lmit	tte	r	Ba Te	l. st	Fu	nct	ior	าธ	Cti	:Co	ntr	rol	Μ.	Q.		Ge	ene	ral	st	ora	age		F	actor	Stora	ge		Mi	sc.	
Program	Out 6	out 5,	Out 4, In 3	Out 3, In 4	Out 2, In 5, 1	In 6	5	4	0	1	tep.Sup.	.U.Sel.	ult. +	Divide	🛃 adj.	ero Test tr.Beset	R&R	RO	RI +	RI -	RI	RO	GSI RO	GS2 RO	GS3 RO	GS4 RO	GS1 RI	GS2 RI	GS3 RI	GS4 RI	FS1 & 2 RO	FS3 & 4 RO	FS1 & 2 RI	FS3 & 4 RI	.U.Prog Rpt.	Group uppress P.U.or D.O.	uppress Type	Notes
1												P44	<u> </u>							X						1	†		1			х				01	A	
2																	Х																	Х			A	
3		<u> </u>												_					1.4 N	T T			х												х		В	
4	1	ļ									Ť						_		N N	2.4 T				<u> </u>	X							L				ļ	В	
5				 				L_									_		,						x		х	ļ									с	7
6	1	ļ	<u> </u>			X		ļ		Б.1 Т							5.1 N		'n⊥									Ļ	х								D	6
7		<u> </u>	ļ		<u> </u>	Ń				-									X	-		N.			1.3 T		ļ										B	
8	X				2.2						-			-			X			-				-		-				X							B	
9	<u> </u>				T														X	-			-	-		-		–			<u> </u>	v	1.3				E	
10	0 1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-													Ň						-			[<u> </u>		-						A	7.2	7.2			F	
12	ក្តីខ្ល	├			-					ļ				_			-X		v	\vdash	-					v		–					T	N			F C	
13	000	-					x				Y								A.	Y		<u> </u>				<u>^</u>											r G	
14	4 C C C C	1	1.				x												x	A				†			1										G	
15	to to	<u> </u>					1	Х												x		-								· · ·							С	
16	ped.																х											X	0								G	
17	Tan-																	Х			Х								R								Н	
18	00	X	ļ																X									-				Х					С	
19	X		-		-	-											X											N.1	4 7					Ť			I	
20					X					X								_		L	X	-	<u> </u>			-			le								G	
21	-					v	-		7.]										X			-	X			-			Ino		7.1						G	
23	x	-		-		A			Ň							-	v		A			-	-			-			0	v	T						J	
24	<u> </u>		<u> </u>			-		-					_	-	-		~		v			-		v													G	
25	\vdash	<u> </u>		-	-		-			v	v								<u>^</u>	v				<u>^</u>				-	-								u c	
26	1	-		x	1					X	- <u>^</u>				[-					X																c	
27					1					х	Х					_				Х																	С	
28			X							Х											X																С	
29		_	ļ							X	х									X			ļ														С	
30		X	<u> </u>			<u> </u>	-			X			-	Ц							X	ļ					<u> </u>		_								C	
31		2.1								X 2,1	X					-				X	2.5	2.5											7.3				C	
32	+	T.	-							N v				-							N	Т											T				F'	
34	1	\vdash			-					^	^		-	+		-	x														-						G	
35	1	+											2.3	2.3		-+				-		-										x				P.U.3	В	
36	1	1	x							-			1	-		-		x			x															1.010	G	
37		X	1															X			Х																C	
38	Х																Х																3.2 Ť	3,2 Ň			В	
39		ļ												_		_			X	L								ļ			-6.3 N	6.3 Ť					D	
40	ļ		X													_	X			_														Х			G	
41	<u> </u>	-		-	<u> </u>			-		$\left - \right $		-				-	_				Х			Х													H	
42	+	-	X			-				H			Y	-	_	-+			X										_		x						-G u	
44	<u> </u>	x	+										Λ			-	-			-	6.2		-						_			6 ₁ 2	y				D	
45	1	X	1														-		х		IN	x										<u> </u>						
46	1	1	1			X													x												х							
47				Х											8 N																						К	
48	X				L											T		x																х				
49	Ļ		ļ	ļ														х			X														4N		L	
50			ļ				 							\neg		_				X		X														P.U.4	L	<u>.</u>
50		<u>X</u>			-										_		X							$\left - \right $									X			D U 2		
53		-	x					-		$\left \cdot \right $					_	\neg	v		X						-			v				X				P.U.2	v	
54	+	x	A .	-	-		-	-								\neg	^		x	\vdash		\square		y				^									ĸ	
55		+			-	x										x				-		[**	
56	t	1	1		1	[\square								-			Х							x										D.0.3		
57										X										Х																	L	
58																	X													x								
59		-												_																	X			Х		P.U.3	L	
Lon	L	L	1																X	L			L									X					М	

(n.m means calculate selector n, mth point)

FIGURE 6. WIRING DIAGRAM FOR NACA FLOATING-POINT CPC CONTROL PANEL

6. One of the program exits of program step 6 is made equivalent to two exits by the use of 6H6 diodes. The diode wiring is as follows:



7. The use of diodes in suppressions type C, E, F, and I is shown in the suppression diagram (Figure 5).

8. General storage 3 read-in is coupled to "read units into 6th." On program step 6, with calculate selector 3 normal, counter read-out and reset prevents "in 6th" from being effective, and the read-in to general storage 3 is not affected.

9. "Drop-out of group suppressor 4" is coupled to "read units out of 6th."

402 Diagrams. To simplify wiring and discussion of the 402 control panels, the wiring has been illustrated in three parts (Figures 7-9).

Wiring of Numerical Input and Clearing (Figure 7). Co-selectors 1 and 2 control the input of a factor from storage or a card, or the re-use of the previous result to channel A. Co-selectors 3 and 4 perform the same function for channel B. Co-selectors 7 and 8 permit the unconditional substitution of a number in the "s" field of a card for a zero result, or the conditional substitution of a number in the "s" field for a non-zero result. Counter group 7 of the accounting machine is used for the entry of a factor from the table field by use of spread read-in. Pilot selector 1 controls detail printing of the 402, and pilot selector 8 controls "machine stop" and causes a list cycle for each stop.

The early field selector controls entry of the table field sign. The sign impulse for the number in the "s" field is synthesized by combining a "hot 10" with the coupling exit impulse of pilot selector 10 which transfers when "s" is negative (that is, there is an X in column 53). This sign is wired to the transferred points of pilot selector 14 which transfers on a conditional 7 code.

Operation Control (Figure 8). Co-selectors 5, 6, 9, 10, 11, and 12 control which of the three independent program fields is to be read. Negative balance selector 2 controls which of the conditional code columns will be read. When this selector is normal, the N column is read; when trans-



FIGURE 7. NACA FLOATING-POINT CPC CONTROL PANEL, WIRING OF NUMERICAL INPUT AND CLEARING



FIGURE 8. NACA FLOATING-POINT CPC CONTROL PANEL, OPERATION CONTROL

ferred, the T column is read. The pickup of this negative balance selector is controlled by pilot selector 15. When this pilot selector is normal the pickup impulse comes from the counter sign; hence, the discrimination is "negative-positive." Pilot selector 15 is transferred by a conditional 0 code. When this occurs the general storage 4 sign provides the pickup impulse, and the discrimination becomes "zeronot zero."

Nth Rooting Routine (Figure 9). The transferred positions of the field selector accept the emitted programming which controls the Nth rooting routine. At a specific field selector position the channel A and B addresses for the following calculate cycle are read as well as the R codes for the channel C address, and the shift code for the previous calculate cycle.

The order of pickup of the field selector is ordered by the code word by wiring the last four positions of channel B to the pickup control.

Special program is begun by a 1 code read conditionally or unconditionally.

Also shown on this diagram is the method used to obtain the correct factor in the multiplier quotient for exponent adjustment. Calculate selector pickup 2 is wired to the units position of the multiplier quotient, an emitted 5 is wired to the ten's position, and calculate selector pickup 3 is wired to the sign position.

List Wiring. The list wiring appears in all three diagrams. The operation codes list from the calculate selector pickup hubs. Pilot selectors 11 and 12 are wired so that a 5 or 6 will list to indicate when one of the transferred program fields has been read.

Examples of Programming. The programming shown on the CPC program code sheet (Figure 10) is that required for the calculation of the *n*th root of a when no initial estimate is available. The functions of the card coding are as follows:

Card Number	Function
001	Read a into storage unit 26.
002	Read code number into storage unit 28.
003	Cause one card cycle delay before storage 26 may
	be read out.
004	Form the "sgn of a." (If a is negative, sgn of $a = -1$; if a is positive, sgn of $a = +1$; if $a = 0$, sgn of $a = 0$.) The 0.2 punch in column
	4 and the 6 in column 5 cause storage unit 26 to



FIGURE 9. NACA FLOATING-POINT CPC CONTROL PANEL, NTH ROOTING ROUTINE

	Clear X's 1 2 3 4 5 6 7 50																															
Card		Pr	ogra	n Fie	eld			Factor I	ield A		Facto	or Fi	ield B		Table I	leld	S1	bstituti	on Field	Pr	ogram	Fiel	a 50)	Pı	ogram	Field	1 60	E		List	Π
no.	Α.	0.	в.	c.	S.N	.T.		8	_			b		-	t			s		A5 0	5 B5	C5	S ₅ N	$5^{T}5$	A ₆	06 B6	C6	S6N6	T6 [™]	II	dent.	
11213	141	5 6	7[8	9/10	<u>1111</u>	213	1415	16[17]18[.	19:20:21:22:23	2425	2627	28(29	30313233	3435	3637383	940414243	44/45	46/47/48/49	50515253	54555	<u>6 57 5</u>	8 <u> 59 60</u>	6116	263	6465	666768	16970	7172	73 74	757	77 78	7980
001	0	0		26	2			а	1				i																			
002	0	0		28	2			code	number				1 1					1	8				ŀ								Π	
003	5			(1	ntei	po	sed	Operation)											1				Π									
004	02	6 ² 5	26	2 2	2 2 2 X X						1				F 1																	
005		5	0 2 6		0						1							1 1 · · ·														
006	0	0	R		2	(8	07	000	0 0 0 1 0									 														
007	0	0	R		2			m			1																					
008	R	2	0 0		4						1	N	 	İ					 													
009	R	3 5	22		2						<u> </u>		<u> </u>					 														
010			•		2	1									1				1													
011	R				2			1					1		1				 													
012	2																		 													
											1																					
											 							1														
																								•			\Box					
	1				ΓT															ΙT	1									Ιſ		$ \top $

Figure 10. Calculation of the Nth Root of a

60

Card Number Function (continued)

be read out with the sign punched in column 23. That is, we obtain -|a| over channel A. This is divided by -a, giving the sgn of a. The X's in columns 12 and 13 prevent a machine stop in case a = 0, and the division 0/0 gives a zero result.

- 005 Form |a| to use in calculating the initial estimate. The channel B address code causes a to be read out over channel B with the sign punched in column 33 The 0 shift code brings the factor in with the first exponent digit placed as the first numerical digit and numerical digits 7 and 8 dropped.
- 006 Add 07;00000010 to the result of card 005. When this is done all the numerical digits of a are dropped, and a number $01;1e_1e_2$ is obtained, where e_1 and e_2 are the first and second exponent digits of a, respectively. The X in column N instructs the calculator to treat 07;00000010 as a number in normal notation. The 8 in column T is required only when even roots are being extracted. If the sgn of a, calculated on card 004, is negative, column T will be read on card 006, causing a machine stop.
- 007 A number m is added to the exponent digits. The number m has been calculated to give a good first estimate of the desired roots and has the following values for the different roots:

Root	т	
2	-00;4854	
3	00;0260	(X in column 12)
4	00;5340	
5	01;1045	
6	01;1556	
7	01;2067	

The presence of an initial numerical digit of zero in m for the cube root requires an X in column 12 when this estimation is used.

- 008 Divide the sum from the previous card by n, the order of the root (that is, 51;2 for square root), and shift the result 4 places to the left. This puts the exponent digits of the estimate in their correct position.
- 009 Multiply the estimate by the sgn of *a*. This is necessary to obtain a negative estimate of the odd root of a negative number.
- 010 The 1 codes in column 12 and 13 start the *n*th rooting procedure unconditionally.
- 011 List the *n*th root.

Some of the steps described above, obviously, are not always necessary. If an estimate of the root is known, it is only necessary to read it out over channel A or B on the card (009) preceding the 1 code which starts special program. If the number a is known to be positive, formation of the sgn of a is unnecessary.

Use of Auxiliary Program Fields and Conditional Codes. Examples of the use of the auxiliary program fields and some of the conditional codes are shown in Figure 11. The first six cards illustrate a transfer of control to program field 50 under the control of the sign of a result. In the following example it was desired to approximate the operation $a_{k+1} = a_k e^{da/a}$ by the use of only a few terms of the Taylor series for $e^{da/a}$. It is known that the Taylor series for e^x converges more rapidly for positive x than for negative x; therefore, for positive da/a, the operation is approximated by

$$a_{k+1} = a_k \left[1 + \frac{da}{a} + \frac{1}{2} \left(\frac{da}{a} \right)^2 \right],$$

and for negative da/a, by

$$a_{k+1} = \frac{a_k}{\left[1 - \frac{da}{a} + \frac{1}{2}\left(\frac{da}{a}\right)^2\right]}$$

The machine operation forms the absolute value of da/a, and computes the first three terms of the series for $e^{|da/a|}$. A transfer of control is used to "remember" the sign of da/a and thus carry out a multiplication if da/a is positive, or a division if da/a is negative.

This program illustrates the timing of conditional codes relative to the command whose result controls the conditional operation. Conditional codes are read from second reading and are effective on the cycle following their reading. Therefore, it follows that the quality of the result of the command on card n controls the reading of conditional codes on card n+2 which in turn modifies the results of the command ordered on card n+1.

Card by card the programming for this example is as follows:

Card	
Number	

Number	Function
001	Simulate the calculation of da/a by reading it from a card. The 7 codes in columns 12 and 13 on card 002 cause the result da/a to be stored unconditionally as its absolute value.
002	Form $\left \frac{da}{a}\right ^2$. The R codes for the channel A and
	B addresses, and the 3 operation code causes re-use of this factor.
003	Form $\frac{1}{2} \left \frac{da}{a} \right ^2$. The 5 code in the T column causes
	a transfer to program field 50 if the result of card 001, da/a , was negative. The same calcula-

tion occurs on either program field.

Card	Program Field Factor Field A					leld A	T	Factor Fi	leld B	Τ	Clear X Table F	' <u>s[1]2[3</u> ield	4 5 Śt	bstitutic	on Field	Program Field 50 Program Field 6					. 60			List	T					
no.	A.	0. B.	l d.	IS.J	I.T.				<u> </u>	h		1	t		 	e		Ac 10	el Be	C- B	-01		2 104	B ₂	621	SANA	TAP		ident	.
123	4 5	6 7	8 9 1	011	1213	141	51617 1819	2021222	2425	2627 28 29	3031323	33435	36373839	40 41 42 43	4445	46474849	50 51 52 53	54555	65758	59606	162	5364	65 666	768	6970	7172	737	47517	67717	817980
001	00		7 2	2 2			da/a																					Π	T	TT
002	R	3 R		2	7 7		1										 											Π	T	
003	00	3 R		2	5	50	5			 							 	0 0 3	3 R	2									T	T
004	82	R	_	2	_		 			 							<u> </u>	8 2	R	2	5	5								
005	00	R	1	2	_	51	1	ļ	ļ	ļ	<u> </u>							0 0	R	2	5	5				_				
006	00	3 R	1 1	2			ak						 				i 	0 0 2	R	112	5	5								
883	-	(1	nter	pos	eđ	Ope:	rations)		<u> </u>	 																		Ц	$\downarrow \downarrow$	44
010	00	500) 1 1	2	1	51	1	ļ	51	1	ļ	1	<u> </u>									_	\rightarrow					\downarrow	11	$\downarrow\downarrow$
011	Ŗ		1 2	2	0 C		<u> </u>		ļ		<u> </u>		1	ļ			<u> </u>					_	_					$\downarrow \downarrow$	44	\downarrow
012					3 2		ļ	<u> </u>			ļ	5 1	3		51	4	i 					_	$\rightarrow \rightarrow$				_	1.	$\downarrow\downarrow$	$\downarrow \downarrow$
013	00	500	13	2	_	51	1	<u> </u>	51	2	<u> .</u>		 	ļ			i 											\downarrow	$\downarrow\downarrow$	
014	R		14	2	00					ļ			<u> </u>	1			 					_				_	_		++	·
015					3 2		 		 	1	<u> </u>	5 1	5	ļ	51	6	 		+		+									++
016	11			2	_			 		ļ		<u> </u>									$\left \right $	+				+		++	++	┿╋
017	12			2			<u> </u>		<u> </u>		- 						 				+	+						┼┼	++	++
018	13			2	+			<u> </u>		 	1			1			ļ		+			+	\rightarrow			+		++	++	++
019	14			2	+-		<u> </u>						 								++							++	++	++-
020	77			2			<u> </u>			 		<u> </u>	1								+	+	++				_	++	++	++
021	0 0	<u>5 0 c</u>	1 5	2	_	21	1	ļ -	22	2	ļ	ļ	ļ	. 			<u> </u>				+							$\left \right $	++	
022		(Inte	rpo	sed	Op	rations)		 			ļ	ļ			1			-	$\left \right $	_					_	$\downarrow \downarrow$	$\downarrow\downarrow$	$\downarrow\downarrow$
023	0	<u></u>		$\left \right $	4-		<u> </u>	ļ			<u> </u>	ļ							4			\perp					_	$\downarrow\downarrow$	$\downarrow \downarrow$	$\downarrow\downarrow$
024	<u>1 5</u>	5 1 5		2	x x		<u> </u>				ļ	ļ	i !													+	_	\downarrow	$\downarrow \downarrow$	$\downarrow \downarrow$
025							<u> </u>	ļ	L		ļ	 	i I	 							\parallel	_					\downarrow	$\downarrow \downarrow$	$\downarrow \downarrow$	44-
·							ļ					}																		

FIGURE 11. USE OF AUXILIARY PROGRAM FIELDS AND CONDITIONAL CODES

Card Number Function (continued)

- 004 Form $\left|\frac{da}{a}\right| + \frac{1}{2}\left|\frac{da}{a}\right|^2$ by adding the content of counter group 2 to the previous result. Reset counter group 2. If program field 50 is being read, the same operation is performed, and the 5 codes in the N₅ and T₅ columns cause the continued reading of program field 50.
- 005 Form $1 + \left|\frac{da}{a}\right| + \frac{1}{2}\left|\frac{da}{a}\right|^2$ by adding 1 to the previous result.
- 006 Form the product $a_k \left(1 + \left|\frac{da}{a}\right| + \frac{1}{2}\left|\frac{da}{a}\right|^2\right)$, or the quotient $a_k \left/ \left(1 + \left|\frac{da}{a}\right| + \frac{1}{2}\left|\frac{da}{a}\right|^2\right)$, and store in storage unit 11. The first instruction comes from the basic program field which is read if da/a is positive. The second instruction comes from program field 50 which is read if da/a is negative. (Normally, a_k would have been a computed quantity, but for purposes of illustration it has

been entered from a card.)

Cards 07-09 are blank. The programming in cards 010 to 025 illustrates the use of a "zero-not zero" discrimination, and the use of an involuntary reference to the "s" field to store the sign of a "computer zero." The timing of conditional codes is again illustrated; the reading of codes 3N and 2T on cards 012 and 015 depends on the quality of the results commanded on cards 010 and 013, respectively. The control is made to depend on the "zero-not zero" condition of the results by 0N and 0T punches on cards 011 and 014.

Card by card the programming for this example is:

Card

Number	Function
010	Form $A - B$ (which in this case is an exact
	zero) and store the result in storage unit 11.
011	Re-read the result of card 010. If it is non-zero,
	store it in storage unit 12. If it is zero, store the
	number punched in the "s" field on the next card
	in storage unit 12.
012	Read the "s" field conditionally (see card 011) and
	also read the conditional codes in the T and N
	columns. Because the result of card 010 is zero,
	the T code 2 is read, and the N code 3 is ignored.
	(No operation code is used on this card.)

Card		
Number	Function	(continued)

013-015	Repeat	the	preceding	operation	for	a	non-zero
	result.						

016-020 Cause read-out of the stored results to illustrate the effectiveness of the controls.

Cards 021 to 024 illustrate involuntary reference to the "s" field.

- 021 Multiply A times B. The numbers chosen in card fields a and b produce a negative result which is treated as zero because its exponent digits are negative. The stored result is a 1 in the low order position of channel C. This 1 is read from the "s" field of card 022 and is stored with the sign of the zero result.
- 022-023 Mark time to allow the result of card 021 to be stored in storage unit 15.
 - 024 Form sgn of the content of storage unit 15. This is done by punching X-23 which causes A to be - | content of storage unit 15|, and by performing a negative division with the 2 and 5 codes. The XN and XT punches suspend the automatic formation of a zero result, and the machine stop that would ordinarily result from division with both factors in abnormal notation.

DISCUSSION

Mr. R. W. Smith: Why did you use a calculate selector rather than zero test in order to stop indefinite program repeating?

Mr. Patton: A non-zero result of a zero test is used to suppress program repeat when the high-order digit is nonzero. In the case of an identically zero result, left shifting and program repeating would occur indefinitely so that the negative balance selector of the 402 is used to pick up a calculate selector to suspend 604 calculation in such cases.

Dr. Yowell: These floating decimal control panels are very fine, but I am a basic unbeliever! I think the floating decimal drives problems underground in many instances, and Mr. Patton's first remarks as to the disadvantages of the floating decimal must be taken very seriously. There are times when one runs into trouble with a fixed decimal control panel, and when a floating decimal panel is useful in cases where one has decimal troubles. If one is trying to solve a system of equations and obtains wide fluctuations over the numbers, it generally means there is a near zero determinant and one is "getting back" accuracy out of the solution, no matter what is done. It is the problem of analysis of results of a floating decimal panel that causes a great deal of trouble. One can get out of a floating decimal procedure eight digits which look very good and may mean absolutely nothing.

So the fact that it is convenient to use floating decimal panels at times does not avoid the basic difficulty of sitting down and thinking about results from the very beginning. I would like to stress at this time that the convenience of the floating decimal panel does not, in most cases, really help out in critical numerical cases of small determinants in matrices where the scale factors fluctuate widely. One has to think about them sooner or later; the floating decimal panel just glosses over this point and is deceptive to someone who is uninitiated in computing procedures.

Mr. Patton: However, if we have a problem in which great accuracy is required, the floating point panel turns out to be wonderful for finding the magnitudes of the number that will occur in the subsequent fixed-point operation which will give the accuracy desired. We do not use the floating point panel indiscriminately.

Dr. Arnold: Very briefly, the 604 control panel that we have has only addition, multiplication, and division, with the change of sign of one of the factors. Previous results are read back into the 604 through the use of the selectors on the 417. One feature of our panel is the manner of shifting before addition, which we do essentially through repeated division by two. The difference in the exponents is divided three times by two to give one additional shift in another way.

Addition is handled entirely without program repeat addition with or without carry-over or addition with loss of digits. The zero prints nothing. We have two operations on the 417 panel to convert an exponent to a number or a number to an exponent.

Dr. Thomas: As to Dr. Yowell's remarks on the use of floating decimal techniques, I think there is really quite general agreement as to what they will accomplish. If there are as many significant digits in the floating decimal numbers as in the fixed decimal numbers, nothing is ever lost by using the floating decimal. But, of course, one may get results which have no sense! However, any result obtained using a fixed decimal can be obtained using the floating decimal with the same number of digits; and a floating decimal system would eliminate the headache of deciding exactly where overflow would occur and where to program shifts.

Of course, it is necessary to store the index; so there are fewer digits in a floating decimal than in a fixed decimal. Hence, the decision of which is preferable, in any case, depends upon the accuracy desired in the particular problem.

The only danger in using the floating decimal, in general, is that one may attach significance to results which have no accuracy left. The disadvantage is not having spent time working out exactly where overflow will occur.

I think that when we have stored-program machines in which arbitrary changes of program can be made, one will normally work nine-tenths of the time with fixed decimals. But they will be placed within a longer computation where floating decimal will be used.

Dr. Hamming: I have been using floating decimal on machines at Bell Telephone Laboratories for about five years, and my response to Dr. Yowell is this: To use floating decimal is like playing with fire! On the other hand, that is no argument at all against fire, which is a very useful device to humanity.

Mr. Madden: To overcome the annoyance of intermittent machine errors, we at RAND have developed a generalpurpose setup, which uses the familiar principle of casting out nines. However, we cast out 99's and carry a count of the number along beside the number itself on the channel and in the storage. We have to sacrifice an operation in order to accomplish this. We have addition, subtraction, multiplication, and division.

Dr. Hurd: There are eight-digit numbers?

Mr. Madden: Eight-digit numbers, two-digit counter. We divide by 99. All the numbers coming into the 604 are checked against their counters to make sure there hasn't been a failure in the reading. All of the operations taking place in the 604 are also checked by this casting-out-99's principle.

Further, we carry the count plus one; if there is complete failure to read so that there are zeros in the number and zeros in the count, this error is detected.

Catapult Takeoff Analysis

JOHN R. LOWE

Douglas Aircraft Company, Inc.

×

DURING World War II, the load-carrying ability of naval carrier aircraft was exploited to the fullest. Obviously, any increase in bombs, ammunition or fuel capacity was an increase in striking power for an airplane capable of becoming airborne. In fact design overload conditions on combat aircraft were exceeded continuously; the only requirement on the resulting combination flying arsenal and tanker was that it take off from a carrier deck.

These increases in airplane gross weight required increased takeoff distances. Eventually it became necessary to assist heavy airplanes to attain flying speed, and the carrier catapult, as it is known today, came into being.

With the advent of jet propulsion, shipboard catapults became even more important. Jets, with their slow accelerations and necessarily high fuel capacity, required catapulting from their inception. At the present time, nearly all carrier based aircraft are catapulted from the carrier deck; pilots no longer are required to "lift off."

A carrier catapult derives its force from air, under pressure, stored in an air-oil accumulator. The air acts on a column of oil and forces it into a hydraulic cylinder. The force and motion of the hydraulic ram are transmitted through a cable system to a shuttle in a slot on the carrier deck. Another cable—or bridle as it is called in this case connects the shuttle to the airplane.

The firing operation of a catapult consists of opening a valve which allows oil, backed up by air pressure, to enter the hydraulic cylinder. The hydraulic ram moves the shuttle which, in turn, pulls the airplane along the deck for the length of the catapult power stroke. At the end of the power stroke, the ram and shuttle are braked, allowing the airplane to drop the bridle and pass on over the edge of the deck.

Prior to each catapulting the airplane must be "tensioned" in order to remove all slack in the bridle and cable system. In the "tensioning" operation a small force is applied to the shuttle which, through the bridle, pulls the airplane forward and tensions a restraining link which connects the airplane to the carrier deck. When full catapult power is applied, this restraining link—or "holdback unit" as it is called-breaks, and the airplane starts its accelerated takeoff run.

Catapult end speed is not the only requirement for flight. Lift is proportional to the aerodynamic angle of attack; hence, there is the additional requirement that the airplane rotate to the correct angle of attack in the vicinity of the end of the power stroke. Since the airplane is pulled only by the catapult shuttle, provisions for rotation to the nose-up angle necessary for flight must be accomplished in the design stage of the airplane.

There is one more serious problem concerned in the catapulting phase of aircraft takeoff. The problem is that of excessive "porpoising" or oscillating of the airplane both vertically and rotationally during the catapult power stroke. Oscillations of this sort are objected to strongly by pilots and in addition are usually damaging to the airplane.

The purpose of the catapult analysis is to determine, during the design stage of the airplane: (1) proper location of the airplane's catapulting accessories, (2) loads imposed during the catapult operation, and (3) the effect of the many variables (gross weight, center of gravity [c.g.], location, wind over carrier deck, etc.) on the flight path of the airplane.

Basic Equations

The flight path of the airplane can be described by two simultaneous second-order ordinary non-linear differential equations.

The basic equations of motion are shown in Figure 1.

The catapult operation may be considered in two phases. The first phase covers the period from the end of tensioning to holdback release. It is assumed that 0.15 seconds are necessary for the horizontal component of the load in the holdback unit to increase lineally from the value obtained statically with full thrust and 1500 pounds tensioning force to the load required to break the holdback unit. This period is represented by negative time, -.15 sec. to 0 sec. The second phase deals with the period from holdback release to the end of the catapult power stroke. This period is represented by positive time.







(7)

$$\frac{K_{22}}{K_{22}} = \frac{-[K_7 \sin \alpha + K_6 \cos \alpha - K_{14} \cos ((24 + \alpha) - K_{21} \sin ((24 + \alpha) + K_{32} \cos ((24 + \alpha + T)))][K_5] - [Y - (K_{25} - \vartheta_1)][K_3][K_5] + [HB][h]}{K_{22}}$$

$$\frac{-K_{23}\left[(K_8+K_9t)^2(\alpha-\angle 2)-K_{29}\left(K_8+K_9t\right)(\dot{Y})\right]-K_{31}\left(K_8+K_9t\right)(\dot{\alpha})}{K_{22}}$$

$$\begin{split} \theta &= \sin^{-1} \left[\frac{Y - K_{11} \cos \alpha + K_{10} \sin \alpha}{K_{12}} \right] \\ \phi &= \sin^{-1} \left[\frac{Y - K_{21} \cos \alpha + K_{10} \sin \alpha}{K_{23}} \right] \\ \kappa_{5}' &= K_{5} (1 \pm K_{35}) \begin{cases} \text{when } \overline{Y}_{n} < \overline{Y}_{n-1} K_{35} \text{ is POSITIVE} \\ \text{when } \overline{Y}_{n} > \overline{Y}_{n-1} K_{35} \text{ is POSITIVE} \\ \text{when } \overline{Y}_{n} > \overline{Y}_{n-1} K_{36} \text{ is POSITIVE} \\ \text{when } \overline{\overline{Y}}_{n} < \overline{\overline{Y}}_{n-1} K_{36} \text{ is POSITIVE} \\ \frac{\overline{Y}}{F} &= \frac{K_{7} \cos \alpha - K_{19} \sin \alpha + Y}{\cos (\sqrt{5 + \alpha})} \\ \overline{Y} &= \frac{K_{15} \sin \alpha + K_{7} \cos \alpha + Y}{\cos (\sqrt{5 + \alpha})} \\ \overline{Y} &= \frac{K_{16} \sin \alpha + K_{7} \cos \alpha + Y}{\cos (\sqrt{5 + \alpha})} \\ HB \cos \phi &= K_{30} + K_{34} t \end{cases} \\ k_{15}' &= K_{00} \left(K_{52} - \overline{\overline{Y}} \right) + K_{61} \text{ when } \overline{\overline{Y}} < K_{52} \\ h &= K_{20} \sin (\phi - \alpha) - K_{27} \cos (\phi - \alpha) \\ \end{array}$$

FIGURE 2. DETAILED EQUATIONS FOR CATAPULT TAKEOFF ANALYSIS

Definition of Symbols for Detailed Equations in Figure 2	
	UNIT lbs
K_1 = Aerodynamic drag coefficient (polynomial in β)	$\cdot \frac{108}{(\text{ft./sec.})^2}$
$K_2 =$ Engine thrust $K_3 =$ Main gear drag coefficient (= 0 for $t \leq 0$)	. lbs. . lbs./lb.
$K_4 =$ Horizontal inertia force (= 0 for $t < 0$) $K_5 =$ Main gear ground reaction (polynomial in \overline{Y})	. lbs.
$K_0 =$ Distance along fuselage reference line (1.r.l.) from c.g. to intersection of main gear strut $K_7 =$ Perpendicular distance from c.g. to f.r.l	. in. ft /sec
$K_8 =$ Velocity of carrier $K_9 =$ Horizontal airplane acceleration (three different linear functions of t) $K_{10} =$ Distance parallel to f.r.l. from c.g. to catapult hooks	. ft./sec. ² . in.
$K_{11} = \text{Distance perpendicular to first from e.g. to catapute nooks}$. in. lbs.
$K_{12} = \text{Aerodynamic lift coefficient}$	$\frac{1}{\text{deg. (ft./sec.)}^2}$
$K_{14} =$ Perpendicular distance from main gear strut center line to offset	<i>.</i> in.
$K_{15} =$ Tail bumper ground reaction (normally, polynomial in \overline{Y}) $K_{16} =$ Gross weight of airplane	. lbs. . lbs.
$K_{17} = Mass \text{ of airplane}$	ibs. sec. ²
K_{18} = Distance along f.r.l. from c.g. to intersection of nose gear strut	. in.
K_{19} = Distance along f.r.l. from c.g. to intersection of tail bumper strut	. in.
$K_{21} = Distance from f.r.l. to main gear strut offset in line of action of main gear strut$. in.
$K_{22} =$ Moment of inertia in pitch	. lbs. in. sec. ²
$K_{23} = \text{Aerodynamic moment coefficient}$	$\frac{1}{\text{deg. (ft./sec.)}^2}$
$K_{24} = \text{Tail bumper tire radius}$. in.
$K_{26} = Distance parallel to f.r.l. from c.g. to holdback pivot point$. in.
K_{27} = Distance perpendicular to f.r.l. from c.g. to holdback pivot point	. in.
$K_{30} = \text{First term in equation for } HB \cos \phi$. lbs.
$K_{\mathfrak{s}\mathfrak{l}}=\operatorname{Coefficient}$ of aerodynamic moment due to $\dot{\alpha}$	$\frac{111. \text{ lbs.}}{\left(\frac{\text{ft.}}{1}\right)^2 \left(\frac{\text{rad.}}{1}\right)}$
K_{32} = Radius of main gear arc as strut extends	\sec./ \sec./
$K_{34} = \text{Second term in equation for } HB \cos \varphi$.	. lbs./lb.
$K_{30} = \text{Tail bumper strut friction}$ $K_{30} = K_{30}^2 - K_{30}^2$	1000000000000000000000000000000000000
$K_{51} = 2K_{21}$.in.
$K_{52} = \overline{\overline{Y}}$ for tail bumper bottomed	. in.
$K_{00} = $ Spring rate of fuselage in vertical bending	. lbs./in. lbs
$\angle 1$ = Horizontal intercept of C_L versus α curve	. deg.
$2 =$ Horizontal intercept of C_{M} versus α curve	. deg.
$\angle 4$ = Angle between perpendicular with f.r.l. and main gear strut	. deg.
$\angle 6 =$ Angle between perpendicular with f.r.l. and tail bumper strut	. deg.
$\Delta t = \text{Time increment}$. sec.
Y = Distance from f.r.l. to ground in line of action of main gear strut	. in.
Y = Distance from f.r.l. to ground in line of action of nose gear strut	. in.
Y = Distance from f.r.l. to ground in line of action of tail bumper strut $\alpha = \text{Angle of pitch of f.r.l.}$. in. . deg.
Y = Vertical distance from c.g. to deck	. in.
$\phi =$ Angle of catapult holdback unit with deck	. deg.
$\delta_1 = Main \text{ gear tire deflection (rational function of } \overline{Y})$. in.
$F_B = \text{Catapult bridle torce.}$ $HB = \text{Catapult holdback unit force } (= 0 \text{ for } t \ge 0)$	Ibs. Ibs.
\ddot{Y} = Vertical acceleration of airplane at c.g	. in./sec. ²
$\dot{Y} =$ Vertical velocity of airplane at c.g.	.in./sec.
$\ddot{a} =$ Angular velocity of an plane about c.g. $\ddot{a} =$ Angular acceleration of airplane about c.g. T = Angle between perpendicular to main gear strut through offset and main gear radius of extension	. rad./sec. ²
h = Catapult holdback unit force moment arm about c.g	.in.
$\beta = \text{Angle of attack}$	deg.

Method of Static Balancing to Establish Boundary Conditions

It is desirable that the airplane be in its static attitude for the first point of the analysis; that is the ΣF and ΣM should equal zero when the airplane is at rest. One may see from the equations that it would be very difficult to express Y and α explicitly so that they may be determined such that \ddot{Y} and $\ddot{\alpha}$ would equal zero. For this reason, a trial-and-error method of static balancing the airplane is used. This method is accomplished by successive changes of Y and α until the values of both \ddot{Y} and $\ddot{\alpha}$ are of such magnitude that they may be neglected. The values of Y and α so determined then become the boundary conditions.

Determination of Time Increment

As in any step-by-step integration problem, the choice of time increment is very important. After a preliminary investigation of the effect of varying the time increment, it was decided that a .01 second increment was necessary to maintain the desired accuracy. This time increment was determined by computing approximately one-third of a case at various time increments. When two different time increments produced approximately the same results, the larger of the two was selected as the time increment to be used in the final analysis.

The problem is afflicted with several discontinuities such as change in sign of coefficient of friction as the main gear changes from extending to compressing. The most severe of these discontinuities is occasioned by the bottoming of the tail bumper. When this happens, a change from the spring rate of the air pressure in the tail bumper to the relatively enormous spring rate of the fuselage itself is occasioned.

To minimize the effect of tail bumper bottoming, the time increment is reduced to one-tenth of its normal value during this critical time in this manner:

- (a) Extrapolate for Y and α . Compute $\overline{\overline{Y}}_{n+1}$.
- (b) Check sign of $(\overline{\overline{\overline{Y}}}_n K_{52})(\overline{\overline{\overline{Y}}}_{n+1} K_{52})$.
- (c) If positive, proceed with calculation.

(d) If negative, extrapolate again, using $.1\Delta t$ and continue from that point.

Determination of \overline{Y}

From Figure 1, it may be seen that the main gear extends by rotating on an arm offset from the strut centerline. Because of this condition, it is possible to express \overline{Y} as shown in equation 4. Because of the complexity of the expression, Newton's method of iteration was used to determine \overline{Y} rather than to attempt to express \overline{Y} explicitly in terms of Y and α . \overline{Y}_{n-1} is used as a first approximation to \overline{Y}_n .

Approximations of Empirical Functions

Four of the quantities involved in the equations are known only as empirical functions of other quantities which can be computed directly. They are given graphically. These functions are approximated by Gram-Tschebycheff polynomials of degree five or less. Computing the coefficients of one such polynomial requires about two minutes on the IBM Card-Programmed Electronic Calculator.

Some of these functions exhibit large change of slope. A device to make them easier to fit is to transform the independent variable, x, as follows:

$$z=\frac{x+a}{bx+c},$$

where a, b and c are chosen so as to make the function as nearly as possible linear in z.

No attempt was made to find optimum approximations as it was felt that time so expended would delay the job out of all proportion to the possible saving in CPC time.

Printing Form

The following results are printed in the arrangement as shown in Table I.

Sequence of Computation

The sequence of computation used in the solution of the problem is as follows:

- 1. Evaluate differential equation.
- 2. Print.
- 3. Extrapolate.





FIGURE 3. F3D-1 CATAPULT TAKEOFF ANALYSIS PATUXENT TEST RESULTS

Typical Results

The curve on Figure 3 shows typical test results obtained on the Douglas Model F3D-1 airplane and the predicted results obtained by the method outlined herein.

Computation

It required about two weeks to accumulate the detailed information necessary to write the equations and evaluate the constants. Several hours were spent in finding approximations to the empirical functions. Next, a sequence of computation was planned, storage units assigned, and the number of decimals to be carried in the various quantities established. This planning took about a day. Then the actual coding and checking required about two days. Several errors were passed, so a check against the hand calculation took most of another day. Finally, it was determined that three iterations were required for \overline{Y} , the time interval was established and computation started. The final program deck contains 311 cards and requires about seven hours to run.

It is perhaps of interest to note that tabulating plays several roles in this problem which are not obvious. The mass, center of gravity and moments of inertia are derived from weight control card records. The aerodynamic characteristics come from wind tunnel tests, the results of which are recorded directly on cards and reduced to final form by punched card computing.

