NASA SP-5913 (02)

TECHNOLOGY UTILIZATION

MATHEMATICAL COMPUTER PROGRAMS

A COMPILATION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Foreword

The National Aeronautics and Space Administration and the Atomic Energy Commission have established a Technology Utilization Program for the dissemination of information on technological developments which have potential utility outside the aerospace and nuclear communities. By encouraging multiple application of the results of their research and development, NASA and AEC earn for the public an increased return on the investment in aerospace research and development programs.

The computer programs, routines, and subroutines presented in this compilation, appear in two sections. Section one consists of items that should aid engineers, scientists, and mathematicians in direct problem solving. Section two is made up of a group of items that should afford the same users greater flexibility in the use of software.

Additional information on individual items can be requested by circling the appropriate number on the Reader Service Card included in this compilation; or from: COSMIC, 112 Barrow Hall, University of Georgia, Athens, Georgia 30601.

Unless otherwise stated, NASA and AEC contemplate no patent action on the technology described.

We appreciate comment by readers and welcome hearing about the relevance and utility of the information in this compilation.

Jeffrey T. Hamilton, Director
Technology Utilization Office
National Aeronautics and Space Administration

NOTICE • This document was prepared under the sponsorship of the National Aeronautics and Space Administration. Neither the United States Government nor any person acting on behalf of the United States Government assumes any liability resulting from the use of the information contained in this document, or warrants that such use will be free from privately owned rights.

For sale by the National Technical Information Service, Springfield, Virginia 22151.

Table of Contents

Section 1. Programs for Problem Solving

MULTI-DIMENSIONAL REAL FOURIER TRANSFORM

These programs compute the one-dimensional and multi-dimensional Fourier transforms for real data, the multi-dimensional complex Fourier transforms, and the multi-dimensional sine, cosine, and sine-cosine transforms.

There are four subroutines, each of which performs the required calculation for one of the four problems given. Each subroutine uses the Cooley-Tukey Fast Fourier Transform (FFT), and, in all but the one-dimensional case, the transforms are calculated in up to six dimensions.

The Cooley-Tukey algorithm is an efficient computational procedure for evaluating expressions of the form:

$$X_j = \sum_{k=0}^{N-1} a_k W^{jk}, \quad \begin{array}{l} j = 0,1..N-1 \\ W = \exp(2\pi i/N) \\ i^2 = -1 \end{array}$$

where Xj and a_k are complex, and N is composite. If the calculations were to be carried out in a straightforward manner, with an operation defined as one complex multiplication and addition, a total of N(N−1) operations would be required; i.e., N−1 operations for each value of j. The Cooley-Tukey algorithm, however, organizes the calculations in such a way that only (N $\log_2$N) operations are required. This represents a significant savings for large N.

These programs are written in FORTRAN V for use with the UNIVAC-1108 computer. The specific subroutines are as follows: NPO-11648, Multi-Dimensional Real Fourier Transform; NPO-11649, One-Dimensional Real Fourier Transform; NPO-11651, Multi-Dimensional Complex Fourier Transform; and NPO-11652, Multi-Dimensional Sine, Cosine, and Sine-Cosine Transforms.

Source: Fred T. Krogh of
Caltech/JPL
under contract to
NASA Pasadena Office
(NPO-11648, 11649, 11651, 11652)

Circle 1 on Reader Service Card

FORTRAN IV PROGRAM FOR SYMBOLIC SOLUTION OF UP TO 20 SIMULTANEOUS LINEAR EQUATIONS

The symbolic (as opposed to the numeric) solution of linear equations is useful in understanding the influence of the coefficients on the solution, and in determining formulas that can be used in further analyses. This program is, basically, a symbolic manipulator that provides a means for obtaining an algebraic solution to a set of up to 20 simultaneous linear equations.

In general, the program operates by reading in the linear equations in matrix form as the matrix equation [A] [X] = [B] [Y] where the X's are the dependent variables, the Y's are the independent variables, and the A's and B's are the constant coefficients of the equations, all in symbolic form. Also read in is the list of X solutions desired since it is not usually required to solve for all X's. The program then operates on the matrices to develop the matrix equation [X] = $[A]^{-1}$ [B] [Y] during which operation the particular X solutions requested are obtained.

