Computer Code Abstract

TORT

A Three-Dimensional Discrete Ordinates Neutron/Photon Transport Code

W. A. Rhoades and R. L. Childs

Oak Ridge National Laboratory, P.O. Box 2008
Oak Ridge, Tennessee 37831-6363

Received January 11, 1990
Accepted April 10, 1990

1. Problem Solved: TORT calculates the flux or fluence of particles throughout a two- or three-dimensional geometric system from particles incident on the system from extraneous sources or generated internally as a result of interaction with the system. The principal application is for the deep-penetration transport of neutrons and photons. Certain reactor eigenvalue problems can also be solved. Numerous printed edits of the results are available, and many results can be transferred to output files for subsequent analysis.
2. Method of Solution: The Boltzmann transport equation is solved using the method of discrete ordinates to treat the directional variable and recursion relationships to treat spatial variables. Energy dependence is treated using a multigroup formulation. Time dependence is not treated. Starting in one corner of a mesh, at the highest energy, and with starting guesses for implicit sources, boundary conditions and recursion relationships are used to independently sweep into the mesh for each discrete direction. Integral quantities such as scalar flux are obtained from weighted sums over the directional results. The calculation then proceeds to lower energy groups, one at a time.

   Iterations are used to resolve implicitness caused by scattering between directions within a single energy group, by scattering from an energy group to another group previously calculated, by fission, and by certain boundary conditions. Methods are available to accelerate convergence. Anisotropic scattering is represented by a Legendre expansion of arbitrary order; methods are available to mitigate the effect of negative scattering estimates resulting from finite truncation of the expansion. Direction sets can be biased, concentrating work into directions of particular interest.

   Fixed sources can be specified at either external or internal mesh boundaries, or distributed within mesh cells. Either cylindrical (r-θ-z) or Cartesian (x-y-z) geometry is supported. A novel adaptation of the nodal method provides high-order accuracy in x-y-z geometry. Additional details are given in Refs. 1, 2, and 3 and in the descriptive material included in the source file.
3. Related Material:
   a. Data files:
      (1) macroscopic cross-section file (required)
      (2) external boundary source file (optional)
      (3) distributed source file (optional)
      (4) flux guess input file (optional)
      (5) flux output file (optional)
      (6) response summary output file (optional).
   b. Programs:
      (1) DRV—a driver program that coordinates the execution of worker modules
      (2) GIP—a cross-section file preparation code
      (3) DOTTOR—a code that prepares external boundary sources based on a two-dimensional calculation.
4. Restrictions and Limitations: External forces and nonlinear physical effects cannot be treated. Penetration through large, nonscattering regions may become inaccurate because of ray effects. Problems with scattering ratios near unity or eigenvalue calculations with closely spaced eigenvalues may be quite time-consuming. Flexible dimensioning is used throughout, so that no fixed limits on group, problem size, etc., are applicable.

   The development of TORT was supported entirely from problem-solving funds provided by the Defense Nuclear Agency, Department of Defense, and the focus has been on problems of interest to that sponsor. Certain features found in DORT have not been added or completed in TORT. The code is provided on an *ad hoc* basis, and direct consultation with the originators will require a valid contract arrangement.
5. Computers and Operating Systems: TORT is designed to be adaptable to most large-scale computers that support direct (random)-access disk storage or the equivalent. Extensive tailoring to the architecture of vector processors has been done, and such computers provide important efficiency

gains over scalar computers. Because of the self-adaptive nature of the trade-off of fast memory versus external storage, no special changes are required to take full advantage of virtual memory if available. Insofar as possible, machine-dependent features or features that may give trouble on a particular system have been concentrated in interchangeable compatibility packages. Correct operation has been demonstrated on Cray 1, Cray X-MP, and Cray 2 computers using the CTSS operating system, and on a Cray X-MP using the UNICOS version 5.0 system. IBM compatibility is not currently maintained, although it could be restored for a willing customer.

6. Languages, Compilers, and Libraries: On Cray systems, TORT can operate in a 100% FORTRAN configuration, using calls to the system subroutine libraries for necessary system facilities such as input/output. Optional assembler language routines are available to enhance performance if desired. Either Cray CFT 1.14, CFT 1.15, or CFT 77 version 3.0 compilers can be used with TORT. The FORTRAN 77 language standard is followed closely. An exception is the use of the traditional Hollerith data type as described in the CFT77 document. On Cray-CTSS systems, either the Los Alamos National Laboratory CFTLIB or the Lawrence Livermore National Laboratory NMFECC FORTLIB library can be used. The standard library distributed with the Cray UNICOS version 5.0 can also be used.

7. Running Time: CPU time used by the flux sweep is roughly proportional to the number of flux calculations:

$$\text{space} = \text{mesh cells} * \text{directions} * \text{energy groups} * \text{iterations/group}.$$

Efficiency on vector processors depends somewhat on the number of intervals in the first space dimension. A string length of 127 is ideal for Cray, while strings <20 will show a loss of efficiency. Note that TORT problems use twice as many directions for a given order of quadrature compared to DORT. A large problem with 104 247 mesh cells and 60 directions completed an iteration every 0.75 min using the vector nodal flux routine. The flux calculation used >90% of this total. The weighted difference option performs much faster, of course.

8. Machine Requirements: The load length on Cray is ~90 000 words before space for user data is allocated. The large problem cited above can be run in ~ 800 000 words of memory, including buffers, program, and data space. External data storage for nine scratch files, e.g., cross sections, must be supplied on sequential-access files.

9. Material Available: Available material includes a user's manual; source file in update form, including comment files and documentation updates; job control language for unloading the source and installing the code; and sample problem solutions and tutorial description. All questions regarding the code or its availability are to be directed to:

   Radiation Shielding Information Center (RSIC)
   P.O. Box 2008
   Oak Ridge, Tennessee 37831-6362

10. Authors: The originators of TORT are W. A. Rhoades and R. L. Childs. Since TORT is directly based on the DOT codes, major contributors to that work should also be recognized: W. W. Engle, Jr., M. L. Gritzner, F. R. Mynatt, R. J. Rodgers, D. B. Simpson, and E. T. Tomlinson.

11. References:

[1] W. A. RHOADES and R. L. CHILDS, "The DORT Two-Dimensional Discrete Ordinates Transport Code," *Nucl. Sci. Eng.*, **99**, *1*, 88 (1988).

[2] W. A. RHOADES and R. L. CHILDS, "The TORT Three-Dimensional Discrete Ordinates Neutron/Photon Transport Code," ORNL-6268, Oak Ridge National Laboratory (Nov. 1987).

[3] W. A. RHOADES and R. L. CHILDS, "An Updated Version of the DOT 4 One- and Two-Dimensional Neutron/Photon Transport Code," ORNL-5851, Oak Ridge National Laboratory (July 1982).