NUCLEAR SCIENCE AND ENGINEERING: 29, 314-316 (1967)

Computer Code Abstracts

EXTERMINATOR-2*

A. Name of Code: EXTERMINATOR-2[1].

B. Computer for Which Code is Designed: EXTERMINATOR-2[1] is a machine-independent code in the sense that if a particular computer has a FORTRAN-IV compiler, then the code can easily be made to run on the machine.

C. Nature of Problem Solved: The multigroup, two-dimensional, neutron diffusion equations are solved in XY, RZ, or $R\theta$ geometry.

D. Method of Solution: The Equipoise method[2] is employed to solve the finite-difference analogs of the multigroup neutron diffusion equations.

E. Restrictions on the Complexity of the Problem: Since variable dimensioning techniques available in FORTRAN-IV where used in this code, the only restriction on the size of problem is the available core storage. The code examines the size of each problem and stores fluxes and equation coefficients (except for scattering matrix coefficients which are recalculated at each iteration) according to the core size of the machine in one of four ways:

1) All fluxes and equation coefficients are contained in core and no I/O devices are used during the iterative part of the calculation

2) All equation coefficients are contained in core and I/O devices are used to store the fluxes

3) The fluxes are contained in core and I/O devices are used to store the equation coefficients

4) Both the coefficients and fluxes are used from I/O devices.

For a problem which has I rows and J columns in the mesh, K energy groups, M different materials, N nuclides, and L sets of specifications of composition locations, the amount of core storage required for variables, when all fluxes and equation coefficient are used from I/O devices, is

$$4I + 24J + 16K + 2M + 6N + (2M + N + 1)K^2 + 8IJ + 14JK + 15MK + MN + 4NK + 5L + 8$$

words.

The core storage required to be able to contain the fluxes and coefficients in core is

$$4I + 24J + 16K + 2M + 6N + (2M + N + 1)K^2 + 8IJ + 4JK + 15MK + MN + 4NK + 5L + 8IJK + 10.$$

If all cross sections are in macroscopic form, N is set to 1 in the above expressions.

F. Typical Running Time: The running time of a problem will depend upon the size of the problem, the computer, the compiler and operating system being used, and the amount of I/O required during the iterative part of the calculation. The following table gives the running times and rates of some typical problems run on the IBM-360/75 computer using the IBM FORTRAN-IV compiler with level-2 optimization and Operating System/360. The machine has a 512K-byte core memory plus 1024K bytes of Large Capacity Storage.

Problem	Mesh Size $I \times J$	Number Groups	2311 Disk I/O	Number of Iterations	Time Min	Rate sec/point/group/ Iteration
1	31 × 31	3	No	83	3	0.0007
2	42 × 25	9	No	151	17	0.0007
3	31 × 81	9	No	92	28	0.0008
4	51 × 51	25	Yes	149	255	0.0016

G. Unusual Features of the Code:

1) Three outer-boundary conditions may be imposed: zero flux, zero normal derivative, or periodic. The logarithmic boundary condition may also be specified; either along boundaries or internal to the mesh.

2) The code allows neutron scattering from any group to any other group.

3) Eigenvalue problems, constant source problems, poison search problems, and nuclide density search problems may be solved by direct iteration, with the unknown treated as the eigenvalue of the problem. Indirect search by solution of succeeding eigenvalue problems is also included. Special neutron problems can be solved in which some of the fluxes are negative.

4) The effect on the multiplication factor and the fluxes due to pointwise equilibrium xenon concentrations may be taken into account.

5) The code will calculate adjoint fluxes and do perturbation calculations.

6) Flux-weighted broad-group microscopic cross sections may be calculated.

7) Succeeding cases require only those input data which are different from the input from the preceding case.

8) Optional output includes point-group fluxes as n/(cm^2 sec), point neutron density as n/cm^3, point source density as fissions/(cm^3 sec) or neutron productions/(cm^3 sec), nuclide reaction rates, total and composition neutron balances, and cumulative heating along defined coolant channels.

*Research sponsored by the USAEC under contract with the Union Carbide Corporation.

H. Related and Auxiliary Programs: This code is a FORTRAN-IV version of the code EXTERMINATOR[3], with major improvements.

I. Status: In use.

J. Machine Requirements: A machine with a minimum of about 64K words of core storage and 5 I/O devices for temporary storage in addition to those for input data and printed output. Some problems may require 4 additional I/O devices.

