

CONF-750312--1

Progress Report C00-2280-21 May 1975

QUANTITATIVE AND COMPUTERIZED ERROR ANALYSIS

By

Erwin H. Bareiss and Keith L. Derstine

Northwestern University and
Argonne National Laboratory

CS-75-5

ES-75-5

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT
ADMINISTRATION UNDER CONTRACT AT(11-1)-2280.

MASTER

DISTRIBUTION OF THIS DOCUMENT UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

QUANTITATIVE AND COMPUTERIZED ERROR ANALYSIS

Erwin H. Bareiss & Keith L. Derstine

Northwestern University and Argonne National Laboratory

Abstract

A method is described how a quantitative measure for the robustness of a given transport theory code for coarse network calculations can be obtained. A code, that performs this task automatically and at only nominal cost, is described. This code generates also user oriented benchmark problems which exhibit the analytic behavior at interfaces.

Introduction

The problem which we address here is concerned with ascertaining the reliability of neutron transport calculations by high-speed computers. Today's general approach is to check one computer code against another, often for mathematically not identical problem settings. The best way of checking approximate numerical calculations is to compare the results against mathematically exact solutions. Unfortunately, in transport theory such solutions are, in general, not available, or if available, they are very expensive to obtain. There is however the possibility to create benchmark problems to given meaningful exact solutions.

What we have started is a new discipline in Numerical Analysis, namely Quantitative and Computerized Error Analysis. Clearly, this

discipline can be (and is) applied to other operator equations than the transport equation. Traditionally, a new numerical technique was conceived, implemented in a computer code, and then distributed. An alternative approach is to make a systematic analysis of the mathematical properties of the solution and the new technique, and then test the theoretical predictions on inexpensive well designed small numerical benchmark problems. The basic difference between classical or qualitative error analysis and the quantitative error analysis is that in quantitative error analysis we aim at obtaining realistic error bounds for coarse mesh calculations and performance predictions before a code is implemented. What we do, can be described by mathematical expressions, but the formulas are very lengthy, complicated and difficult to evaluate. For all practical purposes analytic coarse mesh error analysis would be much too expensive and too time consuming. Our computer approach yields not only new insight, but simultaneously provides numerical results that give a quantitative assessment of the approximation method under investigation. It is a new type of analysis that combines modern and classical analysis with computer technology and computer graphics. A very high efficiency is possible because we exploit the use of dimensional analysis and invariant theory. This is important, since the operators we deal with are linear. Indeed we have used a great many of the properties which E. Inönü presented in his lecture on "A Scale Group for the Linear Boltzmann Equation and its Representation." We are very pleased to have now available a systematic and complete tabulat-

ion of all scaling, translation, rotation and other group properties of the transport equation.

To give a more concrete understanding of our research, we shall briefly outline the major parts of our project and then describe an actual code, BEAPAC-1T, which is running at Argonne National Laboratory and at Northwestern University. We will also show how to use this code and present some samples of the output options. The present code is designed for the one-dimensional transport equation and supplements well the theoretical results presented at the Fourth National Conference on Transport Theory, since they were concerned mainly with one-dimensional problems, and BEAPAC-1T makes use of them. A code for problems in two space-dimensions is now being implemented, both for the transport and diffusion equations.

The entire code, when the project is finished, will consist of three parts with the following functions.

- A. Cell Calculations (including 2-Cell Calculations)
- B. Global Analysis
- C. Computational Complexity

- A. The code for part A is again subdivided in three major parts.
 - a) Creation of Benchmark Problems
 - b) Numerical Solution of the Benchmark Problems by Approximate Methods
 - c) Calculation of the Error. The user will specify the appropriate error norm.

Part A can also be characterized by calling its function Basic Quantitative Error Analysis. It is designed to evaluate the desired error norms for calculations over a coarse network or with large finite elements. To our knowledge there exists no other systematic approach to deal with the coarse mesh error analysis. As will become evident, such an analysis is practical only by automation and use of the computer.

B. The codes for part B will conduct a global analysis for multiregion, multigroup calculation. The algorithms will be subject to:

- a) Global Error Analysis
- b) Stability Analysis
- c) Convergence Analysis

We note that the classical concept of consistency analysis does not appear explicitly. The codes will supply absolute error bounds. However the main goal is to provide Probabilistic Error Estimates. Research in all areas is under way. The tools employed are borrowed from functional analysis, perturbation-, matrix-, operator-, and probability theory and statistics.

C. The codes for part C are concerned with computational complexity. By this we understand:

- a) Performance Prediction
- b) Code Evaluation
- c) Data Management Analysis

Basically, part C is cost accounting. It will be based on the total operations count (inclusive iterations count), error estimates, and memory requirements. It will also provide a ratio of the actual operations count of the implemented transport code to the theoretical minimum operations count as determined from the mathematically defined algorithm. If this ratio is much greater than one, the entire program (computer code as well as theoretical background) should be investigated for potential significant reductions in computer time.