OTHER PROBLEMS

- 1. Complex Eigenvalues of fourth-order matrices of complex elements. Since our machine is equipped with two storage units, we can store all the elements of such a matrix directly. The characteristic equation is then formed by expanding the determinant. The first two roots, usually those of smallest modulus, are then found by Newton's method, and the residual quadratic solved by formula. Usually these matrices are flutter matrices, and in this case velocity, damping and frequency are calculated from the roots. Presently, this process requires about fifteen minutes per matrix; however, we are working on a new scheme involving the direct solution of the fourth degree equation which should cut this time to five minutes. Also being coded is a scheme for the direct solution of fifth-order complex matrices.
- 2. Trajectories of various kinds. Some of these are:
 - a. Two-dimensional trajectories of long-range guided missiles, taking into account variation of atmospheric pressure and speed of sound with altitude, variation of lift and drag coefficients with Mach number and angle of attack, and turning rate as a function of any one of several variables.
 - b. Tactical trajectories, where the flight characteristics of the missile, a programmed course of evasive action by the target, and the characteristics of a computer are simulated.

- c. Beam rider trajectories, in which the missile carries a control unit, guiding it always toward the center of a radar beam.
- d. Short range trajectories, in which the effects of such random disturbing factors as thrust misalignment and winds are studied.
- e. Trajectories of an escape capsule exploded from an airplane traveling at high speed. This study is to insure that the aerodynamics of the capsule are such that it will not fall into the empennage of the airplane or gyrate so wildly as to injure the pilot.
- 3. A study of an air conditioning system for a small, highspeed airplane. This involved the solution of three simultaneous algebraic non-linear equations, with several hundred different sets of coefficients. The unknowns in these equations occurred with eleven different fractional exponents. These equations were solved by successive substitution, which required about fifteen minutes per case.
- 4. A study of the initiation of an airplane turn. This involved the solution of eleven simultaneous, non-linear differential equations. This problem was solved while only one storage unit was available—and required some juggling to keep within the storage limitations.
- 5. *Differential equations* describing the action of a hydraulic servomechanism, with valve overlap and friction taken into account. These were especially interesting because about fifty determinations of alternative courses of computation were made at each point.
- 6. *Polynomial equations* with complex coefficients of the eighth and lower degree are solved by Newton's method. Polynomials with real coefficients of the twelfth and lower degree are solved by a combination of Lin's and Bairstowe's methods. In this work it is often necessary to do some computing with the roots, or to determine the coefficients.
- 7. Other differential equations:
 - a. Maneuvering tail loads.
 - b. Landing gear spin-up.
 - c. Supercharger vane design.
 - d. Aircraft stability analysis.
- 8. Miscellaneous:
 - a. Calculation of transfer functions of dynamic systems.
 - b. Optical properties of radomes of one, two or three sandwiches.
 - c. Aircraft and engine performance calculations.
 - d. Wind tunnel data reduction.
 - e. Calculation of loft lines.
 - f. Studies of the propagation of sound.

- g. Heat transfer equations.
- h. Radar data reduction.
- i. Flight test data analysis.
- j. Studies of the aeroelastic properties of airfoils.

Methods

In my opinion it is seldom profitable to wire a CPC control panel for a particular problem. Almost all of our work is done with a few general-purpose panels. We usually find that if the CPC is not an economical machine to use for a particular problem with these simple control panels, then the problem is most likely a good application for the IBM Type 604 Electronic Calculating Punch or the IBM Type 602-A Calculating Punch. Of course, some of our thinking has been conditioned by the fact that we have had only one CPC and have, therefore, used it only for problems where it shows an enormous advantage over standard equipment.

Our 604 CPC control panels are very simple. The one we use most handles all numbers as four whole numbers and six decimals, and provides addition, subtraction, multiplication, division and square root.

To enable the CPC to use alternative procedures, we have two complete sets of instruction codes in the program card form. Field 1 is normally read, but on a special code the machine tests the sign of the result for that card, and if it is negative starts reading its instructions from field 2. A following test command will cause the machine to shift back to field 1, if negative, or stay in field 2 if positive. Another special code transfers control to field 1 unconditionally. We have found this device extremely useful.

Printing from any one of eight sets of type bars directly from channel C through the field selector is under the control of a special code field and can take place at any time.

Certain other special codes are used, such as absolute value and spread read-in. We have found that a second storage unit makes the machine much more useful than one storage unit. On the other hand, we have seldom felt the need for three storage units, and have not encountered problems absolutely requiring a third unit, excepting those of large matrices.

We use a 604 panel which checks CPC program decks for errors such as:

- 1. Calling a number from 941 storage sooner than two cards after it was sent to storage.
- 2. Calling a number from a 417 counter on the card after a number was sent to the counter.
- 3. Sending numbers to the same 941 storage unit on two successive cards.
- 4. Cards out of sequence.

This panel has proved to be valuable in reducing the time required to check out new programs.

DISCUSSION

Mr. Maginniss: Did you consider a differential analyzer such as those at MIT, the University of Pennsylvania, or General Electric for this problem? I would roughly estimate that fifteen or twenty minutes per parameter would be required.

Mr. Lowe: We tried several approaches to the problem: a graphical solution, building a model with rubber bands for the catapult, and two different differential analyzers. Being pressed for time, we solved the problem on our own CPC in seven hours. Perhaps we should also investigate the MIT machine.

Mr. Bell: When you change your time scale, does the machine or the operator make the change?

Mr. Lowe: The machine.

Mr. Bell: You mentioned checking coding. Mr. Lowe has developed a very clever setup for checking a program before it gets near the CPC.

Mr. Lowe: There is really not too much to say about it. I don't have a copy of the programming with me, but would be very happy to send a copy to anyone who is interested.

Mr. Thompson: We have a panel for the 402 that has approximately the same functions, and we would also be happy to supply information to anyone who is interested.

Mr. Patton: To make sure the operator has the cards in order, we do the following: We have several large decks in which only the loading cards are changed and onto which information from a master code deck is directly reproduced. Then, when the cards are collated and passed through the CPC, the serial number of both cards n and n+1 appears. We use the comparing brushes of the 402 to ensure that the cards are in sequence, comparing from third reading on card n and from second reading on card n+1. Then, if any cards are missing or out of order we can automatically stop the machine.

Mr. Lowe: We do the same thing. It is a very good point.

Mr. Lesser: We, too, do this, but that still brings up a problem where multiple decks are involved. Depending upon the type of problem, these decks are exposed in a particular order. In general, in order to keep them straight, one will want to start the sequence order each time with the multiple deck, because it is never known which one of the others is also included. Hence, there is still a problem, even with a sequence control, of making sure that the problem the machine is working is still the particular problem.

Mr. Lowe: Not that it is any panacea, but we also incorporate the deck number as well as the card number in our sequence control. This immediately raises the question: What happens when you interchange the sequence of decks? Usually we simply change the n+1 deck number in the last card of the deck.

Dr. Evans: When there are groups of instruction decks, we program the accounting machine in order to avoid operator trouble. The general idea is to digit-emit the operations from the digit-emitter and run them through the field selector so that the field selector acts as the various programs for the accounting machine. We usually run with instruction decks of four cards, and thereby can use a major program level on the accounting machine and cut out these instruction decks altogether. Lately we have done such things as pick up a special program device and sometimes have instruction decks of up to six cards replaced by digitemission by picking up the special program control on certain cards to bring in the extra programs.

Dr. Grosch: One might think of setting up restricted subprograms (this, by the way, is someone else's idea) where, instead of having access to a large number of storage units and counters, one would restrict oneself to just three or four of them and very few operations. In this way, the total size of the code required on a card could be decreased from perhaps eleven to five columns; then two instructions could be obtained from one level of the field selector. This would effect a 22-card sequence on a Model I.

Computation of Loan Amortization Schedules on the IBM Card-Programmed Electronic Calculator

CHARLES H. GUSHEE

Financial Publishing Company

×

IN THIS PAPER I am not concerned with any complicated problem of analysis, but with the contrasting problem of mass production of relatively simple results. In approaching such problems I bring to you the experience of twenty-five years as director of what I call a "number factory" as distinguished from the "computing laboratory" with which you are all familiar. In other words, I have been concerned with the production of a very large volume of mathematical results of a low order of difficulty (but a high order of accuracy). In this effort I have used just about all the commercially available computing equipment, up to and including the IBM Card-Programmed Electronic Calculator.

My subject is the problem of schedules for the amortization of loans. The equal payment necessary to amortize a loan of 1 is represented by



where i is the interest rate per period and n is the number of periods. Tables of the function have long been published. However, in the decades of the 1930's and 1940's this method of payment was widely popularized for home loans by the Federal Housing Administration and by the Veterans Administration. This resulted in a widespread demand for complete schedules showing the allocation of each payment, and the unpaid balance at the end of each period.

The payment can alternatively be expressed as

$$i + \frac{i}{(1+i)^n - 1}$$

where i is the interest on 1 or the amount of the loan, and the second item is the first term of a geometric progression in the ratio (1+i) whose sum is unity. This fact was taken advantage of by the FHA whose amortization schedules showed the successive payments of principal computed as a geometric progression.

However, in my own approach I felt it important to duplicate the procedure of the typical lending institution. Interest is calculated on the amount of the loan. This interest is subtracted from the payment. The difference is the amortization, to be subtracted from the old balance of loan to produce the new balance. This sequence of simple operations is repeated until the loan is fully paid.

Perhaps an historical review of our methods will be of interest. In the early 1930's such a schedule would have been prepared by two computers working together, one with a rotary calculator and the other with a two-crossfooter adding machine. One computer read the interest from the rotary calculator and the other subtracted to obtain amortization and new balance. The results were then typewritten. A great advance occurred in 1939 when we received our first multi-register bookkeeping machine with motor repeat lugs. Thereafter a single computer obtained the amortization (as payment minus interest) on a rotary calculator and entered it in both the bookkeeping machine and the calculator. The bookkeeping machine completed the line automatically, and the rotary calculator contained the amortization figure for the next line. The speed of this operation is remarkable, the mechanical operation requiring only five seconds per line, plus the time for entering the amortization on the keyboard. We still operate six units of this equipment, primarily for loans with frequent changes of payment or interest rate or other irregularities where the control panel wiring time would be excessive for the unique problem.

In fact, in some problems we now have a single computer operating two rotary calculators with the left hand and entering results in a bookkeeping machine with the right hand. In textile mills the vernacular would call this the "stretch-out," and it is admittedly not a happy condition for the human computer.

We first came to punched card computing with the IBM Type 602 Calculating Punch. This gave us the advantage of punched-card computing and minimized the human element, but the cost was not significantly lower than previous methods, the 602 requiring four or five seconds per line, with later tabulation on an IBM Type 405 Accounting Machine. Our 602 has been supplanted by an IBM Type 602-A Calculating Punch which still is turning out schedules, but especially for large loans, up to balances of 99 millions or 10 digits, multiplied by an interest rate to 12 digits.

The next advance was the IBM Type 604 Electronic Calculating Punch. This produced 100 lines per minute, as compared with 12 or 15 by previous methods, but our card cost was large. With a theoretical capacity of 48,000 cards in an eight-hour day, and with each card only used twice, we were using perhaps six or seven million cards a year at a cost of around \$7,000 to feed this one machine alone.

Finally the CPC arrived, and immediately took over the job virtually without the use of cards. In the simplest application, a speed of 150 lines per minute is obtained. (By this time you may have noticed my preoccupation with speed, the typical outlook of the factory manager.) All the data are retained in electronic storage in the 604 and read out, not via channels, but by direct outlet wires to the type bars. The IBM Type 941 Auxiliary Storage Unit is used only for storage of identifying data, such as rate, term, etc., for use in printing page headings. Incidentally, on this form we were able to print two schedules side by side within the 89 type bar limit, thus obtaining two original copies instead of one original and one carbon copy. Clarity is improved and form cost reduced.

The versatility of the CPC on the problem is illustrated by one of those trivial conditions which stimulate ingenuity. Our standard amortization schedule (an example is shown in Figure 1) contained two sets of columns on a single page, 60 lines for each column. In other words, line 61 appeared on the same line as and to the right of line number 1. It is the solution to this peculiar, and perhaps trivial, problem which I will give in greater detail as illustrating the versatility of the CPC.

A single card is key punched with the data for the loan, including the amount (8 digits), payment (6 digits), interest factor (8 digits); and for use in printing headings only, the years, months and annual interest rate.

To explain the "interest factor," in our early control panels: for a 4% loan with monthly payments we multiplied by the annual rate of .04 and divided by 12 (via digit emission). Quarterly loans were handled by setup change. However, the method broke down for loans with weekly payments, fortnightly, bimonthly, etc.; and especially for loans with odd decimal rates. Therefore, we now key punch the interest on "1" for the period at which payments are to be made. That is, for 4% and monthly payments we key punch .0033333334. (Note the final digit to assure uprounding of critical half-cents of interest.)

Our detailed procedure is as follows: The first card resets all the counters in the IBM Type 417 Accounting Machine. The second card loads by spread entry. The next five cards transfer the five elements on the loading card to the 941 for subsequent use. The next is a cycle card necessarily preceding the next four cards which print the heading information for the 941 on channel B. Printing is facilitated by wiring channel B exits to the common of the field selector, thus avoiding much split wiring.

Since our special problem is to print line 1 and line 61 on the same line, our next card throws us into program repeat on the 604, stopped by negative item count against a "59," set up by our loading card in the seventh and eighth positions of factor storage 1 and 2 containing the payment. In about seven seconds the 604 completes 60 steps and contains the 60th balance. It is now prepared both to compute line 1 and line 61 and so proceed alternately left and right down the page.

This is accomplished by retaining the earlier balance in general storage 1 and 2 and the later balance in general storage 3 and 4, from which we read out alternately by split wire to channel C shift entry.

Lacking electronic storage capacity we read the interest factor on every line from the 941 into factor storage 3 and 4 by channel A controlled by the special program. However, on the program repeat operation, we could not do this and maintain electronic speed. Therefore, on program repeat we do not alternate but use general storage 3 and 4 only, transferring the interest factor back and forth from factor storage 3 and 4 to general storage 1 and 2.

Thus, we produce a full page of an amortization schedule at a speed of 150 periods per minute, or 75 double lines per minute, on special program using only one card keypunched with data and 16 control cards.

The end of the page is signalled by a negative balance condition in counter group 1 of the 417. At spread entry a "119" has been entered and "1" subtracted on each special program step.

The control cards for the second and subsequent pages are essentially the same as for page one, but omitting spread entry of new data, and inserting a special 604 program to rearrange data in the 604 (e.g., balance 60 being no longer needed, and balance 120 being shifted to balance 60 position).

The ending of the schedule, upon completion of the number of payments necessary to pay the loan in full, is the most intricate problem. Observe that the schedule may end at the bottom of the left-hand column, or at the bottom of the right-hand column, or at some intermediate point in either. If the schedule ends in the middle of the left-hand column we must not print the right side; but if it ends in the middle of the right-hand column, then the left-hand column must continue to be printed for the entire page.

We can readily determine whether the schedule ends in the left-hand column. If the negative balance selector is picked up during the program repeat operation, the schedule must end in the left-hand column. By using the negative balance selector to pick up a pilot selector which is held transferred, we have a continuing means of control.
Schedi	ale of Direct Re	duction Loan					
Preba	RATE % 00.00	PAYMENT \$ 112.77 I Publishing Compared	LOAN \$ 8,250.00		TERM: YE	ARS MONTH	S PERIODS .8 4
PAYMENT	PAYM	ENT ON	BALANCE	PAYN	IENT	PAYMENT ON	BALANCE
NUMBER	INTEREST	PRINCIPAL	OF LOAN		BER INTE	REST PRINCIP	AL OF LOAN
2	27.22	85.55	8,079.18	62	8.31	104.46	2388.10
3	26.93	85.84 8613	7,993.34	63 64	7.86	104.81 105.16	228329 217813
5	26.36	86.41	7.820.80	65	726	105.51	2072.62
6	26.07 25.78	86.70 86.99	7673410	66 67	6.81 6.56	105.86 106.21	1,966.76
8	25.49	8728	7,559.83	68	6.20	106.57	1753.98
10	2 4.91	87.86	7,384.40	70	5.65	107.28	1539.78
11	24.61	88.16	7,296,24	71	5.13	107.64	143214 132414
13	2 4.0 3	88.74	7,119.05	73	4.41	108.36	1.215.78
14	23.73	89.04 89.34	7.030.01 6,940.67	74	4.05 3.69	108.72	1107.06 997.98
16	23.14	89.63	6,851.04	76	333	109.44	888.54
18	22.54	90.23	6,670.88	78	2.60	110.17	668.56
19	2224	90.53	6,580.35 6,489.51	79 80	2.23	110.54 110.91	558.02
21	21.63	9114	6,398.37	81	1.49	111.28	335.83
22 23	21.33 21.02	91.44 91.75	6,306.93 6,215.18	82 83	1.12	111.65 112.02	22418 11216
24	20.72	92.05	6,123.13	84	.37	11216	112.53*
26	20.10	92.67	5,938.10				
27 28	19.79 19.48	92.98 93.29	584512				
29	1917	93.60	5,658.23				
30 31	18.86	93.91 94.22	5,564 <i>52</i> 5,470 1 0				
32	18.23	94.54	5,375.56				
34	17.60	95.17	5,185.54				
35 36	17.29 16.97	95.48 95.80	5,090.06 4,994.26				
37	16.65	9612	489814				
39	16.01	96.44 96.76	4,801.70 4,704.94				
40 41	15.68 1536	97.09	4,607.85				
42	1 5.0 3	97.74	4,412.70				
43 44	14.71 14.38	98.06 98.39	4,314.64				
45	14.05	98.72	4,117.53				
46	1 3.39	99.04	4,018.49 3,9 1 9 1 1				
48	13.06	99.71 10004	3,819.40 371936				
50	12.40	100.37	3,618.99				
51 52	12.06 11.73	100.71 101.04	3,5 1 8,28 3,417,24				
53	11.39	101.38	3,315.86				
55	10.71	102.06	3,112.08				
56 57	10.37 10.03	102.40	3,009.68 2,906.94				
58	9.69	103.08	2,803.86				
59 60	9.35 9.00	103.77	2,596.67				
The final a	aument is news	llu somewhat diffare	nt from the resular				
payment, a	nd is shown eith	her starred on the las	line, or on the follow	ving line.			
з в б, '	735.61			1	222.44	8,250.00	8,250.00
Su Bal	n of ances			:	Sum of Interest	Sum Principal	of Payments

FIGURE 1

If the schedule ends on the left, we suppress both control impulses and output on program levels 1 and 3 which are used for the right-hand side, and thus suppress all printing on the right.

If the schedule ends on the bottom line of the left, it stops automatically by negative item count. Otherwise it stops from the negative balance selector picked up on special program impulsed to minor program stop.

If the schedule ends on the right on the bottom line, no special wiring is necessary, as we shall have a full page like any other. If it ends earlier, the negative balance selector will pick up and suppress program levels 1 and 3, thus preventing printing on the right side but not stopping the lefthand side until the bottom line has been completed.

One important step is a proof for accuracy. The interest, amortization and balances are all accumulated in counters of the 417, and final totals are printed manually at completion of each schedule. Obviously, the total of the amortization column must equal the original amount of the loan. Since each item of the interest column is the interest on a single balance, the total of the interest column is the interest on the total of the balance column, with a minor variation which is due to possible random rounding up or down of fractions of a cent in each interest item. The proof at present is done on a rotary calculator by an inspector.

We use continuous forms, with a tape-controlled carriage for heading, first body line, advance to proof line at the end of each schedule or advance to subsequent page. The hopper will hold sufficient cards for continuous automatic operation for about forty-five minutes. Thus, the CPC constitutes an automatic production unit running continuously all day with no human intervention other than loading new instruction cards about every half-hour.

I believe the details of this unique commercial application of the CPC indicate the applicability of this type of equipment to the problem of mass computation of large volumes of relatively simple work. The CPC every day turns out several hundred amortization schedules at a lower unit cost than ever before achieved.

Perhaps a few general remarks on commercial tablemaking would be useful. We like to use progressive calculations where each element is computed from its predecessor. If the final result is correct, it is an excellent proof of the intermediate results. Where multiple solutions of a function are needed, as in table-making, it is often possible to develop methods for progressive calculation well adapted to punched card computing.

In commercial work there is usually no demand for an optimum interval table. The desired interval is the commercial unit involved. Thus, if the unit of price for trading bonds on the New York Stock Exchange is $\frac{1}{8}$ point, the trader wishes a table showing the price argument in $\frac{1}{8}$ point, regardless of the number of orders of differences in various parts of such a table.

However, the "critical" table is growing in acceptance. In many cases for limited applications a critical table can be made in a small fraction of the bulk of the usual table.

The critical table also can be used on a large scale instead of methods of iteration or subtabulation. As an example, we are now making a critical table of mortgage yields at 502 rates from 2.495% to 7.505% at intervals of .01%from which a table of yields to the nearest .01% can be read directly by inspection. The actual 604 time may be two or three days, which is very much less than by using more refined methods perhaps less suited to machine calculation.

Iteration with us is still a neglected field. Prior to electronic computation we could always get results cheaper some other way. In table-making there is frequently part of the area where convergence diminishes, and a uniform procedure may not be appropriate.

In the old days, we would even prefer, where possible, to form the inverse table and use Lagrangian interpolation itself iteratively to approach a result. Iteration is on our program for future study.

Subtabulation likewise is a field not exactly neglected but still not fully exploited. In the simpler table it works well. For tables with higher order of differences, involving "throwbacks" or similar devices, we have not operated efficiently with bookkeeping machines, but we have not tried the CPC.

To summarize, I have reported to you the manner in which the CPC is used to compute a large volume of relatively simple work. I expect that the CPC will alter greatly our approach to the computation of financial tables, as well as make the production of many new tables commercially possible.

DISCUSSION

Mr. Ferber: On the first line of your loan amortization schedule, do you know in advance that the period is going to be equal to 84? You must look that up.

Mr. Gushee: We usually compute this in advance as a check on the machine. In spite of this completely automatic operation, we find it desirable to have a human inspector look at things. One of the tests of error would be the failure to come out in a predetermined term, and so we usually predetermine the term and include that in our data; our inspector will make sure that the term will be that which is given. This will help catch egregious errors, such as the wrong interest rate punched in the master card.

Mr. Ferber: To be sure that you obtained the even penny, do you adjust the interest rate in advance? Can you carry more places, for example, than the nearest cent in your calculation?

Mr. Gushee: Oh, yes; the interest factor is punched in the card to ten decimals; in this case, it is .0033333334. We adjust by rounding up or down in all critical offsets, round-

ing up in accordance with interest practice. Our interest is always perfect to the nearest cent.

Mr. Ferber: Suppose your interest rate were divided by 12 and were an even amount. Would you adjust that at all when you start multiplying? I notice you put a 4 at the end of the 3333...'s.

Mr. Gushee: That is to ensure that the interest will be rounded by a 5 when the interest is an exact half-cent; commercial practice is to round half-cents up. Banks will usually round up any half-cent that is even.

Dr. Brown: Do you not round off at each stage of the game in these computations?

Mr. Gushee: Yes, we do.

Dr. Brown: Then I would merely like to point out as a matter of interest from a strictly mathematical point of view your table may be in error all the way down the line.

Mr. Gushee: That is right. It would be much more desirable to compute some other element to a great number of decimal places and round off. Admittedly this is a commercial application. The original FHA method was theoretically far superior to ours.

Dr. Herget: If you just multiply the monthly payment by the number of payments and subtract from the totals at the bottom, there is a difference of \$.24.

Dr. Hurd: He has to calculate this and have it pay on the basis of a month. So he has no opportunity to avoid the bias which Dr. Brown mentioned.

Mr. Gushee: This 112.77 is necessarily an inaccurate figure. It is rounded up in all cases, so as to be sufficient. The theoretical figure might be 112.76629.

Dr. Herget: But is it not what the customer pays?

Mr. Gushee: No. The final payment is different from the others and is tabulated separately at the end.

Dr. Herget: But the sum of all the interest and the sum of the principal should equal the sum of the payments that were made.

Mr. Gushee: Yes, and they do, because instead of multiplying the payment by 84, it should be multiplied by 83 and the odd final payment of \$112.53 added to it.

Mr. Bell: I would like to ask a question. The question is whether you have considered internal programming on the 417 with only 16 program levels. I do not think there would be much difficulty in eliminating the insertion of cards.

Mr. Gushee: We have not considered it.

Dr. Petrie: I have one suggestion I might offer for consideration: Instead of punching the rate in decimal form, punch it in fractional form and have the machine actually make the change. As the cards are processed, the interest rate is calculated, the amount deducted from the payment, the deduction from the principal, and the new balance of the loan punched on the card. In this way, it can always be kept current.

Techniques for Handling Graphical Data on the IBM Card-Programmed Electronic Calculator

WILLIAM D. BELL

Telecomputing Corporation

X

MANY IMPORTANT forms of engineering computations utilize data in the form of graphs. Often this data is empirical in nature and not amenable to analytic representation. Such calculations have usually been considered impractical for punched card computing techniques.

The field of aviation presents many excellent examples of problems of this type. Particular examples are jet engine performance and aerodynamic performance for power available and power required. Jet engine analysis methods for a single data point—that is, a given altitude, velocity, and power setting—require relating the data from six to fifteen graphs and involve one-half to one hour per point for manual solution. We have evolved techniques for the IBM Card-Programmed Electronic Calculator which efficiently cope with this type of problem, and require two to three minutes in which to complete a solution.

Setting up such problems on the Card-Programmed Calculator involves representing the graph by a mesh of points. Since typical graphs are either two- or three-dimensional, the associated mesh of points may also be two- or threedimensional. These data must be placed upon cards which are introduced into the program deck. The calculator should then be capable of locating the closest point in the mesh and calculating the corresponding value of the function by interpolation.

Experience gained over a period of time using a number of different procedures and methods has indicated that necessary conditions for a useful method of setting up and handling graphical data should include the following requirements:

1. It should not be necessary to compute a fictitious argument for the purpose of table look-up. It is possible to transform the arguments to a new set of variables consisting of a series of sequential integers, n, n + 1, $n + 2, \ldots, m$. The comparing unit of the IBM Type 417 Accounting Machine can then locate a desired mesh point. The disadvantages of the method include the additional computing time and, more important, the restriction placed on the location of the mesh points when reading the initial graphical data.

- 2. The interval chosen for the argument should not be restricted; in particular, a constant interval in the argument should not be necessary.
- 3. The general procedure should be set up for threedimensional graphs, treating two-dimensional data as a simplified case. The interpolation procedure should be the same regardless of the number of variables and sensibly independent of the size of the numbers in the various arguments and functions.
- 4. There should be no necessity of an artificial phasing so that arguments in one card pertain to data in following cards. Such methods can simplify the machine setup but complicate setting up the data.
- 5. A minimum number of points for a required accuracy should be used to establish a mesh. Since these data are included in the program deck and must go through the CPC once for each data point, an excess number of data points can greatly lengthen the time required for calculation.

The CPC method described below meets the above requirements. It has three logical parts—reading the graphical data to establish mesh points, preliminary calculations to set up interpolation data, and the basic CPC operation. Only one restriction is inherent in the method, and that is a requirement that all arguments be positive. No actual limitation is involved, since a negative argument can always be translated by adding a simple constant.

Setting Up Graph Data

Two-dimensional graphs are usually presented as z = f(x). More complicated three-dimensional relationships, z = f(x, y), are plotted as families of curves on a two-dimensional graph. The interpolation process utilizes a second degree equation for the variation in x (the abscissa on the graph) and a linear interpolation for the y variation (the family variation). Since the intervals in the argument are not constant, divided differences are necessary for the interpolation formula. The usual forward interpolation process requires knowledge of the n + 1 point as well as the *n*th. By using a backward interpolation formula, the machine can identify a point as it goes by and then interpolate back into the correct region. To fix these ideas and clarify the mathematics, the formulas involved are shown below.

Considering the general case of z = f(x, y) for a constant value of y, we can define divided differences as:

where

$$[x_0 \quad x_1] = \frac{z_0 - z_1}{x_0 - x_1} , \qquad (2)$$

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} = \frac{z_1 - z_2}{x_1 - x_2} , \qquad (3)$$

$$\begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix} = \frac{\begin{bmatrix} x_0 & x_1 \end{bmatrix} - \begin{bmatrix} x_1 & x_2 \end{bmatrix}}{x_0 - x_2} .$$
(4)

Similarly, for constant x,

where

$$\begin{bmatrix} y_0 & y_1 \end{bmatrix} = \frac{z_0 - z_1}{y_0 - y_1} . \tag{6}$$

Combining 2, 3, 4 and 6 in the form of Newton's backward interpolation formula, we have

$$\begin{aligned}
s(x, y) &= s_2 + [x_1 x_2] (x - x_2) & (7) \\
&+ [x_0 x_1 x_2] (x - x_2) (x - x_1) + [y_1 y_2] (y - y_2), \\
\text{with} & x_1 < x \leq x_2 \\
& y_1 < y \leq y_2.
\end{aligned}$$

When reading the graph data, it is necessary to select points for intervals over which the interpolation formula 7 will produce adequate accuracy. It is also necessary to hold one variable constant for reading data at varying intervals of the second argument. Both the arguments and function are written so that the numbers will be as close to the region of 1 to 10 as possible with an associated power of 10. By this means, the size of the divided difference is also fairly well controlled.

Some experience is necessary in learning to evaluate the intervals over which interpolation will hold. The principles are simple, and with a little practice the necessary points can be quickly read and listed. Typical two-dimensional graphs are represented by 10 to 15 points or even as few as two for linear functions. For three-dimensional graphs, 50 or 60 points may be required.

For small jobs, we read and record the points manually. For larger jobs which may involve a large number of points, we utilize a Telereader-Telecorder of our own manufacture to read and record data points rapidly and directly into IBM cards. This machine is a projection reader utilizing an electronic counter which makes possible a high degree of accuracy at a considerable saving in time and effort.

Preliminary Calculations for Interpolation Data

One IBM card is required for each data point. These are either key punched from the manual list or prepared automatically by the Telecorder. Each graph is assigned a different identification number. By sorting the cards to the proper sequence, it is possible to compute the first and second divided differences in x. Then the cards are rearranged, and the first divided difference in y is computed. The IBM Type 604 Electronic Calculating Punch is used for these operations. Each final card has seven necessary pieces of information, which are:

x_1		
x_2		
y_2		
z_2		
$[x_0$	$\dot{x}_1]$	
$[x_0$	x_1	$x_2]$
[y0	y1]	•

Card-Programmed Calculator Operations

The program card deck used in the CPC consists of the necessary intermediate computing steps with the graphical data interspersed through the file. It would appear obvious, knowing a value of x and y, to recognize one data card from a group of cards representing a particular graph, and by control of the spread read-in hub on the 417 control panel, cause the desired interpolation data to add into the mechanical counters. However, the circuit of the spread read-in function unfortunately precludes this method of operation. If the spread read-in impulse is absent but the sign X of a field is present, the associated 417 counter group will be cancelled to zero. This circuit deficiency could be corrected by elementary changes within the CPC and make possible much simpler three-panel control wiring than is now required for the system described here.

The 604 control panel contains these operations:

Code 1	Add
Code 2	Subtract
Code 3	Multiply
Code 4	Divide
Code 5	\ldots Store x and y arguments
Code 6	Table look-up
Code 8	Square root

Codes 5 and 6 require explanation. When a value of x and y have been computed, and are to be entered into a table look-up operation, they are routed over channel A and B, and operational code 5 transfers them to two eight-place

general storage units for storage. A code 6 table look-up operation is then called out on all of the data cards, representing the mesh points which define the graphical data. Values of x_2 and y_2 from each mesh point are read from the second reading brushes on the 417 on channel A and B into the electronic computer. Differences are calculated between the stored x and the stored y and the data point (x_2, y_2) . For a particular table look-up, the first card for which the two comparisons indicate that the table data are greater than or equal to the arguments, a test energizes the negative test relay, which, in conjunction with 417 wiring, makes all counters of the 417 operative on the following card cycle. Thus, as the same card reaches the third reading brushes, if it is the desired point, all interpolation data are set up within the machine. Only the first table card which satisfies the test conditions is permitted to add into the counters, since, obviously, all following cards will also satisfy the conditions.

Four program cycles are then necessary to complete the interpolation calculation, using the formula of equation 7. As particular arguments are computed, they are set up with the power of 10 required by the table data. The decimal position of the interpolation formula is, therefore, constant for every look-up operation.

The method which has been described here for handling graphical data for engineering calculations has proved simple to apply and has had important results in application.

DISCUSSION

Mr. Maginniss: I should like to ask two questions: First, about how many cards do you need for an average table? Second, can you always arrange your variables in such a way that linear interpolation is good in one direction, while quadratic interpolation is needed in the other direction?

Mr. Bell: Answering the second question, when linear interpolation does not hold, more points are read to get the grid closer together.

As to the number of points required, we have very rarely ever had to use more than 60 or 70 points for three-dimensional cases.

Mr. Koch: We handle graphical data on the 604 by writing equations for all the data and using sums of products of polynomials of second order. We find that we can get about one per cent accuracy on the graphs, which is certainly good enough, because, as Mr. Bell pointed out, the graphs themselves are only accurate to about five per cent.

At times, particularly at the points where a graph starts out with a sharp curvature and ends up a straight line, we have arbitrarily to break the graph in two parts and write an equation for each part, controlling our setup to pick out the correct set of equations. It required approximately 100 hours to write the equations for about 30 graphs. We are now programming the setup for the CPC and hope that we can write equations for three-dimensional graphs of this nature in one to two minutes on the CPC. If this can be done, it will mean an hour on the CPC as compared to 100 hours on the 604.

Dr. Herget: Considering the number of cards which must be put in the table and the number of operations needed for the interpolation, what is the possibility that you could write a quadratic expansion in two dimensions, have fewer cards in the table and still get the same accuracy? In other words, a Taylor series expansion of about six points would cover what you cover with sixty points.

Mr. Bell: If you try to do it analytically, you may run into trouble. There will be special problems where it will be better to use other means.

I agree with Mr. Koch that you should use a 604 for small programs, but the CPC will be faster for 100 cases because of control panel wiring for machine setup.

Mr. Porter: If you have a problem where data curves initially are not going to change, which is very rarely, then the analytical approach offers advantages, unless you have to discriminate at too many breaks. Here your card deck would run up to quite an excessive amount per point on any kind of large-scale problem.

Mr. Lesser: Most of the problems that we have tackled in aerodynamics for the machine have been those for which it was fairly easy to find analytical approximations. In other words, you could look at the curve and practically tell whether or not it will fit your backlog of analytical approximations. However, in the case of jet-engine power data where most of the curves have a bad hook down at the left-hand corner, it is very difficult to approximate this part of the curve analytically; and it is this area of the curve that is the most important to us.

Mr. Horner: We have done a good many of the problems that have been described, and the best approach to me seems to mix the detail cards in with the table points and calculate the result on the CPC.

Where iteration is to be run over and over again, analytical representation is useful, because the number of detail cards would be decreased.

Mr. Bell: These methods can be combined to give one that is sometimes quite superior. Suppose that one has set up his program to do problems of the order of 100 pieces of data. Then he is given 1400 to do. He can set up the 1400 data points and expand them by the number of table lookups. He would then sort this deck of cards into his table look-up cards and gang punch the data; this is done as an independent operation. When these cards are put into the program and are run, the card volume will have been decreased to where it is now fractions of a minute instead of two or three minutes. This employs the same procedure, the same methods.

Dr. Brinkley: The thermodynamic properties of a working fluid are conventionally represented by a Mollier chart, in which the enthalpy, H, is plotted as a function of the entropy, S, for specified values of the temperature, T, and pressure, p.



The punched card analogue consists of a table containing H(p, T), S(p, T) as well as certain other functions of the state variables.

Power cycles are analyzed by a routine consisting of successive inverse interpolations in the table for given values of H and p or of S and p. The tables are in such form that linear orthographic interpolations are unavailable.

At our laboratory, we routinely use a standard card-program, consisting of about 50 cards, with general-purpose panels to compute Lagrangian five-point interpolation coefficients. This deck is followed by a standard program to carry out the interpolation of each of the functions tabulated on the card, six in number, to the required value of the specified variables. This program also contains about 50 cards. The proper table cards for a given interpolation are independently selected by a standard collator routine.

This procedure has proved to be more efficient than methods based on special control panels for the problem at hand.

We believe strongly in the general utility of interpolative formulas of Lagrangian form as opposed to formulas involving differences. Such formulas are generally most economical as regards program length and storage requirements.

Calculation of the Flow Properties in an Arbitrary Two-Dimensional Cascade

JOHN T. HORNER

Allison Division, General Motors Corporation

×

THE GENERAL PROBLEM of cascade flow may be stated as follows: Given the initial conditions of flow entering a cascade and the cascade geometry, determine the properties (velocity, pressure and density) of the flow field at every point. From this information, the circulation around a given blade and the blade lift may be calculated. The possibility of flow separation from the boundary layer may also be detected by comparison of the results with parameters derived from experimental data.

The inverse cascade problem determines the blade geometry when the entrance conditions are given and the variation of some property of the flow field (velocity, pressure, or density) over the blade surface is postulated. This problem may be solved in much the same way as the direct problem. The two calculations enable the designer to (1)derive families of blade shapes based on established parameters and (2) find the useful operating range of blades within those families.

DEVELOPMENT OF FLOW EQUATIONS

Consider an arbitrary blade cascade as shown in Figure 1 where the flow pattern has already been constructed. The lines parallel to the blade surfaces represent streamlines; those normal to the streamlines are velocity potential lines. The streamline spacing is determined by satisfying continuity of flow across a potential line and then dividing the total flow into a number of equal portions which pass between the streamlines. The spacing between the velocity potential lines may also be arbitrarily selected. Certain restrictions should be imposed here, however, as will be shown later. The present method of calculation determines the location of mesh points, using the selected mesh spacing, and finds the velocity at every point of the mesh. From this information, as was previously mentioned, the blade circulation, lift and other physical properties may be immediately calculated.

To facilitate the study, several assumptions are made regarding the behavior of the fluid throughout the selected region. It is assumed that at some point sufficiently far removed from the entrance to the cascade, the flow is moving with constant velocity in a given direction with respect to the cascade and that the fluid obeys the adiabatic law for a perfect gas. These conditions are sufficient to ensure the existence of a velocity potential function¹ which may be defined by:

$$\frac{\partial \phi}{\partial s} = V$$
$$\frac{\partial \phi}{\partial n} = 0$$

Steady state flow conditions are also assumed, or

$$\frac{\partial V}{\partial t} = 0 \ .$$

In addition, a laminar boundary layer is assumed to cover the entire surface of the blade so that flow separation does



FIGURE 1. BLADE CASCADE

not occur. The nomenclature used in the following study is defined in Table I.