K. Programming Language Used: FORTRAN-IV.

L. Monitor System: Not specified.

M. Programming Information: The code can be made to conform to machines of different core sizes simply by adjusting the fixed dimension of only one variable in a short master program.

The code compiled under the IBM FORTRAN-IV compiler on the IBM-360 computer is about 35 000 words long, but can be shortened by as much as one-half this length by removing some of the subroutines which do optional calculations. Using the overlay feature of the IBM-360 operating system, the storage requirements can be reduced to about 15 000 words.

N. Material Available: FORTRAN deck, sample problem input deck, and the report which includes the input and output of the sample problem.

O. *References*:

[1]T. B. FOWLER, M. L. TOBIAS, and D. R. VONDY, "EXTERMINATOR-2: A FORTRAN-IV Code for Solving Multigroup Neutron Diffusion Equations in Two Dimensions," ORNL-4078, Oak Ridge National Laboratory (April 1967).

[2]M. L. TOBIAS and T. B. FOWLER, "The Equipoise Method—A Simple Procedure for Group-Diffusion Calculations in Two and Three Dimensions," *Nucl. Sci. Eng.*, 12, 513 (1962).

[3]T. B. FOWLER, M. L. TOBIAS, and D. R. VONDY, "EXTERMINATOR—A Multigroup Code for Solving Neutron Diffusion Equations in One and Two Dimensions," ORNL-TM-842, Oak Ridge National Laboratory (February 1965).

T. B. Fowler
M. L. Tobias
D. R. Vondy

Oak Ridge National Laboratory
Oak Ridge, Tennessee 37830

Received January 31, 1967

WIGL2

1. Name of Program: WIGL2

2. Computer for Which Program is Designed and Programming Language Used: CDC-6600, FORTRAN IV.

3. Nature of Problem Solved: WIGL2 is a one-dimensional, two-group, space-time diffusion-theory program with zero, one, or six delayed-neutron groups. The program will treat slab, cylindrical, and spherical geometries and includes nonboiling heat transfer. It accounts for xenon feedback and feedback effects due to fuel and coolant temperature. Control-rod motion and control-system feedback based on total core power or outlet coolant temperature can be simulated. Transients may be excited by prescribed changes in inlet coolant temperature, coolant flow rate, or rod position.

4. Method of Solution: The neutron equations along with the various feedback processes form a nonlinear system of equations. The solution is obtained by computing the changes in nuclear parameters during a given time step by using the neutron fluxes at the beginning of the time step. The time-difference form of the diffusion equations is obtained by replacing continuous time by a sequence of times and introducing a set of internally computed time-dependent parameters (θ's) to increase the stability and accuracy of the numerical method. The θ-method is discussed in detail by Henry and Vota[1]. At each time step, the spatial equations are solved by a simultaneous inversion of the two-group operator; thus no spatial iteration is required. The coefficients of these equations are functions of local coolant temperature, fuel metal temperature, and control motion.

5. Restrictions on Complexity of the Problem: No restrictions are placed on single parameters related to problem size. The total amount of storage set aside for all parameters is 14 000 memory locations. This allows the user to decide which area requires a more detailed representation.

6. Related Programs: WIGL2 for the CDC-6600 is a slightly modified version of the FORTRAN II program, WIGL2, written for the Philco-2000.

7. Typical Running Time: A problem with 50 spatial points, 20 thermal-hydraulic regions, and 20 nuclear regions would run less than one second a time step. The time increases with an increase in these parameters. The amount of printing requested also affects running time.

8. Unusual Features of the Program: The θ-method used in the difference equations is unique in WIGL2. Also, the program performs an automatic criticality rod search and contains rather complete feedback.

9. Status: The program is currently in use and may be obtained by domestic users from the Argonne Code Center.

10. Machine Requirements: WIGL2 requires $130\,000_8$ central memory words and one disk for storage of permanent files. The number of central memory words required could be decreased by decreasing the amount of storage set aside for parameters. This would also further limit problem size. If the ability to store data on disk for succeeding WIGL2 problems is not desired, the disk is not needed.

11. Operating System: SCOPE 2.0.

12. Other Programming Information: The required software environment is described by Pfeifer[3]. It includes routines for program loading, input conversion, and the storage, retrieval, and processing of permanent files.