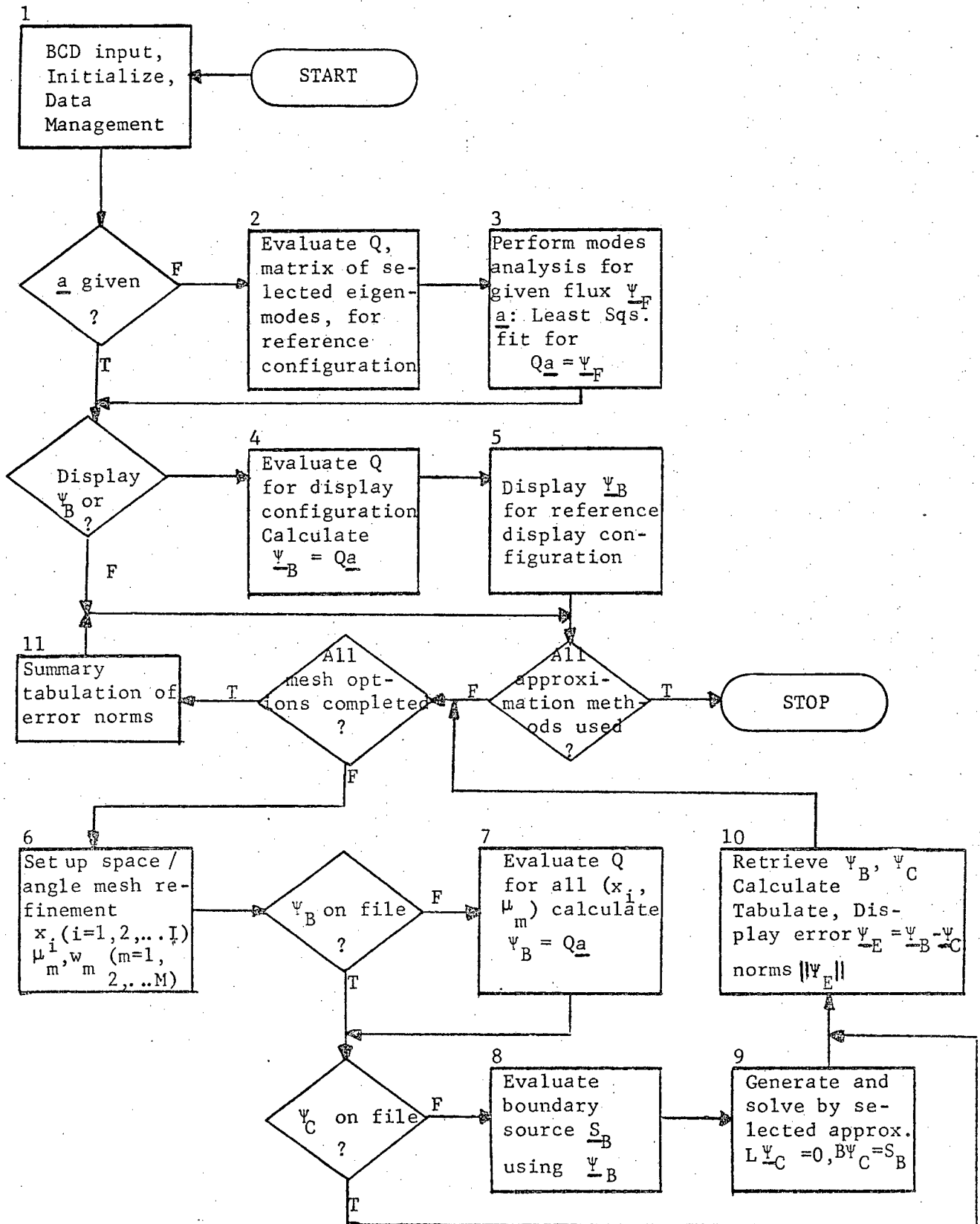
Description of BEAPAC-1T

Since it is very often easier to understand new concepts by considering a set of examples, we describe here an actual code, BEAPAC-1T, show how it can be used by citing concrete examples, and state the underlying motivation for the particular implementation.

BEAPAC-1T stands for Benchmark Error Analysis Package for 1-dimensional Transport theory calculations. It is an implementation of part A outlined above. Figure 1 shows the masterflow chart. We shall discuss the function of the different boxes, but for a more detailed description we refer to the Appendix.

The first tenets for the implementation of BEAPAC-1T are to give the user Freedom to set his own Standards and Flexibility in Applications. BEAPAC-1T is therefore an open-ended (expandable) collection of subroutines. If the user wishes a special feature not provided in the package, he can write his own subroutine and add it by ob-

Figure 1: BEAPAC-1T MASTER FLOW DIAGRAM



serving very simple interface requirements. The user chooses his own performance criteria for his particular problem by calling available options.

Although the linear neutron transport is mathematically uniquely defined, there are several forms of the transport equation which are mathematically equivalent. At this time, the user has the option to base his tests on the following stationary operator equations:

Stochastic Transport Equation (Standard form)

Symmetrized Transport Equation (Canonical form, Vladimirov equation)

Integral Transport Equation

The guiding principle of the project is to store on tape for easy access and in computable form the theoretical knowledge of numerical analysis which is pertinent to the quantitative (numerical) solution of neutron transport problems, algorithms for approximate solutions and special functions with instructions for their proper use. The output is in tabular form and in visual displays, meaningful to the engineers and scientists who are not specialists in numerical analysis. A systematic analysis of a code requires detailed attention, knowledge, and time. As we will demonstrate, a computer can do this tedious work fast and cheaply.

Creation of Benchmark Problems

To specify benchmark problems is in general not a too difficult job for an engineer. However to design a meaningful benchmark problem for which the exact answer is known, proved

rather difficult. It is necessary to have knowledge of the analytic behavior of the solution at interfaces, boundaries, and corners (for multi-dimensional calculations).

Historically, the solution of differential equations was based on "Hard Analysis." The problem was considered solved when the general solution of the operator equation was known. For example, the solution of the Bessel equation is a linear combination of the appropriate Bessel functions of the first and second kind. In other words, these two functions span the solution space. Modern analysis often is satisfied with "Soft Analysis." The solution is imbedded in a given Sobolev space. This approach is good for asymptotic error analysis, for proving existence -, convergence-and consistency theorems. However, this approach is not sufficient for "coarse network" calculations. We have shown early in our research by pilot calculations that some "low" order algorithms gave better results than "high" order methods. An explanation of this fact will be given at the end of this paper.

At this time BEAPAC-1T can generate exact benchmark problems for the homogeneous transport equation using the following exact solutions

$$(1) \quad \psi_{\pm v_0} = \frac{c}{2} \frac{v_0}{v_0 \pm \mu} e^{\pm x/v_0} = \phi_{\pm v_0} e^{\pm x/v}$$

$$(2) \quad \psi_t = \int_{-1}^{+1} A(v) \phi_v e^{-x/v} dv$$

$$(3) \quad \Psi_0 = \begin{cases} 0 & \mu < 0 \\ x^{-\mu} + \mu e^{-x/\mu} + A_2 \Psi_3 + A_3 \Psi_3 + \dots & \mu \geq 0 \end{cases}$$

where

$$(4) \quad \Psi_n = \sum_{k=0}^n \mu^{n-k} E_k(x) + \dots + e^{-x/\mu} [\mu^{n-1} \log(\frac{1}{\mu} - 1) + \dots]$$

Equations (1) are the regular elementary functions introduced by Case, and are known as asymptotic solutions of the transport equations. Equation (2) represents the transient solutions in the notation of Case. They span the solution space of our operator equation. Analytic expressions for (2) are stored for a set of $A(\nu)$ which are piecewise polynomial functions in ν . Equation (3) is an exact solution for a vacuum boundary at $x=0$ ($c=1$) as given by Abu-Shumays and Bareiss. Thus, to create a benchmark problem for a slab with a vacuum boundary, we can calculate $\Psi(a, \mu)$ for all values of μ at the slab width a (or cell width a) which are necessary to solve the problem by a given approximation method. Note, $\Psi(x, \mu)$ can be calculated for any desired location in $\{0 \leq x \leq a, -1 \leq \mu \leq 1\}$ as accurately as desired. For practical purposes, we call this an exact solution.