This program is written in FORTRAN IV language for use with the IBM-7094/7044 DCS computer.

Source: D. R. Packe
Lewis Research Center
(LEW-10439)

Circle 2 on Reader Service Card

FORTRAN IV PROGRAM FOR COMPUTATION OF WEBER FUNCTIONS AND THEIR FIRST DERIVATIVES

A FORTRAN IV program was developed for computing $D_v(x)$ and $(d/dx)\ D_v(x)$, which are the Weber function of order v and argument x and its first derivative, respectively. The range of values is x (0,30) and v (1,200). The computations are performed in double precision and the values are accurate to seven significant figures over most of this region. However, in the range x (5.5, $\sqrt{50}$) and $(x^2/4) + 1 < v < (x^2/4) + 13$, the accuracy is reduced. Included are tables of D_v (x) and $(d/dx)D_v$ (x) with values listed at intervals of 1 in x and 5 in v; the tables begin with x = 0 and v = 4.5, but they omit part of the region covered by x (5.5 $\sqrt{60}$).

The program is so written to take advantage of asymptotic forms, recursion relations, and series computations at combinations of v and x that yield the desired accuracy in the shortest machine time.

This program is written in FORTRAN IV language for use with the IBM-7094 computer.

Source: A. Priede and G. Allen
Lewis Research Center
(LEW-10898)

Circle 3 on Reader Service Card

FORTRAN H SUBROUTINE SUBPROGRAM

This subroutine, a generally useful program for mathematics analysts, is used to solve an Nth order system of first order ordinary differential equations. It is used to solve differential equations of the form:

$$y_1' = f_1\ (x: y_1, y_2, \ldots, y_N), \text{etc.}$$

The method of calculation is the one-step fourth-order Runge-Kutta procedure with built-in (automatic) variable incrementing. All arithmetic calculations in the subroutine and its associated dependent subprograms are performed in double precision. The calling sequence of the subroutine requires double precision arguments.

This program is written in FORTRAN H language for use with the IBM-360, Release II computer.

Source: North American Rockwell Corp.
under contract to
Marshall Space Flight Center
(MFS-18564)

Circle 4 on Reader Service Card

SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

This program solves sets of simultaneous linear equations and should be of interest to mathematicians and others involved with such problems. The program (SLEQ) has been written in FORTRAN IV for the IBM-360 computer.

The program accepts equation input in essentially algebraic form in a free field format. This form of input is much more readily prepared than that in a fixed field format containing the coefficients only, and represents the program's major advantage. In its original form, the program could provide solutions to a maximum of 100 equations in 100 variables. Extensive checking of the input is performed by the program and, in the event of errors, descriptive messages are printed out.

Source: D. G. Michaels of
Aerojet-General Corp.
under contract to
Lewis Research Center
(LEW-11019)

Circle 5 on Reader Service Card

DOUBLE PRECISION, REAL SIMULTANEOUS LINEAR EQUATION SOLVER

The function of the computer program is to solve "n" real simultaneous linear equations in "n" unknowns for "k" vectors. The problem is solved by the "Factorized Inverse" process. [Y] (n,n), the coefficient matrix, and B_K, the column vector(s) formed by the right-hand side of the set of equations, are input. [A] (n,n), the "Factorized Inverse Table," is formed from [Y] (n,n) by systematic row and column matrix operations. In general,

$$A_{ii} = (Y_{ii} - \sum_{k=1}^{i-1} A_{ik}A_{ki})^{-1} \qquad (i=j)$$

$$A_{ij} = Y_{ij} \sum_{k=1}^{j-1} A_{ik}A_{kj} \qquad (i > j)$$

$$A_{ij} = A_{ii} \quad (Y_{ij} - \sum_{k=1}^{i-1} A_{ik}A_{kj}) \; (i < j)$$

Once [A] (n,n) is formed, it is used to compute the forward and backward solution(s). The equations for the forward solution(s) are, in general,

$$f_i = B_i - \sum_{j-1}^{i-1} A_{ij}d_j$$

$$d_i = A_{ii}f_i$$

The equations for the backward solution(s) are, in general,

$$D_i = d_i - \sum_{j=i+1}^{n} A_{ij}D_j$$

This method requires n^2 arithmetic operations per solution whereas the conventional matrix inversion technique requires n^3 operations. The program is limited to 80 equations in 80 unknowns for 10 solutions. Solution vectors are output, and other output is optional. While a complete timing analysis has not been performed, a 21 unknowns solution required 6 seconds with all optional output and 1.5 seconds with output restricted to the solution vector. Source language is FORTRAN II, version 3 for use with the IBM-7094 computer.