The flow equations are developed in terms of the flow field coordinates s, n. Consider a fluid element moving between two streamlines as shown in Figure 2. Since a velocity potential exists, the circulation around this fluid element must be zero, or

$$\Gamma = \oint V \, d\sigma = 0 \ . \tag{1}$$

This equation may be applied whenever σ is any path in a simply-connected region of the flow field. Summing contributions to the circulation around the fluid element alone,

$$\left(V + \frac{\partial V}{\partial n} \Delta n\right) \left(\Delta s + \frac{\partial (\Delta s)}{\partial n} \Delta n\right) - V \Delta s = 0$$

or, upon dropping second order terms,

$$\frac{\partial V}{\partial n} = -V \frac{1}{\Delta s} \frac{\partial (\Delta s)}{\partial n} = +VC , \qquad (2)$$

where C is the curvature of the streamline. Equation 2 may be integrated immediately to yield

$$V_2 = V_1 \exp \int_{n_1}^{n_2} C \, dn \, . \tag{3}$$

TABLE I

NOMENCLATURE

- x, y... Cascade coordinates, inches
- s, n... Flow coordinates, inches
- V....Velocity, feet per second
- T....Temperature, °R.
- ρDensity, pounds per cubic foot
- P....Pressure, pounds per square foot
- W....Weight flow, pounds per unit length per second
- C...Curvature, inches⁻¹
- γRatio of specific heats
- ϕVelocity potential, feet² per second
- θRatio of total temperature to standard temperature, $T_{\text{stan}} = 518.4^{\circ}\text{R}$.
- δ....Ratio of total pressure to standard pressure, $P_{\text{stan}} = 2115.9 \text{ psf.}$
- ΓCirculation, feet-inches per second
- t....Subscript indicating stagnation temperature
- c_pSpecific heat of air at constant pressure, BTU per pound per °R.
- R...Gas constant, feet per °R.
- J....Conversion factor, 778.16 foot-pounds = 1 BTU
- g...Gravitational constant, feet² per second²



FIGURE 2. FLOW ELEMENT

Equation 3 relates the velocities to the mesh geometry at all points along a velocity potential line. Physically it means that fluid elements may not cross the streamlines. The fluid element moving between the streamlines may be accelerated in the s direction only by a difference in pressure acting on the two ends of the element. Equating the differences in pressure forces, to the product of the mass and acceleration, the equation of motion for the fluid element is obtained. This is

$$\frac{dV}{dt} = \frac{-g}{\rho} \frac{\partial p}{\partial s} \quad . \tag{4}$$

Upon applying the condition that $\partial V/\partial t = 0$, this becomes

$$V\frac{dv}{ds} = \frac{-g}{\rho}\frac{d\rho}{ds} \ . \tag{5}$$

For adiabatic expansion or compression, the pressure and density are related as follows:

$$\frac{P}{P_t} = \left(\frac{\rho}{\rho_t}\right)^{\gamma}.$$
(6)

Substituting from equation 6 into 5 and integrating along a streamline, Bernouilli's equation is obtained which may be written as

$$\left(\frac{\rho}{\rho_t}\right) = \left(1 - \frac{V^2}{2gJc_pT_t}\right)^{\frac{1}{\gamma-1}}$$
(7)

in which the equation of state

$$\frac{P}{\rho} = RT \tag{8}$$

is used. The weight of the fluid per unit blade length per unit of time passing across a velocity potential line is given by

$$dW = \rho V \, dn \ . \tag{9}$$

Introducing now a new variable, $V/\sqrt{\theta}$, the weight flow across a velocity potential line becomes

$$\frac{W\sqrt{\theta}}{\delta} = \frac{P_{\text{stan}}}{RT_{\text{stan}}} \int \left[\left(1 - \frac{(V/\sqrt{\theta})^2}{2gJc_pT_{\text{stan}}} \right)^{\frac{1}{\gamma-1}} V/\sqrt{\theta} \right] dn$$
$$= \int f(V/\sqrt{\theta}, T_t) dn . \qquad (10)$$

All other equations may now be redefined in terms of $V/\sqrt{\theta}$, merely by substitution of this quantity in place of V.

The calculating procedure can now be readily understood. In order to obtain the cascade geometry, a large scale plot must be drawn showing two or three blades in a cascade. Sketch velocity potential lines and streamlines upon this plot. Choose a line of zero velocity potential far enough in front of the cascade that the velocity remains constant across every part of this line. Then integrate along each potential line using equations 3 and 10 and iterating until $W\sqrt{\theta}/\delta$ agrees with the given value. After this is complete, interpolate, shifting coordinates to get equal weight flow between each pair of streamlines. Then, integrate along the streamlines from the zero velocity potential line to get a value of ϕ for each potential line where

$$\phi = \int \frac{V}{\sqrt{\theta}} \, ds \, . \tag{11}$$

Another interpolation is now performed, where the coordinates are shifted to get lines of equal velocity potential. The entire procedure is repeated until the mesh points cease to shift. The third trial ought to yield values sufficiently close for engineering purposes.

The procedure outlined above determines the flow mesh, assuming fixed gas boundaries entering and leaving the cascade. This procedure must be modified in order to locate the front and rear stagnation points on the blade and to find the actual outflow pattern of the gas. One method of relaxing the boundaries is to carry out the weight flow integration along the velocity potential line across two spaces at the cascade entrance and exit, adjusting the upper and lower boundary to correspond with the middle. This is the method used in the present procedure. In carrying this out, it is convenient to determine the velocity potential spacing in the following manner. First find the velocity potential from the zero potential line to the leading edge of the middle blade in the cascade. Then divide this value of the velocity potential by the number of assumed spaces used in finding it. This is the value used to obtain the velocity potential spacing, and it ensures that the mesh pattern is the same at the leading edge of all three blades when the final mesh is determined.

Modification of Equations for Machine Calculations

The actual calculation uses cascade coordinates x, y instead of the flow coordinates. In carrying out the integration for ϕ , the curvature C and the absolute value of the secant of α , the angle between the x axis and a tangent to the streamline, are obtained. These equations are:

$$C = \left(\frac{\frac{d^2 y}{dx^2}}{\sqrt{1 + \left(\frac{dy}{dx}\right)^2}}\right)^3 \tag{12}$$

$$\sec \alpha = \sqrt{1 + \left(\frac{dy}{dx}\right)^2} \tag{13}$$

$$\phi = \int \frac{V}{\sqrt{\theta}} \sec \alpha \ dx \ . \tag{14}$$

The equations for the weight flow integration along a velocity potential line become

$$(V/\sqrt{\theta})_2 = (V/\sqrt{\theta})_1 \quad \exp \int_{y_1}^{y_2} C \sec \alpha \ dy \ , \quad (15)$$

$$\frac{W\sqrt{\theta}}{\delta} = \int f(V/\sqrt{\theta}, T_t) \sec \alpha \ dy \ . \tag{16}$$

The indicated differentiations and integrations were programmed by using the Lagrangian interpolation formula for a three-point interpolation.

Letting:

$$\Psi(x) = \Psi(x_0) \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + \Psi(x_1) \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + \Psi(x_2) \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} , \qquad (17)$$

then:

$$\frac{d\Psi(x)}{dx} = \Psi(x_0) \frac{(x-x_1) + (x-x_2)}{(x_0 - x_1)(x_0 - x_2)} + \Psi(x_1) \frac{(x-x_0) + (x-x_2)}{(x_1 - x_0)(x_1 - x_2)} + \Psi(x_2) \frac{(x-x_0) + (x-x_1)}{(x_2 - x_0)(x_2 - x_1)} , \qquad (18)$$

$$\frac{d^2\Psi(x)}{dx} = \frac{2\Psi(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{2\Psi(x_1)}{(x_1 - x_0)(x_1 - x_2)} + \frac{2\Psi(x_2)}{(x_2 - x_0)(x_2 - x_1)} .$$
(19)

Carrying out the indicated integrations for two spaces,

$$\int_{x_0}^{x_2} \Psi(x) \, dx = \left(\frac{x_2 - x_0}{6}\right) \left[\Psi(x_0) \left(2 - \frac{x_2 - x_1}{x_1 - x_0}\right) + \Psi(x_1) \left(2 + \frac{x_2 - x_1}{x_1 - x_0} + \frac{x_1 - x_0}{x_2 - x_1}\right) + \Psi(x_2) \left(2 - \frac{x_1 - x_0}{x_2 - x_1}\right)\right].$$
(20)

Equations 12, 13 and 14 are programmed in one deck containing 43 cards for each point. The integration is taken over two spaces at a time; thus, two values of the integral are rotated in storage as the integration proceeds. A preliminary calculate deck into which the values of the first three points are entered precedes this calculation. After this calculation is completed, an interpolation for equal values of ϕ is carried out in two parts, the first part getting interpolated values for x, y and $V/\sqrt{\theta}$, the second part getting interpolated values for C and sec α .

Equations 15 and 16 are programmed in a second deck containing 41 cards together with an 18-card table. The integration is carried out in the same manner as the previous one, with $f(V/\sqrt{\theta},T_t)$ being found by a table look-up. Second degree interpolation is used, with T_t being held constant for a particular problem and given by the entrance conditions. This calculation is followed by a final deck which computes a better value of $V/\sqrt{\theta}$ for the first streamline to enter in the next iteration. When this calculation is complete, an interpolation for equal weight flow between the streamlines is carried out, interpolated values being obtained for x, y and $V/\sqrt{\theta}$. By choosing 320 points, the entire calculation may be carried out in 18 hours, assuming an average of three iterations for the weight-flow integration and assuming that three runs are necessary for convergence. The flow sheets for the calculation and the calculate instructions are given in the Appendix.

Notes on the

IBM CARD-PROGRAMMED ELECTRONIC CALCULATOR SETUP

The ability of the IBM Card-Programmed Electronic Calculator to handle the problem outlined cannot be completely understood without a description of the manner in which the machine is programmed for the problem. Basic setup 2 is used. For this setup the 604 Electronic Calculator is scheduled as follows:

Code	Notes
1	· · · ·
2	•
3	Round-off.Drop5digits.
4	Round-off. Add 5 digits to dividend in dividing.
5	
6	-6.70000 < x < 6.70000
7	.01000< <i>x</i> <1.99000
6–8	-6.70000 < x < 6.70000
7–8	.01000 < x < 1.99000
	Code 1 2 3 4 5 6 7 6–8 7–8

The IBM Type 402 Accounting Machine is instructed to perform the special operations as shown in Table II.

	Т	able II	
Operation	Code	Column	Notes
Counters clear	1	12	
Auxiliary storage clear	2	12	
Spread read-in	3	12	
Arctangent shift	4	12	Divides $\tan x$ into 1 and adds π into counter if $\tan x < 1.00000$.
Machine stop-Negative balance	e 5	12	
Machine stop - Two successive	•		
negative balances	6	12	
Select one of two factors	7	12	Compares two factors, selecting larger or smaller as desired.
Decimal shift for log _e x	8	12	Shifts x one unit to right and adds $\log_e 10$ for $x > 1.99000$.
Absolute value	9	12	
Table look-up	9	13	



FIGURE 3

All factors are entered originally into the standard card form shown in Figure 3 and, in summary punching, are punched ordinarily from a given counter field into the corresponding counter read-in field. Read-in and other instructions are emitted during punching.

Appendix

CALCULATE INSTRUCTIONS

1. Sort initial data cards in ascending order of ϕ and then in ascending order of $W_a\sqrt{\theta}/\delta$ using identity numbers in counter 7. Merge cards of same $W_a\sqrt{\theta}/\delta$ with Calculate Deck 1, merging the first three cards into the preliminary deck.

- 2. Calculate C, sec α , and ϕ , punching values as shown in Figure 4. Emit and punch 3 in column 12 (spread read-in instruction).
- 3. Sort summary punched cards in descending order of ϕ . Gang punch ϕ_{n+1} on ϕ_n cards. Merge with $\overline{\phi}$ Interpolate Deck A.
- Calculate x̄, ȳ and V̄/√θ and punch these together with Ident and 3 in column 12. Omit punching in counters 5 and 7.

	COUNTER 1	2	3	4	5	6	7
Initial data	x	у	$V/\sqrt{ heta}$				Ident $W_a \sqrt{\theta} / \delta \phi$ xx xx
Summary punch from Deck 1	x	у	$V/\sqrt{\theta}$	С	sec a	ϕ^*	Ident
Interpolate Deck						$\overline{\phi}$	Ident
Summary punch from Interpolate Deck (two runs) (Initial data for Deck 2)†	\overline{x}	\overline{y}	$\overline{V}/\sqrt{ heta}$	Ē	sec α		Ident
Summary punch from Deck 2	x	y	$V/\sqrt{\theta}$			$\overline{W_a\sqrt{ heta}/\delta}$	Ident
Interpolate Deck						$\overline{W_a\sqrt{ heta}/\delta}$	Ident
Summary punch from Interpolate Deck (initial data for Deck 1)	\overline{x}	$\overline{\mathcal{Y}}$	$\overline{V}/\sqrt{ heta}$				Ident

*Omit punching of ϕ for first calculation.

[†]Omit this interpolation for first calculation.

FIGURE 4. FLOW SHEET

- 5. Merge first summary punched cards with ϕ Interpolate Deck B.
- 6. Calculate \overline{C} and $\overline{\sec \alpha}$ and punch these together with Ident.
- 7. Gang punch \overline{C} and sec $\overline{\alpha}$ on first group of interpolated cards containing \overline{x} , \overline{y} and $\overline{V}/\sqrt{\theta}$.
- Sort cards in ascending order of W_a√θ/δ and then in ascending order of φ, using identity numbers in counter
 Merge cards of same φ with Calculate Deck 2, merging the first three cards into the preliminary deck.
- 9. Calculate $V/\sqrt{\theta}$ and $W_a\sqrt{\theta}/\delta$. Follow last calculate deck by final deck to get a better approximation for $(V/\sqrt{\theta})$. Repeat calculation using new value and continue until convergence occurs. Emit and punch 3 in column 12 when punching.
- 10. Sort summary punched cards in descending order of $W_a\sqrt{\theta}/\delta$. Gang punch $(W_a\sqrt{\theta}/\delta)_{n+1}$ on $(W_a\sqrt{\theta}/\delta)_n$ cards. Merge with $W_a\sqrt{\theta}/\delta$ Interpolate Deck.
- Calculate x̄, ȳ and V̄/√θ and punch these together with Ident and 3 in column 13.
 For first run, omit steps 3, 4, 5, 6 and 7 from the calculating procedure.
- NOTE: The instructions to handle the relaxation of the entering and exit boundaries are omitted here. The above instructions, however, are used for the general calculating procedure.

REFERENCE

1. HORACE LAMB, Hydrodynamics, 6th Edition (Dover Publications, Inc., 1945), Chapter II.

DISCUSSION

Dr. Hurd: Here is an example of a mathematical model at which Mr. Horner has arrived on the basis of certain assumptions followed by a calculating procedure to get specific results. Have you confronted this procedure with some experimental results to gain a notion as to what is the accuracy, the adequacy of the model, etc.? Is this possible?

Mr. Horner: It is possible to check this by experimentation. In fact, the object of this calculation originally was to remove many of the experimental procedures that were necessary. We spend much time and money checking our design of the blade for an engine to see whether it is satisfactory. We can design blades that are between 80 and 90 per cent efficient. We think that using the difference made by the compressibility of the fluid will enable us to design more efficient blades. Most blades are designed now from profiles derived from flow in an incompressible fluid.

Mr. Patton: At the Lewis Flight Propulsion Laboratory we run into similar problems, but we use a completely different method of attack. We set up the non-linear elliptic difference equation that defines the flow and develop a set of simultaneous equations from the difference cofficcients. So far, we have solved three problems of that type. One involved 385 simultaneous equations; another, 379; and the latest one, 191. We used a three-point formula for the first two and a five-point Lagrangian formula for the third. We put all our non-linearities on the right-hand side and start out with the incompressible case; from that, we develop the compressible cases.

Dr. Hurd: Is this a factorization process? Mr. Patton: Yes, it is.

Automatic Calculation of the Roots of Complex Polynomial Equations Using the IBM Card-Programmed Electronic Calculator

JOHN GALLISHAW, JR.

Chance Vought Aircraft

×

COMPLEX polynomial equations, up to the seventh degree, are currently being solved entirely automatically on an IBM Card-Programmed Electronic Calculator which has the special program device, and a total of twelve coselectors, two digit selectors, and sixteen pilot selectors. The method normally requires no card handling from the time the coefficients are loaded into the machine until all the roots have been found.

The solution process is the well-known iterative scheme involving synthetic division and Newton's method. For each root the machine performs only enough iterations to produce the root to the desired accuracy and then proceeds automatically to the calculation of the next root. If any, or all of the roots are approximately known in advance, these values can be introduced as first trials; otherwise, standard trial values are used. Using standard trial values, the CPC takes from twenty to twenty-five minutes to obtain all the roots of a complex seventh degree equation. Equations of lower degree require correspondingly smaller amounts of time.

The process uses eight-digit arithmetic throughout. With proper size adjustment of the coefficients of the original equation the calculations seldom run out of field. The size adjustment is also done on the CPC with a separate set of control panels. No data are required except the unadjusted coefficients, and the process is entirely automatic, taking about thirty seconds for each seventh degree equation.

Theory

If a polynomial f(z) is divided by a trial linear factor $(z-z_n)$ and the quotient $q_1(z)$ is again divided by $(z-z_n)$ the successive remainders are $f(z_n)$ and $f'(z_n)$. We give below the identities involved and at the same time introduce our notation. Note that the degree of f(z) is N - 1.

$$z = x + iy (1)$$

$$f(z) \equiv \sum_{p=1}^{N} (a_p + ib_p) z^{p-1} \,.$$
(2)

$$f(z) \equiv (z - z_n) \sum_{p=2}^{N} (c_p + id_p) z^{p-2} + (c_1 + id_1).$$
(3)

$$q_1(z) \equiv \sum_{p=2}^{N} (c_p + id_p) z^{p-2}.$$
 (4)

$$q_1(z) \equiv (z - z_n) \sum_{p=3}^{N} (e_p + if_p) z^{p-3} + (e_2 + if_2) .$$
 (5)

$$f(z_n) = c_1 + id_1$$
 (6)

$$f'(z_n) = e_2 + if_2$$
 (7)

If z_n is a trial root, then the "improved" trial root z_{n+1} given by Newton's method is

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} = z_n + \Delta z_n$$
 (8)

Using the notation of (1), (6), and (7), the real and imaginary parts of (8) are

$$x_{n+1} = x_n - \frac{(e_2c_1 + f_2d_1)}{(e_2)^2 + (f_2)^2} = x_n + \Delta x_n , \qquad (9)$$

$$y_{n+1} = y_n - \frac{(d_1 e_2 - c_1 f_2)}{(e_2)^2 + (f_2)^2} = y_n + \Delta y_n : \qquad (10)$$

Synthetic division is used when the two divisions by $(z-z_n)$ are carried out with a desk calculating machine. The computation form is as shown in Table I, page 88. TABLE I

$$\begin{array}{ll} (a_N+ib_N) & (a_{N-1}+ib_{N-1})\dots(a_2+ib_2) & (a_1+ib_1) \left| x_n+iy_n \right. \\ (c_N+id_N) & (c_{N-1}+id_{N-1})\dots(c_2+id_2) & (c_1+id_1) \end{array}$$

$$\begin{array}{l} (e_N+if_N) & (e_{N-1}+if_{N-1})\dots(e_2+if_2) \\ \text{where} & (c_p+id_p) = & (a_p+ib_p) + & (c_{p+1}+id_{p+1}) & (x_n+iy_n) \\ & \left. \{ p = N, N-1, \dots, 2, 1; \ c_{N+1} = & d_{N+1} = 0 \right\} \\ \text{and} & & (e_p+if_p) = & (c_p+id_p) + & (e_{p+1}+if_{p+1}) & (x_n+iy_n) \\ & \left. \{ p = N, N-1, \dots, 2; \ e_{N+1} = & f_{N+1} = 0 \right\} \end{array}$$

The calculations performed by the CPC to obtain (c_1+id_1) and (e_2+if_2) are:

(c_N+id_N)	=	(a_N+ib_N)	+	0	$\cdot (x_n + iy_n)$
$(e_{N+1}+if_{N+1})$	=	0	+	0	$\cdot (x_n + iy_n)$
$(c_{N-1}+id_{N-1})$	=	$(a_{N-1}+ib_{N-1})$	$+ (c_N -$	$+id_N)$	$\cdot (x_n + iy_n)$
(e_N+if_N)	==	(c_N+id_N)	$+ (e_{N+1})$	$+1+if_{N+1}$)	$\cdot (x_n + iy_n)$
$(c_{N-2}+id_{N-2})$	=	$(a_{N-2}+ib_{N-2})$	$+ (c_{N-1})$	$-1 + id_{N-1}$	(x_n+iy_n)
$(e_{N-1}+if_{N-1})$	=	$(c_{N-1}+id_{N-1})$	$+ (e_N -$	$+if_N$)	$\cdot (x_n + iy_n)$
• • •					
$(c_2 + id_2)$	_	$(a_2 + ib_2)$	$+ (c_3 +$	$-id_3$)	$\cdot (x_n + i y_n)$
(e_3+if_3)	=	(c_3+id_3)	+ (e4+	$-if_4$)	$\cdot (x_n + iy_n)$
(c_1+id_1)	=	$(a_1 + ib_1)$	$+ (c_2 +$	$-id_2$)	$\cdot (x_n + iy_n)$
(e_2+if_2)	=	(c_2+id_2)	$+ (e_3 +$	$-if_3$)	$\cdot (x_n + iy_n).$

It will be noted that the machine goes through a small number of "zero" calculations.

		Тав	le II				
(A)	Field			a	CP	Ç Ca	odes
Cycle	Selector Level	Operation	Special 402 Control	\overline{A}^{Ch}	ianne B	C^{2l}	Operation
1	9		Total $2 \rightarrow +4$		••		••••
2	8		$ \{ \text{Total } 6 \rightarrow +2 \} $ $ \{ \text{Total } 7 \rightarrow +3 \} $	72	73		
3	7	$(c_{p+1}x_n) - (d_{p+1}y_n) \bigg _{c}$	(1(p)	· 		2
4	6	$a_p \int^{c_p}$		73	72	76	3
5	5	$(d_{p+1}x_n) + (c_{p+1}y_n) \bigg _{d}$		2(<i>p</i>)			1
6	4	$b_p \int d_p$		74	75	77	3
7	3	$(e_{p+2}x_n) - (f_{p+2}y_n) \Big]$		82			2
8	2	c_{p+1}		85	84	72	3
9	1	$(f_{p+2}x_n) + (e_{p+2}y_n) \Big _{f}$		83	.		1
10	0	$d_{p+1} \int^{J_{p+1}}$				75	3

(D)	Table III													
(в)	Field Selector		Special	-C	CP hanne	C C C C C C C C C C C C C C C C C C C	odes							
Cycle	Level	Operation	402 Control	A	<u> </u>	C	Operation							
1	9		Total $2 \rightarrow +4$											
2	8		$ \begin{array}{l} \text{Total } 6 \rightarrow +2 \\ \text{Total } 7 \rightarrow +3 \end{array} $	74	74									
3	7	$(e_2)^2$		75	75		5							
4	б	$(f_2)^2$		74	72	76	5							
5	5	$-(e_2c_1)$		75	73		6							
6	4	$-(f_2d_2)$		76			6							
7	3	$x_n - \frac{(e_2c_1 + f_2d_1)}{(e_2)^2 + (f_2)^2} = x_{n+1}$		84	83		7							
8	2	$-(d_1e_2)$		85	82		6							
9	1	$+(c_1f_2)$		86			5							
10	0	$y_n - \frac{(d_1e_2 - c_1f_2)}{(e_2)^2 + (f_2)^2} = y_{n+1}$				75	8							

MACHINE PROCEDURE

First, the coefficients are loaded into the auxiliary storage by means of precoded loading cards. A "first guess" card is now read to supply $x_n + iy_n$. Next, an iteration card is read to cause a series of special program cycles, and to supply the first value for p, i.e., p = N. Instruction codes are emitted from the field selector and p is emitted from a counter of the IBM Type 402 Accounting Machine.

The two basic series of cycles are shown in Tables II and III.

Cycle series (A) is repeated once for each complex coefficient of the equation, reducing the value of p by one in each series. When p = 1, the following series is of type (B). This performs Newton's correction, zero tests Δx_n and Δy_n , and replaces x_n with x_{n+1} and y_n with y_{n+1} . This completes one iteration.

Another iteration card is read, and the whole process is repeated until Δx and Δy are zero to the desired degree of accuracy. The unused iteration cards are passed through the machine, and the root is divided out by synthetic division with precoded cards which load the coefficients of the depressed equation in place of the old equation. The root is summary punched, and the whole process repeated until all the roots are found.

Coding

For a complete solution of one equation, coefficient loading cards and a prepunched master deck are required. The loading cards, of course, are different for each equation. The master deck differs only with the degree of the equation.

Each root of an equation requires three types of precoded cards—a first guess card, iteration cards, and a set of depressed equation cards.

It should be noted that the channel A and B instructions must be punched in the card preceding the one on which they are to take effect. This is because the wiring is from the lower brushes instead of the upper brushes.

ADDITIONAL CODES

As well as the usual CPC codes, the following additional codes are required:

- 1. A code to set up the field selector for emitted instructions and start special programming. This is a 9 punch in a column not used for anything else.
- 2. A code to allow the machine to go into special program. This is a 9 punch in a column not used for anything else.
- 3. A code to identify the initial value of *p*. This is a digit in a column not used for anything else.
- 4. A code to identify "first guess" cards. This is an X punch.
- 5. A code to clear storage, counters and the 604 at the beginning of each equation. This is an X punch.
- 6. A code to cause the results to list. This is an X punch.

AL	3-	-+ (
7/a	4	D.	=	D	
1.		77	_	5"+1	
7H	*	Fn.	æ	En+1	

GRAM	PR	OG	RAA	A	TESTS	FACTOR	STORAGE	MULT.		COUNTER		GENERAL			
PRO	Svp.	Su Typ	p. 1 xe 3	lon. Sup.		land 2	and 2 3 and 4		RI		RO	land 2	3 and 4		
R						A	В	ā		[]		[X _n]	[Yn]		
9		1		,2					MULT +	[+ ā D]		RO			
10		1	4	1		RO		3							
11						āД				R					
12			-						Милт+	[+ # D]		RO			
13						RO			+ ©	+āD=[AD]					
14			_						/z adj	· · · · · · · · · · · · · · · · · · ·					
15						AD.					RR				
16				1			RO			+ <i>B</i>					
17			_					Б			RR				
18					ļ			L	Mult_2	[± Б Е]			RO		
19			╡				RO	6							
20			-	_			±БЕ				RR				
21		Ц	4	1	 				MULT - 2	[± <u>b</u> E]			RO		
22		Ц	4		ļ		RO		+ 6	± БE=[BE]					
23		H	4						1/2 adj						
24							<u> </u>				RR				
25			4	Ł		RO			+	+AD					
26		1		,2	ļ		RO		+	± BE					
L			1		ļ	·									
30		Д	1	3		RO			+	<u>+ A</u>					
34		Ш	[4			RO		+	+B					
			_							·					
38	i	Д	7	5.6	ļ	ļ	RO		Милт <u>+</u> 5	[±ā B]					
39		1	4	1		RO		æ							
40		μ	_	4	ļ	<u>āB+Č</u> *					RR				
41			+	\downarrow			RO		Mulr_6	[± ≩ B]					
42		Ц	-	+-					1/2 adj						
43				+		RO			+ [@]	±āB+C					
44		1		ł		<u>±AB+C</u>					RR				
45		Д	7	5,6	<u> </u>	RO			+	±AB+C					
46		V	4	7,8		RO		1%	÷						
47		H	+	1	 		Rem				RR				
48		$\left \cdot \right $	-	+	ļ		RO		+ °	+Rem				-	
49			+	+-	├		+ %a	RO							
50		┝┥	+	+-		RO		<u>[%</u>]	÷						
51			4	+					6		RR				
52		μ	+	+-			RO		+ "	+ ^c /a					
53		μ	+	+		<u>ci</u>		RO	+	+ <u>Ya</u> =[YA]					
54		\mathbb{H}		+		4A			6		RO				
55		μ		+	ZT				Ś		RS				
56		$\left \cdot \right $	+	+-		RO			+	<u> </u>					_
57		1	+	+	ļ				+	=[^n+i or Yn+i]		R0 7	R08		
58		Y	-	+-	871 on 7						RR	Xn+17	Yn+1 8		
59	NZ	Ø	-	*	BTzonB				-	-"1"					
60	NZ	V		7,8	L						RR				

* CLEFT IN COUNTER FROM PREVIOUS CALCULATION

FIGURE 1. TYPE 604 PROGRAM CHART

Figure 1 shows the programming used in this solution process. The following notations and abbreviations are used:

- Α is the entry from channel A into FS 1, 2.
- В is the entry from channel A into FS 3, 4.
- С is the counter with exit into channel C.
- is the real part of the trial root in GS 1, 2 with exit into the D 402 type bar for printing purposes.
- is the imaginary part of the trial root in GS 3, 4 with exit into the 402 type bar for printing purposes. E
- A $=\overline{a}+a$

where $\bar{a} = xxx.00000$,

 $\underline{a} = 000.\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}$

- C means quantity C was not entered on the read cycle, but was there previously.
- GS 1 and 2 means GS 1 and 2 are given an 8-6 assignment. [term] means it is a natural result of a calculation; without bracket it means the unit must be wired to read in. RS means reset.
- $\operatorname{Mult}_{-2}^{+1} \bigg\}$ means multiply + when calculate selector 1 is transferred.
 - means multiply when calculate selector 2 is transferred.
 - BT_1 means balance test to pick up negative balance selector 1. NZ means non-zero.

	N. B.	SEL.										CO-SEI	ECTO	RS								:
	то	0	•4"	"4" •	"7" ●	•5•	"7" •	"5" •	"5" •	"4" ●	•5*	"7" •	• 5•	"7" •	•3*	"6" ●	"6" •	0	0	0	0	"6" Фт
	NO	10	"2" ●	"3" ● 1	"1" •	P9N1	0	0	•2"	"3" •	"3" •	"2" ●	P9N1	0	30	#1# ●	"4" ●	#7# ●	"5" • 4	"7" 1 ●	"7" ●	"3" ●N
	cO	0	F8,5	F8,7	F7,4 ●	F7, 5	F7,6	F7, 7	F7, 10	F6, 5	F6, 10) F5,4	F5, 5	F5,6	F5,7	F5,10	F4, 5	F4, 6	F4,7	F4, 8	F4,9	F4,10 C
	<u> </u>	-	*4*	"8" •	"3" •	#7# ●	"2" ●	0	0	"6" ●	"6" ●	"5" •	"8" •	0	0	P151	0	0	0	0	P6C1	от
	P4C2 T●	BAR	"2"	~	0	"2" •	" 4 " ●	"7" ●	"2" ● 6	"3" •	"3" •	•1* ●	"3" •	F7,2 F5,2	F4, 2 7 ●	0	P1C2 ●	ο	0 1	30	0	ON
	NO	20	F3,5	F3,6	F3,7	F3,10	F2,7	F2, 8	F2, 9	F2, 10	F1,5	F1,10	F0, 10	₽13,I ●	₽14, I ●	F0,2	F0,1	0	0	0	ELAC	oc
	"10" C.●	"10"	0	0	0	$\overline{}$	6	0	0	0	0	0	P8N1	FC, 3	0	FC,4	FC,5	FC,6	FC,7	FC,8	FC,9	FC, 10
	···		P1 5C2	0.9	• 0	0		P5C1	0 1	00	0	0	0		(1) 1	CH.AI	B NST	LB CH, BI		LB CH.CI	NST	LB OPER
	NB .	- AC	C11C2	0	0	Õ	0		0	0	0	0	NON	C9C1	P5C2	CHL A	CONT	сн. в с	CONT	сн.с	CONT.	OPER CONT
		SELECT	ORIE	 זי טיג		Ť	PILO	T SELE	CTORS			UB(4)	UB"X"	UB(6)			UB(5)	LB(S)	-			UB"X"
	q	1		ľ		1	0	0	0	°	0	•	•	•	0	0	•	•	0	0	0	●x
	0	. 🎤	Y	A	LL PRO	Ľ	COMP	05(2)	1	>	1	0	0	O SU	0	0	0	0	0	0	0	OD
	•	CDCY	ৎ	<	EPL.		EXIT 2	0	(1-7P	-	-	0	0	CHG	<u> </u>	<u> </u>	0	0	C7C2	,C3	C7T4	01
	1	P31	S)	\)	12	\ °	CDCY	4	"10" •		RI	"10" ●	LIST	4	"1/2"+3		AC ●	AC ●	*9* •	AC ●	"10" ●T
	F9,1	F8,1 2●	521 C		SPLIT O	~	\	þ.	6	0	0	RO •	0	C11T1	ام	0	0	о	0	LB(3) ●	0	N
Ŋ	AC 3	AC	521)P	UNCH	SEL, P.	. .	P10C1	+2B	0	EAR (C8T4	SIGN GS1,2	* AC/	C3N1 C1N4	P1C1		+6A	ТОТ 2А ●	2A EXIT 1	10T 2A	SIGN GS 3, 4 C
CONT	+2,3 OT 6,5	TTON	52	PUNC	TH SEL	0	7	~	0	~	0	0	0	0		\mathbf{V}	0	0	AC	AC	PROG STOP	от
S. P.	0			0	NŌ	。	0	CDCY	0	0	PROG START	0	0	β	•	\mathcal{L}	0	0	0	P2C2	FIELI SEL, P	N NO
	°	~``	ľ,	Ň	- 0		C7N5	P14N2	°	NB2T1	CIIC	3		/ \	2A ENT	1	0	0	+2A	+2A	C9N1	00
	FLD. SE	L. P.U.	P15N2	UB(5)	<u>с о</u> st.c	 L. 1	2 0	OUNTE		us		7	17	S. P. C	CONT	╀		FIELI	SELE	CTION		
	<u> </u>	-0					#211		•		-	-/	0	S. P. C	ONT	<u> </u>	0	0	0	0	0	09
	•	•	•	•		ė		•	0	0	0	•	0	•	نو			" •		0	0	08
	٩	1	. N	. 1	R	R	R	. N	٩	٩	3	20	0	•	•	•	•	•	•	0	0	•7
	0	A	V	A	A	V	A	Y	-	6	Ъ	6	0	0	•5"	•	Ç2C3	•	•2*	•	•	C2C4 ● 6
	<u> </u>	0	0	0	0	0	0	0	0	0	0	"1" •	0	C7N2	2 n4n	C2C5	C3C1	C3C2	C3C3	0	0	C3C4
C ▲	тО	×+ ●	X+ ●	0	0	X+ ●	X+ ●	0	NZ (NB1P	RO	RI	0	C7N3	•3# •	*7* •	C3C5		C4C2	C4C3	C4C4	C4C5 ● 4
L C	I N O 1	10	0	. II 3 ●	IV •	50	0		SUP	, 0	ο	o	0	0	"2" •	"8" •	C5C1	C5C2	C5C3	' o	0	C5C4
s	c			SUP ●							٩		0	0	"1" ●	"8" ●	"5" •	"8" •	C5C5	C6C1	C6C2	C6C3
L E	то	x-	x-	0	0	x-	x-	0	SUP NZ	NB2P	GS34 RO	GS34 RI	0	0	"O"	"8" •	C6C4	0	0	0	0	C6C5
C T	N			Ш	1			1	J			J	C7C5	C7N4	1911 •	0	0	0	0	"7"	"5"	C7C1
O R	SUP		2	SUP	SUP	38	3	SUP	vi	59	57	58	"9"	ELAC	•		:11-TR	ANS-C	12	•	•	•••
5	с●	•	•	•	1 •	•	•	•	•	•	•	•	i 🛡	•	?	-	5	Î	2	3	4	5 C

FIGURE 2. DETAILS OF SELECTOR WIRING FOR TYPE 402

Figure 2 is a wiring diagram of some unusual portions of the 402 control panel. In this diagram the notations and abbreviations are as follows:

- F 9,2 means field selector level 9, column 2.
- "4" means digit 4 is emitted from the digit emitter.
- P9N1 means pilot selector 9, normal of upper point.
- P9N2 means pilot selector 9, normal of lower point.
- C9C1 means co-selector 9, common of point 1, numbered from left to right.
- P1-I means the immediate pickup of pilot selector 1.
- NB2T1 means negative balance selector 2, transfer of point 1.
 - NB1P means negative balance selector 1, pickup.
 - CDCY means card cycles.
 - LB (1) means to wire additional code 1 (see previous section on Additional Codes) from lower brushes.
 - $\frac{1}{2} \rightarrow X$ means half after X time. (58) means program step 58.

DISCUSSION

Mr. C. V. L. Smith: How do you determine the first approximation?

Mr. Gallishaw: Actually, we do not even try to determine it. Because of the transformation which we performed originally on the coefficients, we know that the product of the roots is going to be somewhere in the vicinity of 1; so we choose a trial root. Actually, we choose the trial root, .8 + i(.6), which has a modulus identical to the 1 which is usually attached to one of the middle roots.

We have been equally successful in trying to attack the smallest root by studying an initial trial root of zero.

Dr. Brown: You mentioned that after the first root is calculated, the machine goes on to the second root. Do you get the second root by using the same equation, or do you reduce it to a sixth-degree polynomial from a seventh, etc.?

Mr. Gallishaw: We reduce it to a sixth from a seventh, and replace the coefficient to the seventh with the coefficient to the sixth as we go along.

Dr. Welmers: Recently I was attempting to teach a class in matrix theory that the roots of a polynomial equation could be obtained as characteristic roots of a matrix. I was wondering if this particular scheme has found any very widespread calculational use, or how it compares with the more direct method indicated here.

Dr. Gross: I once had a problem on the CPC involving many cubics with complex coefficients. When I tried to iterate the roots by putting the equation in a matrix form, it sometimes worked out fairly well. In one example, we had to iterate 200 times to get the roots to separate, even though the ratio was distinctly different from 1. Only if you

knew the absolute values of the roots beforehand, could you calculate approximately how many iterations would be needed.

Mr. Lowe: We use quite the inverse process. We derive by setting up the characteristic function of the matrix and then get the roots of the matrix.

Mr. Carter: Do you find it better to start with the smaller root? When you divide out the smaller root to get the polynomial of degree one less, it seems to me your coefficients are going to be more accurate. Consequently, do you not find it better to start with zero?

Mr. Gallishaw: Generally speaking, we have found that to be true. Not only does it help in achieving greater accuracy in the roots but it also tends to make the coefficients of the depressed equation slightly more workable in size.

Mr. Lesser: In connection with programming from counters, I would like to point out that it is probably easier on a Model II CPC, because top counter read-out eliminates the trouble of getting too many digit impulses out of the counters for instruction. One can obtain many more instructions on the field selector than he can when he has to separate the digit impulses.

Dr. Evans: When digit impulsing as instructions from the counter, do you get all of the half-after impulses from the column split control?

Mr. Gallishaw: In this particular case I did not take it from the column split control, but rather from the emitter into the comparing unit, comparing it against nothing.

One last comment is that this control panel is also very useful in evaluating an equation of constant coefficients for a very large number of values of the unknown, say 1,000.

A Recursion Relation for Computing Least Square Polynomials over Moving-Arcs

GEORGE R. TRIMBLE, JR.

Aberdeen Proving Ground

×

A METHOD of computing least square polynomials over moving-arcs is here presented; it requires the retention of a relatively small number of variable factors in the computations over moving-arcs when advancing from one value of the independent variable to the next. The method is based upon a recursion relation which is derived herein. One consequence of this method is the reduction of the number of storage registers required.

LEAST SQUARE POLYNOMIALS OVER MOVING-ARCS

Given the finite set of real points (X_j, Y_j) in a plane, j = 0, 1, 2, ..., N, with equally-spaced arguments X_j , it is required to fit an *r*th degree polynomial,

$$y_i = y(X_i) = A_{p,0} + A_{p,1} X_i + A_{p,2} X_i^2 + \ldots + A_{p,r} X_i^r , (1)$$

to the subset of 2n + 1 points (X_i, Y_i) , i = p - n, $p - n + 1, \ldots, p - 1, p, p + 1, \ldots, p + n - 1, p + n$, by the least square procedure and to evaluate this polynomial at X_p , the midpoint of the interval (X_{p-n}, X_{p+n}) all abscissae of which are on the polynomial given by equation 1. The arcs of equation 1 thus defined depend upon p and will be referred to as moving-arcs. Each interval (X_{p-n}, X_{p+n}) , $p = n, n + 1, \ldots, N - n$, determines a polynomial; the first subscript in the coefficients $A_{p,q}$, $q = 0, 1, 2, \ldots, r$, indicates the polynomial associated with the interval considered (X_{p-n}, X_{p+n}) . A series of such polynomials

$$y_{p} = A_{p,0} + A_{p,1} X_{p} + A_{p,2} X_{p}^{2} + \dots + A_{p,r} X_{p}^{r} , \qquad (2)$$
$$(p = n, n + 1, \dots, N - n)$$

is obtained. The N + 1 - 2n polynomials of equation 2 are the minimizing polynomials over moving-arcs of equation 1 in the least square sense.