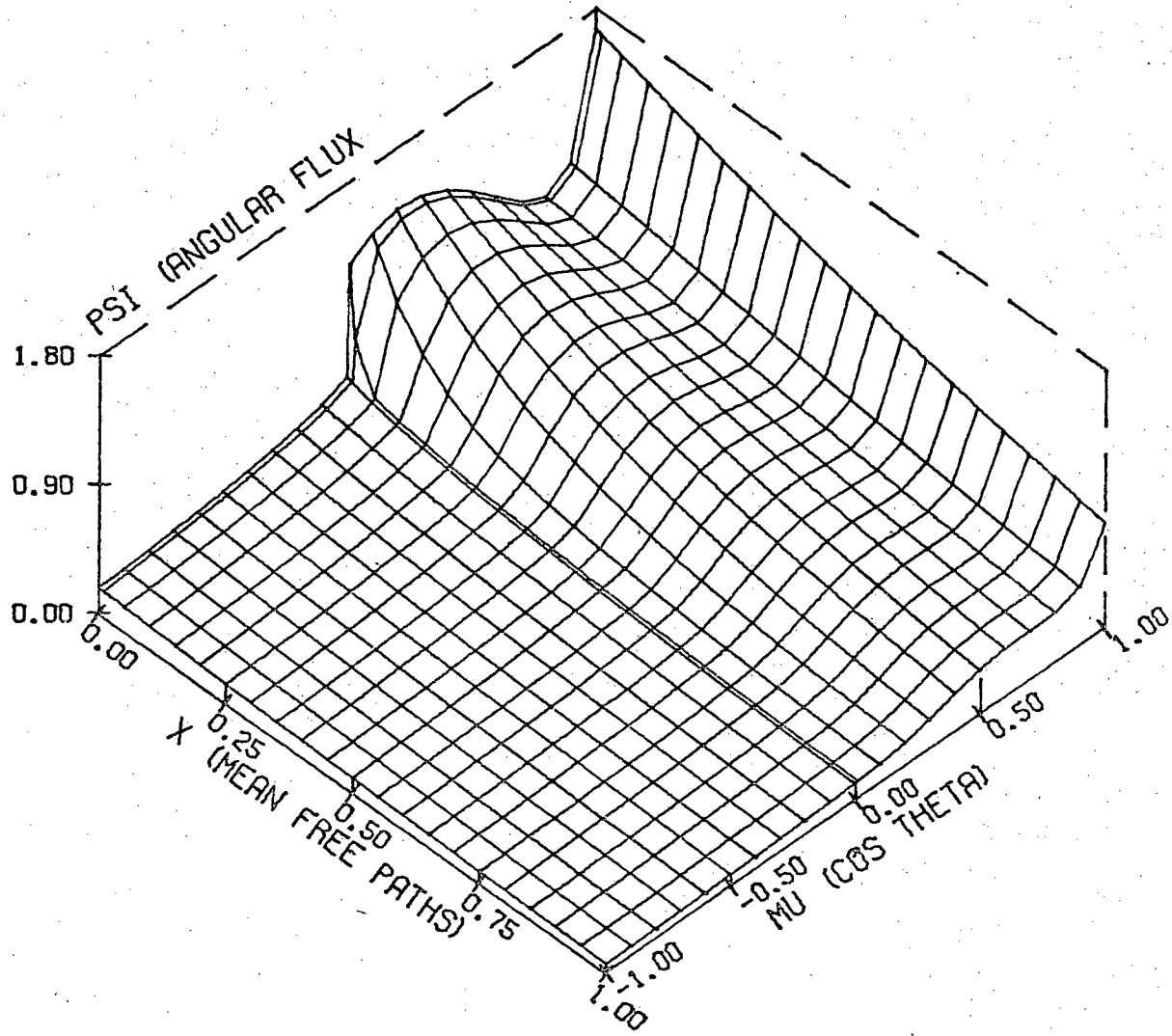
The user may want to simulate an interface condition with an irregular flux shape. This is possible by taking a linear combination of the equations (1) and (2). Figure 2 represents such a flux shape, where

$$(5) \quad \Psi(x, \mu) = a_1 \Psi_{v_0} + a_2 \Psi_{-v_0} + a_3 \Psi_{t1} + a_4 \Psi_{t2}$$

$$a_1 = .75, a_2 = 0, a_3 = .75, a_4 = 1.5$$

Figure 2

GALERKIN (C=.646) COMB. MODE , BC(I,I)



and ψ_{t1} and ψ_{t2} are specified by

$$(6) \quad A^1(\nu) = \begin{cases} 0 & \mu < 0 \\ 1 - \nu & \mu \geq 0 \end{cases} \quad A^2(\nu) = \begin{cases} 0 & \mu < 0 \\ \nu(1-\nu) & \mu \geq 0 \end{cases}$$

This solution contains the jump as well as the logarithmic behavior at $\mu=0$.

BEAPAC-1T contains also a subroutine "MODE ANALYZER."

Assume a transport theory calculation has been made, and the numerical solution looks suspicious, say like Figure 1. Does the code give a wrong numerical answer, or do we have actually such an unexpected flux shape? We use the Mode Analyzer to obtain a discrete least square approximation to the computed Ψ in the form

$$(7) \quad \Psi(x, \mu) \approx a^+ \psi_{\nu_0}^+ + a^- \psi_{-\nu_0}^- + \sum_{k=1}^N \int_{\nu_k}^{\nu_{k+1}} A(\nu) \phi_{\nu} e^{-x/\nu} d\nu$$

Here, $A(\nu)$ is a piecewise linear function, i.e.

$$A(\nu) = \sum_{k=1}^N a_k S_k(\nu)$$

where the $S_k(\nu)$ are chapeau functions (linear splines). The integrals in (7) have been evaluated analytically, and contain logarithmic terms. Thus (7) is

$$(8) \quad \Psi(x, \mu) \approx \underline{q}^T \underline{a} \quad (\underline{q} = \underline{q}(x, \mu))$$

where

$$\underline{a}^T = (a^+, a^-, a_1, a_2, \dots, a_n)$$

$$\underline{q}^T = (\psi_{\nu_0}^+, \psi_{-\nu_0}^-, \dots, \int_{\nu_{k-1}}^{\nu_{k+1}} S_k(\nu) \phi_{\nu} e^{-x/\nu} d\nu, \dots)$$

are vectors. If the computed $\Psi(x,\mu)$ is known at an ordered set of discrete points

$$\{P_i = (x_{k_i}, \mu_{l_i})\}$$

then the set $\{\Psi(P_i)\}$ forms a vector $\underline{\Psi}$ and \underline{a} is defined as the discrete least squares approximation to the matrix equation

$$(9) \quad Q\underline{a} = \underline{\Psi}$$

The row elements of Q are defined from (8) by the relation $\Psi(P_i) = \underline{q}^T(P_i)\underline{a}$. To obtain \underline{a} , we simply apply the Householder factorization to (9). Now a benchmark problem can be generated as above with the exact solution.

$$(10) \quad \Psi_B(x,\mu) \equiv \underline{q}^T \underline{a} \quad \underline{q} = \underline{q}(x,\mu)$$

If the transport code cannot reproduce this solution with sufficient accuracy, we have positive evidence of its unreliability. (We are in the process of implementing (7) with cubic splines and piecewise higher order polynomials to reduce the dimension of \underline{a} . If numerical experience indicates sufficient cost savings, the linear splines for $A(v)$ will be supplemented with higher order approximations.)

Boxes 1 through 5 of the masterflow chart can now easily be explained. If the vector \underline{a} and the corresponding base functions are specified by the input data, boxes 2 and 3 are by-passed, because the Mode Analyzer is not required. To obtain Figure 2, boxes 2 and 3 were by-passed. However to generate the necessary

data for the "three-dimensional" display of Figure 2 of the benchmark problem solution, box 4 is needed to calculate sufficiently many values $\Psi(x_k, \mu_2)$ to assure the desired "smooth" appearance of $\Psi(x, \mu)$. Figure 2 is of intermediate smoothness, which we consider "good enough" for practical purposes.

