Computer Code Abstracts

SPAN-4

1. Program Name (and title): SPAN-4, A Point Kernel Computer Program for Shielding

2. Computer and Language(s): CDC-6600, FORTRAN IV. Some environmental routines are used.[1]

3. Problem Solved: SPAN-4 calculates the fast-neutron dose rate, thermal-neutron flux, gamma-ray flux, dose rate, and energy-absorption rate in rectangular, cylindrical, and spherical geometries by integrating appropriate exponential kernels over a source distribution. The shield configuration is flexible: a first-level shield mesh, using any one of the three geometries, is specified. Regions of this same geometry or of other geometries, having their own (finer) meshes, may then be embedded between the first-level mesh lines, defining second-level shield meshes. This process is telescopic: third-level shield meshes may be embedded between second-level mesh lines in turn. All meshes may have variable spacing. Sources and detectors may be located arbitrarily with respect to any shield mesh. The source is defined by the function:

$$S = S_0 + S_1(a) * S_2(b) * S_3(c) + S_4(a,b) * S_3(c) + S_5(a,c) * S_2(b) + S_6(b,c) * S_1(a) + S_7(a,b,c) \quad ,$$

where a, b, and c represent coordinates. If any factor is missing, the corresponding terms are zero. Cross sections, buildup factors, standard compositions, energy structures, dose-conversion factors, and infinite line source kernels are contained in a library of approximately 10 000 items.

4. Approximations and Methods of Solution: All kernels used assume exponential attenuation. By rigorously analytic ray tracing, the straight-line distances between points in the source and dose points are found to be used in calculating the attenuation. Integrals are evaluated by Gauss or Lobatto quadrature. Accuracy is dependent on the accuracy of the library data and on the orders of quadrature used. In thermal-neutron flux calculations, the dose points must be located within or beyond hydrogenous regions, since removal cross sections are used.

5. Restrictions on the Complexity of the Problem: Dynamic storage is used for large blocks of data. However, the following limits are to be observed:

Maximum Pieces of Data Allowed

75	Shield units
75	Composition matrices
450	Different dynamically-stored arrays. The number of such arrays is 3* (number of shield units) + number of composition matrices + 10 * (number of sources + number of detectors) + number of field point lists + 1.
400	Different composition numbers
200	Mesh lines in any direction in any shield unit
33	Embedded geometries in the path of one ray
9	Sources and detectors
9	Field-point lists
9	Combined field-point lists
33	Integrations in one case
15	Decay-times lists
2000	Items in any one data group, except source strengths
2000	Product of the orders of the three quadrature numbers

Additions to the Library are allowed with these restrictions:

Maximum Additions Allowed

1016	Pieces of cross-section data or 32 new elements
1060	Pieces of composition data or 279 compositions
190	Pieces of energy-structure data or 19 energy structures
350	Pieces of buildup-factor data or 3 new buildup-factor materials

6. Typical Running Time: A conservative estimate of the computer time required for a run is given by

$$T(\text{sec}) = 25 + [0.00042 * N_G * N_I + 0.1 * (E-1) + 0.02\ N_c] * N_F \quad ,$$

where N_G is the product of the orders of the quadratures used in integrating over the source volume, N_I is the number of mesh lines crossed in ray tracing from a typical Gauss point to a typical field point, N_F is the number of field points in the run, E is the number of energy levels in the energy structure, and N_c is the number of different compositions.

7. Unusual Features of the Program: The extreme flexibility of the shield geometry options allows nearly exact representations of very complex reactors, shields, and reactor compartment situations. Complex reactors and other devices are represented by embedding portions of one geometry in another as follows.

A basic geometry (rectangular, cylindrical, or spherical) is defined for the shield configuration, and is used to specify the meshlines in the first-level shield unit. The first-level shield usually encompasses

all of the shields and components which are to be represented in the problem. After the first-level shield unit has been specified, new shield units may be defined (using a different geometry than the first-level shield if desired), and portions of these new shield units may be embedded between the meshlines of the first-level shield unit. These new shield units become second-level shields.

The embedding process is telescopic, such that third-level shield units may be defined and embedded in the second-level shield units, and each new level of shield geometry is embedded in the next lower level. The meshlines which are specified in each shield may have variable spacing.