Since the X_i 's are equally spaced, they can be written as

$$X_j = X_0 + jh,$$
 $(j = 0, 1, 2, ..., N)$

where h is a constant. Then,

or

$$X_{j} - X_{p} = X_{0} + jh - X_{0} - ph$$

 $X_{j} = X_{p} + (j - p)h$. (3)

In the interval (X_{p-n}, X_{p+n}) , consider the variable,

$$x_{p,i} = i - p , \qquad (4)$$

$$(i = p - n, p - n + 1, \dots, p + n - 1, p + n).$$

By means of equation 4 we can rewrite equation 3 as

$$X_i = X_p + x_{p,i} h \,. \tag{5}$$

Substituting the expression for X_i as given by equation 5 into equation 1, the polynomial becomes

$$y_i = a_{p,0} + a_{p,1} x_{p,i} + a_{p,2} x_{p,i}^2 + \dots + a_{p,r} x_{p,i}^r (6)$$

(i = p - n, p - n + 1, ..., p + n - 1, p + n)

where, as above, the interval over which this polynomial is being considered is (X_{p-n}, X_{p+n}) . In terms of the new variable $x_{p,i}$ this interval becomes, $(x_{p,p-n}, x_{p,p+n})$. The value of $x_{p,i}$ at the midpoint of the interval $(x_{p,p-n}, x_{p,p+n})$ is

$$x_{p,p} = \frac{1}{h} \left(X_p - X_p \right) = 0;$$

consequently, the polynomial in equation 6 when evaluated at $x_{p,p}$ becomes

$$y_p = a_{p,0}$$
. (7)

From this it is seen that only $a_{p,0}$ needs to be computed for the moving-arc procedure.

The least square procedure requires that

$$F = \sum_{i=p-n}^{p+n} (y_i - Y_i)^2$$

=
$$\sum_{i=p-n}^{p+n} (a_{p,o} + a_{p,1} x_{p,i} + a_{p,2} x_{p,i}^2 + \dots + a_{p,r} x_{p,i}^r - Y_i)^2$$

be minimized. The well-known necessary condition for F to be a minimum is

$$\frac{\partial F}{\partial a_{p,0}} = \frac{\partial F}{\partial a_{p,1}} = \frac{\partial F}{\partial a_{p,2}} = \dots = \frac{\partial F}{\partial a_{p,r}} = 0 \quad . \tag{8}$$

Taking partial derivatives, it is found that

$$\frac{\partial F}{\partial a_{p,0}} = \sum_{i=p-n}^{p+n} 2(a_{p,0} + a_{p,1} x_{p,i} + a_{p,2} x_{p,i}^2) + \dots + a_{p,r} x_{p,i}^r - Y_i)$$
(9)
$$\frac{\partial F}{\partial a_{p,1}} = \sum_{i=p-n}^{p+n} 2x_{p,i}(a_{p,0} + a_{p,1} x_{p,i} + a_{p,2} x_{p,i}^2) + \dots + a_{p,r} x_{p,i}^r - Y_i)$$

$$\frac{\partial F}{\partial a_{p,r}} = \sum_{i=p-n}^{p+n} 2x_{p,i}^r (a_{p,0} + a_{p,1} x_{p,i} + a_{p,2} x_{p,i}^2) + \dots + a_{p,r} x_{p,i}^r - Y_i).$$

Define S_m as

$$S_{m} = \sum_{i=p-n}^{p+n} x_{p,i}^{m} = \sum_{i=p-n}^{p+n} (i-p)^{m} = \sum_{i=-n}^{n} i^{m},$$

$$(m = 1, 2, \dots, 2r).$$

It is seen that

 $S_{2s+1} = 0,$ (s = 0, 1, 2, ..., r - 1), (10)

$$S_{2s} = 2 \sum_{i=1}^{n} i^{2s},$$
 (s = 1, 2, 3, ..., r),

where S_0 is defined as

 $S_0=2n+1$

Simplifying equation 9 by means of equation 10 and applying condition 8 we obtain the normal equations,

$$S_0 a_{p,0} + 0 \cdot a_{p,1} + \ldots + S_r a_{p,r} = \sum_{i=p-n}^{p+n} Y_i$$
(11)

$$0 \cdot a_{p,0} + S_2 a_{p,1} + \ldots + S_{r+1} a_{p,r} = \sum_{i=p-n} x_{p,i} Y_i$$

$$S_r a_{p,0} + S_{r+1} a_{p,1} + \ldots + S_{2r} a_{p,r} = \sum_{i=p-n}^{p+n} x_{p,i}^r Y_i.$$

In matrix notation, equation 11 becomes $MA_p = K_p,$

where

Similarly, equation 7 may be written

(13)

where

(12)

$$B = (1 \quad 0 \quad 0 \dots 0) \ .$$

 $y_p = BA_p$,

Denote the determinant of M by $\Delta(M)$ and the adjoint of M by F(M). It follows that

$$M^{-1} = \frac{F(M)}{\Delta(M)} \,.$$

Solving for A_p in equation 12, we obtain

$$A_p = M^{-1} K_p = \frac{F(M)}{\Delta(M)} \cdot K_p.$$
(14)

Substituting into equation 13 the expression for A_p given by equation 14, we have

$$y_p = B \cdot \frac{F(M)}{\Delta(M)} \cdot K_p.$$
(15)

Examination of F(M) shows that it has the form,

that is, $f_{k,l} = 0$ when k + l is odd. Also, F(M) is symmetric $(f_{k,l} = f_{l,k})$. Evaluating the matrices in equation 15,

$$y_{p} = \frac{1}{\Delta(M)} (1 \quad 0 \quad 0 \dots 0) \begin{pmatrix} f_{0,0} & 0 & \dots & f_{0,r} \\ 0 & f_{1,1} & \dots & f_{1,r} \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\$$

or

$$y_{p} = \frac{1}{\Delta(M)} (f_{0,0} \quad 0 \quad f_{0,2} \quad \dots \quad f_{0,r}) \begin{pmatrix} \sum_{i=p-n}^{p+n} Y_{i} \\ \sum_{i=p-n}^{p+n} X_{p,i} \quad Y_{i} \\ \vdots \\ \vdots \\ \sum_{i=p-n}^{p+n} X_{p,i}^{r} \quad Y_{i} \end{pmatrix}$$

or

$$y_{p} = \frac{f_{0,0}}{\Delta(M)} \sum_{i=p-n}^{p+n} Y_{i} + \frac{f_{0,2}}{\Delta(M)} \sum_{i=p-n}^{p+n} x_{p,i}^{2} Y_{i} + \dots + \frac{f_{0,r}}{\Delta(M)} \sum_{i=p-n}^{p+n} x_{p,i}^{r} Y_{i}.$$
 (16)

This result given by equation 16 will be utilized later in the derivation of a useful recursion relation. For use in hand computations it will be convenient to express equation 16 as

$$y_p = \sum_{i=p-n}^{p+n} H_{i-p} Y_i$$
, (17)

where

$$H_{i-p} = \frac{f_{0,0}}{\Delta(M)} + \frac{f_{0,2}}{\Delta(M)} x_{p,i}^2 + \dots + \frac{f_{0,r}}{\Delta(M)} x_{p,i}^r$$
$$= \frac{f_{0,0}}{\Delta(M)} + \frac{f_{0,2}}{\Delta(M)} (i-p)^2 + \dots + \frac{f_{0,r}}{\Delta(M)} (i-p)^r.$$

Two cases must be considered: (a) r = 2t, (b) r = 2t + 1. If r = 2t, the coefficients H_{i-p} become

$$H_{i-p} = \frac{f_{0,0}}{\Delta(M)} + \frac{f_{0,2}}{\Delta(M)}(i-p)^2 + \ldots + \frac{f_{0,2t}}{\Delta(M)}(i-p)^{2t}.$$

If r = 2t + 1, the coefficients H_{i-p} become

$$H_{i-p} = \frac{f_{0,0}}{\Delta(M)} + \frac{f_{0,2}}{\Delta(M)}(i-p)^2 + \ldots + \frac{f_{0,2t}}{\Delta(M)}(i-p)^{2t},$$

since $f_{0,2t+1} = 0$. This means that the coefficients H_{i-p} are exactly the same for a (2t+1)th degree polynomial as they are for a (2t)th degree polynomial.

The form of equation 17 is convenient for hand computing in that when the observed values Y_i are listed in one column and the coefficients H_{i-p} are listed in the column next to it, the computing consists of multiplying each term in one column by its corresponding term in the other column and summing the 2n + 1 products obtained to determine the fitted value y_p .

THE RECURSION RELATION

The recursion relation provides a method by means of which results obtained at one value of the independent variable are utilized in the computations for the immediately following value.

Equation 16 is rewritten as

$$y_{p} = \frac{f_{0,0}}{\Delta(M)} D_{p,0} + \frac{f_{0,2}}{\Delta(M)} D_{p,2} + \dots + \frac{f_{0,2t}}{\Delta(M)} D_{p,2t} , \quad (18)$$

where $D_{p,k}$ is defined as

$$D_{p,k} = \sum_{i=p-n}^{p+n} (i-p)^k Y_i \qquad (k = 1, 2, ..., 2t),$$
$$D_{p,0} = \sum_{i=p-n}^{p+n} Y_i \qquad (k = 0),$$

and where r = 2t or r = 2t + 1 as previously observed.

At the point X_{p+1} , equations 18 become

$$y_{p+1} = \frac{f_{0,0}}{\Delta(M)} D_{p+1,0} + \frac{f_{0,2}}{\Delta(M)} D_{p+1,2} + \dots + \frac{f_{0,2t}}{\Delta(M)} D_{p+1,2t} ,$$

with $D_{p+1,k}$ being defined as

$$D_{p+1,k} = \sum_{i=p+1-n}^{p+1+n} (i-p-1)^k Y_i, \qquad (k = 1, 2, ..., 2t),$$
$$D_{p+1,0} = \sum_{i=p+1-n}^{p+1+n} Y_i, \qquad (k = 0).$$

But,

$$D_{p+1,0} = \sum_{i=p-n}^{p+n} Y_i - Y_{p-n} + Y_{p+1+n}$$
$$= D_{p,0} - Y_{p-n} + Y_{p+1+n}.$$

Expanding $[(i-p)-1]^k$ by the binomial theorem, we have

$$D_{p+1,k} = \sum_{i=p-n}^{p+n} (i-p-1)^k Y_i - (-n-1)^k Y_{p-n} + n^k Y_{p+1+n}$$
$$= \sum_{i=p-n}^{p+n} \left[\sum_{r=0}^k (-1)^r \binom{k}{r} (i-p)^{k-r} Y_i \right] - (-n-1)^k Y_{p-n} + n^k Y_{p+1+n}$$
$$= \sum_{r=0}^k \left[(-1)^r \binom{k}{r} \binom{p+n}{(i-p)^{k-r} Y_i} \right] - (-n-1)^k Y_{p-n} + n^k Y_{p+1+n}.$$

Thus, one finds a useful recursion relation

$$D_{p+1,k} = \sum_{r=0}^{k} \left[(-1)^{r} \binom{k}{r} D_{p,k-r} \right]$$
(19)
- (-n-1)^{k} Y_{p-n} + n^{k} Y_{p+1+n},

where $k = 0, 1, 2, \ldots, 2t$. From this equation it is seen that $D_{p+1,k}$ can be expressed as a function of $D_{p,r}$, where $r = 0, 1, 2, \ldots, k$, with the binomial coefficients and the two observed values Y_{p-n} and Y_{p+1+n} . By use of this relation, each $D_{p+1,k}$ with $k = 0, 1, 2, \ldots, 2t$ can be computed, using the values $D_{p,k}$, $k = 0, 1, 2, \ldots, 2t$, computed for the previous value of the independent variable, and the observed values Y_{p-n} and Y_{p+1+n} .

To see the merits of the method based on equation 19, a comparison of the two computing methods must be made. The method employing the expression for y_p , as given in equation 17, requires that each value of Y_i , i = p - n + 1, $p - n + 2, \ldots, p + n$, be remembered explicitly when progressing from X_p to X_{p+1} . The only new value needed to compute y_{p+1} is Y_{p+1+n} . If an automatic computing machine is used to perform the computations, 2n storage registers are required to store the 2n values of Y_i , i = p - n + 1, $p - n + 2, \ldots, p + n$, which must be remembered.

The method employing the recursion relation requires that $D_{p,k}$, $k = 0, 1, 2, \ldots, 2t$, be remembered when progressing from X_p to X_{p+1} , where r = 2t + 1. In this case, two observed values Y_{p-n} and Y_{p+1+n} must be introduced to compute y_{p+1} . At most, r + 1 storage registers are required to remember the values $D_{p,k}$, $k = 0, 1, 2, \ldots, 2t$.

If it is assumed that the various constants needed for the two methods can be generated by the machine and present no problem, a comparison of the number of storage registers required by the two methods shows that the recursion method is best by this criterion, as long as r + 1 is smaller than 2n. Since r is the degree of the polynomial and 2n + 1 is the number of points over which the polynomial is being fitted, r is always less than 2n + 1. The values of r for which the recursion method is not the best method are

$$2n - 1 \le r < 2n + 1. \tag{20}$$

Because in practice the number of points used in determining a best-fit polynomial is usually much larger than the degree of the polynomial, the cases indicated in condition 20 will occur infrequently.

Thus, the recursion method makes it possible to perform on automatic computing machines computations involving large numbers of points on moving-arcs which could not be carried out so effectively using manual methods.

Appendix

Illustrative Example of Use of Recursion Method

For r = 2 (or r = 3), and n = 12, the constants are $f_{0,0} = 467$, $f_{0,2} = -5$, and $\Delta(M) = 5175$. From Table I, the following are obtained:

TABLE I	$D_{13,0} = \sum_{i=1}^{25} Y_{i}$
<i>i V</i> .	$D_{13,0} = Y_i$
<i>i I i</i>	
1 121.121	
2 121.232	= 3067.840
3 121.279	25
4 121.341	$D_{13,1} = \sum (i-13) Y_i$
5 121.402	$\sum_{i=1}^{\prime}$
6 121.489	= 218.428
7 121.575	25
8 121.713	$D_{12,2} = \sum (i-13)^2 Y_i$
9 121.845	= 13,2
10 121.997	- 159 774 308
11 122.163	= 139,774.300
12 122.337	$y_{13} = \frac{1}{5175} (467D_{13,0} - 5D_{13,2})$
13 122.515	5175
14 122.692	= 122.475
15 122.884	$D_{14,0} = D_{13,0} - Y_1 + Y_{26}$
16 123.001	= 3072.198
17 123.206	D = D = D + 13V + 12V
18 123.412	$D_{14,1} = D_{13,1} - D_{13,0} + 15I_1 + 12I_{26}$
19 123.655	= 230.909
20 123.891	$D_{14,2} = D_{13,2} - 2D_{13,1} + D_{13,0}$
21 124.115	$-169Y_1 + 144Y_{26}$
22 124.352	- 160 004 819
23 124.608	= 100,004.019
24 124.883	$y_{14} = \frac{1}{5175} \left(467 D_{14,0} - 5 D_{14,2} \right)$
25 125.132	51/5
26 125.479	= 122.646

DISCUSSION

Dr. Berkowitz: Do you actually compute the coefficients of the least square polynomial?

Mr. Trimble: No, the coefficients of the polynomial are in no way determined directly.

Dr. Herget: This is essentially a smoothing process, is it not?

Mr. Trimble: Yes.

Dr. Herget: I don't quite see where we got from 25 to 3. Do you mean a polynomial of the third order is supposed to go through 25 points?

Mr. Trimble: That is correct.

Dr. Aroian: What do you do if you want to go to higher degrees—say 4, 5, or 6? Do you change your degrees or do you just decide on 3 and stick with it?

Mr. Trimble: We decided on 3, for our particular case, from experimental data.

Dr. Aroian. The point is that you might use orthogonal polynomials which permit you to go to higher degrees without recomputing equations. There are two types of tables prepared for these: either the R. A. Fisher or Gram-Charlier tables.

Mr. Trimble: We have used orthogonal polynomials for this process. But even then you must remember each of the 25 values observed separately to get the coefficients for the polynomials.

Mr. Waddell: We have set up this orthogonal polynomial problem and use it continually. By means of Fisher's tables, we use from 21 to 67 points in remembering all the values inside the machine. It works very fast and very neatly on a Model II CPC.

Dr. Hurd: You mean you have one programmed deck which applies to either 21 or anything up to 67 points?

Mr. Waddell: Yes, that is correct.

Numerical Solution of Second-Order Non-Linear Simultaneous Differential Equations

HENRY S. WOLANSKI

Consolidated Vultee Aircraft Corporation, Fort Worth Division

×

STABILITY problems in aircraft design often involve solution of non-linear simultaneous differential equations. For example, the evaluation of the longitudinal stability characteristics for a towed airplane involve the solution of equations of motion which are of the type mentioned above. Basically, the system can be represented as a compound pendulum with two degrees of freedom, i.e., a rigid body connected by a cable to a fixed horizontal axis. The rigid body has acting upon it not only the force of gravity W, but also the aerodynamic forces of lift L and drag D, a thrust force T and an aerodynamic moment M. The rigid body is so suspended that it is free to rotate about a fixed horizontal axis through point O and about a horizontal axis through point H and normal to the X - Z plane as shown in Figure 1. The assumption is made that some tension always exists in the cable OH; if any slack is allowed in the cable, then the aerodynamic characteristics of the rigid body will be identical to those in free flight, and the equations derived for the system will have no physical significance. The purpose of this paper is to present two numerical methods by which a solution can be attained. A further aim of this paper is to compare the methods used and to show the effect of an interval Δt on the results obtained.

Derivation of the Equations of Motion

In deriving the longitudinal equations of motion for the system, the Lagrange equation 1 of force,

$$\frac{d}{dt} \left[\frac{\partial T}{\partial s} \right] - \frac{\partial T}{\partial s} = \frac{\partial W_e}{\partial s}$$
(1)
where, s = generalized coordinate
 T = kinetic energy of the system

 W_e = virtual work of the system t = time



FIGURE 1. RIGID BODY CONNECTED BY A CABLE TO A FIXED HORIZONTAL AXIS

expressed in terms of energy was used, because it was more convenient to deal with the energy of the system. From Figure 1 it can be seen that the coordinates of the center of gravity c of the rigid body are:

$$x = -l\cos\phi - b\cos\theta \tag{2}$$

$$z = l\sin\phi + b\sin\theta.$$
 (3)

The velocity of the center of gravity c is then obtained by taking the time derivative of these coordinates, i.e.,

$$\dot{x} = -\dot{l}\cos\phi + \dot{l}\dot{\phi}\sin\phi + b\dot{\theta}\cos\theta \tag{4}$$

$$\dot{z} = l\sin\phi + l\dot{\phi}\cos\phi + b\theta\cos\theta.$$
 (5)

$$V^2 = \dot{x}^2 + \dot{z}^2$$
(6)

$$V^{2} = l^{2}\dot{\phi}^{2} + b^{2}\theta^{2} + \dot{l}^{2} + 2lb\dot{\phi}\dot{\theta}\cos\left(\theta - \phi\right) - 2lb\dot{\theta}\sin\left(\theta - \phi\right).$$
(7)

Now, cable length is defined in terms of constant reel-in velocity ω and time t as follows:

$$l = -\int \omega \, dt = -\omega t + l_0 \tag{8}$$

or

 $\dot{l} = -\omega$

so that the velocity of the center of gravity c can be written as

$$V^{2} = l^{2}\dot{\phi}^{2} + b^{2}\dot{\theta}^{2} + \omega^{2} + 2b\omega\dot{\theta}\sin(\theta - \phi) + 2lb\dot{\phi}\dot{\theta}\cos(\theta - \phi).$$
(9)

The kinetic energy of the system is equal to

$$T = \frac{1}{2} \left[mV^2 + I_{yy}\dot{\theta}^2 \right]$$
$$T = \frac{1}{2} \left\{ m \left[l^2 \dot{\phi}^2 + b^2 \dot{\theta}^2 + \omega^2 + 2b\omega\dot{\theta}\sin\left(\theta - \phi\right) \right. \right. \\\left. + 2lb\dot{\phi}\dot{\theta}\cos\left(\theta - \phi\right) \right] + I_{yy}\dot{\theta}^2 \right\}.$$
(10)

By use of the assumption of small virtual displacements, the expression for virtual work can be written as

$$\begin{split} \delta W_e &= -L \cos \gamma \; (\delta z) - L \sin \gamma \; (\delta x) - D \cos \gamma \; (\delta x) \\ &+ D \sin \gamma \; (\delta z) + W \; (\delta z) + T \cos \theta \; (\delta x) \\ &- T \sin \theta \; (\delta z) + M \; (\delta \theta) \; , \end{split}$$

where

$$\delta x = l \sin \phi (\delta \phi) + b \sin \theta (\delta \theta)$$

$$\delta z = l \cos \phi (\delta \phi) + b \cos \theta (\delta \theta).$$

Simplifying equation 11 gives the following expression for virtual work:

$$\delta W_{\theta} = \begin{bmatrix} -Ll \cos (\gamma - \phi) + Dl \sin (\gamma - \phi) \\ -Tl \sin (\theta - \phi) + Wl \cos \phi \end{bmatrix} \delta \phi \\ + \begin{bmatrix} -Lb \cos (\gamma - \theta) + Db \sin (\gamma - \theta) \\ + Wb \cos \theta + M \end{bmatrix} \delta \theta.$$
(12)

Combining the kinetic energy equation 10 and virtual work equation 12 in the Lagrange equation 1 will give the final equations of motion.

For the degree of freedom in ϕ , e.g., $s = \phi$,

$$-\frac{\partial T}{\partial \phi} = m\omega b \dot{\theta} \cos \left(\theta - \phi\right) - m l b \dot{\phi} \dot{\theta} \sin \left(\theta - \phi\right)$$
(13)

and

$$\frac{\partial T}{\partial \dot{\phi}} = m l^2 \dot{\phi} + m l b \dot{\theta} \cos\left(\theta - \phi\right), \qquad (14)$$

from which

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\phi}} \right) = -2m\omega l \dot{\phi} + m l^2 \ddot{\phi} - m l b \dot{\theta} \sin \left(\theta - \phi \right) \left(\dot{\theta} - \dot{\phi} \right) + m l b \ddot{\theta} \cos \left(\theta - \phi \right) - m \omega b \dot{\theta} \cos \left(\theta - \phi \right)$$
(15)

so that the equation of motion becomes

$$ml\ddot{\phi} = -m[-2\omega\dot{\phi} - b\dot{\theta}^{2}\sin(\theta - \phi) + b\ddot{\theta}\cos(\theta - \phi)] + W\cos\phi - L\cos(\gamma - \phi) + D\sin(\gamma - \phi) -T\sin(\theta - \phi). \quad (16)$$

Similarly, the equation of motion for the degree of freedom in θ , e.g., $s = \theta$, can be found to be

$$[I_{yy} + mb^{2}]\ddot{\theta} = -m[lb\dot{\phi}^{2}\sin(\theta-\phi) + lb\ddot{\phi}\cos(\theta-\phi) - 2b\omega\dot{\phi}\cos(\theta-\phi)] + Wb\cos\theta - Db\sin(\theta-\gamma) - Lb\cos(\theta-\gamma) + M.$$
(17)

The induced angle γ is defined as

$$\gamma = \frac{-\dot{z}}{V_0 + \dot{x}},\tag{18}$$

whereby, using equations 4 and 5, we arrive at the expression

$$\gamma = \frac{\omega \sin \phi - l\dot{\phi} \cos \phi - b\dot{\theta} \cos \theta}{V_0 + \omega \cos \phi + l\dot{\phi} \sin \phi + b\dot{\theta} \sin \theta} .$$
(19)

The values of lift L, drag D and pitching moment M used in the preceding equations are defined as

$$L = (C_{L_{\alpha}} \alpha \ C_{L_{0}} + C_{L_{\theta}} \theta) qs \tag{20}$$

$$D = (C_{D_{\alpha}}\alpha + C_{D_{0}})qs \tag{21}$$

$$M = (C_{M_{\alpha}} \alpha + C_{M_0} + C_{M_{\dot{\theta}}} \dot{\theta}) qs \qquad (22)$$

where

$$C_{L_{\alpha}} = \frac{\partial C_L}{\partial \alpha} =$$
lift curve slope

 $C_{L_0} =$ lift coefficient at zero angle of attack

$$C_{L_{\dot{\theta}}} = \frac{\partial C_L}{\partial \dot{\theta}} =$$
 rate of change of lift coefficient with pitching velocity

$$C_{D_{\alpha}} = \frac{\partial C_{D}}{\partial \alpha}$$
 = rate of change of drag coefficient with angle of attack

$$C_{M_{\alpha}} = \frac{\partial C_{M}}{\partial \alpha} =$$
 static longitudinal stability
 $C_{M_{0}} =$ pitching moment coefficient at zero angle of attac

$$C_{M\dot{\theta}} = \frac{\partial C_M}{\partial \dot{\theta}} = \text{damping in pitch}$$
$$q = \frac{1}{2}\rho V^2 = \text{dynamic pressure}$$
$$s = \text{wing area}$$

It can be seen that the equations of motion (16 and 17) are second-order non-linear differential equations which have to be solved simultaneously in order to obtain a timehistory plot of the motion. To express a solution in elementary analytical form would be very difficult, if not impossible. It then becomes necessary to use a series or some other approximate method to obtain a particular solution.

METHODS FOR NUMERICAL SOLUTION

The two methods used in obtaining a numerical solution are based on those outlined in detail in reference 1. Basically, both methods are step-by-step solutions where the values of the dependent variables are calculated one after another for a sequence of equally-spaced values of the independent variable t. It is assumed that the functions f_1 and f_2 satisfy all requirements necessary to insure the existence of unique, continuous, differentiable solutions of the form $\theta = f_1(t)$ and $\phi = f_2(t)$ throughout the interval considered.

The problem consists of a pair of simultaneous differential equations of the second order

$$\hat{\theta} = f_1(\theta, \phi, \dot{\theta}, \dot{\phi}, \ddot{\phi}, t)$$
(23)

$$\ddot{\boldsymbol{\phi}} = f_2(\theta, \, \boldsymbol{\phi}, \, \dot{\theta}, \, \dot{\boldsymbol{\phi}}, \, \ddot{\theta}, \, t) \tag{23a}$$

in which the dots denote differentiation with respect to the independent variable t. The iterative and more accurate method of the two consists essentially of predicting the values of the first derivatives using the "prediction" equations which are numerical integration formulas for a fourth-degree polynomial:

$$\dot{\theta}_{n+1} = \dot{\theta}_{n-3} + \frac{4}{3} \Delta t (2\ddot{\theta}_n - \ddot{\theta}_{n-1} + 2\ddot{\theta}_{n-2})$$
(24)

$$\dot{\phi}_{n+1} = \dot{\phi}_{n-3} + \frac{4}{3} \Delta t (2\ddot{\phi}_n - \ddot{\phi}_{n-1} + 2\ddot{\phi}_{n-2}) . \qquad (24a)$$

Next, the dependent variables are computed by numerical integration using the expressions

$$\theta_{n+1} = \theta_{n-1} + \frac{1}{3} \Delta t (\dot{\theta}_{n+1} + 4\dot{\theta}_n + \dot{\theta}_{n-1})$$
(25)

$$\phi_{n+1} = \phi_{n-1} + \frac{1}{3} \Delta t (\dot{\phi}_{n+1} + 4\dot{\phi}_n + \dot{\phi}_{n-1})$$
(25a)

which are based on Simpson's rule. It is noted that the solutions for the values of θ_{n+1} , ϕ_{n+1} , $\dot{\theta}_{n+1}$ and $\dot{\phi}_{n+1}$ are obtained independently for each of the differential equations 23 and 23a just as though it were a single equation. But now in computing the second derivatives of θ_{n+1} and ϕ_{n+1} from the differential equations, the solutions are carried out simultaneously. Having found the values of $\ddot{\theta}_{n+1}$ and $\ddot{\phi}_{n+1}$, the first derivatives are again computed, this time using the expressions for numerical integration given by Simpson's Rule

$$\dot{\theta}_{n+1} = \dot{\theta}_{n-1} + \frac{1}{3} \Delta t (\ddot{\theta}_{n+1} + 4\ddot{\theta}_n + \ddot{\theta}_{n-1})$$
 (26)

$$\dot{\phi}_{n+1} = \dot{\phi}_{n-1} + \frac{1}{3} \Delta t (\ddot{\phi}_{n+1} + 4\ddot{\phi}_n + \ddot{\phi}_{n-1}) .$$
 (26a)

These formulas are known as the "correction" equations. The values of $\dot{\theta}_{n+1}$ and $\dot{\phi}_{n+1}$ obtained by the "prediction" equations are subtracted, respectively, from the ones obtained by the "correction" equations. If the difference is significant within the desired accuracy, the above process is repeated using equations 25, 26 and 23 with the exception of computing new "predicted" values of $\dot{\theta}_{n+1}$ and $\dot{\phi}_{n+1}$, because the previous values obtained from the "correction" equations are used as the new "predicted" ones. This iterative process continues until two successive values of the first derivatives are the same, that is, until convergence is attained. This completes the necessary computations to obtain the values of the dependent variables at one value of the independent variables t.

To apply this method to the system represented by Figure 1, it is necessary to have a set of initial values of θ , ϕ , $\dot{\theta}$, $\dot{\phi}$, $\ddot{\theta}$ and $\ddot{\phi}$ at some time t_0 and also at the times t_{-1} , t_{-2} and t_{-3} . These initial conditions are arrived at as follows: the rigid body is displaced from its equilibrium position to some arbitrary position where θ and ϕ will have the same value and all the derivatives of these functions are then considered to be zero.

The instant at which the body is released from the arbitrary position is considered to be the initial time t_0 . Therefore, the initial conditions and those previous to the time t_0 are known. For example, in the case under consideration, the specified conditions are:

at
$$n = 0$$
, $t_n = 0$,
 $\theta_n = \theta_{n-1} = \theta_{n-2} = \theta_{n-3} = .1$ radian
 $\phi_n = \phi_{n-1} = \phi_{n-2} = \phi_{n-3} = .2$ radian
 $\dot{\theta}_n = \dot{\theta}_{n-1} = \dot{\phi}_n = \dot{\phi}_{n-1} = 0$
 $\ddot{\theta}_n = \ddot{\theta}_{n-1} = \ddot{\phi}_n = \ddot{\phi}_{n-1} = 0$.

Since this is an iterative process, the assumption of $\hat{\theta}_0$ and $\ddot{\phi}_0$ equal to zero is valid, and convergence to the correct values will be attained. The step-by-step computations are carried out until a sufficient range of the independent variable *t* is obtained to show the time-history plot of the dependent variables θ and ϕ .

The second method is a much simpler, approximate one which is based on the numerical integration equations

$$\theta_{n+1} = \theta_{n-1} + 2\Delta t \,\theta_n \tag{27}$$

 $\dot{\phi}_{n+1} = \dot{\phi}_{n-1} + 2\Delta t \, \ddot{\phi}_n \tag{27a}$

$$\theta_{n+1} = \theta_{n-1} + 2\Delta t \, \dot{\theta}_n \tag{28}$$

$$\phi_{n+1} = \phi_{n-1} + 2\Delta t \, \dot{\phi}_n \tag{28a}$$

The computing procedure is self-evident, once the initial conditions have been established. The initial conditions at the time t_0 are identical to those given for the first method, with the exception that the second derivatives of θ and ϕ cannot be considered as zero. Using the differential equations 16 and 17, these second derivatives are computed at the time t_0 , and then the values for the functions and the first derivatives at t_{n+1} are computed using equations 27 and 28. Thus, at each value for the independent variable, the dependent variables and their derivatives are computed without involving any iterative process. As in the first method the step-by-step calculations are carried out until the desired range of t is obtained.

Control Panels

The solution of these differential equations involves a small card volume and many mathematical operations so that the use of an IBM Type 604 Electronic Calculating Punch is inadvisable for this problem. The nature of the problem is such that it can be readily planned and solved using the IBM Card-Programmed Electronic Calculator. The mathematical operations necessary are the simple arithmetic operations of addition, subtraction, multiplication and division, including the trigonometric functions of sine and cosine. A standard CPC-function panel performing these operations using eight-digit factors was available so that it was not necessary to wire special control panels. Thus, setup time for this problem involved only planning the equations and method of solution for card programming. The design of the CPC-function panel is based on an IBM diagram with a modification in computing the sin θ/θ and $\cos \theta$ series. The IBM diagram predetermines the number of terms to be computed for each series, while this panel is wired to compute as many terms as necessary until the nth term is zero. The decimal positioning is such that six decimals are obtained in handling eight-digit factors; therefore, the sin θ/θ and cos θ functions are accurate to the sixth decimal.

101

Discussion of Results

For the particular condition investigated when the values for force of gravity, aerodynamic forces of lift, drag and pitching moment were introduced, the equations of motion (16 and 17) become

$$l\ddot{\phi} = 2\omega\dot{\phi} + 18.76 \ \dot{\theta}^{2} \sin(\theta - \phi) - 18.76 \ \dot{\theta} \cos(\theta - \phi) + 32.2 \cos\phi - 175.5 \ \alpha \cos(\phi - \gamma) - 2.19 \cos(\phi - \gamma) - 15.5 \ \alpha \sin(\phi - \gamma) + .3225 \sin(\phi - \gamma) - 1.862 \sin(\theta - \phi)$$
(29)

$$\hat{\theta} = -.0485 \, l \phi^2 \sin (\theta - \phi) - .0485 \, l \phi \cos (\theta - \phi) + .097 \, \omega \phi \cos (\theta - \phi) + 1.563 \cos \theta - .752 \, \alpha \sin \alpha + .01567 \sin \alpha - 8.5 \, \alpha \cos \alpha - .1061 \cos \alpha - .31 \, \alpha + .0531 - .0536 \, \dot{\theta} \,,$$
(29a)

where the term $(C_{L_{\theta}} qs) \theta$ from the lift expression was omitted from both equations of motion since $C_{L_{\theta}} \simeq 0$. From Figure 1, α is defined as being equal to $\theta - \gamma$; it has also been introduced in the above expressions. The constant reel-in velocity ω was considered to be zero so that the equation for γ was simplified to the form

$$\gamma = \frac{-l\dot{\phi} - 18.76\dot{\theta}}{383} \tag{30}$$

with ω being zero; the cable length remained a constant.

To obtain the best possible accuracy, the iterative method was tried first. In planning the instruction cards, enough iterations were used so that convergence would be attained. This amounted to an instruction deck of 625 cards for which one data load card for the initial condition was necessary. At each t_n all the values of θ , ϕ , $\dot{\theta}$, $\dot{\phi}$, $\ddot{\theta}$, and $\ddot{\phi}$ for n, n-1, n-2 and n-3 were listed so that the computing could be stopped at any point for any reason and started again by merely reloading the listed values.

Computations were carried out for intervals of .1 and .01 seconds for the independent variable t. These results in the form of time-history plots of the motion are shown in Figures 2 and 3, respectively. From Figure 2 it is apparent that because of too large an interval in the independent variable t, the numerical integration of the first and second derivatives of θ and ϕ is inaccurate. This is especially true when the derivatives approach maximum or minimum values, since three-point integration does not approximate the curve well enough. Improvement in the results was obtained by decreasing the interval to .01 seconds. These re-



sults are shown in Figure 3. Although the curves in Figure 3 show no scatter, some inaccuracy does exist but is of such a small magnitude that it is not detected on the scale to which this plot is made. Increasing the accuracy of the results by decreasing the interval is done at the expense of increasing the amount of computing tenfold since ten times the number of points have to be calculated to cover a given range in the independent variable t. At times this may be necessary in order to obtain accurate enough results from which a period of oscillation and other stability characteristics of the motion can be determined.

An alternate approach to increase the accuracy in the final results is to use more accurate formulas for integration rather than decreasing the interval. Five-term integration formulas exist; these are exact if the polynomial is of the seventh degree. This approach was not tried because the method using the three-point integration equations made use of all the storage capacity available in the CPC. Any other higher order integration equations would have needed more storage capacity than is available.

Another method which can be used to improve accuracy and eliminate any excessive scatter is to apply an averaging process to the results. This can be accomplished by one of two means. One approach is to stop the calculating and plot the results in the region of the scatter, and then "fair out" the inaccuracies. This has to be done for the functions and their derivatives. The faired values are then used as new initial values, and the computations are continued until excessive scatter appears again. The other approach is essentially the same with the difference that the "fairing out" is incorporated in the instruction cards as part of the computing procedure. The method of least squares can be used as the means for "fairing out" scatter. The former method was used because it was felt that changing the instruction cards was not warranted.

Since the first method for the solution of the second-order non-linear simultaneous differential equations used a large number of instruction cards, an investigation was made to see if a much simpler method could be used which would give the desired accuracy and decrease the computing time. Computations for the second method were carried out for the same two intervals as for the first method. These results are shown in Figures 4, 5 and 5A. Although decreasing the



FIGURE 3



FIGURE 4









interval improved the final results, the simple integration equations were still inadequate to give accuracy throughout the whole range of the independent variable t. The error is inherent in the integration formula used, so that attainment of any great accuracy is impossible. Then, too, this being a non-iterative process, any error accumulates until the results become of no significance. The second method was card programmed in an instruction deck of 110 cards. As in the first method, only one data load card for the initial condition was necessary.

SUMMARY

Two methods for the solution of second-order non-linear simultaneous equations are presented. They are, basically, step-by-step solutions where values of the dependent variables are computed at equally spaced values of the independent variable. One of the methods is an iterative process of predicting and correcting the dependent variables and their derivatives at a chosen value of the independent variable until convergence is attained. The other is a much simpler method which involves no iterative process, but whose accuracy wholly depends on the interval chosen.

A derivation for the equations of motion for the case of a towed airplane is presented to show how the physical system can be interpreted mathematically. Reasons are also given for using a particular CPC-function panel and IBM computer.

The final results showing the effects of the interval chosen and the methods used are presented as time-history plots of the dependent variables θ and ϕ . From these plots it can be surmised that in order to obtain the desired accuracy it is necessary to use the iterative method which employs at least three-point integration formulas. But then, too, it may be necessary to choose a smaller interval for the independent variable or use an averaging process to "fair out" the scatter of results. Greater accuracy can be attained if higher order integration equations are used.

REFERENCE

1. W. E. MILNE, Numerical Calculus (Princeton, N. J.: Princeton University Press, 1949).

DISCUSSION

Mr. Sheldon: I would like to make a remark about stability. I know of three types of instability which can occur when you solve a differential equation step by step.

One is that the original differential equation may have solutions which tend to infinity in the direction in which you are integrating, when you want a solution which tends to zero. Thus, any small error you make will increase exponentially, and eventually obscure the solution.

Another type of instability is that which is obtained when you replace the differential system by a difference system of higher order than the differential system. In this case, the difference system may have solutions which have nothing whatever to do with the differential system, and these may increase exponentially.

The third type of instability is the type you sometimes get in partial differential equations, when you have hyperbolic and parabolic systems.

You have to be very careful to make sure that the instability you find as a result of your solution is not just due to your numerical technique. It may not really be inherent in the physical problem. Mr. Carter: I would like to add to Mr. Sheldon's remarks that the stability may depend upon different difference equations that you might choose.