Evaluation of Numerical Methods

We are now ready to test a numerical method to solve the transport equation with the solution Ψ_B , say that of Figure 2. BEAPAC-1T contains a set of subroutines for the numerical solution of the transport equation by methods which are widely used. They are all in a standard form. The numerical methods are reduced to the problem of solving a linear matrix equation of the form

$$A \underline{v} = \underline{b}$$

Hence, a particular subroutine contains instructions to calculate the elements of the matrix A and the vector \underline{b} . The process is as follows. First, the desired network in the (x, μ) -domain is set up (box 6). Then the necessary boundary values are calculated (box 8). With the information from boxes 6 to 8, the elements of A and \underline{b} are calculated, and the linear equation is solved accurately by a direct method, usually by Gaussian elimination with iterative improvement. In most cases, the elements of \underline{v} represent the discrete directional fluxes, and are denoted

by $\psi_c(x_k, \mu_l)$. If this is not so, a further calculation is necessary to arrive at ψ_c .

The following numerical methods are available as sub-routines.

Derived from the Stochastic Equation:

First, third and fifth order Hermite-Birkhoff approximation in space, Gaussian quadrature in angle. The matrix A of the first order method is identical to the latest version of the S_N -method.

Galerkin method in space with chapeau functions as basis, and discrete ordinates in space (including Gaussian quadrature).

Derived from the Symmetrized Equation:

First order difference equation in space, discrete ordinates in angle.

Piece-wise bilinear Finite Element Method in phase space.

Derived from the Integral Equation:

Nystrom Method.

The user can also interface an existing code for the numerical solution of the transport equation with BEAPAC-1T. This is equivalent to by-passing boxes 8 and 9 and is indicated in Figure 1 by the decision box " ψ_c on file."

The next step is to calculate the errors by the formula

$$\psi_E = \psi_B - \psi_c$$

where Ψ_E is the pointwise difference between the exact solution Ψ_B and the approximate solution Ψ_C . The user can specify a number of options for error norms which he wants to display graphically or print in tabular form.

To illustrate two options, we solved the benchmark problem specified by Figure 2 by the Galerkin-discrete ordinate method mentioned above. Figure 3 shows the absolute error $\Psi_E(x_k, \mu_l)$ at the nodes of the network for which the calculations were done. Table 1 shows the maximum absolute error

$$\|\Psi_E\| = \max_{k,l} |\Psi_E(x_k, \mu_l)|$$

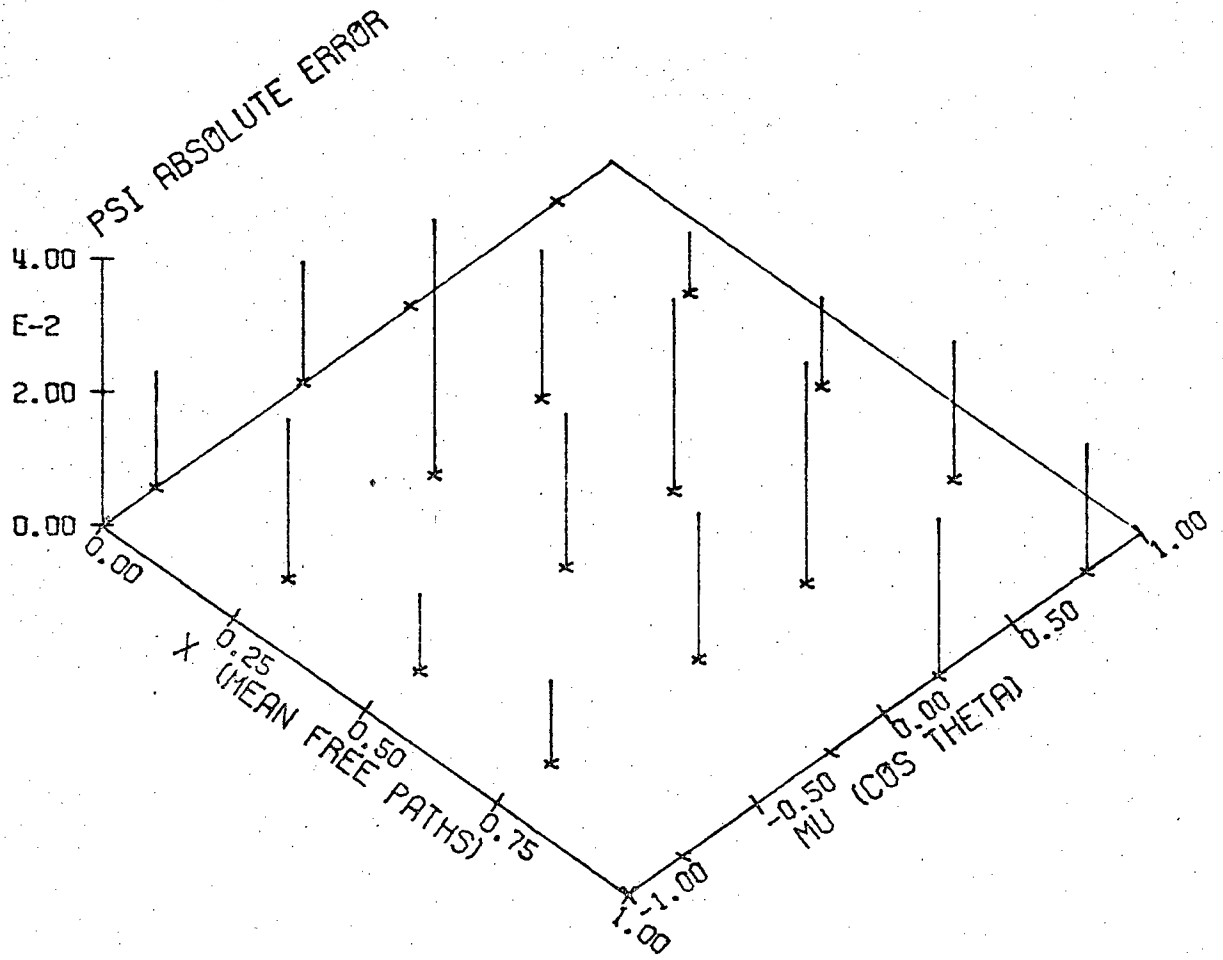
for a run of twenty different mesh configurations and gives additional information. The number 4 in the I column indicates that the calculation was carried out with 4 intervals per mean free path in the x-direction. The number 4 in the J row indicates that 4 double-Gauss abscissae were used. The entry at (4,4) is 3.81E-2. This means the absolute maximum error is 0.038, the signed extreme error is +.0381. The next three entries below mean that the exact solution where the extreme error occurred is 0.227 (rounded to 3 decimals) and the location is $x = .25; \mu = -.211$, i.e.