Source: R. F. Jonk of
Chrysler Corp.
under contract to
Marshall Space Flight Center
(MFS-1905)

Circle 6 on Reader Service Card

CALCULATION OF EIGENVALUES AND EIGENVECTORS OF ARBITRARY MATRICES

This program uses a modification of the Greenstadt method to calculate the eigenvalues and eigenvectors of an arbitrary complex matrix. The matrix is reduced to upper triangular form. Elements in the lower triangle (called pivot elements) are driven to zero through the application of a sequence of unitary transformations. The sequence of operations upon the set of pivot elements is called a pass. The algorithm continues until a specified maximum number of passes has been made, or until the average modulus of the set of pivot elements has been reduced to a particular value. The maximum order of the input matrix is 40.

The triangularized matrix (labeled "Eigenvalue Matrix") is output in two parts, the real part and the imaginary part. The eigenvalues appear on the main diagonal, real part and imaginary part. The vector matrix of the triangularized system (in similar format) follows. Finally, the eigenvector matrix of the original system is output.

The program is written in FORTRAN H language for use with the IBM-360 computer.

Source: North American Rockwell Corp.
under contract to
Marshall Space Flight Center
(MFS-2368)

Circle 7 on Reader Service Card

VARIABLE ORDER INTEGRATORS FOR THE NUMERICAL SOLUTION OF ORDINARY ˙)IᶠFERENTIAL EQUATIONS

A series of computer subroutines, developed to integrate systems of ordinary differential equations, may also be used for numerical quadrature. SVDQ is a single precision subroutine, and DVDQ is double precision. VODQ keeps the derivative and performs most of the calculations in single precision, and accumulates the independent and dependent variables in double precision.

The subroutines use linear multistep predictor-corrector formulas of the Adams type, with formulas available for treating directly, differential equations of orders 1 to 4. The subroutine automatically selects the integration orders (which may be different for different equations in the system) and the stepsize. The integration orders are selected so as to maximize the stepsize while maintaining numerical stability and meeting the user's requested local accuracy. Special returns are available based on the number of steps, the value of the independent variable, values of dependent variables, and values of user defined auxiliary functions.

This program is written in FORTRAN IV and FORTRAN V for use with the IBM-7094 (IBSYS) computer or the UNIVAC-1108 (EXEC 8) computer.

Source: F. T. Krogh of
Caltech/JPL
under contract to
NASA Pasadena Office
(NPO-11643)

Circle 8 on Reader Service Card

COMPUTER SUBROUTINE ISUDS ACCURATELY SOLVES LARGE SYSTEM OF SIMULTANEOUS LINEAR ALGEBRAIC EQUATIONS

Since the accuracy of a solution for a set of simultaneous equations decreases as the order of the system increases, it is necessary to obtain double-precision accuracy while using a single-precision coefficient matrix to conserve memory storage. This is accomplished by a computer program which is an Iterative Scheme Using a Direct Solution (ISUDS), which obtains the desired accuracy.

ISUDS finds a solution to a system of equations and increases its accuracy while using a single precision coefficient matrix. The equations are written in matrix form as $AX=B$, where A is a square non-singular coefficient matrix, X is a vector, and B is a vector. The values of X that are found are substituted into the equations and the residuals are calculated, using double-precision arithmetic.

The system of equations is then solved again, except with the residuals of the equations as the right-hand sides. The first solution (X_1) satisfies the equations with the right-hand side equal to the vector B, minus the residuals R, while the second solution (X_2) satisfies the same system with the residuals on the right-hand side. Hence, X_1+X_2 satisfies the same system of equations, and since $(B-R)+R=B$, the sum of X_1+X_2 will give an accurate solution to $AX=B$. A solution to any desired accuracy may be obtained on a digital computer, depending on the word size.