Mr. Von Holdt: At Los Alamos we solve quite a few differential equations and simultaneous systems. We usually start them off with the Runge-Kutta method of smaller intervals, which involves four iterations to get fourth-order accuracy. Then we switch to the Milne method. We also have a little trick; we know that each of the predicted and corrected values has a certain value, which is a particular coefficient times the fourth derivative. If the interval is assumed small enough so that these fourth derivatives, at two intermediate points somewhere, are approximately the same point or the same value is obtained, then the error of the corrected term is 1/29th in the fourth derivative. This error is the difference between the predicted and corrected term and can be added to the corrected term to get a still more accurate one. If the estimate of the error gets too large, we cut down to a smaller interval, and usually have a Runge-Kutta deck right at hand to do this. When the interval has been cut, we start off again with about four points by the Runge-Kutta method and switch into the Milne method in a smaller interval. To double the interval is much simpler, of course.

Dr. Herget: Taking a look at the number of oscillations that there were in the solution, I am quite confident that 200 points are sufficient to do a good job. The intuition with which I approach these things is concerned with the radian measure of an interval of time, and what is needed is something of the order of magnitude of between a tenth and a hundredth of a radian for one of these oscillations; so I am confident that 2,000 points were not needed, as against 200.

I am always irritated by mathematicians who teach their students Simpson's rule and point out how tremendously accurate it is, because it is accurate to 1/90th of the fifth difference. What should be pointed out is that there always exist completely general formulas, which were known to Gauss and to which all those who highly respect Gauss in every other field never pay any attention!

Aside from those remarks, I would like to make another: In astronomy it is generally a theorem that the best way to solve problems is in rectangular coordinates, the reason being that $x = r \cos \theta$ and $y = r \sin \theta$ and it is not necessary to look into any trigonometric functions if you deal with x and y.

Mr. Madden: We had a problem a good deal like this which gave rise to equations like these, and we had considerable trouble working at them blindly. We found that one of our angles was oscillating from minus 50,000 radians to plus 50,000 radians in a space of five hundredths of a second. Because of the nature of the problem, we were unable to cut down the interval; we had to keep our interval at this .05, and in the light of Dr. Herget's remarks we found that when we converted to rectangular coordinates, this oscillation dropped out immediately, and apparently a solution was quickly obtained.

Matrix Inversion and Solution of Simultaneous Linear Algebraic Equations with the IBM Type 604 Electronic Calculating Punch

GEORGE W. PETRIE, III

International Business Machines Corporation

×

CURRENT computational literature is replete with many descriptions of methods for matrix inversion (see References). Gutshall⁶ points out the necessity for further statistical study of types of matrices which may be subjected to numerical inversion and suggests, also, a comparative study of known methods of inversion. Dantzig² demonstrates in his simplex method a recurring need for obtaining the inverse of certain matrices to compute optimum programs. In fact it is his request for a simple, fast procedure for inversion of large order Leontief-type matrices that led to the technique presented in this paper. The procedure, while using the standard elimination method, 3,4,9,11,12,15 allows for simple and continuous processing of cards through a reproducer, the IBM Type 604 Electronic Calculating Punch, an accounting machine (for checking), and a second reproducer. This cycle is repeated N times to invert an Nth order matrix. The size of the matrix is not limited, and the four machines are operated continuously without change of control panels.

Consider the matrix equation, AX = I, as describing a system of simultaneous linear equations in the x's. By simple operations (multiplying equations by appropriate constants and combining with other equations) it is possible, provided A is non-singular, to reduce the system to one in which the coefficient of X is I. In this case, the right-hand member of the matrix equation is reduced to the inverse of A. Thus, in the elimination method one starts with an augmented matrix composed of the original matrix, A, and the unit matrix, I. As the original matrix is reduced to the unit matrix, the original unit matrix is simultaneously reduced to the inverse of the original given matrix. This reduction may be carried out one column at a time. After k - 1 columns have been reduced, the procedure involves dividing the kth row by a_{kk} (to obtain 1 in the kth column) and then subtracting the product of this result with the appropriate constant, a_{ik} , from the *i*th row so as to obtain zero in the

kth column of the ith row. This latter subtraction is carried out for all $i \neq k$. Columns of the unit matrix numbered greater than k are unaffected. The kth column is, in general, completely changed. Thus, N^2 elements enter the computation at any one cycle. In the present procedure the kth column (or vector) of the left-hand part of the reduced matrix is not written, because its value is known automatically. Instead, it is replaced by the kth column (or vector) of the right-hand part of the augmented matrix after this step in the reduction. By selecting the appropriate formula to use (as listed in step 3 of the next section), the 604 actually brings in vectors from the unit matrix as needed. The same machine is used to allow the matrix elements to be reordered after each step in the elimination so that the new pivotal row is on top and the new pivotal column is at the left. Thus, intermediate row and column sorting is entirely eliminated. It is through the elimination of all collating and this type of sorting that the present more rapid machine procedure becomes possible.

The accounting machine is used for checking purposes. The checking operation must be looked upon as occupying one step in the continuous flow of cards through the four machines. Processing would not be appreciably accelerated if this check were omitted, because all machines are running simultaneously. It is important to point out, however, that the checking procedure augments the original N^2 cards to $N^2 + N$ cards.

MACHINE PROCEDURE

The original Nth order matrix a_{ij} is punched on N^2 cards, one element together with the identifying *i* and *j* to a card. An additional row of check sums is appended to the original matrix. These values are defined by

$$s_j = 1 - \sum_i a_{ij}$$

A ten-digit fixed decimal system is used (two integers and eight decimals). Data are punched as follows:

Columns	Data
1-3	Number of row (i)
4-6	Number of column (j)
10-19	Value of element (a_{ii})
20-29	Value of element (a_{11}) —First card only.

In all cases minus signs are carried as X-punches over the right-hand digit. In addition, elements of the first row are identified with X and Y punches in column 1 (denoted XY1) to indicate the pivotal row; elements of the first column have XY7 punches to indicate the pivotal column. Elements of the row of s_j are identified by XY13 punches. The 521 punch unit is used to supply subsequent X and Y punches as needed.

After the cards are prepared and sorted to row within column, the first column is gang punched as listed in step 6 below. After this preliminary step, the following six steps are repeated N times to obtain the inverse:

1. Reproduce the values of *i* (columns 1-3) and a_{ij} (columns 10-19) from the leading N+1 cards (with identifying XY7) into each N+1 of the remaining N^2-1 cards.

Read (XY7)	· · · ·	Punch (NXY7)			
Digits of 1-3	-	1-3			
Digits of 10-19		20-29			

This means that an element in the *i*th row of the leading column will now appear on the same card with all other elements of the *i*th row of the matrix. As a variation, the value of a_{ij} to be reproduced may be read from either columns 10-19 of XY7 cards or from columns 20-29 of NXY7 (no XY7) cards which have already passed through the punch side of the reproducer for this step. Thus, in practice, the operator starts with the N+1 cards bearing the designator XY7 in the read feed and all others either in the punch feed or immediately available thereto. After reproducing the values from the N+1 cards, both stackers are emptied. The 2(N+1) cards are placed together in the read feed and 4(N+1) cards are next obtained from the two stackers. As soon as a sizable group of cards is generated, one may take the cards from the read stacker to the next operation process at the 521-604 while continuing to reproduce values from the cards which came from the punch stacker of the reproducer.

In both cases a comparison check is carried on all reproductions. In addition, the reproduced value of i (columns 1-3) is checked for double punches. The reason for the reproduction of i is apparent at step 3 following.

2. Place a blank card having a different distinguishing color and XY1 punches after the last card processed in step 1.

3. Calculate values of b_{ij} and t_j according to the following formulas. For complete generality assume that k-1steps in the inversion process have been completed, and that the matrix is in order of row within column as given in the following array:



Values of
$$b_{ij}$$
 and $t_j = 1 - \sum_i b_{ij}$ are computed by:

XY-conditions

$b_{k,k} = \frac{1}{a_{k,k}}$		pivotal el	ement XY1, XY7
$b_{i,k} = - rac{1}{a_{k,k}} \cdot a_{i,k}$,	for $i \neq k$	pivotal colu	ımn NXY1, XY7
$b_{k,j}=+\frac{a_{k,j}}{a_{k,k}},$	for $j \neq k$	pivotal	row X Y 1, NXY 7
$b_{i,j} = a_{i,j} - \frac{a_{k,j}}{a_{k,k}} \cdot a_{i,k}$,	for $i \neq k$,	$j \neq k$	NXY1, NXY7
$t_k = -\frac{1}{a_{k,k}} \cdot s_k$	check	, pivotal colu	ımn NXY1, XY7

$$t_j = s_j - \frac{a_{k,j}}{a_{k,k}} \cdot s_k$$
, for $i \neq k$ check, other NXY1, NXY7.

One observes that the computation for t_j is identical with that for $b_{i,j}$ where $i \neq k$. The values computed are punched directly into columns 30-39 on all NXY1 cards. For XY1 cards, however, the values are stored and punched on the following XY1 card. This results in the array of values shown in Table I.

Notice that no b is punched on the first card. The additional card of step 2 was added to obtain the last value. In addition to the above calculations, the following operations are performed simultaneously on the 521:

- a. offset gang punch digits of columns 4-6 into 7-9,
- b. offset gang punch XY1 into XY4,
- c. gang punch XY7 into XY7 on the XY1 card following an XY7 card,

```
TABLE I
               b_{k,k}
                                             b<sub>k,j-1</sub>
                                                                      b_{k,N-1}
                                                                                       b_{k,N}
                                                                                                                b<sub>k,k-2</sub>
                                                                                                                                   b_{k,k-1}
                                                              . . .
               b_{k+1,k+1}
                                                                                       b_{k+1,1}
                                             b_{k+1,j}
                                                                       b_{k+1,N}
                                                                                                                b_{k+1,k}
                                                             . . .
                                                                                      b_{i,1}
b_{i,k}
                                                                      b_{i,N}
                                                                                                                b_{i,k-1}
b<sub>N.k</sub>
                                                                                                                t_{k-1}
t_k
b_{1,k}
                                             b_{k-1,j}
                                                                             1. N
                                                                                      b_{k-1,1}
                                                                                                                b_{k-1,k-1}
```

 d. emit XY10 into cards following the XY1 which follows an XY7; stop the XY10 emission after the next XY1 card.

It will be observed that these operations result in a column identification of the last card (the card added at step 2). As yet there is no row identification, but this will be supplied by step 1 of the following cycle. If needed for a visual check, it is easy to remember that the unidentified row is one greater than the column number which is already on the card. It would also be possible to prepunch the row identification before step 2. This would involve a simple consecutive-number deck.

4. Tabulate and list N lines with the following data:

k (read from first card of column)



This tabulation is used to check the accuracy of the previous steps. The last value should differ from -1 by no more than an acceptable rounding error. One has a choice here of checking each column separately or else the entire matrix as a whole.

5. Remove the first card, for which no further use is made. Take the next N+1 cards and place them at the end of the deck. This leaves the cards in the following array:

$b_{k+1,k+1}$	• • •	$b_{k+1,\vec{N}}$	$b_{k+1,1}$	•••	b _{k+1,k}
•		•			
•		•			
•		•			
$b_{N,k+1}$					
t_{k+1}		t_n	t_1		t_k
$b_{1,k+1}$					
•		•			•
•		•	•		•
•		•	•		•
$b_{k,k+1}$	• • •	$b_{k,N}$	$b_{k,1}$	• • •	$b_{k,k}$

6. Reproduce into a new set of cards (or tumbled cards already half utilized).

Punch	Read
Digits of 1-3	1-3
Digits of 7-9	4-6
30-39	10-19 on all cards and also into
	20-29 from XY4, XY10
XY4	XY1
XY10	XY7
XY16	XY13

At the same time, gang punch from columns 10-19 of XY1, XY7 card into columns 20-29 on all NXY1, XY7 cards.

The new set of cards will be in such order that the pivotal row is on top, the pivotal column is to the left, and elements of both are marked with proper identifying X and Y punches. Thus, these cards are ready to process through step 1 of the following cycle. After N cycles the values obtained at step 5 give the required inverse.

Solution of Equations

To solve a set of linear simultaneous algebraic equations, the procedure outlined above is only slightly modified. The same control panels are used without any changes or additional wiring. The original matrix of coefficients is augmented by the vector of constants and row of negative sums of column elements for checking. Thus, one starts with a matrix of $(N+1)^2$ elements. After step 1 on each cycle, the first N+1 cards (XY7) are discarded. Therefore, the cards, which pass through the 604, number N(N+1), $(N-1)(N+1), (N-2)(N+1), \dots, 1(N+1)$. At step 4 the check value is approximately $-1, -2, \ldots, -N$ on the successive steps and should differ from these negative integers by no more than an acceptable rounding error. At step 5 the first card is removed. No other rearranging occurs. Finally, one obtains the solution of the simultaneous equations as a vector accompanied by a check sum. This procedure should take approximately half as long as the inversion procedure.

SUMMARY

The inversion procedure outlined above is believed to be faster and easier to perform than other methods now in common practice. Further inquiry should be made of the applicability of this procedure to a floating decimal calculation; also to matrices involving complex numbers.^a

For large-order matrices this procedure may be used to process cards continuously from one machine to another, keeping all machines in operation simultaneously. For small-order matrices, the number of cards is insufficient to keep all machines running. In this case, all cards would be processed on one machine at a time. Several matrices could be inverted at the same time.

It is possible to perform steps 1 and 6 on a single IBM Type 519 Document Originating Machine with only one control panel, if a control punch has been emitted onto the cards at step 3. This modification would be useful for handling low-order matrices.

If desired, it is also possible to handle a row check similar to the column check. In this case, the check sum is defined by:

$$s_i = -1 - \sum_j a_{ij} \; .$$

The Appendix contains the program for the 604 and the wiring diagram for the 521.

ACKNOWLEDGEMENTS

The writer wishes to express his acknowledgements to Dr. George B. Dantzig, U. S. Air Force Comptroller's Office; Dr. Jack Sherman, The Texas Company; and Miss Caroline J. Cooper, IBM. Dr. Dantzig requested the development of a fast method of inversion of Leontief-type matrices. He also suggested the technique whereby the unit matrix is generated vector by vector as needed, resulting in the equations for $b_{i,j}$ in step 3. Dr. Sherman developed the reproduction used in step 1 and communicated the concept to the author in a paper soon to be published concerning matrix inversion with the IBM Type 602-A Calculating Punch¹⁴. The assistance of Miss Cooper was realized in aiding with the programming and executing the entire plan into successful operation.

REFERENCES

- 1. JUSTUS CHANCELLOR, JOHN W. SHELDON, and G. LISTON TATUM, "The Solution of Simultaneous Linear Equations Using the IBM Card-Programmed Electronic Calculator," Proceedings, Industrial Computation Seminar, September, 1950 (IBM), pp. 57-61.
- GEORGE B. DANTZIG, "Maximization of a Linear Function of Variables Subject to Linear Inequalities," Chapter 21, Activity Analysis of Production and Allocation (Wiley, 1951). 2.
- 3. PAUL S. DWYER, Linear Computations (Wiley, 1951), pp. 190-191.
- 4. R. A. FRAZER, W. J. DUNCAN, and A. R. COLLAR, Elementary
- Matrices (Macmillan, 1947), pp. 119-120.
 HERMAN H. GOLDSTINE and JOHN VON NEUMANN, "Numerical Inverting of Matrices of High Order II," Proceedings of the American Mathematical Society, Vol. 2, No. 2 (April, 1951), pp. 198-202. pp. 188-202
- WILLIAM D. GUTSHALL, "Practical Inversion of Matrices of High Order," Proceedings, Computation Seminar, December, 6. 1949 (IBM), pp. 171-173.
- 7. HAROLD HOTELLING, "Some New Methods in Matrix Calcula-tion," Annals of Mathematical Statistics, Vol. 14, No. 1 (March, 1943), pp. 1-34.
- HAROLD HOTELLING, "Further Points on Matrix Calculation and Simultaneous Equations," Annals of Mathematical Statistics, Vol. 14, No. 4 (December, 1943), pp. 440-441. KAISER S. KUNZ, "Matrix Methods," Proceedings, Computation Seminar, December, 1949 (IBM), pp. 37-42. 8
- IRVING C. LIGGETT, "The Gauss-Seidel Method of Solution of Simultaneous Linear Equations," Proceedings, Industrial Com-putation Seminar, September, 1950 (IBM), pp. 62-65. 10
- 11. JOHN LOWE, "Solution of Simultaneous Linear Algebraic Equations Using the IBM Type 604 Electronic Calculating Punch," Proceedings, Computation Seminar, December, 1949 (IBM), pp. 54-55.
- 12. WILLIAM EDMUND MILNE, Numerical Calculus (Princeton University Press, 1949), pp. 15-29.
- JOHN VON NEUMANN and H. H. GOLDSTINE, "Numerical Inverting of Matrices of High Order," Bulletin of the American Mathematical Society, Vol. 53 (November 11, 1943), pp. 1021-1099.
 JACK SHERMAN, "Computations of Inverse Matrices by Means of IBM Machines," Applied Science Department Technical Newsletter No. 3 (IBM).
- FRANK M. VERZUH, "The Solution of Simultaneous Linear Equations with the Aid of the 602 Calculating Punch," Math-ematical Tables and Other Aids to Computation, III, No. 27 15. (July, 1949), pp. 453-462.

APPENDIX

NOTE ON PROGRAM SUPPRESSION: Programs are suppressed without balance test through calculate selectors as follows:

Programs	Calculate Selector
1-37	<u>6</u> T
38-43	2N
44-56	6N

On programs marked suppress PB, the wiring is for suppress on positive balance through calculate selector 6N. Suppress without balance test is wired through 6T. Corresponding wiring occurs for NB.

^aThe author understands that members of Dr. Dantzig's group are investigating the problem of which order of elements allows of the determination of the most accurate inverse. Certainly more work is necessary in this area.

NO.	SUP.	SEL.	FACTOR STORAGE		QUOT.	COUNTER	GENERAL STORAGE		CONDITIONS				
	See Nore		3 (6-4)	2	4			2 (8-6)	3 (8-6) 4	XY I XY 7	x y 1 N 5 Y 7	NXY I XY 7	NXYI NXY7
READ			0 Cors 20-24	L		Cors 25.29		Cors 10-11	18,19 000000	L			
1			EMIT J				RI-4			<u> </u>	X	x	
2			RO	BAL TE	57		RI	+		X	X	<u>×</u>	
3	· · ·		20				RR			X	X	X	
-			KU_			PO	DI				×	×	
6						RI	A 4		RO	X	x	x	_
7	PB						RR		RI	X	x	x	
8	NB						R0 3		RI	x	x	x	
_ 9	NB						RI-3		RO	x	X	x	
_10	NВ		RI			ļ	RR			<u> </u>	X	x	
11							RI-	RO		×	X	X	
12	PB						RI		RO	<u> </u>	X	<u> </u>	
13	NB			L	L	<u> </u>	RI 3		RO	X	X.	X	
14	DD		FU NE	G. BAL I	FL	1	PIC	PO				X	
16	NR						RI 4	PAD			X		
17	PB					RO	RI 4			x	x	x	
18	NB					RO	RI 2			X	X	x	
19			DIVIDE						RO	x	x	x	
_20							RR	RI		x	x	x	
21				ļ			RI 6	RO	· · · · · · · · · · · · · · · · · · ·	x	X	x	L
22					1	RO		RI		LX_	X	×	
			DIVIDE	I	r				RO	X	X	X	
24	11-		00	<u> </u>		1				<u> X</u>	X	X	
25	NB		DT				PP			⊢× , , ,	X	× v	
27	NP	<u> </u>	RA			1	RIA	-		1 V	×	T v	1
28			RI			RO	<u>, , , , , , , , , , , , , , , , , , , </u>			x	x	x	
29	NB		DIVIDE						RO	X	x	X	
30							RR			x	x	x	
31	NB		MULT-		·····			RO		X	x	x	L
32	NB						RR 6		RI	x	X	x	<u> </u>
33	NB						RI	_	RO	X	X	X	
34	NB				├──		<u>RR 2</u>		RI	X	X	X	
	NB	5N 7N8A				+	RI		RO	X	X	<u> </u>	
	NB	NTT	· · · ·				RI_		RO	X	X		-
	NR	NTN					RI-		RU		X		
	NR	TNN					RI-	-	RO		- <u>^</u> -	x	
	NB	7 T T					RI -		RO			X	
	NВ	TNT					RI		RO			x	
	NB	TTN					RI		RO	ļ		x	
36		5N 7N8	RO				RI			x	x		
<u> </u>		NTT	RO				RI			x	X		
		NNT	RO				RI-			X	x		
		NTN	RO			+	RI-			X	X		
		TTT	RO				$R_{I} =$	+		<u> </u>		X	
		TNT	RO				RI					x	
		TTN	RO				RT					x	
37		5N 7N 8N	,				R1 6	RO		x	x		
		NTT					RI 6	RO		x	X.		
		NNT			L	ļ	RI-6	RO		X	x		
		NTN				ļ	RI-6	RO		X	<u>×</u>		
		TNN					RI-6	RO		<u> </u>		X	
		TTT.				<u>↓</u>	RF-6	RO				Ă	
		TT					RI 6 RI 4	RU				×	
38		IIN			····-		ROF	RU PT		v		×	
39						RI	RR			X	Ŷ		_
40					RO		RI			x	x		
41				RO			RI 6			x	x		
42					RI	RO				x	x		
43				RI_				RO		x	x		
44			MULT -		RO	-							<u>x</u>
45							RI 4		RO				X
46							RR 6		RI				X
47			A	00			RI		RO				X
48			PO	KU.		DT		+					×
 50			MIL +-		Rn	A 1		+					-4-
51			1/2 An	UST			RI 1	1					x
52			~				RR 4		RI				x
53							RI		RO				x
54							RI 3	RO					x
55				RO			<u> </u>	RI 3		-	L		X
56			MULT	-				RO					X

Type 604 Programming


Type 521 Wiring

DISCUSSION

Dr. Aroian: Could you tell us something about the roundoff error? You started with eight significant figures in a 40-by-40 matrix. What do you have left at the end?

Dr. Petrie: May I quote von Neumann and Goldstein? I believe they state if you start with a 19-by-19 matrix you lose eight digits. That appears to be the worst case. People who are inverting high-order matrices seem to have better luck. I do not know whether it is a matter of luck or something that is not yet known.

Dr. Brown: What happens when you get eleven digits?

You say the machine stops. Then what do you do?

Dr. Petrie. In this case, it would be necessary to perform some scaling of the information; that is, divide one column by a power of 10, or something of that type. However, it is decidedly not a machine procedure.

Mr. Schlieser: Could you give me an estimate of the time required to invert, say, a 30-by-30 matrix, using your method?

Dr. Petrie: This method is quite new and has not been fully tested. I wish I could give you specific figures of what has been done. Theoretically, the 30-by-30 matrix would involve approximately 900 cards which can be processed in

SEMINAR



DETAIL OF TYPE 604 WIRING

any one machine in nine minutes. We must repeat that process thirty times.

Professor Verzuh: Do you use just one 604 control panel for this operation?

Dr. Petrie: Yes.

Professor Verzuh: With regard to checking, our experience indicates that it is well to include not only check rows but check columns. When you do encounter an error you have an indication of the spot that is in error.

Mr. Tillitt: Sometimes you might get a zero divisor. I wonder if it would be worth while to have a 604 "look" every time there is division to see if division is being used.

Dr. Petrie: This is programmed in this particular control

panel. There is a check to see immediately if we do have a zero divisor.

Dr. Brinkley: You commented that the choice of which of the four metric operations you have is determined by X punching in the card. At what stage in the procedure are those X punches imposed and how is that done?

Dr. Petrie: Originally the matrix is prepared with the first column and the first row with significant X punches. Each time in the reduction in passing through the 604, new X punches are introduced in the row following the pivotal row and into the column following the pivotal column. The X punches at step 6 are then reproduced into the new set of cards, where they will serve as instructions to the 604 in the following cycle.

111

The Determination of Eigenvectors and Eigenvalues of Symmetric Matrices*

EVERETT C. YOWELL

Institute for Numerical Analysis, National Bureau of Standards

×

THERE ARE several basic properties of matrices and vectors which underlie all the common numerical methods of finding eigenvectors and eigenvalues. It might be advisable to state these explicitly before examining any of the numerical methods.

The most fundamental property is that a vector remains a vector under the operation of multiplication by a matrix. If we consider an *n*-dimensional vector as a one-column matrix of order *n*, then by applying the definitions of matrix multiplication we can show that the result of multiplying an *n*th-order square matrix by an *n*-dimensional vector is another *n*-dimensional vector. We may write this symbolically as

$$Ay = z . (1)$$

(In this and all subsequent equations, capital letters represent square matrices, and small letters represent vectors. The letter x will be reserved for eigenvectors, and the symbol λ will be used only for eigenvalues.)

Secondly, this process of matrix multiplication is distributive. That is

$$A(y+z) = Ay + Az .$$
 (2)

Thirdly, the process of matrix multiplication is associative. If A and B are two square matrices, then

$$A(By) = (AB)y . (3)$$

In particular, if k is a constant,

$$A(ky) = k(Ay) . \tag{4}$$

These first three properties hold for all matrices and all vectors as long as the process of multiplication is definable.

Let us now define an eigenvector. If the resulting vector z of equation 1 is parallel to the initial vector y, then y is an eigenvector of the matrix A. Or symbolically, if

$$Ax = \lambda x, \tag{5}$$

where λ is a constant, then if x is not of zero length, x is an eigenvector, and λ is its associated eigenvalue.

From this definition, we can show that every matrix of order *n* has precisely *n* eigenvalues. Let us rewrite equation 5 by transferring the λx term on the right-hand side of the equation to the left side. This gives

$$(A-\lambda I)x=0.$$

Here I is the unit matrix, the matrix with 1's on the main diagonal and 0's everywhere else. In this form, it is easy to see that the matrix equation can be written as a system of nsimultaneous equations in n unknowns. (We consider the components of x as unknown and λ as a known parameter.) Further, this system is homogeneous. Now a homogeneous system of equations can have a non-zero solution if and only if the determinant of the coefficients is zero. This condition is

$$|A - \lambda I| = 0$$

If we expand this determinant, we obtain a polynomial of degree n in λ , and the roots of this polynomial are the values of λ for which eigenvectors x can be found. Since a polynomial of degree n has precisely n roots, it follows that there are precisely n eigenvalues λ of the matrix A.

It can also be shown that associated with each eigenvalue of a symmetric matrix is an eigenvector. This is more complicated than the demonstration that there are n eigenvalues, because of the possibility of two or more eigenvalues being equal. If two eigenvalues are equal, then it is not possible to specify the two eigenvectors uniquely, but only the plane in which they lie. For any linear combination of the two will also satisfy the definition of an eigenvector. Suppose we have a case where $\lambda_1 = \lambda_2$ and have by some means or other found two vectors x_1 and x_2 which satisfy the eigenvector definition. Then we know that $Ax_1 = \lambda_1 x_1$ and $Ax_2 = \lambda_2 x_2$. Let us now take a linear combination of x_1 and x_2 ,

$$z = c_1 x_1 + c_2 x_2 \; .$$

Applying our matrix, we obtain

 $A(z) = A(c_1x_1 + c_2x_2)$ $= c_1Ax_1 + c_2Ax_2$ $= c_1\lambda_1x_1 + c_2\lambda_2x_2 .$

^{*}The preparation of this paper was sponsored (in part) by the Office of Air Research.

Since
$$\lambda_1 = \lambda_2$$
, this reduces to

$$A(z) = \lambda_1(c_1x_1 + c_2x_2) = \lambda_1z .$$

Hence, z is just as good an eigenvector as x_1 or x_2 .

From our basic definition of an eigenvector, it is apparent that only the direction of the vector is important. Its length is not determined. Therefore, the length can be chosen in the most convenient manner. This usually means either choosing the length of the vector to be unity, or choosing the largest component to be unity.

What we can say about the length is how the length changes under the matrix multiplication. If we define |y| to be the length of y and define λ_I as the absolutely largest eigenvalue and λ_N as the absolutely smallest eigenvalue, then under multiplication by the matrix A we can show that

$$\lambda_N \leq \frac{|z|}{|y|} \leq \lambda_I$$
, where $z = Ay$ (6)

Another property of the set of eigenvectors of any matrix is that, if there are n eigenvectors, they span the space of the matrix—that is, they can be used as a coordinate system to express any other vector of the same dimensionality. If we let x_i be the *i*th eigenvector, then any arbitrary vector y can be written as a linear combination of the eigenvectors; That is,

$$y = \sum_{i=1}^{n} c_i x_i . \tag{7}$$

There is one last property of the eigenvectors. For a symmetric matrix, these vectors are mutually orthogonal. That is,

$$x_i \cdot x_j = 0 \quad (i \neq j) \tag{8}$$

Now that we have briefly surveyed the mathematical theory of eigenvectors and eigenvalues, we can examine some of the numerical methods for finding them. There are three, namely: the power method, the method of the characteristic polynomial, and the gradient method that we shall now discuss.

The first of these, the power method, is perhaps the best known method for finding eigenvectors. Mathematically, it is based on these three facts:

(A)
$$y = \sum_{i=1}^{n} c_i x_i$$
.
(B) $A(y+z) = Ay + Az$.
(C) $Ax_i = \lambda_i x_i$.

Suppose we choose any arbitrary vector y_0 . By statement (A), this can be written as a linear combination of the eigenvectors of the matrix. If we multiply this vector by the matrix A, we get another vector y_1 as the result. Now by

statement (B), we see that multiplication of y_0 (which is a linear combination of the eigenvectors) by the matrix A is equivalent to taking the same linear combination of the products of the eigenvectors by the matrix. By statement (c), we see that each product of eigenvector and matrix is equal to the product of the eigenvector and eigenvalue. Putting this symbolically,

$$y_0 = \sum_{i=I}^N c_i x_i$$
$$Ay_0 = y_1 = \sum_{i=I}^N c_i Ax_i$$
$$Ay_0 = y_1 = \sum_{i=I}^N c_i \lambda_i x_i .$$

We now perform the operation again on the vector y_1 . The resulting vector y_2 can, by the same arguments, be shown to be equivalent to

$$\sum_{i=I}^N c_i \,\lambda_i^2 x_i \; .$$

After a large number of iterations, our iterated vector y_n can be written

$$y_n = \sum_{i=1}^N c_i \,\lambda_i^n x_i \;\;.$$

It is now necessary to show that, for sufficiently large values of n, y_n approaches the absolutely largest eigenvector x_I . (The short phrase "largest eigenvector" will be used to mean the eigenvector associated with the largest eigenvalue; the phrase by itself has no meaning as the length of the eigenvectors is arbitrary.) Let us divide both sides of the equation by λ_I^n . This gives

$$\frac{y_n}{\lambda_I^n} = c_I x_I + \sum_{i=II}^N c_i \left(\frac{\lambda_i}{\lambda_I}\right)^n x_i \; .$$

Since λ_I is the eigenvalue of greatest absolute value, the fraction λ_i/λ_I lies between +1 and -1. Hence, sufficiently large powers of the fraction approach zero arbitrarily closely. Thus, for sufficiently high powers of n, y_n/λ_I^n arbitrarily closely approaches the vector $c_I x_I$. And we do not have to worry about the constants λ_I^n and c_I , for we have seen that an eigenvector is defined in terms of its direction alone, and any multiplicative constant is admissible.

It is possible that the original trial vector was chosen in such a manner that c_I was zero. In this case, the power method may converge to the second largest eigenvector

rather than the largest. The probability of such an unfortunate choice is small, but it has occurred in our experience at the Institute for Numerical Analysis.

The two disadvantages of the power method are apparent from this discussion. The first is the need for normalization. We saw that the *n*th iterate of *y* became, for all practical purposes, $\lambda_I^n x_I$. Now if λ_I is greater than 1, the length of y_n will become very large as *n* increases, thus eventually leading to an overflow. And if λ_I is less than 1, the length of y_n will become very small as *n* increases, thus eventually leading to a zero vector. It is necessary, then, to normalize the length of the vector after every few iterations. For convenience, we have generally normalized the largest component to unity at each step of the iteration.

The second disadvantage is the possible slow speed of convergence of this method. The fraction $(\lambda_i/\lambda_I)^n$ approaches zero as *n* increases, but for values of λ_i/λ_I near to 1, *n* must be very large in order to reduce the contributions of the smaller eigenvectors to a preassigned amount.

The following table gives the smallest exponent *n* such that the *n*th power of a given ratio $r = \lambda_{II}/\lambda_I$ will be less than a preassigned small number ϵ . It will serve as a rough guide to the number of iterations needed to change an arbitrary vector into an eigenvector of a given degree of purity when some knowledge is available as to the relative sizes of the two absolutely largest eigenvalues.

r	10-4	10-5	10-8
0.40	11	13	21
0.50	14	17	. 27
0.60	19	23	37
0.70	26	33	52
0.80	42	52	83
0.90	88	110	175
0.95	181	226	361

From this table, it is apparent that the power method converges very slowly if the ratio of the two absolutely greatest eigenvalues is much greater than 0.50.

This slow rate of convergence makes the power method impractical unless some method is available for accelerating the convergence. Fortunately, such methods have been devised. Let us consider the case of two close eigenvalues. By one acceleration scheme¹ three successive iterates, $A^n y$, $A^{n+1}y$, and $A^{n+2}y$ are formed, choosing *n* sufficiently large so that we are reasonably sure $A^n y$ is a linear combination of only two eigenvectors. It should be noted that $A^{n+1}y$ and $A^{n+2}y$ should not be normalized in length. Now choose any two vectors *s* and *t* and form the scalar products

$a_{12}=s\cdot A^n y$	$a_{13} = t \cdot A^n y$
$a_{22} = s \cdot A^{n+1}y$	$a_{23} = t \cdot A^{n+1}y$
$a_{32} = s \cdot A^{n+2} y$	$a_{33}=t\cdot A^{n+2}y.$

COMPUTATION

Then the two eigenvalues that we are seeking are the roots of the determinantal equation

$$egin{array}{cccc} 1 & a_{12} & a_{13} \ \lambda & a_{22} & a_{23} \ \lambda^2 & a_{32} & a_{33} \end{array} = 0 \; .$$

This can be demonstrated to be true as long as $A^n y$ is a linear combination of only two eigenvectors. In case the determinantal equation yields zero coefficients, a different choice of s and t will give a solution in nearly all cases.

This scheme can be easily generalized to cases where more than two eigenvalues are close together. If we consider the case of three close eigenvalues, we find that we need four successive non-normalized iterates, $A^n y$, $A^{n+1} y$, $A^{n+2} y$, and $A^{n+3} y$. We choose three arbitrary vectors s, t, and u. We form the twelve scalar products

$a_{12}=s\cdot A^n y$	$a_{13} = t \cdot A^n y$	$a_{14} = u \cdot A^n y$
$a_{22} = s \cdot \mathcal{A}^{n+1} y$	$a_{23} = t \cdot A^{n+1}y$	$a_{24} = u \cdot A^{n+1} y$
$a_{32}=s\cdot A^{n+2}y$	$a_{33} = t \cdot A^{n+2}y$	$a_{34} = u \cdot A^{n+2} y$
$a_{42} = s \cdot A^{n+3}y$	$a_{43} = t : A^{n+3}y$	$a_{44} = u \cdot A^{n+3}y$

and solve the following determinantal equation for the three close values of λ :

$$\begin{vmatrix} 1 & a_{12} & a_{13} & a_{14} \\ \lambda & a_{22} & a_{23} & a_{24} \\ \lambda^2 & a_{32} & a_{33} & a_{34} \\ \lambda^3 & a_{42} & a_{43} & a_{44} \end{vmatrix} = 0$$

If one is using the power method, this acceleration scheme is only used when some close eigenvalues slow the rate of convergence down below a practical level. However, the closeness of the eigenvalues is not necessary for the use of this scheme. In its most general form, this acceleration method leads directly to the method of the characteristic polynomial.

The method of the characteristic polynomial, briefly, consists in setting up a polynomial of degree n whose roots are the eigenvalues of the matrix. There are several ways of setting up this polynomial. One follows the acceleration scheme outlined above. Since we are looking for all n eigenvalues, we need n + 1 successive non-normalized iterates and n arbitrary vectors. As before, we form the scalar products

a_{12}	$= s \cdot y$	a_{13}	= t	·у	• • •	$a_{1,n+1}$	$= z \cdot y$
a_{22}	$= s \cdot Ay$	a_{23}	= t	$\cdot Ay$	••••	$a_{2,n+1}$	$= z \cdot Ay$
a_{32}	$= s \cdot A^2 y$	a_{33}	= t	$\cdot A^2 y$	•••	$a_{3,n+1}$	$= z \cdot A^2 y$
	•		•		• • •		•
	•		•		•••		•
	•		•		• • •		•
$a_{n+1,2}$	$= s \cdot A^n y$	$a_{n+1,3}$	= t	$\cdot A^n y$	• • •	$a_{n+1,n+1}$	$= z \cdot A^n y$

and the determinantal equation

The left-hand side of this equation is expanded by any practical method to give the characteristic polynomial.

A second way of finding the characteristic polynomial is algebraically more straightforward. It is to write down the condition (which we have done previously) that a non-zero eigenvector exists, namely

$$|A-\lambda I|=0.$$

This determinantal equation, when expanded by any practical method, again yields the characteristic polynomial.

Either of these two methods leads to the evaluation of a high-order determinant. A method which does not demand this is one which makes use of the theorem of Sylvester and Cayley which states that a matrix satisfies its own characteristic equation. If the characteristic equation is

$$\lambda^n + \sum_{i=0}^{n-1} a_i \lambda^i = 0$$
 ,

then the matrix equation

$$A^n + \sum_{i=0}^{n-1} a_i A^i = 0$$

is also true, as is the equation in any iterated vector

$$A^n y + \sum_{i=0}^{n-1} a_i A^i y = 0 .$$

Making use of this last equation, we choose any vector and form the n+1 successive iterates $y, Ay, A^2y, \ldots, A^ny$. No normalization of length can be used in this sequence. Then taking the *j*th component of each vector, we have a set of *n* equations of the form

$$\sum_{i=0}^{n-1} a_i(A^i y)j = -(A^n y)j .$$

This is a set of n linear equations—one for each component of the vector—in n unknowns, the coefficients a_i . It can be solved by standard techniques to give the coefficients of the characteristic polynomial.

Once the characteristic polynomial is found by any one of these proposed methods, the eigenvalues, which are the roots of the polynomial, are very easily found. A straight solution by Newton's method will produce the roots. The eigenvectors are then calculated from the matrices $A - \lambda_i I$.

The components of the *i*th eigenvector are the cofactors of any row or any column of the determinant $|A - \lambda_i I|$.

This method of the characteristic polynomial has one great disadvantage. It demands that a large number of digits be carried throughout the computation. In setting up the n+1 non-normalized vectors, a large number of digits must be carried to insure some significance in the last vector. And the coefficients of the characteristic polynomial must be carried to a large number of digits in order to insure significance in all coefficients and hence in all the roots.

We have seen that both the power method and the method of the characteristic polynomial have certain computational difficulties. A method which presents fewer numerical difficulties is a gradient method which is applicable, however, only to symmetric, or more properly to Hermitian, matrices.^a

A gradient approach to a problem is an approach that leads to maxima or minima. We look for some function that has a maximum or a minimum at an eigenvector. Such a function is

$$\mu(x) = \frac{x \cdot Ax}{x \cdot x} \; .$$

When x is the largest eigenvector, $\mu(x)$ is a maximum and is the algebraically largest eigenvalue. When x is the smallest eigenvector, $\mu(x)$ is a minimum and is the algebraically smallest eigenvalue. When x is any other eigenvector, $\mu(x)$ is a critical value as well as an eigenvalue, and the derivative of $\mu(x)$ is zero at that point.