$$\Psi_B(.25, -.211) = .227$$

The total computing cost for generating the benchmark problem, solve the stochastic transport equation 20 times for the dif-

Figure 3

GALERKIN (C=.646) COMB. MODE , BC (I,I)



LEGEND

C = 0.6462
ANGULAR = 4
SPATIAL = 5
MAX ERR = 0.03814260
ASSOC FLX= 0.2268

Table 1

BENCHMARK ERROR ANALYSIS PACKAGE FOR 1D NEUTRON TRANSPORT METHODS
 1.2.7.6.1 BC STUDY X(0,1) MU(-1,1) (C= .6462, COMB, STD) GLK BC(I,I) 1/23/75

BP 7.10 1.2.7.6.1 BC STUDY X(0,1) MU(-1,1) (C= .6462, COMB, STD) GLK BC(I,I)
 MAX. ABS. VECTOR FLUX ERROR AND LOCATION (GBL)

		I: NO. OF X MESH INTERVALS		J: NO. OF MU POINTS		
		2	4	8	16	
					
ERROR	1	-7.20E-02	1.19E-01	2.87E-01	3.54E-01	
EXACT		0.249	0.166	0.131	0.124	
X LOC		0.0	1.000	1.000	1.000	
U LOC		-0.500	0.211	0.069	0.020	
ERROR	2	-4.01E-02	5.68E-02	1.42E-01	2.08E-01	
EXACT		0.249	0.365	0.229	0.214	
X LOC		0.0	0.500	0.500	0.500	
U LOC		-0.500	0.211	0.069	0.020	
ERROR	4	-3.69E-02	3.81E-02	5.97E-02	1.62E-01	
EXACT		0.249	0.227	0.342	0.294	
X LOC		0.0	0.250	0.250	0.250	
U LOC		-0.500	-0.211	0.069	0.020	
ERROR	8	-3.61E-02	3.15E-02	1.25E-02	1.09E-01	
EXACT		0.249	0.195	0.526	0.357	
X LOC		0.0	0.375	0.125	0.125	
U LOC		-0.500	-0.211	0.069	0.020	
ERROR	16	-3.64E-02	3.11E-02	3.68E-03	4.20E-02	
EXACT		0.228	0.210	0.244	0.433	
X LOC		0.063	0.313	0.063	0.063	
U LOC		-0.500	-0.211	-0.069	0.020	

ferent mesh configurations by the Galerkin method, search for the extreme error and print Table 1 is 7¢ for using low priority on Argonne's IBM 370-195. The cost is 14¢ using high priority. The total computing cost to generate and print Table 1, 20 error plots like Figure 3, and 1 exact solution like Figure 2 is 85¢ for high priority accounting; 40¢ for a low priority job.

These cost figures show that computerized error analysis is feasible. It is possible to have a series of benchmark problems solved for a set of different numerical methods, tabulated, and the tables evaluated again by an additional small computer subroutine. We have actually done this to determine optimal mesh configurations and corresponding error performance for the methods incorporated in BEAPAC-1T at Northwestern University.

WE DO NOT PLACE JUDGEMENT ON THE MERITS OF CODES. But we will publish experimental data. It is important, that the experiments can be duplicated. The user of transport codes, with the help of BEAPAC-1T should form his own opinion with respect to his own problems. Without a code such as BEAPAC, a systematic comparison of methods is out of reach for the engineer.

Two space dimensions

A two-dimensional version of BEAPAC-1T is now being implemented. The output will be very similar to the tables and graphs in a very recent paper by Madsen. It is interesting to note that the Central Difference Scheme he describes coincides with one of

four different schemes which we used to demonstrate the procedure of computerized error analysis in a semi-automated proto-type code when we started this research project. Since we shall include, as in BEAPAC-1T, two-cell (i.e. 2-region) benchmark problems, the code can do what was generally accepted practice in introducing new transport codes. But it will do more, namely base the evaluation on exact solutions of benchmark problems. It will also be able to evaluate numerical methods to solve the neutron diffusion equation.

I shall very briefly mention a representative type of difficulty which we encountered in preparation for the theoretical foundation of the benchmark problems. In order to investigate the analytical behavior of the mathematical solution of the neutron diffusion equation near corners and its importance to coarse mesh calculations, we solved the two-region wedge problem of a circular disc with Neumann boundary conditions and regionwise constant parameters. Details and calculations of what follows were reported by Bareiss and Vickery. In dimensionless form, the problem reduces to solve the following system.

$$-\nabla^2 u_1 + c_1 u_1 = S \quad (|\theta| < \alpha)$$

$$-\nabla^2 u_2 + c_2 u_2 = 0 \quad (\alpha < |\theta| < \pi)$$

$$u_1 = u_2, \quad d \frac{\partial u_1}{\partial \theta} = \frac{\partial u_2}{\partial \theta} \quad (\theta = \alpha)$$

$$\frac{\partial u_1}{\partial \theta} = 0 \quad (\theta = 0), \quad \frac{\partial u_2}{\partial \theta} = 0 \quad (\theta = \pi), \quad \frac{\partial u_1}{\partial r} = \frac{\partial u_2}{\partial r} = 0 \quad (r = 1)$$

where (r, θ) are polar coordinates, 2α is the vertex angle, and c_1, c_2, d, S are constant parameters.

If $c_1 = c_2 = 0$, the solution is given by

$$(11) \quad u = \sum_{i,j=0}^{\infty} a_{ij} \psi_i(\theta) J_{\nu_i}(\mu_{ij} r)$$

where

$$\psi_i = \begin{cases} \cos \nu_i (\pi - \alpha) \cos \nu_i \theta & (u=u_1; \quad 0 \leq \theta \leq \alpha) \\ \cos \nu_i \alpha \cos \nu_i (\pi - \theta) & (u=u_2; \quad \alpha \leq \theta \leq \pi) \end{cases}$$

The ν_i are the solutions of

$$(12) \quad d \tan \nu \alpha = -\tan \nu (\pi - \alpha)$$

and the μ_{ij} are the zeros of the first derivatives of the Bessel functions $J_{\nu_i}(x)$. For this case, it is easy to calculate u for $r=0$ as accurately as desired.

If $c_1 \neq c_2 \neq 0$, we were not able to calculate u for $r=0$ by the expansion method to more than one or two significant digits with the limit of computation cost we had imposed. The reason is that the ansatz (11) for this problem is wrong from a computational point of view. An investigation shows that

$$(13) \quad u = \sum_{\ell=0}^{\infty} \sum_{k=0}^{\infty} \sum_{i=0}^{\infty} \psi_i J_{\nu_k}(\mu_{k\ell} r)$$

if all $\nu_k \neq 2j$ ($j=1,2,\dots$). The ν_k 's are again solutions of (12). Each time $\nu_k = 2j$, we must add terms of the form

$$r^{\nu_k + 2j} (\ln r)^m$$

to (13). If ν_k is close to $2j$, the numerical process is extremely

unstable. This is a very unsatisfactory situation, since a small change of α in (12) can change the analytic character of (13) completely. We are trying out several other approaches for the analytical representation of u , one of which we hope will be mathematically satisfactory, and show uniform computability. This function will then be a base function for the generation of benchmark problems.