The digital computer code ISUDS is written in FORTRAN IV language for use with the IBM-7094 and is based on the use of ISIMEQ, a 7094 FORTRAN simultaneous linear equation subroutine. A storage capacity of approximately 32K is required.

Source: G. Collier
of Westinghouse Astronuclear Laboratory
under contract to
AEC-NASA Space Nuclear Systems Office
(NUC-10051)

Circle 9 on Reader Service Card

COMPUTER PROGRAM VARI-QUIR III PROVIDES SOLUTION OF STEADY-STATE, MULTIGROUP, TWO-DIMENSIONAL NEUTRON DIFFUSION EQUATIONS

A Gauss-Seidel type of solution, with inner and outer iterations solves the steady-state, multigroup, two-dimensional neutron diffusion equations in X–Y, R–Z, and R–θ geometries. In nuclear reactions having complex geometries it is necessary to calculate the neutron flux as a function of energy and of the spatial coordinates. Economic one-dimensional (slabs, cylinders, spheres) programs are in use and several two-dimensional programs have been developed. Among the two-dimensional programs, the PDQ and CURE codes are the most popular. However, each of these codes consumes large amounts of computer time.

A Gauss-Seidel type of solution with inner and outer iterations solves that problem. The source is held constant during the inner iterations.

Although the running time for this new program (VARI-QUIR III) is approximately one-tenth that required for the CURE code, results compare favorably. The saving in computer time is accomplished through the use of a variational approximation. The program has no restrictions (within reason) on any of the input parameters such as the number of groups, regions, or materials. The parameters are restricted only by a maximum amount of storage and are completely arbitrary at execution time. The program will handle X–Y, R–Z, and R–θ geometries. A double interpolation subroutine is required to operate the code and this routine is furnished with the program deck.

The program is written in FORTRAN IV for use with the IBM-7094 computer (32K memory).

Source: G. Collier
of Westinghouse Astronuclear Laboratory
under contract to
AEC-NASA Space Nuclear Systems Office
(NUC-10052)

Circle 10 on Reader Service Card

Section 2. Programs for Making Determinations and Evaluations

PROGRAM FOR CONVERSATIONAL APPROACH TO MATRIX CALCULATIONS

Engineers and scientists, who require the use of a computer but are unfamiliar with the various languages can make use of this program to create a "dialogue" with the computer in order to have it perform many matrix problems. Potential users can quickly learn the command language to which the computer will respond. The program, called MATAR, interacts with users in a conversational manner and has been successfully used in a time-sharing system.

MATAR provides for simple input/output of matrices, matrix arithmetic, plus formations of the transpose, inverse, determinant, eigenvalues, and exponentiation. The programs issues specific notice of input it cannot process. For example, if the user introduces a matrix not previously defined, the program notifies him that the particular calculation request must be cancelled. The user can promptly rectify his errors.

This program is written in FORTRAN H for use with the IBM-360/67 computers.

Source: R. B. Canright
Lewis Research Center
(LEW-10830)

Circle 11 on Reader Service Card

DIGITAL FILTER SYNTHESIS COMPUTER PROGRAM

A digital filter synthesis computer program has been developed to express any continuous function of a complex variable in approximate form as a computational algorithm or difference equation.

Given an analog prototype of the desired filter and a sampling interval, analog filter critical frequencies are computed along with a new frequency-scaled filter function, F(s), which is the transfer function for the real-time analog equivalent network.

The bilinear transform method is used on F(s) to make the transformation from the s to the z plane. The resulting function F(z), is the digital equivalent of a transfer function and can be represented by the expression:

$$F(z) = \frac{\Sigma A_n z^{-n}}{\Sigma B_n z^{-n}} = \frac{0(z)}{I(z)},$$

where 0(z) is the output data list, I(z) is the input data list, and A_n and B_n are, respectively, the numerator and denominator coefficients of the digital transfer function.

Solving for the function 0(z), which represents the desired output digital data list, the program obtains the difference equation that represents the nearest digital equivalent to the analog prototype within the limitations allowed. The coefficients, A_n and B_n, of this difference equation, are listed for each filter configuration.