If we choose an arbitrary vector y, we can compute $\mu(y)$ and its derivative, and we shall most probably find that the derivative is not zero. Thus, we know that y is not an eigenvector and $\mu(y)$ is not an eigenvalue. However, we can compute the gradient of $\mu(y)$, that is, the direction of change of y which produces the largest change in $\mu(y)$. The direction of the gradient is the direction in which $\mu(y)$ locally increases most rapidly, or it is the direction of steepest ascent. It can be shown that the gradient of $\mu(y)$ except for a scale factor, is given by

$$\xi(y) = Ay - \mu(y)y \; .$$

Knowing the direction of the gradient, we can then construct a new vector y which more closely approximates an eigenvector. If we change y in the direction of steepest ascent,

$$y_1 = y_0 + \alpha \xi(y) ,$$

 y_1 tends to approximate the algebraically largest eigenvector. If we change y in the direction of steepest descent,

$$y_1 = y_0 - \alpha \xi(y),$$

 y_1 tends to approximate the algebraically smallest eigenvector. The size of the constant α is important. In order to

^aThis method was developed by Dr. M. R. Hestenes and Dr. W. Karush² of the Institute for Numerical Analysis.

insure convergence of $\mu(y)$ to an eigenvalue and y itself to an eigenvector, α must be chosen less than 2/M, where $M = \lambda_1 - \lambda_n$. (Arabic subscripts will denote the algebraic ordering of the eigenvalues; λ_1 is the algebraically largest and λ_n is the algebraically smallest eigenvalue.) It can further be demonstrated that if α is less than 1/M, ξ converges in direction to the second eigenvector (x_2 or x_{n-1} according to a positive or negative choice of α).

Although this gradient method seems quite different from the power method, the two are very closely related. In the power method, the relation between a new approximation and an old one is

$$y_{r+1} = Ay_r \; .$$

In the gradient method, the relation is

$$y_{r+1} = y_r + \alpha \left[Ay_r - \mu(y_r) y_r \right] .$$

If we choose α as $[1/\mu(y_r)]$, this becomes

$$y_{r+1} = y_r + \frac{Ay_r}{\mu(y_r)} - y_r$$
$$= \frac{Ay_r}{\mu(y_r)} .$$

Thus, the gradient method with this choice of α is the same as the power method with a normalizing factor of $[1/\mu(y_r)]$. This equivalence means that the amount of work necessary to produce an eigenvector by either of these two methods is roughly comparable.

Since the gradient method is about equivalent to the power method, it also demands an acceleration scheme to insure convergence for close pairs of eigenvalues. We shall first show that the gradient method tends to remove the lower eigenvectors early in the iteration. Let us assume that we have chosen α to be

$$\frac{\beta}{\lambda_1-\lambda_n}$$

where $0 < < \beta < 1$. Our iteration scheme then becomes

$$y_{r+1} = y_r + \alpha \xi(y_r)$$

= $y_r + \alpha [Ay_r - \mu(y_r)y_r]$
= $\alpha \left(\frac{1}{\alpha}y_r + Ay_r - \mu(y_r)y_r\right)$
= $\alpha \left[\left\{\frac{1}{\alpha} - \mu(y_r)\right\}I + A\right]y_r$.

After several iterations, $\mu(y_r)$ very closely approaches λ_1 and can be approximately replaced by λ_1 . Hence,

$$y_{r+1} = \alpha \left[\left\{ \frac{1}{\alpha} - \lambda_1 \right\} I + A \right] y_r$$
.

If we now replace $1/\alpha$ by $(\lambda_1 - \lambda_n)/\beta$, we find that

$$y_{r+1} = \alpha \left[\left(\frac{\lambda_1 - \lambda_n}{\beta} - \lambda_1 \right) I + A \right] y_r .$$

Since we have chosen α so that β is close to 1, we can approximately replace β by 1, getting

$$y_{r+1} = \alpha \left[A - \lambda_n I \right] y_r .$$

This is the formula for the power method using the matrix $A - \lambda_n I$ and the normalizing factor α . Let ν_i be the eigenvalues of the matrix $A - \lambda_n I$. Then the relation between the eigenvalues of A and those of $A - \lambda_n I$ is

$$v_i = \lambda_i - \lambda_n$$
.

Thus, we have effectively transformed our scale so as to make the algebraically smallest eigenvalue about zero. The power method then kills off the smallest eigenvector very rapidly, since the ratio ν_n/ν_1 will be very small. It should be noted that, while the gradient method effectively uses the matrix A - kI, the value of k depending on the choice of α , the results that it produces are the eigenvectors and eigenvalues of the matrix A.

After some steps of this nature, the lower eigenvectors will be removed from the trial vector. We apply our acceleration process by increasing α . Suppose we were to choose $\alpha = (1/\lambda_1 - \lambda_5)$. The gradient process would be equivalent to the power method with the matrix $A - \lambda_5 I$, and would very quickly remove the fifth eigenvector x_5 . In general, we would start with an α close to our original limit 1/M, and increase α as the vector converged to an eigenvector. Constancy of direction of the gradient $\xi(y)$ would be our criterion for increasing α .

This acceleration method is not normally applied by increasing α monotonically; rather, one intersperses a larger value of α among several smaller ones. The reason for this is that large values of α tend to magnify the contribution from any small eigenvector that may be present, and round-off error always introduces small portions of all eigenvectors. Hence, one uses several small values of α to keep the smaller eigenvectors out of the trial vector and then uses a large value of α to remove a portion of the larger eigenvectors.

There is a fairly simple geometrical picture that may make this acceleration scheme a little clearer. The function $\mu(v_r)$ has only one maximum and one minimum. The rest of the critical points are saddlepoints. We now start our procedure by choosing an arbitrary vector y, computing the function $\mu(y)$ and the direction of steepest ascent $\xi(y)$. We want to change y by moving in the direction of $\xi(y)$, but we dare not move too far or we may find ourselves over a saddle and down the slope on the other side. So we take a small step in the direction of $\xi(y)$ (we choose a small value of α), we stop and re-evaluate the direction of steepest ascent. When we get to a point where the direction of the gradient changes very little from step to step, we know we are nearly on a ridge and can follow the ridge a long distance toward the peak. So we take a long step forward (we choose a large value of α). However, our direction was not precisely correct. Although we are much closer to the peak, we are now down on the side of the ridge and must slowly make our way to the ridge again by small steps and by constantly recomputing the direction of steepest ascent. Thus, step by step, we make our way to the peak.

So far, we have talked about finding the largest eigenvalue by the power method and the gradient method. To find the other eigenvalues and eigenvectors is not difficult if the matrix A is symmetric. We make use of the mutual orthogonality of the eigenvectors, and seek by either method the largest eigenvector which is orthogonal to the one we know. It is very easy to make one vector orthogonal to another. If x is our known eigenvector, y is any arbitrary vector, and \overline{y} is that component of y which is orthogonal to x, then \overline{y} can be found from the equation

$$\overline{y} = y - \frac{y \cdot x}{x \cdot x} x$$

In either the power method or the gradient method, the second eigenvector can be found by choosing any trial vector y, and keeping it orthogonal to the first eigenvector during the computing process. If y is made orthogonal to x, it should theoretically stay orthogonal to it. However, round-off error will continuously bring in portions of x, and the numerical operations will magnify these small portions. Hence, every once in a while the iterated vector must be re-orthogonalized to keep out the largest eigenvector. It will then converge to the second eigenvector.

We have spent a great deal of time talking about theoretical ways of finding eigenvectors and eigenvalues. At the Institute for Numerical Analysis, we have tried all three of these methods on our IBM Card-Programmed Electronic Calculator and have some practical experience that may be of value in assessing the relative merits of these three methods.

We have solved quite a few eigenvector problems by the power method. Our operating procedure has been as follows. We set up a control deck of four parts. Part one loads the 941 storage unit with the components of the trial vector. Part two of the control deck orthogonalizes the trial vector to all known eigenvectors. Part three normalizes the length of the trial vector to unity. Part four multiplies the trial vector by the matrix to get the next approximation.

The loading deck consists of spread-read-in cards which carry the components of the trial vector and instruction cards which transfer these components to the 941 storage unit. This section of the control deck is used only at the start of the run, if the matrix is of low order. It must be used at each step if the order of the matrix is large. For it is apparent that no component of the old trial vector can be thrown away until all but one component of the new trial vector has been computed. As the components of the old trial vector are used in computing the last component of the new trial vector, they can be thrown away. So, if the matrix is of high enough order that two trial vectors cannot be simultaneously retained in the storage system of the machine, then the components of the new trial vector must be punched out and read back into the storage system at the beginning of the next iteration.

The orthogonalization deck is not needed when solving for the largest eigenvector. It is needed when solving for the smaller eigenvectors, and its size increases as the number of known eigenvectors increases. When finding the second eigenvector, the trial vector must be orthogonal to one known vector; when finding the third eigenvector, the trial vector must be orthogonal to two known vectors; when finding the *n*th eigenvector, there are n - 1 known eigenvectors against which to orthogonalize. Notice that it is unnecessary to solve by iteration for the *n*th eigenvector. The orthogonality conditions are sufficient to define its direction uniquely.

The orthogonalization deck first computes the scalar product of the trial vector, which is stored in the 941, and the known eigenvector, which is punched into the instruction cards. The scalar product is accumulated in one of the counters of the IBM Type 417 Accounting Machine. This is then divided by the product $x \cdot x$, which is a constant and is punched on one of the instruction cards. The quotient replaces $x \cdot y$ in the counter of the 417. This new constant is then used as a multiplier of the components of the eigenvector x. As soon as the component, which is carried on one of the instruction cards, is multiplied by this constant, it is subtracted from the corresponding component of y, which is in the 941, and the difference replaces the old y component in the 941. This operation requires only one counter in the 417, and hence causes no strain on tight storage space.

It is unnecessary to orthogonalize at every step. We generally print out the product $y \cdot x$. As long as this product is less than a hundred in units of the last place carried, we feel that no great lack of orthogonality has occurred between successive orthogonalizations. If the product stays as small as three or four units in the last place, then we are orthogonalizing too frequently.

The normalization deck is the first deck that is always needed. This deck sums the squares of the components of yin one of the counters of the 417. The square root of the sum is taken to give the length of y, and this quantity replaces the sum in the 417 counter. This length is then divided into the components of y, and the quotients replace the old components of y in the 941 storage units. This operation, like the orthogonalization, requires only one additional storage space, a 417 counter, and does not generally add to the storage problem. We print out the length of the vector as a check on the convergence of the eigenvector.

The multiplication deck multiplies the elements of the matrix by the components of the trial vector. The elements

are punched on instruction cards, while the components of the vector are stored in the 941. The components of the new vector are accumulated in the counters of the 417. If the matrix is of sufficiently low order, the new elements are transferred to blank spaces in the 941 until all elements are computed and available in storage. They are then all transferred to the spaces of the 941 occupied by the corresponding elements of the old vector. If the matrix is not small enough, the elements of the new vector are punched out of the 417 counters as soon as these counters are full.

We have used this method successfully in some cases of non-symmetric matrices with well separated eigenvalues, and used it unsuccessfully in one case of non-symmetric matrices with close sets of eigenvalues. It is a method with some advantages, and one that should be considered when looking for a way to solve an eigenvector problem.

The method of the characteristic polynomial is one that has proved the least successful at the Institute for Numerical Analysis. We have tried it only once, and that was as a training program for a class of students. We tried to solve a fourth-order matrix by this method, and ended up with only four significant figures. We set up our characteristic polynomial using the Sylvester and Cayley approach. The coefficients of the polynomial dropped off rapidly in size. We solved for the roots of this polynomial by Newton's method. The accuracy of the roots was limited by the accuracy of the coefficients of the polynomial. And we computed the components of the eigenvectors by taking minors of the determinant $|A - \lambda_i I|$. From this one sample, we feel that this approach is not desirable in the case of one or two matrices, as there must be a great deal of attention given to the shifting to preserve the significance of the coefficients of the polynomial. We feel that this may have better uses as a way of solving a large number of matrices on a machine capable of handling longer numbers.

We have, of course, tested the gradient method extensively. Our procedure has been to set up a control deck of five parts. Part one is the loading deck. Part two is the orthogonalizing deck. Part three is the deck that computes the function $\mu(y)$. Part four computes the gradient $\xi(y)$. Part five computes the next approximation y_{r+1} .

The loading deck and the orthogonalization decks are exactly analogous to those used in the power method. The first new deck is the deck which computes $\mu(y_r)$. We store the vector y_r in one half of the available storage. This deck multiplies the vector by the elements of the matrix, which are punched on the instruction cards. As soon as a component of the vector Ay_r is completed, it is transferred to the free half of the memory. Now the two scalar products $y_r \cdot y_r$ and $Ay_r \cdot y_r$ are computed and stored in two counters of the 417. The function $\mu(y_r)$ is then computed and replaces one of the now unnecessary scalar products in the 417. This value of μ is then printed out to enable us to check on the convergence of the eigenvalue.

The fourth deck computes $\xi(y_r)$. The components of y_r are multiplied by the value of $\mu(y_r)$ and subtracted from the components of Ay_r . The difference is the component of $\xi(y_r)$ and it replaces the corresponding component of Ay_r in the memory.

The last deck computes the new approximation y_{r+1} . Both vectors y_r and $\xi(y_r)$ are available in the memory. The constant α is read in from an instruction card, and the component of the new vector is computed by adding α times the component of $\xi(y_r)$ to the component of y_r . The new component replaces the corresponding component of y_r in the memory. The components of $\xi(y_r)$ and those of y_{r+1} are printed out, for the components of $\xi(y_r)$ go to zero as y_{r+1} approaches an eigenvector.

In practice, we have found all six eigenvalues and eigenvectors of a sixth-order symmetric matrix in six hours. The accuracy of the vectors was better than one part in ten million in the length, and the accuracy of the eigenvalues was about the same. For matrices of the order of 5 to 8, we figure that an hour per eigenvector is a reasonable time estimate. We have also solved matrices of higher order.^b

The choice of a particular computing method depends a great deal on the problem at hand and the equipment available. It is our opinion that the gradient method is preferable to the power method in finding the eigenvectors of a few matrices. Its advantages are mainly its flexibility. It needs no normalization, its acceleration process is very easy to apply, its gradient converges to give a two-digit approximation to the next eigenvector, it will find either the highest or the lowest eigenvector-these are all advantages. However, the method needs the watchful care of an operator, and may not be well adapted to finding all the eigenvectors of a large number of matrices until fully automatic computers (ones which store the programming inside the internal memory) are available. And if the matrices are not symmetric, then the choice has to be between the power method and the method of the characteristic polynomial.

The eigenvector problem is by no means solved from the point of view of a computer. While the gradient method offers a method that will give an efficient solution to the eigenvector problem for symmetric matrices, no matter how closely the roots are spaced, there is no method which will always efficiently find the eigenvectors of a non-symmetric matrix. Once a reasonable method is devised to handle the case of the non-symmetric matrix with close eigenvalues, then we can say that this method, together with the gradient method, will allow us to solve the eigenvector problem for all practical cases.

^bOne of order 17 was solved by Mr. Robert Hayes, one of our graduate fellows, who has been very active in testing the numerical process. He is confident that he can solve a matrix of order 24 by this method.

REFERENCES

- 1. J. B. ROSSER, C. LANCZOS, M. R. HESTENES, W. KARUSH, "The Separation of Close Eigenvalues of a Real Symmetric Matrix," *National Bureau of Standards Journal of Research* (forthcoming).
- ing).
 M. R. HESTENES and W. KARUSH, "A Method of Gradients for the Calculation of the Characteristic Roots and Vectors of a Real Symmetric Matrix," *National Bureau of Standards Journal of Research* (Washington, D. C.: National Bureau of Standards, July, 1951).

DISCUSSION

Dr. Welmers: Do you have any impressions as to whether this method could be extended to a non-symmetrical matrix or whether you have to start off on an entirely different method for those?

Dr. Yowell: Dr. Hestenes has been trying to answer that question now for about a year and a half.

Professor Verzuh. I wonder whether some of the people in the aircraft industry have any practical comments on non-symmetrical matrices.

Dr. Yowell: I can make a comment for John Lowe in his absence, because I have discussed the problem with him many times.

He expands the non-symmetric matrix, finds the characteristic polynomial and solves for the root by Newton's method. I believe this is a widely-used method.

Mr. Bell: We have used the method that Dr. Yowell has just described. We have used another one that I would like to call a guessing-game, in which you take the matrix, assume a value for the eigenvalue, place it in all elements except 1, rearrange the matrix so that the element which does not have an eigenvalue is in the upper corner, reduce the matrix to a triangular one so that the determinant is now equal to the product of the main diagonals, and you adjust this one element so that the problem will be zero. You have a product of terms, none of which is equal to zero. You want it to be equal to zero; so, by adjusting the last term, you have a value. If it agrees with the initial guess, you have the eigenvalue. This supposes that you have some knowledge of where the eigenvalue is. It has the advantage in the aircraft field that very often they do have quite good knowledge of where the root lies, and can go after a specific value without having to grind through a number, working from either the low or the high end.

Since we do a great deal of work of this type, working on the average with eighth-order complex matrices, we have an average of about three guesses per eigenvalue. It requires something of the order of about an hour—not of machine time but of elapsed time—to get out one trial; so, about three trials will give an eigenvalue, with the assumption that you know where you are looking.

An Application of the IBM Card-Programmed Electronic Calculator to the Analysis of Airplane Maneuvering Horizontal Tail Loads

LOGAN T. WATERMAN

Fairchild Aircraft Division of Fairchild Engine and Airplane Corporation

×

A METHOD is shown by means of which the design maneuvering horizontal tail loads of an airplane can be calculated rapidly on the IBM Card-Programmed Electronic Calculator. This method is based upon use of the Laplace transform to derive a solution of the differential equations which is suited to machine computation techniques.

The methods shown in references 1 and 2 were devised primarily for manual computation and are very cumbersome to apply directly to machine computation procedures. Also, these methods are limited to the use of a trapezoidal time history of the elevator deflection, while the method shown in this paper is good for any shape of the elevator deflection time history. This means that experimental flight test data can be checked and correlated directly with the machine calculation method shown here.

Two possible cases are obtained for the solution of the differential equations. Case I represents a stable airplane configuration, while Case II represents an airplane configuration having a heavily-damped short-period oscillation. Whether Case I or Case II (described mathematically under the section headed *Theory*) is required to obtain the solution of the differential equations depends largely upon the fore and aft location of the center of gravity of the airplane for the loading condition being analyzed. The numerical example shown in this paper is a Case I solution. It will be seen that the Case II solution can be obtained in a manner very similar to that shown in this paper for Case I, if the proper programming revisions are incorporated to take care of the differences in the equations for the two cases.

The problem upon which this paper is based is a typical airplane design analysis required to establish the magnitude of the dynamic loads which will result when a given load factor is imposed upon the airplane. It should be apparent that other dynamic loads analyses such as landing loads, gust loads (near critical flutter speeds), aileron loads, and vertical tail loads can be calculated in a manner very similar to the methods shown in this paper.

The estimated CPC operating time for a production run for 35 time intervals based upon the method shown in this paper is about one hour per loading condition. The same analysis setup parallel to the manual computation method shown in reference 1 takes three hours per loading condition. In addition, more re-runs are required for this latter method, because more errors in card handling occur in this more complicated procedure. Hence, the method shown in this paper results in better than a 70 per cent saving in CPC running time.

Before discussing the actual theory involved, it might be well to define the symbols which will be used. The nomenclature is given on pages 121 and 122.

Theory

The equations of motion used are taken directly from reference 1. The discussions behind the assumptions leading to the neglect of certain stability derivatives in order to obtain the simplified equations shown below are beyond the scope of this paper. The reader is referred to references 1 and 2 for elaboration on this phase of the problem.

The equations of motion are

$$\dot{w} - Z_w w - U_0 q = 0 , \qquad (1)$$

$$M_{w} w + M_{iv} \dot{w} + M_{q} q - \dot{q} = -M_{\delta e} \delta e ; \qquad (2)$$

other required parameters are

$$A_{z} = \dot{w} - U_{0} q , \qquad (3)$$

$$\alpha_w = \frac{w}{U_0} , \qquad (4)$$

$$\dot{\alpha}_w = \frac{iw}{U_0} , \qquad (5)$$

(Continued on page 123)

Nomenclature

	10	
1	$\cdot * \frac{dC_L}{d\alpha}$	= Slope of lift curve of the wing (per radian)
2	. 0	= mass density of the air
3	. Sw	= Area of wing (sq. ft.)
4	. Uo	= Linear velocity of center of gravity along the x-axis
	. 00	(Positive when in plus x-direction) (ft./sec.)
5	. Constant	= 2.00
6	. <i>M</i>	= Mass of airplane (slugs)
7	$\underline{dC_{M}}$	= Slope of the moment coefficient curve for the whole airplane for
	$d\alpha$	a given weight and power configuration (per radian)
8	C = MAC	= Mean aerodynamic chord (ft.)
9	I_{u}	= Moment of inertia about the y-axis which is through the center
	8	of gravity and perpendicular to the plane of symmetry $(X-Z)$ (slug-ft. ²)
10	$\left(\frac{dC_L}{dC_L}\right)$	= Slope of lift curve of the horizontal tail (per radian)
	$\int d\alpha J_t$	
11	S_t	= Area of the tail (sq. tt.)
12	. n	= Tail efficiency = $\frac{(\text{dynamic pressure})_{tail}}{(1 + 1)^{1/2}}$
	,	(dynamic pressure) free stream
13	$\frac{d\epsilon}{d\epsilon}$	= Change in angle of downwash with respect to change in angle of
	aα	attack of the wing
14	K	$\underline{I_y}$
1,		$-Ml_{t^2}$
15	. Constant	= 1.25
16	$\frac{dC_M}{dC_M}$	= Change in the moment coefficient for the whole airplane with
	dδe	respect to a change in elevator deflection (per radian)
18	. Constant	= 57.3
24	. <i>t</i>	= Time (sec.)
26	$\cos \omega t$	
27	$\sin \omega t$	
20	at	
20	2	
	$\frac{at}{2}$	
33	. e*	10
36	Constant	= 1.0
41	. K _T	4
42	Δn	$=\frac{A_z}{32.2}$ = Required load factor
43	Δn	$= \frac{A_z}{32.2} = $ Calculated maximum load factor on first run of 1165 deck
44	l_t	= Distance, parallel to the <i>x</i> -axis, from airplane center of gravity to the aerodynamic center of the tail (ft.)
		(Continued on next page)

*The number symbols are shown on the listing to identify the calculated data obtained from the CPC.

		NOMENCLATURE (CONTINUED)
		1.27
501.	Z_w	$=\frac{1}{M}\frac{\partial Z}{\partial z_{\rm H}}$
		N1 07U
502.	Man	$=\frac{1}{2}\frac{\partial M}{\partial M}$
0021	w	$I_y \partial w$
502	Μ.	$1 \ \partial M$
505.	WI w	$=\frac{1}{I_{y}}\frac{\partial \dot{w}}{\partial \dot{w}}$
504	17	$1 \ \partial M$
504.	M_{q}	$=\frac{1}{I_{u}}\frac{1}{\partial q}$
505	M.	•
505.	a/2	
507	u/2 b	
508	C.	
	01	$\overline{(a)^2}$
512.	ω	$= \sqrt{b} - \left(\frac{a}{2}\right)$, (real for Case II)
542	A_{z}	- Linear acceleration at the center of gravity along the gravic (gra)
572.	32.2	- Enter accortation at the center or gravity along the 2-axis (g s)
543.	α_w	= Angle of attack of the wing (radians)
548.	q	= Angular pitching velocity of the airplane (rad./sec.)
552.	$\dot{\alpha}_w$	
562.	\dot{q}	
579.	α_t	= Angle of attack of the tail (radians)
580.	Δl_t	= Incremental maneuvering horizontal tail load
590.	$(1-d\epsilon/d\epsilon)$	α) (positive when upward) (ibs.)
501	l_t	
391.	$\overline{U_0\sqrt{\eta}}$	
500	$l_t(d\epsilon/d\alpha)$	
592.	$\frac{U_0}{U_0}$	
593.	$d\alpha_t/d\delta e$	
594.	m	
595.	n	
	C1	
596.	$\frac{1}{m-n}$	
	C Im	
597.	$\frac{C_1}{m}$	$\left(\frac{1}{1}-\frac{1}{2}w\right)$
	m = n	
598.	$\frac{C_1}{(n-1)}$	$\frac{-Z_w}{2}$
	m - n	
500	Δt	
399.	2	
600.	Constant	$=\frac{1}{1222}$
601	mt	<i>52.2</i>
602	c ont	
602.	$S_{\rho}(t)$	- Elevator deflection (radians)
604	701	- Linear velocity of center of gravity along the gavis (ft /sec.)
605	701	- Amoun velocity of center of gravity along the 2-axis (11./ Sec.)
000.	~	

where

$$Z_w = -\frac{(dC_L/d\alpha) \rho S_w U_0}{2M}$$
(6)

$$M_w = \frac{(dC_M/d\alpha) \rho S_w U_0 C}{2 I_y} \tag{7}$$

$$M_{i\nu} = -\frac{(dC_L/d\alpha)_t \rho S_t \eta (d\epsilon/d\alpha)}{2 KM}$$
(8)

$$M_{q} = -\frac{1.25 \, (dC_{L}/d\alpha)_{t} \, \rho \, S_{t} \, U_{0} \, \eta}{2 \, KM} \tag{9}$$

$$M_{\delta e} = \frac{(dC_M/d\delta e) \rho S_w U_0^2 C}{2 I_y}$$
(10)

$$\Delta l_t = K_T \, \alpha_t \tag{11}$$

$$K_{T} = \left(dC_{L}/d\alpha \right)_{t} \frac{1}{2} \rho \ U_{0}^{2} S_{t} \eta$$
(12)

$$\alpha_{t} = \left(1 - \frac{d\epsilon}{d\alpha}\right)\alpha_{w} + \frac{l_{t} q}{U_{0}\sqrt{\eta}} + \frac{l_{t} (d\epsilon/d\alpha) \dot{\alpha}_{w}}{U_{0}} + (d\alpha_{t}/d\delta e) \delta e .$$
(13)

The initial conditions are

at
$$t = 0$$
, $\begin{cases} w = \dot{w} = 0 \\ q = \dot{q} = 0 \end{cases}$.

The Laplace transforms of equations 1 and 2 subject to these initial conditions are

$$s\lambda(s) - Z_w \lambda(s) - U_0 \phi(s) = 0, \qquad (14)$$

$$M_{w}\lambda(s) + M_{w}s\lambda(s) + M_{q}\phi(s) - s\phi(s)$$

= - Mde Lde. (15)

Equations 14 and 15 are solved for $\lambda(s)$ and $\phi(s)$ and the following substitutions made:

$$a = M_q + Z_w + U_0 M_{\dot{w}}$$
(16)

$$b = M_q Z_w - U_0 M_w \tag{17}$$

$$C_1 = \bigcup_0 M_{\delta e} \tag{18}$$

$$m = \frac{a}{2} + \sqrt{\left(\frac{a}{2}\right)^2 - b} \tag{19}$$

$$n = \frac{a}{2} - \sqrt{\left(\frac{a}{2}\right)^2 - b} \,. \tag{20}$$

Hence, the Laplace transforms of the coordinates are

$$\lambda(s) = \frac{C_1}{(s-m)(s-n)} \mathcal{L}\delta e, \qquad (21)$$

$$\phi(s) = \frac{C_1 \left(s - Z_w\right)}{U_0 \left(s - m\right) \left(s - n\right)} \mathcal{L}\delta e \,. \tag{22}$$

Using partial fractions or the Bromwich integral the inverse transforms can be shown to be:

Case I,
$$\left(\frac{a}{2}\right)^2 > b$$

$$w = \frac{C_1}{m-n} \left[e^{mt} - e^{nt}\right] * \delta e \qquad (23)$$

$$q = \frac{C_1}{U_0 (m-n)} [(m-Z_w) e^{mt} - (n-Z_w) e^{nt}] * \delta e.$$
 (24)

By the theorem of convolution, equations 23 and 24 can be written as follows:

$$w = \frac{C_1 e^{mt}}{(m-n)} \int_0^t \frac{\delta e(\tau)}{e^{m\tau}} d\tau - \frac{C_1 e^{nt}}{(m-n)} \int_0^t \frac{\delta e(\tau)}{e^{n\tau}} d\tau$$

$$q = \frac{C_1 (m-Z_w) e^{mt}}{U_0 (m-n)} \int_0^t \frac{\delta e(\tau)}{e^{m\tau}} d\tau$$

$$- \frac{C_1 (n-Z_w)}{U_0 (m-n)} \int_0^t \frac{\delta e(\tau)}{e^{n\tau}} d\tau. \qquad (26)$$

Equations 25 and 26 are evaluated numerically for analyses that require a Case I solution. The additional parameters are calculated by

$$A_z = Z_w w \tag{27}$$

$$\dot{w} = A_z + U_0 q \tag{28}$$

$$\dot{q} = M_w w + M_{\dot{w}} \dot{w} + M_q q + M_{\delta e} \delta e \qquad (29)$$

$$\alpha_w = \frac{w}{U_0} \tag{30}$$

$$\dot{\alpha}_w = \frac{\dot{z}\dot{v}}{U_0} \,. \tag{31}$$

Where complex roots of the quadratic are obtained, the following substitutions are made:

$$\omega = \sqrt{b - \left(\frac{a}{2}\right)^2} \tag{32}$$

$$m = \frac{a}{2} + i\omega \tag{33}$$

$$n = \frac{a}{2} - i\omega \,. \tag{34}$$

The transforms of the coordinates shown by equations 21 and 22 become

$$\lambda(s) = \frac{C_1}{\left(s - \frac{a}{2} - i\omega\right)\left(s - \frac{a}{2} + i\omega\right)} \mathcal{L}\delta e \qquad (35)$$

$$\phi(s) = \frac{C_1 (s - Z_w)}{U_0 \left(s - \frac{a}{2} - i\omega\right) \left(s - \frac{a}{2} + i\omega\right)} \mathcal{L}\delta e.$$
(36)

As shown under Case I, the inverse transforms are:

Case II,
$$b > \left(\frac{a}{2}\right)^2$$

$$w = \frac{C_1 e^{\frac{at}{2}}}{\omega} \sin \omega t * \delta e \qquad (37)$$

$$q = \frac{C_1 \left(\frac{a}{2} - Z_w\right) e^{\frac{at}{2}}}{U_0 \omega} \sin \omega t * \delta e$$
$$+ \frac{C_1 e^{\frac{at}{2}}}{U_0} \cos \omega t * \delta e .$$
(38)

Again, by the theorem of convolution, equations 37 and 38 can be written as follows:

$$w = \frac{C_1 e^2}{\omega} \sin \omega t \int_0^t \cos \omega \tau \, \delta e(\tau) d\tau$$
$$- \frac{C_1 e^2}{\omega} \cos \omega t \int_0^t \sin \omega \tau \, \delta e(\tau) d\tau \qquad (39)$$

$$q = \left[\frac{C_1 e^{\frac{at}{2}}}{U_0} \sin \omega t - \frac{C_1 \left(\frac{a}{2} - Z_w\right) e^{\frac{at}{2}}}{U_0 \omega} \cos \omega t}\right] \int_0^t \sin \omega \tau \, \delta e(\tau) d\tau$$
$$+ \left[\frac{C_1 \left(\frac{a}{2} - Z_w\right) e^{\frac{at}{2}}}{U_0 \omega} + \frac{C_1 e^{\frac{at}{2}}}{U_0} \cos \omega t}\right] \int_0^t \cos \omega \tau \, \delta e(\tau) d\tau \,. \tag{40}$$

Equations 39 and 40 are evaluated numerically for analyses that require a Case II solution. The additional parameters are given by equations 27 through 31, inclusive.

The integrals shown in the equations under Case I and Case II are evaluated numerically by multiplying Δt by the average ordinate (integrand). The scheme used is shown as follows:

Let I_i = value of integral at any station,

 A_i = product of the integrand and Δt at any station.

Written explicitly, the first three values of the integral are

$$I_{1} = A_{0} + \frac{A_{0} + A_{1}}{2} = \frac{3}{2}A_{0} + \frac{A_{1}}{2}$$

$$I_{2} = I_{1} + \frac{A_{1} + A_{2}}{2} = \frac{3}{2}A_{0} + A_{1} + \frac{A_{2}}{2}$$

$$I_{3} = I_{2} + \frac{A_{2} + A_{3}}{2} = \frac{3}{2}A_{0} + A_{1} + A_{2} + \frac{A_{3}}{2}.$$

By inspection, the equation for the integral at the *i*th station is

$$I_i = \frac{3}{2}A_0 + A_1 + A_2 + \ldots + A_{i-1} + \frac{A_i}{2}.$$
 (41)

Of course, almost any desired refinement can be obtained by the use of more elaborate integration schemes. However, the accuracy required in the analysis shown in this paper is satisfactorily achieved by means of equation 41.

DISCUSSION OF RESULTS

No attempt was made in the numerical calculation to obtain extremely refined results. Although better answers could be obtained easily by using more elaborate numerical integration methods, it is considered that the results shown



x COMPUTED BY METHOD SHOWN IN THIS PAPER

Figure 1. Angular Pitching Acceleration Time History \dot{q} vs t





FIGURE 2. LIMIT INCREMENTAL TAIL LOAD TIME HISTORY ΔL_t vs t

are well within the practical limitations of the problem. Two typical calculated parameters are shown plotted in Figures 1 and 2. In general, better agreement was obtained for the other parameters not shown. Figure 1 shows a plot of the time history of the angular pitching acceleration of the airplane. Figure 2 shows a plot of the time history of the limit incremental tail load for the airplane.

It is noted that the time interval used in the analysis shown in this paper is exactly the same as that used in reference 1. If a direct numerical solution of the differential equations had been attempted, a much smaller time interval would have been required. It is estimated that this would lead to more complicated programming which would exceed the storage capacity of the machine as well as to require more CPC running time for each condition analyzed.

PROCEDURE FOR CALCULATING MANEUVERING HORIZONTAL TAIL LOADS—CASE I

The calculations are accomplished by the proper sequencing of several decks.^a These decks are interrelated—some storing data and others punching data for the decks that follow. The deck numbers are listed below in the proper order to complete one condition for one set of factual data.

The listing shown in Table I is a complete set of factual data cards for one weight and center of gravity condition. The deck from which this listing was made is called a basic factual data deck. It is preserved for the purpose of printing listings such as the one shown in Table I into which factual data are directly written. Also, it is used to reproduce factual data decks (channel A blank), into which the copied factual data are key punched.

Table II is explained as follows:

- Deck 1154 calculates $\Delta \delta e/\Delta t$, δe , and Δt and punches these values in table look-up cards for use in the 1124 deck. Pull out old factual data on column 46 and sort in new factual data on sequence numbers in columns 45, 44, and 43.
- Deck 1124 calculates δe time history and punches the values with all other necessary information in cards for use in the 1165 deck. Change factual data cards and table look-up cards by hand. It is noted that card number 3 will have the specified load factor for the first calculation (i.e., card 3 and card 2 will have the same channel A entry). After the first run of the 1165 deck, the maximum load factor obtained will be key punched into channel A of card number 3 for the final calculation.
- Deck 1126 calculates Δl_t equation coefficients and punches these values along with all other necessary information in cards for the 1165 deck. Pull old factual data out on column 46 and sort in new factual data on columns 3-2.

In Table III there is only one deck used:

Deck 1161 calculates the stability derivatives and the integral coefficients and stores this data for use in the 1162, 1163, and 1165 decks. Pull out old factual data on column 46 and sort in new data on columns 3-2.

Finally, Table IV makes use of the following decks:

- Deck 1162 calculates e^{mt} and punches these values—along with all other necessary information such as programming, decimal locations, identification, and symbols—in cards for use in the 1165 deck. No sorting is necessary.
- Deck 1163 calculates e^{nt} and punches these values, along with all other necessary information, in cards for use in the 1165 deck. No sorting is necessary.
- Deck 1165 calculates time history for Δn , α_w , $\dot{\alpha}_w$, \dot{q} , \dot{q} , and Δl_t . Pull old factual data out on column 46 and sort in new factual data on columns 3-2, 42-41. For the first trial, a maximum load factor will be obtained at a specified time which is not necessarily correct. Enter this load factor in card number 3 of deck 1124 and repeat decks 1124, 1161, and 1165. This second run of deck 1165 will give the correct values for all the parameters obtained.

DESCRIPTION OF CPC CONTROL PANELS

The same control panels were used throughout the analysis shown in this paper. However, it is recommended that special panels similar to those shown in reference 6 (Continued on page 129)

^aA sample listing of all decks used in the analysis is not shown. Those omitted are considered to be elementary calculations which the reader can find in several sources of literature on the subject.