Since we know that u can behave in the radial direction as

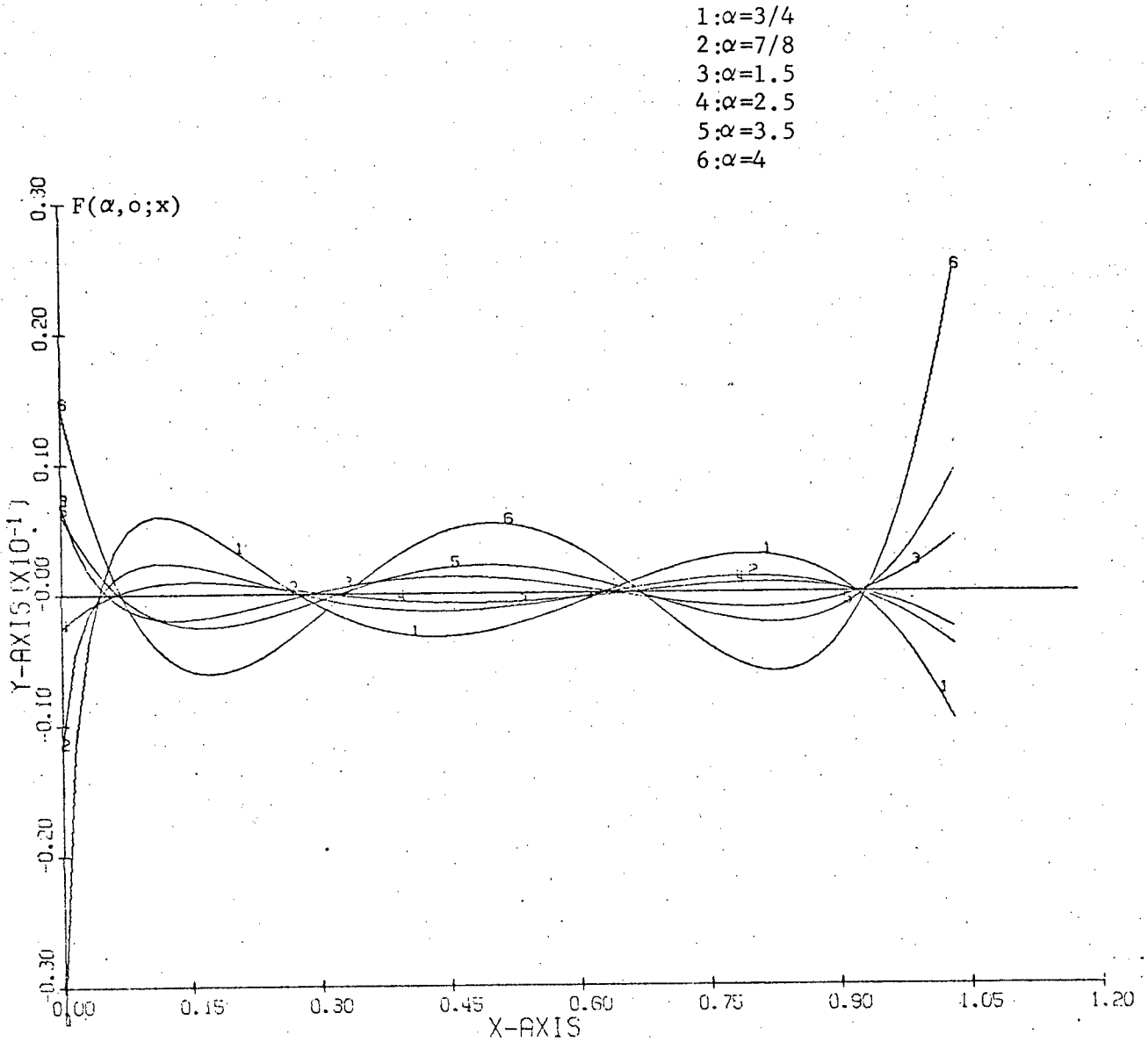
$$a_0 + r^{v_0} (a_{10} + a_{11} r^2 + \dots) \\ + r^{v_1} (a_{20} + a_{21} r^2 + \dots) + \dots$$

the asymptotic theory requires only that the set of base functions for the approximation space contains r^{v_0} , the smallest eigenvalue of (12). For coarse mesh calculations this is not sufficient. In problems of practical interest, there exists only a finite number of v_k such that

$$v_k \neq v_i \pmod{m} \quad (i < k)$$

(m is an integer). Thus the $r^{v_i + m}$ must be included in the base functions of the approximation space for coarse mesh calculations. To show the effect of neglecting to do so, we refer to Figure 4, which shows the error F of least squares approximations of x^α by cubic polynomials, for α in the range from $3/4$ to 4 , and over an interval $h=1$. If the interval differs from unity,

Figure 4



Error function $F(\alpha, 0; x)$ for cubic least squares approximation of x^α

then the error $E_h(x, \alpha)$ is related to the error $F(\alpha, 0; x)$ of Figure 4 by the relation

$$E_h(x, \alpha) = h^\alpha F(\alpha, 0; x)$$

Thus, if $h \ll 1$, then the error for $\alpha = 4$ (curve 6) is negligible with respect to the error $\alpha = 3/4$ (curve 1). We see also that the error of curve 1 is extremely large at the origin. This explains why u in the wedge problem for the diffusion equation is difficult to calculate at the origin $r = 0$. Figure 4 shows also that a collocation method can yield good results when we place collocation points close to those 4 points where the error $F(\alpha, 0, x) = 0$.

The purpose of these last remarks is to show that Quantitative Error Analysis opens a challenging new field of research with important implications for successful large-scale computation.

Bibliography

- Abu-Shumays, I. K., and E. H. Bareiss, "Adjoining Appropriate Singular Elements to Transport Theory Computations," J. Math. Anal. Appl. 48(200-222) 1974
- Bareiss, E. H., and S. R. Vickery, "The Interior Corner Problem for the Neutron Diffusion Equation," AEC Progress Report COO-2280-17, Technological Institute, Northwestern University, November 1974
- Case, K. M. and P. F. Zweifel, "Linear Transport Theory," Addison-Wesley, Reading, MA, 1967
- Inönü, Erdal, "A Scale Group for the Linear Boltzmann Equation and its Representation," Proceedings, Fourth Conference on Transport Theory, Blacksburg, VA. (1975)
- Madsen, N. K., "Weighted Central Difference Equations for the Neutron Transport Equation," Nucl. Sc. Eng. 54(374-386) 1974

APPENDIX

Description of Master Flow Chart BEAPAC-1T (Figure 1)

BEAPAC-1T is a stand-alone modular system designed for quantitative error performance analysis of numerical methods applied to the one dimensional neutron transport equation in slab geometry.

$$(A1a) \quad L\Psi = \mu \frac{\partial \Psi(x, \mu)}{\partial x} + \sum_t \Sigma_t(x) \Psi(x, \mu) - \frac{\Sigma_s(x)}{2} \int_{-1}^1 d\mu' \Psi(x, \mu') = 0$$

$$(A1b) \quad \Psi(x, \mu) = S_B(x, \mu) \quad \text{for } x \text{ on boundary.}$$

There are three major components to the system. In the first component a one or two cell benchmark problem is defined and the neutron flux solution $\Psi_B(x, \mu)$ is generated. A flux vector $\underline{\Psi}_B$ is obtained by evaluating $\Psi_B(x, \mu)$ over ordered set of space-angle mesh points.