Once the difference equation has been developed, digital filtering can be performed by the program on any input data list.

This program is written in FORTRAN IV for use with the IBM-7090/94 computer with IBSYS Executive monitor.

Source: R. M. Munoz and R. A. Moyer
Ames Research Center
(ARC-10130)

Circle 12 on Reader Service Card

DETERMINATION OF RADIATION INTERCHANGE FACTORS

This method of solution is compatible with digital computer analysis and consists of a generalized computer program to carry out the computations.

The program computes thermal and solar radiation interchange factors among surfaces having any combination of the following properties:

1. Diffuse emittance and reflectance,
2. Diffuse emittance and specular reflectance,
3. Diffuse emittance and components of diffuse and specular reflectance, and
4. Directional emittance and bidirectional reflectances.

Enclosures which contain only the first three types of surfaces may be simulated by a maximum of 1500 surfaces. Fully bidirectional enclosures are limited to 38 surfaces or less. The maximum number of surfaces in a mixed enclosure is governed by:

$$NM + M - N$$

where M = total number of surfaces, and N = number of bidirectional surfaces.

In addition to the standard node-to-node matrix formulation, a mean-to-local approximation is available as a user's option. The method utilizes average property values for groups of surfaces. This can result in significant reductions in the size of the matrices and in the time required to invert them.

The program input includes surface properties and geometrical data defining the shape, size, and location of each surface in the enclosure. The program printout includes input data, form factors, exchange factors, the transfer matrix and its inverse, nodal areas, and interchange factors.

Source: R.P. Bobco, F.L. Egendorf,
and R.J. McGrath of
Hughes Aircraft Co.
under contract to
Manned Spacecraft Center
(MSC-13475)

Circle 13 on Reader Service Card

PROGRAM FOR BINOMIAL PARAMETER DETERMINATION

The function of the program is to determine the parameter (root) of the cumulative binomial probability distribution. That is, given the limits on the binomial summation and the numerical value of the sum, the program arrives at the probability root P using the Newton-Raphson iterative procedure. P is determinable for the summation whose limits are 0 to C (C.LE. 1000) as well as the summation whose limits are R to N (R.LE.N., N LE. 1000). Six-place accuracy in P is maintained. The program requires a storage capacity of 2900 36-BIT words, 2000 of which are used to store the logarithms of all factorials from 1 to 1000 in double precision.

The program is written in FORTRAN IV for use with the IBM-7094 computer.

Source: D. Bien
Lewis Research Center
(LEW-11004)

Circle 14 on Reader Service Card

SUBROUTINE ALLOWS EASY COMPUTATION IN EXTENDED PRECISION ARITHMETIC

This is a program that allows relatively simple computation of very large numbers or very small fractions with extreme accuracy. Ordinary double precision FORTRAN arithmetic is limited to numbers between 10^{-38} and 10^{+38}, and to 16 digits of accuracy.

The subroutine, called NPREC, can perform mathematical operations in extended precision floating point arithmetic. This subroutine handles numbers that consist of 35 binary bits (1 word) for the exponent and 70 bits (2 words) for the fraction. A programmer can perform a variety of mathematical operations by writing standard FORTRAN statements within the NPREC routine instead of calling a separate routine to perform each operation.

NPREC can be used on any of the IBM 709/7090/7094 computers. A statement, CALL NPREC, is used to enter the routine, and subsequent statements are interpreted by the routine until the exit statement, CALL NPOUT, is encountered. Any statement that is interpreted in NPREC will take longer to execute than if it were executed outside of the routine, and thus all normal single or double precision statements should be executed outside of the routine.

The NPREC routine incorporates subroutines in converting to and from extended precision numbers and also includes the following extended precision elementary functions: square root, sine, cosine, arc tangent, natural log, and exponential. The only "built-in" library functions that may be used in NPREC are ABSF, MAXIF, MINIF, AND SIGNF. There is limited use of FORTRAN exponential notation in NPREC, and extended precision numbers may be raised to a fixed point power of 2, 3, 4, 5, or 6 only. Any other type of exponential operation must be performed using the exponential and natural log routines within NPREC. No more than 15 operations should appear in any single statement in NPREC.