		TABLE I			
	LISTING OF A	ALL FACTUAL DATA CARDS	WITH CHANNEL A BL	ANK	
				Pull out	t Column 46
Card Number ar	nd Programming	Decimal Location Channel A	Decimal Location Channel C	Deck Number	Channel Channel A C
$\begin{array}{c} 5 & 1 \\ 7 & 1 \\ 8 & 1 \\ 9 & 1 \\ 7 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\$	11 13 14 15 15 16 17 23 26 128 24 128 24 128 224 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 228 128 1	52 512 524 54 572 524 54 522 54 52 54 52 54 52 54 52 54 52 51 22 22 22 51 53 52 51 53 53	52 51 52 52 52 52 52 52 52 52 52 52 52 52 52	1 11 6 1 1 11 6 6 1 1 11 5 4 6 1 1 1 1 5 4 6 7 0 1 1 1 5 5 4 1 11 5 5 4 1 11 5 5 4 1 11 2 6 6 1 11 1 2 6 1 11 2 6 6 1 11 2 6 1 11 2 6 1 11 2 6 6 1 11 2 6 1 11 1 2 6 1 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	$\begin{bmatrix} 9 & 19 \\ 9 & 7 & 10 \\ 9 & 16 & 7 \\ 9 & 13 & 16 \\ 9 & 4 & 509 \\ 9 & 6 & 13 \\ 9 & 9 & 6 & 13 \\ 9 & 14 & 8 \\ 9 & 12 & 22 \\ 9 & 109 & 14 \\ 9 & 109 & 14 \\ 9 & 109 & 14 \\ 9 & 109 & 14 \\ 9 & 2 & 14 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\ 9 & 201 \\$

Table III

Evaluation of the Stability Derivatives and Integral C	OEFFICIENTS
Weight	MAC
62000 431	1161
5 1 11 514000000*2 51400000*52 6 1 12 41000000*52 51400000*52 7 1 13 56100000*52 56100000*52 8 1 14 156000000*52 56100000*51 9 1 15 40000000 2 156000000*51 10 1 16 1925000000 54 40000000 52 10 1 16 1925000000 54 560730000 57 11 1 7 560730000 54 1477000000 54 12 1 18 1477000000 52 324880000 54 13 21 3248800000 52 324880000 54 14 1 23 126400000 52 1364000000 52 16 1 24 15000000 50 12640000 52 16 1 26 4170000000 54 15000000 50 18 1 24 72 14770000000 54 15000000<	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{c} 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\$
53 83 4 82 15 9991802469960 112146000 58 54 24 1 11 123341560*52 730969451*52 55 27 1 12 6303747*50 123341560*52 56 6 1 39663032836949 6303747*50 6303747*50 57 85 1 14 9821514671952 136967163*49 58 14 9821514671952 178485328*52	1161 505 1161 501 1161 502 1161 502 1161 503 1161 504
	$ \begin{array}{c} 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161 \\ 1161 \\ 1161 \\ 506\\ 1161\\ 507\\ 1161\\ 510\\ 1161 \\ 509\\ 1161\\ 509\\ 1161\\ 509\\ 1161\\ 509\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161 \end{array} $
741 869859759 52 77 83 24 75665600 52 999999999 51 78 82 1 74 12 869859751 869859759 51 79 16 1 75 179471099*52 86985970 52 80 24 1 85 12 869859751 179471099*52 81 84 1 25 9907514870952 86985970 52 82 85 1 26 9733542930952 92485129*52 92485129*52 82 3 26 9733542930952 266457069*52 266457069*52	$\begin{array}{c} 1161\\ 1161\\ 1161\\ 512\\ 1161\\ 1161\\ 1161\\ 1161\\ 1161\\ 594\\ 1161\\ 595\\ 1161\end{array}$
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
93 61 60132100 53	1161 598

127

								Т.	ABI	LE	IV			
					C	alculation of Til	ИĘ	Hı	ST	OR	ΥF	OR N	r $n, a_w, \dot{a}_w, q, \dot{q},$ and L_t	
							Τı	MI	: =	= ;	1 s	EC.	с.	
						Weig	ght		c.c	3.]	Lo	atic	tion % MAC Time	
1 2 3		1 1 1		21 18 22		$\begin{array}{r} 62000 \\ 1161 \\ 911663000 51 \\ 5036671 * 52 \\ 766089000 51 \end{array}$					4 3	•	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
4 5 6 7 8	18 18	4 4 3 3	21 22 82 83	72 73 23 24	22	$\begin{array}{r} 5036671 \pm 52 \\ 5036671 \pm 52 \\ 50000000 51 \\ 50000000 51 \end{array}$	9 9 9 9 9	11	66 08 75 25	30 90 29 47	00 00 29 53	51 51 951 951	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
9 10 11 12 13	23 24 12 16	1 1 3 3	21 21	76 77 72 73	1 2	27623500*50 32872600*50 17520871*56 12964800*53	9	11	66	30 30	000	51 51	32872600*50 1 1165 1 1165 27623500*50 1 1165 51 32872600*50 1 1165 51 15973130*56 1 1165 596	
14 15 16 17 18	76 762 17 77	3 3 3 3 3 3	8 3 8 3 8 3 8 3 8 3 8 3 8 3 8 3	74 75 72 73 85	4 1 2 4	$\begin{array}{r} 9972376499950\\ 9972376499950\\ 17520871*56\\ 60132100\\ 996712739950\\ 996712739950\\ 995712739950\\ \end{array}$	99 98 7 99 99	84 81 66 86	02 80 80 80 80 80 80 80 80 80 80 80 80 80	68 90 90 74 54	69 99 00 49 00	956 952 51 956 52	56 118195300*52 1 1165 52 441230000 52 1 1165 51 326497 52 1 1165 51 13422550*56 1 1165 598 56 460665400 52 1 1165 52 441230000 52 1 1165	
20 21 22 23 23	84 85 24 28	1 1 1 3	25	25267772		99999999999999952 1840824 52 27623500*50 32872600*50 123341560*52						52	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	604 548
25 26 27 28 29	11 27 14	4 3 3 3 3 3	11 72 26 25 26	74 72 73 73	2	1000000 54 310559006 50 41700000 54 6303747*50 178485328*52	4	1	00 84 84	00 08 08	00 24 24	54 52 52 52 52	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	542
31 32 33 34	25 72 82 15	333	84 13 18	22 73 73	2 1 2	7676200 54 7676200 54 730969451*52 32479510 52	1	23 36 5	98 96 03	087166	156371	50 \$49 \$52	50 52 1 1165 50 52 1 1165 52 1051390+52 1 1165 605 52 1051390+52 1 1165 505 36816500 52 1 1165	543 552
36 37 38 39	60	1 3 3 3 3	21 26 22 18	72 72 72 72	2	600000000 50 1260439942 50 460431650 50 47800000 52		1 1 5	84 84 03	08 08 66	24 00 71	52 52 52 * 52	52 32479510 52 1 11165 5.90 52 52 1 11165 5.91 52 232025 52 1 11165 5.92 52 84756 52 1 11165 5.93 52 84756 52 1 11165 5.93	562
41	81	3	8 3			14487928 58	99	97	90	92	8 O	952	52 3029*60 1 1165 3029*60 1 1165	580
1 2 3 4		1 1 1		21 18 22		831129430 51 10073341+52 586892360 51							$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
5 6 7 8 9	18 18	4 4 3 3	21 22 82 83	72 73 23 24	88	10073341*5210073341*5250000000 515000000 51	98 98 98	31 86 78 28	12 89 79 36	94 23 93 13	30 60 70 53	51 51 951 951	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
10 11 12 13 14	23 24 12 16 76	1 1 3 3	21 21 82	76 77 72 73 74	1 2 4	60600300*50 85819300*50 17520871*56 12964800*53 9884152699950	8 8 9 9	31 31 85	1 2 1 2 4 3	94 94 78	30 30 89	51 51 956	60600300+50 2 1165 51 45819300+50 2 1165 51 1456210+56 2 1165 56 107754300+52 2 1165	
15 16 17 18	76 12 17 77 77	3 3 3 3 3	83 22 82 83	75 72 73 85	1 2 4	9884152699950 17520871*56 60132100 53 9848435499950 9848435499950	9 8 5 9 9 7	92 86 86 89 52	24 89 89 71 91	56 23 23 71 07	0 0 6 0 6 0 6 0	51 51 956 52	52 1646980000 52 2 1165 51 1248304 52 2 1165 596 51 10242870*56 2 1165 598 56 352910700 52 2 1165 52 1554520000 52 2 1165	
21223	85 23 24 28	1 1 1 3	25	266 772 724		6597177 52 60600300*50 85819300*50 123341560*52	1	28	46	0.0	00	5 2 5 4	128460000 52 2 1165 6597177 52 2 1165 60600300*50 2 1165 52 85819300*50 2 1165 54 1584466*54 2 1165	604 548
26 27 28 29	11 27 14	+ 3 3 3 3 1	11 72 26 25 26	7273	2 2	310559006 50 417000000 54 6303747*50 178485328*52	99	98 28 28	41 59 459 50	5571 071 071	5370775	954 52 52 52 52	54 23940415 50 2 1165 600 52 49206*54 2 1165 4 52 27510200 54 2 1165 502 52 80978*52 2 1165 504 50 11775000*52 2 1165 604	542
31 32 33 34	5 72 82 15	333	84 13 18	222 73 73	2 1 2	25925754 54 25925754 54 730969451+52	1	23	98 96 07	0 H 7 1 3 3	15 63 41	50 *49 52	50 306058 52 2 1165 49 6217200 52 2 1165 605 52 3550940*52 2 1165 505 73633000 52 2 1165 52 1165	543 552
36 37 38 39	с <i>э</i>	3 3 3 3	21 26 22 18	72 72 72 72	2	600000000 50 1260439942 50 400431650 50 47800000 52		6 6 1 0	30 59 21 07	80 71 72 33	5 R 7 7 0 0 4 1	52 52 52 * 52	52 58226042 52 2 11165 590 52 184835 52 2 11165 591 52 831535 52 2 11165 592 52 286260 52 2 11165 593 4815100 *52 2 1165	562
41 42	81	3	83			14487928 58	99	96	48	75	29	952	52 2 11165 9 41 5089*60 2 1165	580

...

128

be wired to calculate and punch trigonometric functions and *e*-functions.

The control panel for the 604 electronic calculating unit is the same as that shown in reference 7.

The control panel for the 521 punch unit is wired to punch either table look-up or programmed cards through the punch selectors. Programmed cards are punched out of the normal side of the selectors, and table look-up cards are punched out of the transferred points. The punch selectors are picked up by means of the DI-impulse on the control panel of the IBM Type 417 Accounting Machine wired through a column split on the 521 panel to pick up pilot selectors which in turn pick up the punch selectors with a card cycles impulse through the transferred points. The punching is done out of three 417 counters into table look-up cards and four 417 counters into programmed cards. Both table look-up and programmed cards received two additional ten-place numbers from the 604 counter and general storage.

The 417 control panel is wired to utilize all the features available on the 604 control panel. In addition, a two-position shift to the right into channel A before calculation is provided; also, a two-position shift to the left into channel B before calculation is provided.

A table look-up feature is wired into the control panel whereby direct linear interpolation for any value of x can be obtained from table cards containing $\Delta y/\Delta x$, y, and Δx .

The 417 is wired to detail print, controlled by a setup change switch. This can also be controlled by an X-punch.

A negative balance test to "machine stop" is provided through a 604 negative balance selector. Also, a four condition negative balance test using two of the 417 counters simultaneously is provided. This test is used to select calculations. These features were not used for the calculations shown in this paper.

Selection is provided for net-balance or non-net balance operation for individual cards. This is controlled by an X-punch. The non-net balance operation is used for all calculations and punching, except table look-up and the conversion of a number to its absolute value.

All storage units are cleared by a Y-punch which also causes an asterisk to list. Counters 1, 2, and 3 are cleared by an X-punch. Also counters 4, 5, 6, and 7 are cleared by a different X-punch. All counters can be cleared on any card with these two X-punches.

REFERENCES

1. J. KELLEY and J. W. MISSALL, "Maneuvering Horizontal Tail Loads," *AAF TR 5185*, Army Air Forces, Air Technical Service Command, Wright Field, Dayton, Ohio (January 25, 1945).

- L. LEES and C. D. PERKINS, "Maneuver Loads on Horizontal Tail Surfaces of Airplanes," *AAF TR 4852*, Army Air Forces, Materiel Command, Wright Field, Dayton, Ohio (November 19, 1942).
- RUEL V. CHURCHILL, Modern Operational Mathematics in Engineering (New York: McGraw-Hill Book Company, Inc., 1944).
- 4. N. W. MCLACHLAN, Complex Variable and Operational Calculus with Technical Applications (Cambridge, at the University Press, New York: The MacMillan Company, 1947).
- 5. INTERNATIONAL BUSINESS MACHINES CORPORATION, Card-Programmed Electronic Calculator, Model I—Principles of Operation, Form 22-8654-0 (1951).
- 6. B. OLDFIELD, "604 Electronic Calculator Diagrams for the Calculation of Sin x, Cos x, e^x, e^{-x²}, Sinh x, and Cosh x on the Card-Programmed Electronic Calculator," International Business Machines Corporation, Technical Newsletter No. 1, Form 22-8256-0 (June, 1950).
- 7. D. B. MACMILLAN and R. H. STARK, "A General-Purpose 604 Electronic Calculator Control Panel for the Card-Programmed Electronic Calculator," *International Business Machines Corporation, Technical Newsletter No.* 1, Form 22-8256-0 (June, 1950).

DISCUSSION

Mr. Ferber: We have been doing weight and balance reports for many years on IBM equipment, but we have not felt that it was a job for the CPC. We run the finished product on the accounting machine, and from this finished product we have the start of the job for the next time by the weight group, making the changes on one copy—usually the master copy, not the carbon copy—in colored crayon. A key-punch operator will cut new cards, take the cards to a 604, multiply them in one pass, insert them in a collator, and then run the report with all the totals. We actually run many reports from these cards by sorting the cards into different sequences and running many different types of reports.

I think the development of general-purpose control panels has now reached a very high level on the 604. I believe that there is now much room for the development of general-purpose panels on the 417 and even the 521, and I would like to hear some comments, because one thing I find that is very important is the ability to emit a selected value through a co-selector.

Mr. Waterman: In regard to the weight and balance calculations, we have never had a very efficient method prior to this. We did this same job on a 602-A and accounting machines, and it took us 28 hours. We consider this to be a better way than the manual method, but we do it in two hours on the CPC, because we have another advantage when we use the CPC: We get a speed of 150 cards per minute straight through, and on the 604 we would get only 100 cards per minute. Besides, we have no need for reproducing cards or changing anything in the cards except those changes which are made by the weights section in between the monthly reports, or weekly reports, or whatever we happen to run.

Fifth-Order Aberration in an Optical System*

RUTH K. ANDERSON

National Bureau of Standards

X

THE IBM Card-Programmed Electronic Calculator has helped tremendously in the field of optical ray tracing. Until recently, the lens designer resorted to less accurate experimental methods rather than struggle with laborious hand calculations.

Assume an optical system consisting of a series of spherical surfaces of varying curvature whose centers lie in a straight line called the optical axis. Number the surfaces from left to right using the subscript k to denote an arbitrary surface. If there are F surfaces, k may be an integer from 1 to F, inclusive. Let c_k be the curvature of an arbitrary surface; the medium to the right of the kth surface is called the kth medium, and the data associated with it will have the subscript k. Thus, the index of refraction of the medium to the right of the kth surface is N_k . The distance between the kth and k+1th surface, measured along the optical axis, is given by d_k . The quantities c, d and N are the constants of the optical system. When they are known, any arbitrary ray may be traced through the system. In addition, one usually calculates the number v_k , where $\nu_k = (N_{k-1}/N_k).$

First-order data are obtained by tracing two paraxial rays through the optical system. These are the marginal and principal rays. (The subscript pr is associated with the principal ray.) Two numbers are necessary to characterize a paraxial ray; u is associated with the medium and represents the slope of the ray, and y is associated with the surface and represents the height of the ray above the optical axis. Usually u_0 $(u_0)_{pr}$, y_1 and $(y_1)_{pr}$ are given, and u_F , $(u_F)_{pr}$, y_F and $(y_F)_{pr}$ are found by alternate applications of the refraction and transfer equations:

Refraction $u_k = c_k (1-\nu_k) y_k + \nu_k u_{k-1}$ Transfer $y_{k+1} = y_k - d_k u_k$.

Data for third and fifth orders are more complicated to compute. The procedure is not difficult but long and tedious. To simplify the coding of the problem, third and fifth-order data are grouped together. The final results consist of 12 fifth-order aberrations at each surface $(a_1, a_2, \ldots, a_{12})$ and the total aberrations at each surface. Each aberration, in turn,

consists of two parts: hybrid aberration $(\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_{12})$ and intrinsic aberration $(a'_1, a'_2, \ldots, a'_{12})$.

The hybrid aberrations at a given surface are sums of products; these products are of the form, [constant] [third order coefficient] [partial sum]. Third-order coefficients are functions of the constants of both the optical system and the marginal and principal rays. The third-order coefficient B_k is a function of u_k , u_{k-1} , y_k , $(y_k)_{pr}$, c_k , v_k , and N_{k-1} . The partial sum at the *n*th surface is defined as

$$b_n = \sum_{k=1}^{n-1} B_k \, .$$

A typical example of the formula for a hybrid aberration is

 $\bar{a}_{5(k)} = 2 \left[B_k \, l_k + F_k (c_k + 2h_k + 2j_k) + C(4f_k + k_k) + E \, g_k \right],$

where capital letters designate third-order coefficients, and small letters designate partial sums.

The intrinsic aberration at a given surface is expressed in a similar manner: [function of constants of both the optical system and the marginal and principal rays] [sum of fifth-order coefficients].

$$a_{2k}' = 2 \left[c_k (y_k)_{pr} - (u_{k-1})_{pr} \right] \left[2 (A_1 y_{pr}^3 y + A_2 G_{pr}^3 G) + A_3 H_{pr}^3 + A_4 H_{pr} H \right],$$

where $A_1 = [1.5 \nu (\nu - 1) + 1] G c^3 S$

$$A_{2} = \left[\frac{1.5(1-\nu)}{\nu^{2}} + 1\right] y S$$

$$A_{3} = -S G y$$

$$A_{4} = -S \left[c y_{pr}^{2} G + y G_{pr}^{2}\right]$$

$$S = \frac{N_{k-1}(1-\nu)}{2\nu(\nu+1)^{2} I}$$

$$I = N_{k} \left[u_{k}(y_{k})_{pr} - (u_{k})_{pr} y_{k}\right]$$

$$G_{k} = \nu_{k} \left[i_{k} - u_{k}\right]$$

$$H_{k} = G + c_{k} \nu_{k}.$$

^{*}The theoretical work was developed by Mr. Donald P. Feder of the National Bureau of Standards, Optics Division.

A 60-program control panel for the IBM Type 604 Electronic Calculator is wired to perform all basic arithmetic operations with ten-digit numbers. In coding division by a ten-digit number, the approximation

$$\frac{1}{a+b} = \frac{1}{a} \left[2 - \frac{1}{a} \left(a+b \right) \right]$$

is used. Channel A is connected to electronic storage units, factor storage 1 and 2, and general storage 3. Channel B is connected to electronic storage units, factor storage 3 and 4, and general storage 4. Because of the limited amount of storage space, it is necessary to summary punch third and fifth-order coefficients (with necessary instruction) and feed them back into the machine.

Approximately 375 programming cards are needed for each surface. Including the printing of intermediate and final results, summary punching, card handling and checking, about seven minutes is required for each surface in the system.

DISCUSSION

Mr. Tillitt: There are other schemes for tracing rays. One of those is in progress at Inyokern, where a means has been developed using a sort of vector scheme for getting through glass and using a coding deck of around 80 cards. This will allow you to take skew rays or meridional rays through symmetric or asymmetric surfaces. The largest thing that has been done so far is a piece of glass which had, I believe, 140 surfaces. There was some question as to whether any light was going to get through this system; but, at any rate, the computations were carried out.

Mrs. Anderson: This is just one type of ray tracing that we do. We also trace paraxial and skew rays through spherical or aspherical surfaces, but I understand this is the first time any fifth-order work has ever been coded on the machine. I think our regular tracing of skew rays through a spherical surface takes about 80 cards also.

Theory of Elastic Vibrations of Helicopter Fuselages

PETER F. LEONE

Piasecki Helicopter Corporation

X

Summary

AN analytical method for determining the natural frequencies of an elastic helicopter fuselage system has been developed herein, with particular application to the longitudinally tandem rotor type helicopter. The Myklestad method of elastic beam vibration analysis was employed to study the bending-torsion vibrations of a non-uniform freefree U-shaped elastic beam supporting flexibly mounted engines. The U-shaped beam, whose horizontal component comprises the fuselage proper and whose vertical components comprise the pylons, assumed cantilevered at the attachment points, was considered to execute uncoupled bending vibrations in a vertical plane and coupled side bending and torsion vibrations in a horizontal plane, ignoring rotor blade motion, damping, and aerodynamic forces. An application was made to a typical helicopter fuselage system wherein an IBM electronic calculator was employed to effect the repetitive tabular calculations required to determine the natural frequencies and normal modes of the system. The machine computations were made by Mr. William P. Heising of the IBM Technical Computing Bureau of New York, employing the IBM Card-Programmed Electronic Calculator.

INTRODUCTION

Because helicopter rotors generate periodic forces in the forward flight condition, the helicopter fuselage is capable of executing elastic forced vibrations in the normal flight speed range. The fuselage response to the rotor excitations is generally manifested in uncoupled bending vibrations in a vertical plane and coupled side bending and torsion vibrations in a horizontal plane, the latter resulting from the fuselage static mass unbalance about its torsion axis. For a balanced rotor having b number of blades, the rotor excitation frequencies are found to be integer multiples of the bth harmonic of the rotor angular velocity, the major excitation being the fundamental harmonic.

The proximity of the natural frequencies of the helicopter fuselage system to the rotor excitation frequencies will, of course, dictate the degree of the fuselage forced response, the fuselage system being composed of the fuselage proper, the pylons and the flexibly mounted engines. Consequently, a determination of these natural frequencies should be made in the design stage in order that an optimum fuselage design may be attained. A vibration analysis of the helicopter fuselage system during the design stage could be either experimental (employing a dynamic structural model) or theoretical (employing linear elastic vibration theory). The present paper is concerned with the theoretical study and presents an analytical method for determining the natural frequencies of a helicopter fuselage system. In particular, the Myklestad method of elastic beam vibration analysis will be employed in determining the natural frequencies of the fuselage system of a longitudinally tandem rotor type helicopter.

PRELIMINARY ANALYSIS

In the study of the elastic vibrations of a helicopter fuselage, the fuselage system may be described by an elastic beam comprising the fuselage proper, elastic beams cantilevered to the fuselage proper comprising the pylons, and simple mass-spring systems mounted to the fuselage proper comprising the flexibly mounted engines. A schematic representation of the fuselage system of a longitudinally tandem rotor type helicopter is shown in Figure 1.



FIGURE 1

For the longitudinally tandem rotor type helicopter shown in Figure 1, the fuselage proper is capable of executing uncoupled bending vibrations in a vertical plane and coupled side bending and torsion vibrations in a horizontal plane, the latter resulting from the fuselage static mass unbalance about its torsion axis. The pylons can execute bending vibrations in the fore and aft and in the lateral senses.

In addition, the pylons can execute torsional vibrations, which, depending upon their static mass balance about their torsion axes, can couple with bending. The engine, considered to be flexibly mounted in the vertical, lateral, and roll senses, can execute uncoupled vertical vibrations and coupled lateral-roll vibrations, the latter due to a static mass unbalance about its roll axis. The resulting motion of the fuselage system will consist of a coupling of the motions of the three fuselage system components. In the vertical plane, the coupled fuselage system motion will include vertical bending of the fuselage proper, fore and aft bending of the pylons, and vertical engine motion. In the horizontal plane, the coupled fuselage system motion will include coupled side bending and torsion of the fuselage proper, coupled lateral bending and torsion of the pylons, and coupled lateral-roll engine motion. The coupled fuselage system motions are shown, for the case of infinite pylon torsional rigidities, in Figures 2 and 3.



FIGURE 2. UNCOUPLED VERTICAL BENDING



FIGURE 3. COUPLED SIDE BENDING AND TORSION

The simplest analytical study is to decouple the resulting motion of the fuselage system into the motions of its individual components and to decouple further the motions of the individual components. The uncoupled natural frequencies of the individual components are then found, from which an approximation to the characteristics of the fuselage forced response can be made by noting the proximity of the uncoupled natural frequencies to the rotor excitation frequencies. The uncoupled natural frequencies may be approximated from uniform beam theory if weighted mass and stiffness parameters are employed. For this approximation, the fuselage proper may be considered as a uniform free-free beam for bending and as a uniform free-free shaft for torsion. The pylons may be considered as uniform cantilevered beams for bending and as uniform cantilevered shafts for torsion, if both mass and stiffness are distributed. For a concentrated tip mass and distributed stiffness, the pylons may be considered as massless cantilever beams for bending and massless cantilever shafts for torsion. The uncoupled natural frequencies of the engine on its flexible mounts will, in general, be known.

Having found the approximate uncoupled natural frequencies of the fuselage system components, their representations on two frequency spectra, one for vibrations in the vertical plane and the other for vibrations in the horizontal plane, along with the representations of the rotor excitation frequencies, will disclose which component motions of the fuselage system will be of importance in the study of the fuselage forced response. The general locations of the approximate coupled natural frequencies may be found by noting that for any pair of adjacent uncoupled natural frequencies, the coupled frequencies will be such that one is to the left of the lowest uncoupled and the other to the right of the highest uncoupled frequency. This approximation appears valid, however, only if the pair of adjacent uncoupled natural frequencies is relatively isolated from the remainder of the spectrum. Typical frequency spectra for a three-bladed longitudinally tandem rotor type helicopter whose pylon torsional rigidities are infinite are shown in Figure 4, wherein all frequencies are dimensionless, being referred to the normal operating rotor speed.

As shown in Figures 4A and 4B, page 134, the rotor excitation frequencies are considered in band form to account for the range of rotor speed encountered in all possible flight conditions. The first rotor harmonic band accounts for the major excitation because of a rotor unbalance, and the third rotor harmonic band accounts for the major excitation generated by a three-bladed balanced rotor.

It is to be noted that the complete natural frequency spectra would include the quasi-rigid body modes at the low ends of the spectra which are associated with the dynamic stability of the helicopter in flight and include the effects of rotor blade motion and aerodynamic forces. These coupled quasi-rigid body motions of the fuselage system and rotor blades are of little interest in the study of the elastic fuselage forced response since their coupling with the elastic modes appears negligible. Consequently, rotor blade motion and aerodynamic forces are ignored in the present analysis.

Unfortunately, the uniform beam theory uncoupled natural frequency spectra do not accurately describe the uncoupled motions of the fuselage system components due to the non-uniformity in mass and stiffness distributions.



FIGURE 4B. COUPLED SIDE BENDING AND TORSION

LEGEND:

 $w_{T_n} = n$ th mode fuselage vertical bending frequency. $w_{S_n} = n$ th mode fuselage side bending frequency. $w_{T_n} = n$ th mode fuselage torsional frequency. $w_{11} =$ Forward pylon fundamental bending frequency. $w_{bb} =$ Rear pylon fundamental bending frequency. $w_{E_T}, w_{E_S}, w_{E_T}, =$ Engine uncoupled vertical, lateral and roll frequencies, respectively.

 Ω = Normal rotor angular velocity.

FIGURE 4. UNCOUPLED NATURAL FREQUENCIES FROM UNIFORM BEAM THEORY

The exact uncoupled natural frequencies are found to be lower than the approximate uniform beam theory frequencies, the errors being as high as 30 per cent for bending and 40 per cent for torsion. But even the exact uncoupled frequencies do not properly describe the resulting motion of the fuselage system, because the component motions couple with one another. Consequently, a rigorous analysis accounting for non-uniformity and coupling is required in order that an accurate study of the fuselage forced response be made. For this reason, the Myklestad method of elastic beam vibration analysis will be employed to determine the exact coupled natural frequencies of a non-uniform helicopter fuselage system. In particular, the Myklestad method will be applied to the fuselage system of a longitudinally tandem rotor type helicopter. Although the Myklestad method will be applied in a free vibration analysis, in which any one mode may be found independently of all other modes, it may be extended to provide a forced response analysis, the results of which would include the forced modes of response and the dynamic stress distributions.

Myklestad Analysis

In applying the Myklestad method to the analysis of the elastic fuselage vibrations of a longitudinally tandem rotor type helicopter, the fuselage system is approximated by an equivalent U-shaped elastic free-free beam composed of a finite number of sections, each section having its individual mass and stiffness properties. The horizontal component of the equivalent beam comprises the fuselage proper, and the vertical components comprise the forward and rear pylons assumed cantilevered at the attachment points. The engine is assumed to be flexibly mounted to a section of the fuselage proper, the flexibility being in the vertical, lateral and roll senses. The equivalent beam is considered to execute uncoupled bending vibrations in a vertical plane and coupled side bending and torsion vibrations in a horizontal plane, wherein the pylon torsional rigidities are assumed infinite. Rotor blade motion, damping, and aerodynamic forces are ignored, in that their effects upon the elastic motions are negligible.

Uncoupled Vertical Bending Vibrations

Now consider the equivalent U-shaped elastic beam composed of a finite number of sections to be executing free uncoupled bending vibrations in a vertical plane (Figure 5).



As shown in Figure 5, the fuselage proper is divided into a finite number of sections, each section consisting of an elastic beam having a concentrated end mass and a bending stiffness distributed over its length. The engine mass is assumed concentrated and flexibly mounted to a section of the fuselage proper, and the pylon mass centers are assumed coincident with the rotor centers, at which centers originate the constant rotor forces F_{z_0} and F'_{z_0} . Designating the concentrated masses as vibration stations consecutively numbered from forward to rear, then the tip mass of the fuselage proper becomes station 1, and its end mass becomes station b, where b designates the base. The forward and rear pylon masses are designated as stations 11 and bb, respectively. To simplify the analysis, the engine mass is considered to be part of the fuselage station mass to which it is flexibly mounted. In so doing, however, an effective engine mass must be employed whose magnitude is a function of the uncoupled engine natural frequency and the vibration frequency of the fuselage system motion (see Appendix A).

Now consider two adjacent fuselage stations, designated n and n+1. Then the slope of the bending curve, the bending deflection, the shear, and the bending moment at each station are shown in Figure 6.





The symbols employed in Figure 6 are defined as follows:

 α_n = Slope of bending curve at n.

 $y_n =$ Bending deflection at n.

 $S_n =$ Shear at n.

 M_n = Bending moment at n.

 $m_n =$ Mass concentrated at n.

 l_n = Length of *n*th section.

 x_n = Coordinate of m_n from base b.

Employing recurrence equations, the slopes and deflections of the two adjacent stations are related as follows:

$$\begin{array}{c} \alpha_{n+1} = \alpha_n - S_n v_{F_n} - M_n v_{M_n} \\ y_{n+1} = y_n - l_n \alpha_{n+1} - S_n d_{F_n} - M_n d_{M_n} \end{array} \right\}$$
(1)

where, considering the section beam to be cantilevered at n+1 and free at n, the bending elastic coefficients are defined as follows:

- v_{F_n} = Slope at *n* due to a unit force at *n*.
- v_{M_n} = Slope at *n* due to a unit moment at *n*.
- d_{F_n} = Deflection at *n* due to a unit force at *n*.
- d_{M_n} = Deflection at *n* due to a unit moment at *n*.

For a uniform cantilever beam of length l, the bending elastic coefficients become the following:

$$\begin{array}{c} v_F = l^2/2EI \\ v_M = l/EI \\ d_F = l^3/3EI \\ d_M = l^2/2EI \end{array}$$

$$(2)$$

where EI is the bending stiffness assumed constant over the beam length.

During a free vibration, the solutions for the slope, deflection, shear and bending moment at any station n take on the form:

$$\begin{array}{c} \alpha(t) = \alpha e^{i\omega t} \\ y(t) = y e^{i\omega t} \\ S(t) = S e^{i\omega t} \\ M(t) = M e^{i\omega t} \end{array}$$

$$(3)$$

where now α , y, S and M are considered as the amplitudes of the dynamic quantities during the free vibration. Denoting the amplitudes of the slope and deflection at station 1 to be ϕ and δ , respectively, then the assumption of linear elastic vibrations permits the determination of the slope and deflection at station n as linear functions of ϕ and δ . Hence, α_n and y_n may be expressed as follows:

$$\begin{array}{l} \alpha_n = \phi f_{\phi_n} - \delta f_{\delta_n} \\ y_n = -\phi g_{\phi_n} + \delta g_{\delta_n} \end{array} \right\}$$
 (4)

where f_{k_n} and g_{k_n} are the undetermined amplitude coefficients for the slope and deflection, respectively, at station n. In a similar fashion, the amplitudes of the shear and bending moment at station n take on the form:

$$S_{n} = -\phi G_{\phi_{n}} + \delta G_{\delta_{n}} \\M_{n} = -\phi G_{\phi_{n}}' + \delta G_{\delta_{n}}'$$

$$(5)$$

where G_{k_n} and G'_{k_n} are the undetermined amplitude coefficients for the shear and bending moment, respectively, at station *n*. Substituting equations 4 and 5 into the equations 1, then the recurrence equations for the slope and deflection amplitude coefficients become the following:

$$\begin{aligned}
f_{k_{n+1}} &= f_{k_n} + v_{F_n} G_{k_n} + v_{M_n} G'_{k_n} \\
g_{k_{n+1}} &= g_{k_n} + l_n f_{k_{n+1}} - d_{F_n} G_{k_n} - d_{M_n} G'_{k_n}.
\end{aligned}$$
(6)

The amplitudes of the shear and bending moment at station n are found as shown in Figure 7.



FIGURE 7

As shown in Figure 7, S_1 and M_1 are the amplitudes of the shear and bending moment at station 1, respectively, resulting from the motions of the forward pylon and the tip mass of the fuselage proper, yet to be determined, and $m_i y_i \omega^2$ is the incremental shear amplitude at station *i* resulting from the bending vibration. It follows, therefore, that S_n and M_n become the following:

$$S_{n} = S_{1} + \sum_{i=2}^{n} m_{i} y_{i} \omega^{2}$$

$$M_{n} = M_{1} + S_{1} (x_{1} - x_{n}) + \sum_{i=2}^{n-1} m_{i} y_{i} \omega^{2} (x_{i} - x_{n})$$
(7)

Substituting equations 4 and 5 into equations 7, then the recurrence equations for the shear and bending moment amplitude coefficients become the following:

$$\begin{array}{c}
G_{k_{n}} = G_{k_{1}} + \sum_{i=2}^{n} m_{i} \omega^{2} g_{k_{i}} \\
G_{k_{n}}' = G_{k_{1}}' + \sum_{i=1}^{n-1} l_{i} G_{k_{i}}
\end{array}$$
(8)

where

$$x_{i} - x_{n} = \sum_{j=i}^{n-1} l_{j} . (9)$$

The recurrence equations for the slope and deflection amplitude coefficients, given by equations 6, and the recurrence equations for the shear and bending moment amplitude coefficients, given by equations 8, formulate the tabular calculations required in the Myklestad analysis. These tabular calculations may be effected by the wellknown Myklestad tables for uncoupled beam bending vibrations (Tables I and II). The tabular calculations required in the analysis for uncoupled vertical bending free vibrations are formulated in these amplitude coefficients tables, one associated with the tip slope ϕ and the other with the tip deflection δ .

n	m	$EI imes 10^{-8}$	l	$d_F imes 10^8$	$\frac{v_F \times 10^8}{d_M \times 10^8}$	$v_{M} imes 10^{8}$
11						
1						
*						
b						
bb						1



The amplitudes of the shear and bending moment at station 1 are found from a study of the motions of the forward pylon and the tip mass of the fuselage proper, as shown in Figure 8.



FIGURE 8

It follows from Figure 8 that

$$S_{1} = (m_{1} + m_{11})y_{1}\omega^{2}$$

$$M_{1} = y_{11}(l_{11}m_{11}\omega^{2} - F_{z_{0}}) . \qquad (10)$$

The amplitudes of the slope and deflection of the forward pylon, α_{11} and y_{11} , respectively, must now be evaluated. These are found to be as follows:

$$\left. \begin{array}{l} \alpha_{11} = \alpha_1 + v_{F_{11}}(m_{11}y_{11}\omega^2 - \alpha_{11}F_{z_0}) \\ y_{11} = l_{11}\alpha_1 + d_{F_{11}}(m_{11}y_{11}\omega^2 - \alpha_{11}F_{z_0}) \end{array} \right\}$$
(11)

Now, since at station 1

then it follows from equations 4 that

$$\begin{cases} f_{\phi_1} = g_{\delta_1} = 1 \\ f_{\delta_1} = g_{\phi_1} = 0 \end{cases}$$
 (13)

The forward pylon slope and deflection amplitudes then take on the form:

$$\begin{array}{c} \alpha_{11} = \phi f_{\phi_{11}} - \delta f_{\delta_{11}} \\ y_{11} = -\phi g_{\phi_{11}} + \delta g_{\delta_{11}} \end{array} \right\}$$
(14)

where

$$\begin{cases} f_{\phi_{11}} = \{1 + m_{11}\omega^2 (l_{11}v_{F_{11}} - d_{F_{11}})\} / \Delta_{11} \\ f_{\delta_{11}} = 0 \end{cases}$$
 (15)

$$g_{\phi_{11}} = -\{l_{11} + F_{z_0}(l_{11}v_{F_{11}} - d_{F_{11}})\}/\Delta_{11} \\ g_{\delta_{11}} = 0$$
(16)

and where

$$\Delta_{11} = 1 + F_{z_0} v_{F_{11}} - m_{11} \omega^2 d_{F_{11}} .$$
⁽¹⁷⁾

Making substitutions into equations 10, then the amplitude coefficients for S_1 and M_1 become:

$$\left.\begin{array}{l}
G_{\phi_1} = 0 \\
G_{\delta_1} = (m_1 + m_{11})\omega^2
\end{array}\right\}$$
(18)

$$\left. \begin{array}{l} G_{\phi_1}' = g_{\phi_{11}} \left(l_{11} m_{11} \omega^2 - F_{z_0} \right) \\ G_{\delta_1}' = 0 \end{array} \right\}$$
(19)

It now remains only to evaluate the rear pylon boundary conditions. This is effected as shown in Figure 9.





It follows from Figure 9 that in order to meet the requirements for zero net shear and bending moment at the base section b, the following boundary condition equations must be satisfied:

$$S_{b} + m_{bb}y_{b\omega}^{2} = 0$$

$$M_{b} + y_{bb}(L_{bb}m_{bb}\omega^{2} - F'_{z_{0}}) = 0$$
(20)

The amplitudes of the rear pylon slope and deflection, α_{bb} and y_{bb} , respectively, must now be evaluated. These are found to be as follows:

$$\begin{array}{l} \alpha_{bb} = \alpha_b + v_{F_{bb}} \left(m_{bb} y_{bb} \omega^2 - \alpha_{bb} F'_{z_0} \right) \\ y_{bb} = l_{bb} \alpha_b + d_{F_{bb}} \left(m_{bb} y_{bb} \omega^2 - \alpha_{bb} F'_{z_0} \right) \end{array} \right\}$$

$$(21)$$

The solutions for a_{bb} and y_{bb} then become:

$$\begin{array}{l} \alpha_{bb} = \phi f_{\phi_{bb}} - \delta f_{\delta_{bb}} \\ y_{bb} = -\phi g_{\phi_{bb}} + \delta g_{\delta_{bb}} \end{array} \right\}$$
(22)

where

$$\begin{cases} f_{\phi_{bb}} = \gamma_1 f_{\phi_b} \\ f_{\delta_{bb}} = \gamma_1 f_{\delta_b} \end{cases}$$

$$(23)$$

$$\begin{array}{c}
g_{\phi_{bb}} = -\lambda_{1} f_{\phi_{b}} \\
g_{\delta_{bb}} = -\lambda_{1} f_{\delta_{b}}
\end{array} \right\}$$
(24)

and where

$$\gamma_{1} = \{ 1 + m_{bb} \omega^{2} (l_{bb} v_{F_{bb}} - d_{F_{bb}}) \} / \Delta_{bb}$$

$$\lambda_{1} = \{ l_{bb} + F'_{z_{0}} (l_{bb} v_{F_{bb}} - d_{F_{bb}}) \} / \Delta_{bb}$$

$$(25)$$

$$\Delta_{bb} = 1 + F'_{z_0} v_{F_{bb}} - m_{bb} \omega^2 d_{F_{bb}} . \qquad (26)$$

The boundary conditions given by equations 20 may be expressed in the form:

$$\left.\begin{array}{l}
S'_{b} = -\phi V_{\phi} + \delta V_{\delta} = 0 \\
M'_{b} = -\phi M_{\phi} + \delta M_{\delta} = 0
\end{array}\right\}$$
(27)

where S'_b and M'_b are the net shear and bending moment,

respectively, at station b which must be zero concurrently, and where their amplitude coefficients are defined as:

$$\left. \begin{array}{l} V_{\phi} = G_{\phi_b} + m_{bb}\omega^2 g_{\phi_b} \\ V_{\delta} = G_{\delta_b} + m_{bb}\omega^2 g_{\delta_b} \end{array} \right\}$$
(28)

If, now, the tip deflection δ is made unity for normalization purposes, then the tip slope ϕ becomes, from the net shear equation, the following:

$$\phi = V_{\delta}/V_{\phi} \ . \tag{30}$$

Substituting, the net bending moment then becomes:

$$M'_{b} = -M_{\phi}(V_{\delta}/V_{\phi}) + M_{\delta} = 0 .$$
 (31)

The above equation 31, which expresses the zero net bending moment at the fuselage base, becomes the criterion for the existence of a natural frequency in uncoupled vertical bending free vibrations. The tabular calculations must be repeated for assumed values of natural frequencies until the criterion is satisfied. The solutions for the slope, deflection, shear and bending moment at a natural frequency then become simply:

$$\begin{array}{c} \alpha_{n} = \phi f_{\phi_{n}} - f_{\delta_{n}} \\ y_{n} = -\phi g_{\phi_{n}} + g_{\delta_{n}} \\ S_{n} = -\phi G_{\phi_{n}} + G_{\delta_{n}} \\ M_{n} = -\phi G'_{\phi_{n}} + G'_{\delta_{n}} \end{array}$$

$$(32)$$

The foregoing equations 32 express the final solutions for uncoupled vertical bending free vibrations. These solutions are valid only at a natural frequency of vibration which satisfies the criterion of zero net bending moment at the fuselage base given by equation 31.

Coupled Side Bending and Torsion Vibrations

Now consider the equivalent U-shaped elastic beam composed of a finite number of sections to be executing coupled side bending and torsion free vibrations in a horizontal plane, shown in Figure 10.



FIGURE 10

As shown in Figure 10, the individual section beam is now considered to have its concentrated mass offset from the torsion axis and to have both bending and torsional stiffnesses distributed over its length. As before, the engine mass is assumed concentrated and to be flexibly mounted to a fuselage station, the flexibility being in both the lateral and roll senses. The engine mass will again be considered as part of the fuselage station mass to which it is flexibly mounted, where now its effective value will be a function of both its uncoupled lateral and uncoupled roll natural frequencies and the vibration frequency of the fuselage system motion (see Appendix A).