The solutions $\Psi_B(x, \mu)$ are linear combinations of the Case eigen-solutions of (1a). The general form of $\Psi(x, \mu)$ expansion is

$$(A2) \quad \Psi(x, \mu) = a^+ \Psi_{\nu_0^+}(x, \mu) + a^- \Psi_{\nu_0^-}(x, \mu) + \int_{-1}^1 A(\nu) \Psi_{\nu}(x, \mu) d\nu$$

We choose for Ψ_B a linear combination as in (2), but $A(\nu)$ is a piecewise polynomial for a particular discretization of the ν variable, ν_n ($n=0, \pm 1, \pm 2, \dots, \pm N$). Given this exact solution to the infinite slab, source boundary conditions S_B for the finite slab problem are evaluated from Ψ_B at the slab boundaries.

The second component contains a variety of numerical procedures for computing the approximate solution $\underline{\Psi}_C$ to (1) from

$$\underline{L} \underline{\Psi}_C = 0 \quad (1c)$$

$$\underline{\Psi}_C = \underline{S}_B \quad \text{for } x \text{ on boundary} \quad (1d)$$

Usually, a systematic (x,μ) mesh refinement is successively applied to the basic cell configuration. Thus a series of $\underline{\Psi}_C$ vectors are generated for a given problem specification.

The third component takes $\underline{\Psi}_B$ and $\underline{\Psi}_C$ (exact and computed flux vectors) and calculates the error vector $\underline{\Psi}_E$ and associated error norms $\|\underline{\Psi}_E\|$. The flux vectors, error norms and related data are displayed graphically and tabularly.

Next, we describe the Master Flow Diagram, Figure 1. The various functions performed are described in the subsequent paragraphs whose numbers correspond to the flow diagram scheme.

Master Flow Diagram Description

1. The BCD input data are read from punched cards. Included are the following general options:
 - a. General control information
 - job title
 - dynamic storage size
 - edit flags
 - interface file existence flags
 - b. Benchmark problem specifications
 - operator type
 - computation cell geometry parameters
 - mode parameters
 - boundary conditions
 - a systematic set of space-angle mesh points

c. Computational methods

- spatial and angular options approximations
- single or double precision Gauss elimination
- Householder factorization (including over determined systems)

d. Error norms and display options

- Norm types
- Norm domain classifications
- Norm tabulations
- 2D- and 3D- plots on Calcomp plotter

The input is checked for consistency. Dynamic storage requirements are computed and if the user specified container space is exceeded, the job will be terminated. Specifications for the computational cells consist of coordinates and cross sections. The code permits irregular mesh refinement in each cell. One and two cell problems are permitted.

2. This block calculates the matrix elements of Q for the reference configuration $(x_k, \mu_l)_{ref}$ for a computational cell. The columns of Q are the components of a base function (such as a mode) evaluated at all points $(x_k, \mu_l)_{ref}$ for a desired benchmark solution ψ_B . This ψ_B is obtained from an input flux distribution $\psi_F(x, \mu)$ evaluated at the reference configuration to give ψ_F . Thus we seek to obtain $\psi_B = Q\mathbf{a} = \sum_{n=1}^N a_n \mathbf{q}_n$, where \mathbf{a} is determined as the discrete least squares solution of

$$Q\mathbf{a} = \psi_F$$

by Householder reduction in block 3.

Equation (A2) contains two types of modes, the asymptotic and transient modes. The asymptotic modes are of the form

$$q_0^+(x, \mu) = \Psi_0^+(x, \mu) = \frac{c}{2} \frac{v_0 e^{-\sum_t x/v_0}}{v_0 - \mu}$$

where v_0 is determined by

$$1 = \frac{c}{2} v_0 \log \frac{v_0 + 1}{v_0 - 1}$$

and

$$c = \frac{\sum_s}{\sum_t}$$

The transient modes are generated by subdividing the integration interval of v into a small number of subintervals Δv_n where $A(v)$ is approximated piecewise by low order polynomials,

$$A(v) \doteq \sum_{n=-N}^N a_n \theta_n(v) \quad (n=0, \pm 1, \dots, \pm N)$$

$$\theta_n(v) = \begin{cases} p_n(v) & v \in \Delta v_n \\ 0 & v \notin \Delta v_n \end{cases}$$

Thus we have

$$q_{n+N+1}(x, \mu) = \int_{(\Delta v_n)} \theta_n(v) \Psi_v(x, \mu) dv$$

The set of all values $q_{n+N+1}(x_k, \mu_k)$ form the vector \underline{q}_{n+N+1} . The matrix Q consists of the column vectors \underline{q}_k

$$Q = [\underline{q}_0^+, \underline{q}_0^-, \underline{q}_1, \dots, \underline{q}_{2N+1}]$$

The corresponding vector \underline{a} has the components $a_0^+, a_0^-, a_1, \dots, a_{2N+1}$.

A two-cell calculation permits the examination of error performance at interfaces. The benchmark solution ψ_{B1} is generated in the first cell as before. The flux shape in the second cell must conform to the flux shape of the first cell at the interface. Hence, we may specify any flux shape ψ_{B2} which satisfies this condition. We are thus able to analyze the error performance for a variety of flux shapes in cell 2. The flux shapes chosen are based on theoretically obtained results for the analytical behavior of the solution. A complication arises in the second cell when we choose a flux shape which satisfies the interface condition but does not satisfy (1) in cell 2. Here we calculate a distributed source term, $S(x,\mu)$, such that $L_2\psi_{B2} = S(x,\mu)$.

3. This block calculates the vector \underline{a} which satisfies, in the least squares sense the possibly overdetermined matrix equation

$$Q\underline{a} = \underline{\psi}_F$$

by the Householder factorization (see block 2). $\underline{\psi}_F$ can be a user supplied flux distribution vector.

4. This block calculates Q , the matrix of base function values, similar to block 2. However, the geometry configuration is now a fine mesh display configuration, so chosen to obtain a high resolution 3D-plot on the Calcomp plotter. Given the matrix Q and the combining coefficients of vector \underline{a} , we evaluate the flux solution $\psi_B(x,\mu)$ for the given display coordinates configuration

(x_i, μ_m) display.

$$\underline{\psi}_B^{(d)} = Q \underline{\psi}_a^{(d)}$$

5. This block retrieves $\underline{\psi}_B$ and or $\underline{\psi}_F$ and displays them in a 3D projection map. Continuous line segment plotting is applied with hidden lines optionally removed. The observation position may be adjusted to any point in the positive octant of the three dimensional cube of viewing space.