Each cell formed by the meshlines of each shield unit may then be specified to contain either a composition (that is, a void, element, compound, or mixture) or a cell-shaped portion of a higher-order shield unit. The highest-order shield unit(s) in a problem will therefore have a composition in every cell.

Multiple sources and detectors are allowed, thus simplifying some types of studies. Nine source and detector geometries may be defined in a SPAN-4 problem, and again the geometries may be rectangular, cylindrical, or spherical. Sources and detectors may be located arbitrarily with respect to any shield unit, and the regions of integration (or detectors) over which the flux, dose rate, or energy-absorption rate are calculated may be parts of rectangular solids, parts of cylinders, or parts of spheres.

The energy output or particle emission of each source is described by a general three-dimensional distribution function. The generality of the function requires that the integral be estimated by means of Gauss or Lobatto quadratures.

8. Status: In production.

9. Machine Requirements: 64K central memory and one system disk. Microfilm is required if the plot options are to be used.

10. Operating System: SPAN-4 is designed to operate under the SCOPE 3.1 operating system described in Ref. 2.

11. Other Programming Information: The program is structured into distinct parts which are loaded into central memory as required. These parts are referred to as OVERLAYs as described in the CDC SCOPE 3.1 basic reference manual (Ref. 2). Essentially an overlay represents the amount of program text in central memory. With overlay loading, the main overlay is always in memory; and, in addition, a particular primary and one secondary overlay may also be in central memory. The SPAN-4 program contains a main overlay, 6 primary overlays, and 7 secondary overlays. Overall program control is vested in the main overlay and the primary and secondary overlays are loaded and executed through the use of the NEXT subroutine described in Ref. 1.

12. Availability: The program and copies of Ref. 3 may be obtained from

    Argonne Code Center
    Attn: Mrs. Margaret Butler
    Argonne National Laboratory
    9700 South Cass Avenue
    Argonne, Illinois 60440

    Reference 3 may also be obtained from

    Clearinghouse for Federal Scientific and Technical Information
    National Bureau of Standards
    U.S. Dept. of Commerce
    Springfield, Virginia 22151
    ($3.00 - printed copy; $0.65 - microfiche)

13. *References:*

[1]C. J. PFEIFER, "CDC-6600 FORTRAN Programming" in "Bettis Environmental Report," WAPD-TM-668 (January 1967).

[2]"Control Data 6400/6500/6600 Computer Systems SCOPE 3.1 Reference Manual," Publ. No. 60189400A (February 1968).

[3]O. J. WALLACE, "SPAN-4, A Point Kernel Computer Program for Shielding," WAPD-TM-809 (December 1969).

O. J. Wallace

Westinghouse Electric Corporation
Bettis Atomic Power Laboratory
West Mifflin, Pennsylvania

Received February 20, 1970

SPECTER

1. Name of Code: SPECTER

2. Computer for Which Program is Designed: IBM-7094 Programming Language: FORTRAN IV

3. Nature of Physical Problem Solved: SPECTER calculates the energy spectra of particles emerging from reactions of the types (y,n), (y,p), (y,α), $(y,2n)$, (y,np), (y,pn), and $(y,2p)$, where the incident particle y may be either n, p, or α. In addition, the code calculates the cross section for each of these reactions.

4. Method of Solution: SPECTER is an outgrowth of the previously published code EDISN.[1] Whereas EDISN is capable of handling only (n,n') and $(n,2n)$ reactions, SPECTER can treat all the reactions specified above. SPECTER makes use of the Weisskopf-Ewing[2] formula to calculate the energy distributions and reaction cross sections. The cross sections for compound-nucleus formation that are needed in evaluating this formula are calculated by the continuum theory of nuclear reactions.[3] Data required as input are the bombarding energy, the masses of all nuclear species taking part in the reaction, and certain parameters that specify the level density of participating nuclei.

5. Restrictions on the Complexity of the Problem: Ideally the code is suited to reactions that proceed through the formation of a compound nucleus (CN). In cases, however, in which the direct interaction mechanism (DI) is also operative, the code is useful in estimating the relative contribution from direct effects, provided the entire cross section for the process, σ_{CN+DI}, be known, e.g., from experiment. In addition, it is required that the level structures of all residual nuclei be adequately described by continuous level densities.

6. Related and Auxiliary Programs: For completeness, a subroutine for calculating the nuclear level density has been included. This subroutine makes use of