Now consider two adjacent fuselage stations, n and n+1. Then the slope of the bending curve, the bending deflection, the twist, the shear, the bending moment and the torque at each station are shown as in Figure 11.



FIGURE 11

In addition to the symbols employed previously, the symbols of Figure 11 are defined as follows:

- $\theta_n = \text{Twist angle at } n.$
- $T_n = \text{Torque at } n.$
- $J_n =$ Mass polar moment of inertia about torsion axis at n.
- r_n = Vertical offset of m_n from torsion axis.

In addition to the recurrence equations for the slopes and deflections of two adjacent stations given by equations 1, the recurrence equation for the twist angles of the two adjacent stations becomes:

$$\theta_{n+1} = \theta_n - T_n v_{T_n} \tag{33}$$

where, considering the section as a uniform shaft cantilevered at n+1 and free at n, the twist elastic coefficient v_T is defined simply as:

$$v_T = l/GI_p \tag{34}$$

where GI_p is the torsional stiffness assumed constant over the section length.

In addition to the form of the solutions for the slope, deflection, shear and bending moment, given by equations 3, the solutions for the twist and torque at station n take on the form:

$$\begin{array}{c} \theta(t) = \theta e^{i\omega t} \\ T(t) = T e^{i\omega t} \end{array}$$
 (35)

where θ and T are considered as the amplitudes of the dynamic quantities during the free vibration.

Denoting the amplitudes of the slope, deflection and twist at station 1 to be ϕ , δ and ψ , respectively, then the assumption of linear elastic vibrations permits the determination of α , y, and θ at n as linear functions of ϕ , δ , and ψ . Hence,

$$\begin{array}{l} \alpha_{n} = & \phi f_{\phi_{n}} - \delta f_{\delta_{n}} - \psi f_{\psi_{n}} \\ y_{n} = & -\phi g_{\phi_{n}} + \delta g_{\delta_{n}} + \psi g_{\psi_{n}} \\ \theta_{n} = & -\phi h_{\phi_{n}} + \delta h_{\delta_{n}} + \psi h_{\psi_{n}} \end{array}$$

$$(36)$$

where f_{k_n} and g_{k_n} are defined as before and where h_{k_n} are the undetermined amplitude coefficients for the twist at n. In a similar fashion, the amplitudes of the shear, bending moment and torque at n take on the form:

$$S_{n} = -\phi G_{\phi_{n}} + \delta G_{\delta_{n}} + \psi G_{\psi_{n}}$$

$$M_{n} = -\phi G_{\phi_{n}}' + \delta G_{\delta_{n}}' + \psi G_{\psi_{n}}'$$

$$T_{n} = -\phi H_{\phi_{n}} + \delta H_{\delta_{n}} + \psi H_{\psi_{n}}$$
(37)

where G_{k_n} and G'_{k_n} are defined as before, and H_{k_n} are the undetermined amplitude coefficients for the torque at n.

In addition to the recurrence equations for the slope and deflection amplitude coefficients given by equations 6, the recurrence equation for the twist amplitude coefficients becomes, upon substituting θ_n of equations 36 and T_n of equations 37 into equation 33, the following:

$$h_{k_{n+1}} = h_{k_n} - v_{T_n} H_{k_n} \tag{38}$$

The amplitudes of the shear, bending moment, and torque at station n are found as shown in Figure 12.



FIGURE 12

139

As shown in Figure 12, S_1 , M_1 , and T_1 are the amplitudes of the shear, bending moment, and torque at station 1, respectively, resulting from the motions of the forward pylon and the tip mass of the fuselage proper. Also acting at station 1 is the vertical shear F_{z_0} , being the shear transmitted by the constant forward rotor force. Also, the incremental shear and torque amplitudes at station *i* are found to be $m_i\omega^2(y_i-r_i\theta_i)$ and $\omega^2(J_i\theta_i-m_ir_iy_i)$, respectively. It follows, therefore, that S_n , M_n , and T_n become the following:

$$S_{n} = S_{1} + \sum_{i=2}^{n} m_{i}\omega^{2}(y_{i} - r_{i}\theta_{i})$$

$$M_{n} = M_{1} + S_{1}(x_{1} - x_{n}) + \sum_{i=2}^{n-1} m_{i}\omega^{2}(y_{i} - r_{i}\theta_{i})(x_{i} - x_{n})$$

$$T_{n} = T_{1} + \sum_{i=2}^{n} \omega^{2}(J_{i}\theta_{i} - m_{i}r_{i}y_{i}) - F_{z_{0}}(y_{1} - y_{n}).$$
(39)

Substituting equations 36 and 37 into equations 39, then the recurrence equations for the shear, bending moment and torque amplitude coefficients become the following:

$$G_{k_{n}} = G_{k_{1}} + \sum_{i=2}^{n} m_{i}\omega^{2}(g_{k_{i}} - r_{i}h_{k_{i}})$$

$$G'_{k_{n}} = G'_{k_{1}} + \sum_{i=1}^{n-1} l_{i}G_{k_{i}}$$

$$H_{k_{n}} = H_{k_{1}} + \sum_{i=2}^{n} \omega^{2}(J_{i}h_{k_{i}} - m_{i}r_{i}g_{k_{i}}) + \sum_{i=1}^{n-1} F_{z_{0}}\Delta g_{k_{i}}$$

$$(40)$$

where

$$\begin{array}{c}
g_{k_{n}} - g_{k_{1}} = \sum_{i=1}^{n-1} \Delta g_{k_{i}} \\
\Delta g_{k_{n}} = g_{k_{n+1}} - g_{k_{n}}.
\end{array}$$
(41)

The recurrence equations for the slope and deflection amplitude coefficients, given by equations 6, the recurrence equation for the twist amplitude coefficients, given by equation 38, and the recurrence equations for the shear, bending moment, and torque amplitude coefficients, given by equations 40 and 41, formulate the tabular calculations required in the Myklestad analysis. These tabular calculations may be effected by the well-known Myklestad tables for coupled bending-torsion beam vibrations (Tables III and IV).

The tabular calculations required in the analysis for coupled side bending and torsion free vibrations are formulated in the amplitude coefficients tables, their being three such tables, these being associated with the tip slope ϕ , the tip deflection δ , and the tip twist ψ .

	<i>n</i>	m	r	mr	J	$EI \times 10^{-8}$	$GI_p \times 10^{-8}$	
	11							
	1							
	*							
	b							
	bb							
	2 /108	2 /1 08	7.9/108		1 108	$v_F imes 10^8$		10
	$m_{\rm c}^2/10^8$	$mr\omega^2/10^8$	$J \omega^2 / 10^8$	l	$d_F imes 10^8$	$\frac{c_F \times 10}{d_M \times 10^8}$	$v_{\rm M} imes 10^8$	$v_{T} imes 10^{6}$
n	<i>m</i> w / 10							
n 11								······································
n 11 1								
n 11 1 ♥						· · · · · · · · · · · · · · · · · · ·		
n 11 1 ♥ b	<i>m</i> w / 10							

bth Tak	TABLE IV. AMPLITUDE COEFFICIENTS								
nui i di		2	3	4	(5)	6	7		
n	$g_{k} = \begin{array}{c} 1 \\ + \end{array}$ for $n-1$	$h_k = 2 \\ -v_T \times 5$	$G_{k} \times 10^{-8} = \sum_{n}^{\infty^{2}} \frac{m\omega^{2}}{10^{8}} \times (1)$ $-\sum_{n}^{\infty^{2}} \frac{mr\omega^{2}}{10^{8}} \times (2)$	$G'_k \times 10^{-8} = \sum_{\substack{l \\ \text{for } n-1}}^{l \times (3)} $	$H_{k} \times 10^{-8} =$ $\sum_{i=1}^{k} \frac{J_{\omega^{2}}}{10^{8}} \times (2)$ $-\frac{mr\omega^{2}}{10^{8}} \times (1)$ $+ \sum_{i=1}^{k} F_{z_{0}} \times 10^{-8} \times (7)$ for $n-1$	$f_{k_{n+1}} = f_{k_n}$ $+ v_F \times (3)$ $+ v_M \times (4)$	$\Delta g_{k} = l \times \textcircled{6} \\ - d_{F} \times \textcircled{3} \\ - d_{M} \times \textcircled{4}$		
1	g_{k_1}	h_{k_1}	G_{k_1}	G' _{k1}	H_{k_1}	f_{k_2}	Δg_{k_1}		
*									
Ъ									

The amplitudes of the shear, bending moment, and torque at station 1 are found from a study of the motions of the forward pylon and the tip mass of the fuselage proper, as shown in Figures 13A and 13B.

 θ_{11} y_{11}







It follows from Figures 13A and 13B that

$$S_{1} = m_{1}\omega^{2}(y_{1} - r_{1}\theta_{1}) + m_{11}y_{11}\omega^{2}$$

$$M_{1} = 0$$

$$T_{1} = J_{1}\omega^{2}\theta_{1} - m_{1}r_{1}y_{1}\omega^{2} + l_{11}m_{11}y_{11}\omega^{2} - F_{\varepsilon_{0}}(y_{11} - y_{1}).$$
(42)

The amplitudes of the slope and deflection of the forward pylon, θ_{11} and y_{11} , respectively, must now be evaluated. These are found to be as follows:

$$\theta_{11} = \theta_1 + v_{F_{11}}(m_{11}y_{11}\omega^2 - \theta_{11}F_{z_0}) y_{11} = y_1 + l_{11}\theta_1 + d_{F_{11}}(m_{11}y_{11}\omega^2 - \theta_{11}F_{z_0}) .$$
 (43)

Now, since at station 1

$$\begin{array}{c} \alpha_1 = \phi \\ y_1 = \delta \\ \theta_1 = \psi \end{array} \right\}$$

$$(44)$$

then it follows from equations 36 that

$$\begin{cases} f_{\phi_1} = g_{\delta_1} = h_{\psi_1} = 1 \\ f_{\delta_1} = g_{\psi_1} = h_{\phi_1} = 0 \\ f_{\psi_1} = g_{\phi_1} = h_{\delta_1} = 0 \\ \end{cases}$$

$$(45)$$

The forward pylon slope and deflection amplitudes then become:

$$\begin{array}{l} \theta_{11} = -\phi h_{\phi_{11}} + \delta h_{\delta_{11}} + \psi h_{\psi_{11}} \\ y_{11} = -\phi g_{\phi_{11}} + \delta g_{\delta_{11}} + \psi g_{\psi_{11}} \end{array} \right\}$$
(46)

where

$$\begin{cases}
g_{\phi_{11}} = 0 \\
g_{\delta_{11}} = (1 + F_{z_0} v_{F_{11}}) / \Delta_{11} \\
g_{\psi_{11}} = \{l_{11} + F_{z_0} (l_{11} v_{F_{11}} - d_{F_{11}})\} / \Delta_{11}
\end{cases}$$
(47)

$$\left. \begin{array}{l} h_{\phi_{11}} = 0 \\ h_{\delta_{11}} = m_{11}\omega^2 v_{F_{11}}/\Delta_{11} \\ h_{\psi_{11}} = \{1 + m_{11}\omega^2 (l_{11}v_{F_{11}} - d_{F_{11}})\}/\Delta_{11} \end{array} \right\}$$
(48)

and where

$$\Delta_{11} = 1 + F_{z_0} v_{F_{11}} - m_{11} \omega^2 d_{F_{11}} .$$
⁽⁴⁹⁾

Making substitutions into equations 42, then the amplitude coefficients for S_1 , M_1 , and T_1 become the following:

$$\begin{cases}
G_{\phi_1} = 0 \\
G_{\delta_1} = m_1 \omega^2 + m_{11} \omega^2 g_{\delta_{11}} \\
G_{\psi_1} = -m_1 r_1 \omega^2 + m_{11} \omega^2 g_{\psi_{11}}
\end{cases}$$
(50)

$$\begin{array}{cccc}
G'_{\phi_1} = 0 \\
G'_{\delta_1} = 0 \\
G'_{\psi_1} = 0
\end{array}$$
(51)

The rear pylon boundary conditions are now found as shown in Figures 14A and 14B.



FIGURE 14A



FIGURE 14B

It follows from Figures 14A and 14B that in order to meet the requirements for zero net shear, bending moment and torque and the base section b, the following boundary condition equations must be satisfied:

$$\begin{cases}
S_{b} + m_{bb}y_{bb}\omega^{2} = 0 \\
M_{b} = 0 \\
T_{b} + l_{bb}m_{bb}y_{bb}\omega^{2} - F'_{z_{0}}(y_{bb} - y_{b}) = 0 .
\end{cases}$$
(53)

The amplitudes of the slope and deflection of the rear pylon, θ_{bb} and y_{bb} respectively, must now be evaluated. These are found to be as follows:

$$\theta_{bb} = \theta_b + v_{F_{bb}} \left(m_{bb} y_{bb} \omega^2 - \theta_{bb} F'_{z_0} \right) y_{bb} = y_b + l_{bb} \theta_b + d_{F_{bb}} \left(m_{bb} y_{bb} \omega^2 - \theta_{bb} F'_{z_0} \right) .$$

$$(54)$$

The solutions for θ_{bb} and y_{bb} then become:

$$\theta_{bb} = -\phi h_{\phi_{bb}} + \delta h_{\delta_{bb}} + \psi h_{\psi_{bb}} y_{bb} = -\phi g_{\phi_{bb}} + \delta g_{\delta_{bb}} + \psi g_{\psi_{bb}}$$

$$(55)$$

where

$$\begin{array}{l}
 h_{\phi_{bb}} = \xi_1 g_{\phi_b} + \xi_2 h_{\phi_b} \\
 h_{\delta_{bb}} = \xi_1 g_{\delta_b} + \xi_2 h_{\delta_b} \\
 h_{\psi_{bb}} = \xi_1 g_{\psi_b} + \xi_2 h_{\psi_b}
\end{array}$$
(56)

$$\left.\begin{array}{l}g_{\phi_{bb}} = \rho_{1}g_{\phi_{b}} + \rho_{2}h_{\phi_{b}}\\g_{\delta_{bb}} = \rho_{1}g_{\delta_{b}} + \rho_{2}h_{\delta_{b}}\\g_{\psi_{bb}} = \rho_{1}g_{\psi_{b}} + \rho_{2}h_{\psi_{b}}\end{array}\right\}$$
(57)

and where

$$\rho_{1} = (1 + F'_{z_{0}} v_{F_{bb}}) / \Delta_{bb} \rho_{2} = \rho_{1} l_{bb} - F'_{z_{0}} d_{F_{bb}} / \Delta_{bb}$$

$$(58)$$

$$\left. \begin{cases} \xi_1 = d_{F_{bb}} / \Delta_{bb} \\ \xi_2 = \{ 1 + m_{bb} \omega^2 (l_{bb} v_{F_{bb}} - d_{F_{bb}}) \} / \Delta_{bb} \end{cases} \right\}$$
(59)

$$\Delta_{bb} = 1 + F'_{z_0} v_{F_{bb}} - m_{bb} \omega^2 d_{F_{bb}} .$$
 (60)

The boundary conditions given by equations 53 may be expressed in the form:

$$S'_{b} = -\phi V_{\phi} + \delta V_{\delta} + \psi V_{\psi} = 0$$

$$M'_{b} = -\phi M_{\phi} + \delta M_{\delta} + \psi M_{\psi} = 0$$

$$T'_{b} = -\phi T_{\phi} + \delta T_{\delta} + \psi T_{\psi} = 0$$

$$(61)$$

where S'_b , M'_b and T'_b are the net shear, bending moment and torque, respectively, at station b which must be zero concurrently, and where their amplitude coefficients are defined as:

$$\left.\begin{array}{l}
V_{\phi} = G_{\phi_b} + m_{bb}\omega^2 g_{\phi_{bb}} \\
V_{\delta} = G_{\delta_b} + m_{bb}\omega^2 g_{\delta_{bb}} \\
V_{\psi} = G_{\psi_b} + m_{bb}\omega^2 g_{\psi_{bb}}
\end{array}\right\}$$
(62)

$$\left.\begin{array}{l}
M_{\phi} = G'_{\phi_b} \\
M_{\delta} = G'_{\delta_b} \\
M_{\psi} = G'_{\psi_b}
\end{array}\right\}$$
(63)

$$T_{\phi} = H_{\phi_{b}} + l_{bb}m_{bb}\omega^{2}g_{\phi_{bb}} - F'_{z_{0}}(g_{\phi_{bb}} - g_{\phi_{b}})$$

$$T_{\delta} = H_{\delta_{b}} + l_{bb}m_{bb}\omega^{2}g_{\delta_{bb}} - F'_{z_{0}}(g_{\delta_{bb}} - g_{\delta_{b}})$$

$$T_{\psi} = H_{\psi_{b}} + l_{bb}m_{bb}\omega^{2}g_{\psi_{bb}} - F'_{z_{0}}(g_{\psi_{bb}} - g_{\psi_{b}}).$$
(64)

If now the tip deflection δ is made unity for normalization purposes, then the tip slope ϕ and the tip twist ψ become, from the solutions of the shear and torque equations, the following:

$$\phi = (V_{\psi}T_{\delta} - T_{\psi}V_{\delta})/(V_{\psi}T_{\phi} - T_{\psi}V_{\phi}) \psi = (V_{\phi}T_{\delta} - T_{\phi}V_{\delta})/(V_{\psi}T_{\phi} - T_{\psi}V_{\phi}) .$$

$$(65)$$

The solutions for ϕ and ψ must satisfy the zero net bending moment equation, namely

$$M'_{b} = -\phi M_{\phi} + \psi M_{\psi} + M_{\delta} = 0.$$
 (66)

The above equation 66, expressing the zero net bending moment at the fuselage base section, becomes the criterion for the existence of a natural frequency in coupled side bending and torsion free vibrations. The tabular calculations must be repeated for assumed values of natural frequencies until the criterion is satisfied. The solutions for the slope, deflection, twist, shear, bending moment and torque at a natural frequency then become simply:

$$\begin{array}{l} \alpha_{n} = \phi f_{\phi_{n}} - \psi f_{\psi_{n}} - f_{\delta_{n}} \\ y_{n} = -\phi g_{\phi_{n}} + \psi g_{\psi_{n}} + g_{\delta_{n}} \\ \theta = -\phi h_{\phi_{n}} + \psi h_{\psi_{n}} + h_{\delta_{n}} \\ S_{n} = -\phi G_{\phi_{n}} + \psi G_{\psi_{n}} + G_{\delta_{n}} \\ M_{n} = -\phi G_{\phi_{n}} + \psi G_{\psi_{n}} + G_{\delta_{n}} \\ T_{n} = -\phi H_{\phi_{n}} + \psi H_{\psi_{n}} + H_{\delta_{n}} \end{array}$$

$$(67)$$

The foregoing equations 67 express the final solutions for coupled side bending and torsion free vibrations. These solutions are valid only at a natural frequency of vibration which satisfies the criterion of zero net bending moment at the fuselage base given by equation 66.

Application of Myklestad Analysis

The results of an application of the Myklestad analysis to a three-bladed longitudinally tandem rotor type helicopter are now presented. These results include the solutions for the first four normal modes of uncoupled vertical bending and the first three normal modes of coupled side bending and torsion free vibrations of an equivalent U-shaped elastic beam supporting a flexibly mounted engine and having infinite pylon torsional rigidities. The Myklestad solutions for the exact coupled natural frequencies are summarized in the following dimensionless frequency spectra (Figure 15) which include the approximate uncoupled natural frequencies derived from uniform beam theory (Figure 4).



FIGURE 15A. UNCOUPLED VERTICAL BENDING



FIGURE 15B. COUPLED SIDE BENDING AND TORSION

FIGURE 15. COUPLED NATURAL FREQUENCIES FROM MYKLESTAD SOLUTION

It was noted earlier that in order to effect the Myklestad analysis for free vibrations, the tabular calculations must be repeated for assumed values of natural frequencies until the criterion for the existence of a normal mode is satisfied. This criterion is formulated in the boundary condition requiring zero net bending moment at the fuselage base, expressed by equation 31 for uncoupled vertical bending and equation 66 for coupled side bending and torsion. To facilitate the solution, it is found convenient to plot the function M'_b/ω^2 versus ω , its zeros then being the criteria for the existence of natural frequencies. The residual moment functions associated with the results presented herein are shown in Figures 16A and 16B.



FIGURE 16A. UNCOUPLED VERTICAL BENDING



FIGURE 16B. COUPLED SIDE BENDING AND TORSION

FIGURE 16. RESIDUAL MOMENT FUNCTIONS

The asymptotes shown in the foregoing plots of the residual moment functions do not, of course, denote natural frequencies of the system. Rather, they denote those frequencies of vibration at which the amplitudes of the pylon and engine motions, and the amplitudes of the torsion motion of the fuselage proper, become infinite for finite amplitudes of bending motion of the fuselage proper. These asymptote frequencies are associated with the uncoupled natural frequencies of the motions of the fuselage system components other than the bending motion of the fuselage proper. The asymptote frequencies, however, do not correspond exactly with the uncoupled natural frequencies of the aforementioned motions due to their coupling with one another and with the bending motion of the fuselage proper. In the plot of the residual moment function for uncoupled vertical bending, shown in Figure 16A, the first asymptote is associated with the engine vertical motion, the second with the rear pylon fore and aft bending, and the third with the forward pylon fore and aft bending. In the plot of the residual moment function for coupled side bending and torsion, shown in Figure 16B, the first asymptote is associated with the engine lateral motion, and the second with the rear pylon lateral bending. Further calculations would vield two additional asymptote frequencies, the first associated with the first torsion mode of the fuselage proper, and the second with the forward pylon lateral bending motion. Calculations beyond this point would yield asymptote frequencies associated with the higher torsion modes of the fuselage proper. Calculations at the low frequencies should yield an asymptote associated with the engine roll motion. In general, it may be said that the total number of asymptotes equals the total number of uncoupled modes of each of the fuselage system components other than bending of the fuselage proper, when employing the criterion of zero bending moment at the base of the fuselage proper to determine the natural frequencies of the helicopter fuselage system.

One final note to be made is the existence of natural frequencies at the low ends of the frequency spectra. For uncoupled vertical bending, there will exist a natural frequency of the order of .01 to .1 associated with the quasirigid body pitch motion of the fuselage system. For coupled side bending and torsion, there will exist a natural frequency of the above order associated with the quasi-rigid body roll motion of the fuselage system. These quasi-rigid body motions arise from the pendular effects of the constant rotor thrusts.

REFERENCES

- N. O. MYKLESTAD, Vibration Analysis (New York: McGraw-Hill Book Company, 1944).
 J. P. DEN HARTOG, Mechanical Vibrations (New York: Mc-
- J. P. DEN HARTOG, Mechanical Vibrations (New York: Mc-Graw-Hill Book Company, 1940).

Appendix A

EFFECTIVE MASS PROPERTIES OF A FLEXIBLY MOUNTED ENGINE

Uncoupled Vertical Bending Vibrations

Consider an engine of mass m_e to be concentrated at a fuselage station, at which station the concentrated fuselage mass is m_f (less engine mass). Also, consider the engine to be flexibly mounted to the fuselage with a spring of constant k_{ξ} and consider the fuselage station spring constant to be k_y . The system may then be represented as a simple two-degree-of-freedom vibratory system as follows:



FIGURE A1

where, in Figure A1, ξ and y are the linear displacements of m_e and m_f , respectively. The equations of motion become:

$$y: m_f \ddot{y} + k_y y + k_{\xi} (y - \xi) = 0 \\ \xi: m_e \ddot{\xi} + k_{\xi} (\xi - y) = 0. \end{cases}$$
(A-1)

For harmonic vibrations of a frequency ω , the solutions are of the form

$$\begin{array}{c} y = y_0 \sin \omega t \\ \xi = \xi_0 \sin \omega t \end{array}$$
 (A-2)

whence it follows that

$$\xi_0 = \mu_\xi \, y_0 \tag{A-3}$$

$$\mu_{\xi} = 1/(1-\beta_{\xi}^2)$$
 $\beta_{\xi} = \omega/\omega_{\xi}$ $\omega_{\xi}^2 = k_{\xi}/m_e$. (A-4)

The equation of motion with respect to y may then be expressed as

$$m_f + m_e \mu_{\xi}) \ddot{y} + k_y y = 0.$$
 (A-5)

Therefore, in uncoupled vertical bending vibrations, the total effective mass at a fuselage station at which is concentrated a flexibly mounted engine becomes:

$$m = m_f + m_e \mu_{\xi} \tag{A-6}$$

where the amplification factor μ_{ξ} is a function of the vibration frequency ω , and the engine uncoupled vertical natural frequency, ω_{ξ} .

Coupled Side Bending and Torsion Vibrations

(

In a similar fashion consider the engine to be mounted flexibly in both the lateral and torsional senses with the engine torsion axis located above the fuselage torsion axis as shown in Figure A2.





where, in Figure A2, k_y and k_θ are the lateral and torsional spring constants of the fuselage station whose mass m_f (less engine) is located a distance r_f below the fuselage torsion axis and whose mass polar moment of inertia about the fuselage torsion axis is J_f , and k_δ and k_ϕ are the lateral and torsional spring constants of the engine whose mass m_e is located a distance h above the fuselage torsion axis and whose mass polar moment of inertia is J_e about its roll axis likewise offset a distance h above the fuselage torsion axis. Also, y and θ are the linear and angular displacements, respectively, of the fuselage station, and δ and ϕ are the linear and angular displacements of the engine, respectively, taken with respect to the fuselage.

The equations of motion then become:

$$y: \quad \ddot{y}(m_{f}+m_{e}) + \ddot{\theta}(-m_{f}r_{f}+m_{e}h) + \ddot{\delta}m_{e} + k_{y}y = 0 + \ddot{\delta}m_{e} + k_{y}y = 0$$

$$\theta: \quad \ddot{\theta}(J_{f}+J_{e}+m_{e}h^{2}) + \ddot{y}(-m_{f}r_{f}+m_{e}h) + \ddot{\delta}m_{e}h + \ddot{\phi}J_{e} + k_{\theta}\theta = 0$$

$$\theta: \quad \ddot{\delta}m_{e} + \ddot{y}m_{e} + \ddot{\theta}m_{e}h + k_{\delta}\delta = 0$$

$$\phi: \quad \ddot{\phi}J_{e} + \ddot{\theta}J_{e} + k_{\phi}\phi = 0 .$$

$$(A-7)$$

For harmonic vibrations, the solutions take the form:

$$y = y_0 \sin \omega t$$

$$\theta = \theta_0 \sin \omega t$$

$$\delta = \delta_0 \sin \omega t$$

$$\phi = \phi_0 \sin \omega t$$

(A-8)

whence it follows that

$$\phi_{0} = \theta_{0}\mu_{\phi}\beta_{\phi}^{2} \\ \delta_{0} = \mu_{\delta}\beta_{\delta}^{2}(y_{0} + h\theta_{0})$$
 (A-9)

where

 $\mu_{\phi} = 1/(1-\beta_{\phi}^2), \quad \beta_{\phi} = \omega/\omega_{\phi}, \quad \omega_{\phi}^2 = k_{\phi}/J_e \\ \mu_{\delta} = 1/(1-\beta_{\delta}^2), \quad \beta_{\delta} = \omega/\omega_{\delta}, \quad \omega_{\delta}^2 = k_{\delta}/m_e . \end{cases}$ (A-10)

The equations of motion with respect to y and θ may then be expressed as:

$$y: \quad \ddot{y}(m_{f}+m_{e}\mu_{\delta}) + k_{y}y - \theta(m_{f}r_{f}-m_{e}h\mu_{\delta}) = 0$$

$$\theta: \quad \ddot{\theta}(J_{f}+J_{e}\mu_{\phi}+m_{e}h^{2}\mu_{\delta}) + k_{\theta}\theta$$

$$- \ddot{y}(m_{f}r_{f}-m_{e}h\mu_{\delta}) = 0$$
 (A-11)

Therefore, in coupled side bending and torsion vibrations, the total mass, static mass moment and mass polar moment of inertia about the fuselage torsion axis at a fuselage station at which is concentrated a flexibly mounted engine become:

$$\begin{array}{c} m = m_f + m_e \mu_\delta \\ rm = m_f r_f - m_e h \mu_\delta \\ J = J_f + J_e \mu_\phi + m_e h^2 \mu_\delta \end{array}$$
 (A-12)

where the amplification factors μ_{δ} and μ_{ϕ} are functions of the vibration frequency ω and the engine uncoupled natural frequencies in the lateral and roll senses, ω_{δ} and ω_{ϕ} , respectively.

DISCUSSION

[This paper and the next were discussed as a unit.]

Appendix B

UNIFORM BEAM THEORY FORMULAS FOR UNCOUPLED NATURAL FREQUENCIES

Beam Type	Natural Frequent rad./sec.	cy Mode	Frequency Coefficient	Modal Shape
Uniform free-free beam	$\omega_n = a_n \sqrt{\frac{EI}{\mu l^4}}$	1st bending 2nd bending 3rd bending	$a_1 = 22.4$ $a_2 = 61.7$ $a_3 = 121.0$	
Uniform free-free shaft	$\omega_n = b_n \sqrt{\frac{GI_p}{\lambda l^2}}$	1st torsion 2nd torsion 3rd torsion	$b_1 = \pi$ $b_2 = 2\pi$ $b_3 = 3\pi$	
Uniform cantilever beam	$\omega_n = c_n \sqrt{\frac{EI}{\mu l^4}}$	1st bending 2nd bending 3rd bending	$c_1 = 3.52$ $c_2 = 22.4$ $c_3 = 61.7$	
Uniform cantilever shaft	$\omega_n = d_n \sqrt{\frac{GI_p}{\lambda l^2}}$	1st torsion 2nd torsion 3rd torsion	$d_1 = \pi/2$ $d_2 = 3\pi/2$ $d_3 = 5\pi/2$	
Massless cantilever beam with tip mass	$\omega = \sqrt{\frac{3EI}{ml^3}}$	Fundamental bending	unity	<u>}</u>
Massless cantilever shaft with tip inertia	$\omega = \sqrt{\frac{GI_p}{Jl}}$	Fundamental torsion	unity	<u>}</u>
LEGEND: EI = Bending stiffness $GI_p =$ Torsional stiffness m = Mass		 Mass polar momen of inertia Beam length 	$\begin{array}{cc} \mu = m \\ \lambda = J \\ \text{of} \end{array}$	/L, mass per unit length (L, mass polar moment inertia per unit length
Machine Procedure for Computation of Elastic Vibrations of Helicopter Fuselages

WILLIAM P. HEISING

International Business Machines Corporation

×

PRELIMINARY ANALYSIS

T H E system of equations developed in the previous paper is a difference system approximating a system of differential equations. If the fuselage section properties are assumed constant, and the corresponding differential equations are examined, a crude analysis shows that the various dependent variables may be expected to increase in an exponential manner along the fuselage. The curve will be influenced by the value of the parameter, ω^2 .

The modal shapes are developed from the dependent variables f, g, and h (which increase exponentially along the fuselage) by the following equations:

Uncoupled Vertical Bending:

Deflection: $y_n = -\Phi g_{\phi_n} + g_{\delta_n}$

Coupled Side Bending and Torsion:

Deflection: $y_n = -\Phi g_{\phi_n} + g_{\delta_n} + \Psi g_{\psi_n}$ Twist Angle: $\theta_n = -\Phi h_{\phi_n} + h_{\delta_n} + \Psi h_{\psi_n}$.

In the case of uncoupled vertical bending, there are two independent solutions which satisfy all the boundary conditions at the forward pylon. Because of the linear character of the equations, any linear combination of the two solutions is also a solution which satisfies the initial boundary conditions at the forward pylon. These two solutions are called the ϕ solution and the δ solution. There are two boundary conditions to be satisfied at the aft pylon. These conditions state that the residual shear and bending moment are zero at the rear. A parameter Φ is the multiple of the ϕ solution to be added to the δ solution so that the combined solution has zero residual shear at the aft pylon. The residual bending moment is then computed and plotted as a function of ω ; the roots of M' = 0 correspond to the natural frequencies of undamped harmonic vibration of small amplitude.

The case of coupled side bending and torsion is similar except that there are three independent solutions consistent with the boundary conditions at the forward pylon. At the aft pylon, the boundary conditions for a natural vibration frequency are that the residual shear, torque, and bending moments be zero. Two parameters, Φ and Ψ , are chosen so

that the residual shear and torque are zero. These parameters are the relative amounts of the ϕ and ψ solutions which must be added to the δ solution to give zero shear and torque at the rear pylon. The residual moments computed using various values of the frequency parameter, $\omega = 2\pi f$, are plotted against ω , and the roots give the natural frequencies of the system.

Under these conditions, the variables f and g, together with the parameter Φ (also the variable h and the parameter Ψ , in the case of coupled side bending and torsion), determine the modal shapes at the natural frequencies. Since the variables f, g, and h increase exponentially along the fuselage, whereas the appropriate physical quantities describing the modal shape (such as deflection in the case of uncoupled vertical bending) oscillate in sign and remain of the same order of magnitude along the fuselage, there must be very nearly exact cancellation of the component deflections from the various solutions near the rear of the fuselage.

In practice, the first three to five significant figures may be identical before subtraction. Thus, a high degree of accuracy in the dependent variables is necessary to get reasonably accurate modal shape curves. Five to seven figures are necessary to obtain two to three significant figures in the modal shape curve. The experience of the Piasecki engineers lead them to recommend the retention of at least six significant figures at all times, if possible. In order to maintain maximum accuracy and yet avoid the likelihood of counter overflow in the Type 402 Accounting Machine, particular care in the scaling of this problem was essential.

SCALING THE SYSTEM OF UNITS

The system of units in which design engineers express the various structural constants of the helicopter fuselage is gravitational, with the *pound of force*, the *inch*, and the *second* as fundamental units. In the engineer's system of units, the numerical values of the different types of structural constants range from 10^9 to 10^{-9} . By an appropriate

alteration of these units by powers of ten, all the various structural constants become of the order of unity in the new system of units. The various dependent variables (f, g, h, G, G', and H) also become of the order of unity in the new units. Thus, 10^2 inches, 10^{-2} seconds, and 10^7 pounds of force were chosen as convenient units for machine procedure, and the various factors for conversion from the engineers' system of units to the given system are powers of ten.

Examination of sample hand computations for typical cases performed^a over a part of the range of frequencies of interest indicated that the maximum values of the various variables were fairly approximately proportional to ω^2 . Since all the equations are linear and homogeneous in all the other variables, it was decided to use solutions with initial or starting values of $1/\omega^2$ rather than unity for f, g, or h, as the case might be. The analogous quantities in this solution, f, g, h, G, G', and H, are equal to $f/\omega^2, g/\omega^2, h/\omega^2$, etc., respectively.

MACHINE PLANNING

Machine planning for the IBM Card-Programmed Electronic Calculator allowed for seven decimal digits in all constants and variables used in the computations. Thus, the appropriate channel C shift code depends only on the mathematical operation being performed. A setup change switch was wired so that the problem could be instantly converted to the solution of the problem using six instead of seven decimal digits. This setup change switch automatically shifted the reading of the factors from the card one position, and also automatically altered the channel C shift code on the 402 control panel in the proper manner.

The parameter, ω^2 , was entered into the CPC on the first card only. The remaining program cards contained instructions and fuselage constants. To repeat the solution for a different frequency, a different initial card was substituted. Output was in the form of a tabular listing only. At the end of each section, all the counters were impulsed to total without reset on a minor program cycle. This caused the printing of a table of f, g, h, G, G', and H. The machine used was equipped with automatic conversion and reconversion so that true negative values were listed, and complement negative values were retained. The identical program cards were used for different helicopters, or revisions of the same model helicopter. The new structural constants were punched in a different factor field in the same card.

The time of solution on the CPC was approximately two minutes for uncoupled vertical bending (compared to eight hours hand computing). Coupled side bending and torsion required five to six minutes per frequency investigated.

Complete checking was not attempted. Smoothness of the residual moment curve served as a check against obvious errors. These curves were plotted while the CPC was in operation. If a random error was suspected, that frequency was re-run. If the results were identical, and a consistent error was suspected, a setup change switch was utilized to read the structural constants from a different factor field on the cards, which field contained the structural constants of an helicopter model which had previously been extensively investigated.

Throwing of the setup change switch which controlled the number of decimal figures retained in each step of the calculation, should give similar answers; thus, single digit counter overflow could be ruled out. These safeguards against error, together with the expected values of the natural frequences of vibration derived by the design engineers on the basis of uniform beam theory were deemed sufficient guarantees of the correctness of the calculation. The errors arising from truncation of the various numerical quantities after seven decimal places could usually be estimated by comparison of the results obtained using only six decimal digits. As was expected, this truncation error increased rapidly with ω^2 , but was within acceptable bounds for the ranges of interest for design purposes.

This problem is considered an ideal type of problem for solution on the CPC. The counters and storage were practically completely utilized to the maximum efficiency. After the problem had been set up and in operation for some months, the engineers wished to discover what effect, if any, their assumption of a rigidly mounted engine had upon the frequency spectrum. This meant that to investigate the nonrigid engine mounting case, some of the structural "constants" became prescribed functions of the frequency being investigated. The conversion from one problem to the other required the replacement of 5 to 10 program cards; thus, it was very convenient to alternate between the two cases to assess the importance of their original assumption in the various frequency ranges.

It is of interest to compare the procedure used with the recently developed floating decimal procedures. The main body of the solution consists of the accumulation of products to obtain the succeeding values of the dependent variables. In the procedure used, practically every card used the multiplication type of operation, and the addition was performed by the read-in to the 402 counters. Thus, the programming of this problem by floating decimal procedures would require practically twice as many program cards.

On the other hand, floating decimal procedures would guarantee eight significant figures at each step. This compares with six figures (or five in a few cases at higher frequency) in the present procedure. Further, the initial planning would have been only a fraction as complicated, since scaling would have been unnecessary. Floating decimal operation would also have eliminated the necessity for conversion factors of powers of ten.

^aThese computations were carried out in the Vibration Analysis Department of the Piasecki Helicopter Corporation.

DISCUSSION

Mr. Ferber: This is exactly the method that we have been using at Consolidated Vultee in San Diego for the past four or five years. In fact, we used this method on the 601 sometimes with complex numbers. It was really a big problem. We also have more flexibility taken into account and instead of four and six equations, we often have eight and ten simultaneous equations. However, I found again that this was just the problem where you would want to emit the value of ω rather than punch a card each time.

Mr. Conant: Is there a possible application to slightly different problems, that is, vibration within the cabin of an aircraft, due to vibrations of the motor and propeller-mount?

Mr. Leone: One of the easiest ways to remedy the excessive vibrations of the engine, of course, is to vary its flexible mounts, but in this particular problem, one has to be careful that, in so doing, the resulting coupling effect of the fuselage vibration with the engine will not produce a coupled frequency which will fall inside of an excitation range.

Dr. Welmers: Very frequently a rough experimental check, after the airplane or the helicopter has been constructed, can give some results rather conveniently. In one particular instance on a helicopter where we had excessive

cabin vibration, a couple of hours spent in vibrating the ship and in shifting the battery around in the tail structure solved the problem. The total shift of the battery was about one and a half or two feet. It completely eliminated an annoying cabin vibration that had been complained of by the pilots. Much of this work is hardly worth a careful analytical study because the variations from one ship to another may make it almost impossible to apply on a production basis. The sheet-metal and the way in which it is bent on most of these non-structural items is not particularly carefully watched.

Mr. Leone: I would just like to say that we at Piasecki are developing a method for the analysis of the flutter and force response of the rotor blade in which we, too, hope to use complex numbers, but it appears to be a big problem at the moment because of the obscure knowledge of the oscillating air forces on rotating wings.

Mr. Ferber: It may be of interest to mention an additional reference on this subject of Mr. Leone's paper: "Calculation of Coupled Vibration Modes and Frequencies of Aircraft" by Michael Dublin, Consolidated Vultee Aircraft Corporation. This paper was presented at the Applied Mechanics Conference held at Stanford University on June 22-23, 1951.