The NPREC routine will detect 20 errors. When one of these errors is detected, a comment will be written in 1 of 2 formats on output tape 6, after which a core dump is given and the job is finished.

The time it takes to perform an arithmetic operation in NPREC varies with the values and relative values of the operands, and also depends on whether or not the operands are stored in COMMON. As an indication of the time requirements, an extended precision division takes approximately 1.4 milliseconds on the 7090, and a multiplication, approximately 0.6 milliseconds. The natural log and exponential functions require approximately 36 and 48 milliseconds respectively. Multiplying the 7090 time by 6 gives the 709 time.

NPREC and its associated subprograms require 2053 storage locations.

Source: R.J. Berggren and J.C. Gysbers of
North American Rockwell Corp.
under contract to
Marshall Space Flight Center
(MFS-1136)

Circle 15 on Reader Service Card

POLYNOMIAL FACTORING USING SUBROUTINE (BROOT)

This is a fast polynomial factoring routine that has proven useful in a number of instances, other than that for which it was written, i.e., root locus applications. The program should interest anyone needing factoring and root finding.

BROOT is the execution subroutine for another subroutine, DIVIT, which obtains second degree factors. The purpose of BROOT is to remove zero roots, to check the input, to find the roots from the second degree factors furnished by DIVIT, and to keep repeating the division process until the polynomial is exhausted.

The purpose of DIVIT is to obtain a second degree factor for a polynomial of degree three or more (up to 100). It uses the Bairstow procedure in double precision arithmetic. Should this fail, the polynomial is then converted to one in which the roots are the squares of the actual roots and the Bairstow procedure is repeated. This squaring process will be repeated until overflow occurs; the Bairstow procedure is then repeated for the original polynomial and the roots are returned with the flag set.

The program is written in FORTRAN H for use with the IBM-360, Release II computer.

Source: North American Rockwell Corp.
under contract to
Marshall Space Flight Center
(MFS-02374)

Circle 16 on Reader Service Card

STRUCTURAL ANALYSIS AND MATRIX INTERPRETIVE SYSTEM (SAMIS)

This program is designed to solve problems involving matrix arithmetic, and is particularly useful in the analysis of mechanical structures.

The program can execute, either exclusively or sequentially, two basic operations. From input data that define an idealization of a structure, the generation phase of the program generates structural matrices for any type of element available in the program element library. This phase is based upon the structural concepts of the finite element method, in particular the stiffness or displacement method. To enable the program to analyze a range of structural types (truss, plate, shell, composite shell beam, etc.), several elements are programmed and cataloged in the program element library. Contained in the library are the general line element suitable for representing axial, bending, and torsion deformations and the triangular plate element which models membrane and bending deformations. The second basic operation is termed the manipulative phase, in which either generated or input matrices are manipulated according to the rules of linear algebra. In structural problems, the matrix manipulations may be sequenced to compute displacements, stresses, reaction faces, or mode shapes and frequencies. The ability to compute these quantities for structural systems which are described by a large number of simultaneous equations requires greater than incore data access and storage capacity. Because of this requirement, the program was developed as a chain system. Based mainly upon the constraint of computer running time, the SAMIS program operates efficiently with matrices ranging from the 100th to 2500th order.

The SAMIS program is now available in four versions. Version 1 Modification 3 is the final modification to the original FORTRAN II program. Version 2 Modification 1 is a FORTRAN IV version of SAMIS with the same capability that exists in the FORTRAN II version plus some new and extended capabilities which include a new modified Choleski or Gauss wavefront algorithm for solving a set of symmetric simultaneous equations; a new eigenvalue routine to obtain the roots and vectors of a real symmetric matrix up to order 200 by Givens-House holder algorithm; and a new checkout and element data generation routine. Version 3 Modification 1 is a CDC-6600 FORTRAN program. This version is the equivalent of the FORTRAN II, Version 1 Modification 1 program in performance. Version 4 is a UNIVAC-1108 program; it is equivalent to the FORTRAN IV, Version 2 Modification 1 program in performance.