6. This block performs the space or angle mesh refinement for the current mesh option. Spatial refinement is usually accomplished by successively halving the mesh interval size. Angular refinement is dependent on the type of discretization. For single or double Gaussian quadrature, the number of angular nodes are successively doubled. For trapezoidal integration the size of the angular intervals are successively halved. Thus we obtain the space co-ordinates

$$x_i \quad (i = 1, 2, \dots, I)$$

and direction cosines and integration weights

$$\mu_m, w_m \quad (m = 1, 2, \dots, M)$$

or the mesh configuration $(x_i, \mu_m)_C$

7. This block evaluates the corresponding benchmark solution $\underline{\Psi}_B = Qa$ at the mesh configuration $(x_i, \mu_m)_C$. The base functions are the same as in block 4. Specially constructed linear combinations of eigenmodes may be selected to simulate flux and flux derivative discontinuities.

8. In this block the appropriate source vector \underline{S}_B is determined at the boundaries of the slab, using $\underline{\Psi}_B$ for the specified boundary conditions.

Boundary conditions available for the standard operator are:

- | | | |
|--|---|---|
| 1. Vacuum | $\Psi(b, \mu) = 0$ | $\mu \underline{i} \cdot \underline{n} < 0$ |
| 2. Reflective | $\Psi(b, \mu) = \Psi(b, -\mu)$ | $\mu > 0$ |
| 3. Inhomogeneous Dirichlet
(Incident) | } $\Psi(b, \mu) = S(b, \mu)$ { | $\mu \underline{i} \cdot \underline{n} < 0$ |
| 4. Inhomogeneous Dirichlet
(Emergent) | | $\mu \underline{i} \cdot \underline{n} > 0$ |
| 5. Inhomogeneous Neumann
(Incident) | } $\left. \frac{\partial \Psi(x, \mu)}{\partial x} \right _{x=b} = S(b, \mu)$ { | $\mu \underline{i} \cdot \underline{n} < 0$ |
| 6. Inhomogeneous Neumann
(Emergent) | | $\mu \underline{i} \cdot \underline{n} > 0$ |
| 7. Inhomogeneous Reflective | $\Psi(b, \mu) - \Psi(b, -\mu) = S(b, \mu)$ | $\mu > 0$ |

where \underline{i} is the unit vector in the positive x-direction

\underline{n} is the unit outward normal at the slab boundaries

$$|\underline{i} \cdot \underline{n}| = 1$$

$$\underline{b} = \begin{cases} x_1 & \text{left hand boundary} \\ x_I & \text{right hand boundary} \end{cases}$$

Thus we have the following cases for \underline{S}_B

$$\begin{aligned}
 1,2. \quad \underline{S}_B &= 0 && \begin{cases} \mu_m^i \cdot n < 0 \\ \mu_m^i \cdot n > 0 \end{cases} \\
 3,4. \quad \underline{S}_B &= \underline{\Psi}_B|_{(b,\mu)} && \begin{cases} \mu_m^i \cdot n < 0 \\ \mu_m^i \cdot n > 0 \end{cases} \\
 5,6. \quad \underline{S}_B &= \left(\underline{\Psi}_B|_{(b,\mu)} - \underline{\Psi}_B|_{(b',\mu)} \right) \frac{1}{\Delta x} \begin{cases} \mu_m^i \cdot n < 0 \\ \mu_m^i \cdot n > 0 \end{cases} && (b-b'=\Delta x) \\
 7. \quad \underline{S}_B &= \underline{\Psi}_B|_{(b,\mu)} - \underline{\Psi}_B|_{(b,\mu)} && \mu > 0
 \end{aligned}$$

Boundary conditions for the symmetrized operator are:

1. Homogeneous Neumann condition(left boundary)

$$\Psi^+(b,\mu) = S_B^+(x_I,\mu)$$

The symmetrized problem deals only with $0 \leq \mu \leq 1$ since

$$\Psi^+(x,\mu) = \Psi^+(x,-\mu).$$

Thus we have the following cases for $S_B^+(b,\mu)$

1. $\underline{S}_B^+|_{(0,\mu)} = 0$
2. $\underline{S}_B^+|_{(x_I,\mu)} = \underline{\Psi}_B^+(x_I,\mu)$

9. Given $(s_{i,\mu_m})_C$, \underline{S}_B , and macroscopic cross sections Σ_s, Σ_t we generate and solve the matrix equation for

$$L\underline{\Psi}_C = 0 \quad (x,\mu) \in (x_i,\mu_m)_C$$

$$B\underline{\Psi}_C = \underline{S}_B \quad \text{on boundary}$$

We classify the approximation methods by origin of operator type.

Standard Equation (discrete ordinates in angle)

1. 1st order Hermite Birkhoff Method (S_n -method)
2. 3rd order Hermite Birkhoff Method
3. 5th order Hermite Birkhoff Method
4. Galerkin Weighting Method

Symmetrized Equation

1. 1st order Finite Difference Method (discrete ordinates in angle)
2. Piecewise bilinear Finite Element Method in phase space

Integral Equation

1. Nystrom Method

The solution to the matrix equation is usually obtained using block tridiagonal elimination for the standard equation and by Choleski reduction in the symmetrized case. For additional information two options are available. The first provides for a single or double precision solution of the linear system. The second provides the alternative of Householder least squares solution of the linear system. These options provide round off information and stability information.

10. This block computes specified flux errors, error norms, and related information. Two pointwise error options are available

a. algebraic flux errors
$$\underline{\Psi}_E = \underline{\Psi}_B - \underline{\Psi}_C$$

b. relative flux error
$$\underline{\Psi}_E = \frac{\underline{\Psi}_B - \underline{\Psi}_C}{\underline{\Psi}_B}$$

Norms based on ψ_E are:

- a. $\|\psi_E\|_\infty$: maximum absolute ψ_E with associated mesh location, algebraic sign and flux values.
- b. $\frac{\|\psi_E\|_1}{N}$: average absolute ψ_E
N = number of calculated fluxes

The norms may be computed for a variety of domains.

1. Global : all calculated flux nodes
2. Boundary : boundary and interface flux nodes
3. Interior : interior flux nodes (non boundary)

(GLOBAL = BOUNDARY U INTERIOR)

For multicell problems the domain may be restricted to individual cells.

The corresponding scalar flux may be optionally tabulated for each angular flux vector.

A refinement of the pointwise error scheme is anticipated. The idea is to employ natural interpolation at a specified set of nodes independent of the mesh discretization. The objective is to maintain a fixed number of nodes at which the flux error is computed. The interpolation uses the Nyström method in conjunction with the Integral Transport equation.

Edits and 3D to 2D projection plots of the ψ_B , ψ_C and ψ_E are optional. Summary edits of the associated error norms are optional.

11. Upon completion of all mesh options, summary tabulations of each error norm and norm domain are generated. A typical tabulation has the spatial refinements increasing down a column, while angular refinements increase along a row.