Source: Herman Hines
Manned Spacecraft Center
(MSC-13315)

Circle 17 on Reader Service Card

FOURIER-FORTRAN H SUBROUTINE SUBPROGRAM

This subroutine computes the coefficients | a | and | b | of the real Fourier series for a given tabulated real function | f |. The series is then evaluated for real arguments.

$$y(x) = \frac{a_0}{2} + \Sigma a_k \cos kx + b_k \sin kx$$

The method of calculation, based on a real Fourier analysis, is described in two textbooks referenced in the program documentation. All arithmetic calculations are performed in double precision, and the calling sequence requires double precision arguments. The function to be approximated is assumed to be periodic with respect to the given interval, and the given set of functional values is assumed to be equally spaced with respect to the given interval. It is assumed that the initial functional value is located at the leftmost point of the interval.

The program is written in FORTRAN-H language for use with the IBM-360, Release II computer.

Source: North American Rockwell Corp.
under contract to
Marshall Space Flight Center
(MFS-18565)

Circle 18 on Reader Service Card

COMPUTER PROGRAM ETC IMPROVES COMPUTATION OF ELASTIC TRANSFER MATRICES OF LEGENDRE POLYNOMIALS P(0) AND P(1)

Numerical difficulties are encountered with other computer programs in carrying out the double integration that is inherent in the calculation of the P(0) and P(1) matrices. This is particularly true when the elastic scattering cross section depends strongly on the neutron energy, i.e., where resonances occur. Also, other programs are not nearly fast enough for optimization.

Rather than carrying out a double integration numerically, one of the integrations is accomplished analytically and the numerical integration need only be carried out over one variable. This results in a more satisfactory numerical treatment and a fester calculation.

For identical input the ETC program takes about one-third of the time, other known programs take, to calculate the P(0) and P(1) kernels. The ETC program has also been made more flexible, viz., any number (< 200) of energy mesh points may be used per fine group (GAM Groups: 1/4 lethargy intervals, 68 groups from 0.414 eV to 10 MeV). This number may vary from group to group. This feature may be utilized to save additional computer time.

The program is written in FORTRAN IV for use with the IBM-7094 or the CDC-6600 computer. It can be used for the calculation of multigroup cross sections for analysis of any nuclear reactor for any application.

Source: G. Gibson and M. Miller
of Westinghouse Astronuclear Laboratory
under contract to
AEC-NASA Space Nuclear Systems Office
(NUC-10070)

Circle 19 on Reader Service Card

MONTE CARLO DIRECT-VIEW FACTOR AND GENERALIZED RADIATIVE HEAT TRANSFER PROGRAMS

Using the Monte Carlo technique, a package of computer programs was written to find the "black body" view factor from one surface segment to another surface segment defined on a primary surface; and also to define the primary surfaces (collection of simple geometric objects, such as a sphere, a cylinder, a cone, a disk, and a parallelogram) in three-dimensional space. It will also find the radiative transfer coefficients from a surface segment defined on a primary surface to all other surface segments defined in a real enclosure.

The function of the program is numerical evaluation of the integral:

$$A_I F_{IJ} = A_J F_{JI} = \int\limits_{\text{Surface I}} dA_I(\bar{x}_I) \int\limits_{\text{Surface J}} dA_J(\bar{X}_j)\ B(\bar{X}_I,\bar{X}_J)\ H(\bar{X}_I,\bar{X}_J),$$

where $\bar{X}_I$, $\bar{X}_J$ are vectors representing points on surfaces I, J of area $A_{I,\ J}$. $B(\bar{X}_I, \bar{X}_J)$ is a blocking factor which is unity if $\bar{X}_I$ can "see" $\bar{X}_J$, and zero otherwise. $H(\bar{X}_I, \bar{X}_J) = \hat{n}_I . (\bar{X}_J - \bar{X}_I)\ \hat{n}_J . (\bar{X}_I - \bar{X}_J) / \pi \mid \bar{X}_I - \bar{X}_J \mid^4$, where $\hat{n}_I$ and $\hat{n}_J$ denote unit surface normal vectors.

The segment (or node) object numbers are obtained from the view factor command. The nodes are located in the definition table. From the nodes come the primary surface object numbers and K values. The primary surfaces are located in the definition table and copied into temporary storage. An error message is printed if nodes or primary surfaces are missing. The areas of the nodes are calculated and the canonical forms are generated for the subroutines:

1. Reset the iteration sums and parameters
2. N = N + 1
3. Generate random points $\bar{X}_I$ and $\bar{X}_J$.

Two random numbers are obtained from a uniform distribution. These numbers are modified so that the points generated on the different surface window will cover the surfaces evenly. The surface points and normals are obtained using the subroutines.

The Generalized Radiative Heat Transfer Program finds radiative-transfer coefficients from a surface segment defined on a primary surface to all other surface segments defined in a real enclosure. The program permits radiation heat transfer analysis for enclosures containing surfaces which have, in general, both diffuse and specular reflectivities.

A geometric configuration is defined by the customer, consisting of a set of defining primary surfaces and their nodes. A primary surface may be a parallelogram, disc, cylinder, sphere, or cone. Nodes are defined as subsurfaces on the primary surfaces. One input data case consists of a geometric configuration together with commands for finding Script F's between objects within the configuration. Commands are executed as they are encountered. When a Script F command is encountered, the following operations are performed: The node summation areas are cleared to zero (These areas consist of Script F sums for 1 and 2 wavelength nodes along with force and torque sums. The presence of these areas is entirely dependent upon the type of node being encountered). The counter for the total number of rays considered is set to zero. The counter is incremented by 1.

An advantage is the wide range of problems that can be handled and the speed of solution from time of problem conception. The program is written in FORTRAN IV language for use with the IBM-360 computer.

Source: J.H. Scates and J.L. McWilliams
of the Boeing Company
under contract to
Marshall Space Flight Center
(MFS-15051)

Circle 20 on Reader Service Card

PEAK AND INTEGRAL SENSITIVITY PROGRAM

Given a system of linear differential equations, X=AX+B, one is frequently interested in how variations in the matrix of constant coefficients, A, will influence the solution or linear combinations of the solution. The purpose of this program is to generate two measures of the sensitivity of the solution, or linear combinations of the solution, to changes in the matrix A. These measures are peak and integral sensitivity. The order of the system of differential equations must be less than or equal to 20. The number of output vectors must be less than or equal to 100.

The output of the program consists of a listing of the input, the values of the output variables of each system, and the differences between the corresponding output variables of the second system and the first, at regular time intervals. Also, a list giving the name of each output variable, its integral sensitivity, its peak sensitivity, and the time of occurence of its peak sensitivity.

The program is written in FORTRAN IV for use with the IBM-7094 computer.

Source: Caltech/JPL
under contract to
NASA Pasadena Office
(NPO-11209)

Circle 21 on Reader Service Card

SUBROUTINES FOR EVALUATION OF SINGLE AND MULTIPLE INTEGRALS USING MODIFIED ROMBERG METHOD

Double and single precision computer subroutines, ROMBD and ROMBS, respectively, have been developed for numerical quadrature using a modified Romberg procedure with a variable step size, for evaluation of both single and multiple integrals. The package also contains subroutines RMB1, RMB2, and RMB3 evaluating multiple integrals.

The routines represent a "state of the art" in their field. They have been thoroughly tested and found to be equal to or better than comparable routines. The programs have been compared to SQUANK (Lyness, J.: Notes on the Adaptive Simpson Quadrature Routine, ACM Journal, Volume 16, July 1969) and found to be more reliable and capable of solving a larger class of problems.

Although ROMBD and ROMBS are designed to serve as a library "standard" for solving most of the problems of the form $I = \int_a^b f(x)dx$, it must be recognized that with singularities and certain discontinuities in f(x), Gaussian quadrature or other methods may be more appropriate.

Subroutine RMB1 with successive calls to RMB2 and a final call to RMB3 comprise the single precision package for multiple integration. ROMBS is used as the basic integration technique.

This program is written in FORTRAN V for use with the UNIVAC-1108 computer. The routines are easily convertible to FORTRAN IV.

Source: W.R. Bunton and M. Diethelm of
Caltech/JPL
under contract to
NASA Pasadena Office
(NPO-11295)

Circle 22 on Reader Service Card

NASA-Langley, 1972

> *"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*
>
> —NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546