LA-UR-11- 02462

Approved for public release;
distribution is unlimited.

Title:	A Brief User's Guide for PARTISN
Author(s):	Jeffrey A. Favorite, XCP-7
Intended for:	Electronic distribution and archive

Los Alamos
NATIONAL LABORATORY
EST. 1943

Form 836 (7/06)

Los Alamos
NATIONAL LABORATORY

research note

X-Computational Physics Division
Transport Applications Group

Group XCP-7, MS F699
Los Alamos, New Mexico 87545
505/606-0495
Fax: 505/667-3726

To/MS: Distribution
From/MS: Jeffrey A. Favorite / XCP-7, MS F663
Phone/Email: 7-7941 / fave@lanl.gov
Symbol: XCP-7–RN(U)11–02 (LA–UR–11–????)
Date: April 21, 2011 (Rev. 1)
September 29, 2008 (Rev. 0; LA–UR–08–6339)

SUBJECT: A Brief User's Guide for PARTISN (U)

Abstract

This document is intended as a very basic primer on the use of the PARTISN multigroup discrete-ordinates transport code for neutron criticality and time-independent neutron leakage problems. One-dimensional (*r* only) spherical and two-dimensional (*r*-*z*) cylindrical geometries and only the most basic options are discussed. It is assumed that the Nuclear Data Interface (NDI) is the source of nuclear data, that volume sources are isotropic, and that external boundaries are vacuums. The location and meaning of useful information in the PARTISN output is discussed. A sample input deck for a cylindrical problem that stacks a leakage calculation and a k_{eff} calculation is listed. The method for inputting cross sections by hand (e.g., for analytic benchmark comparisons) is presented.

I. Introduction

This document is intended as a very basic primer on the use of the PARTISN multigroup discrete-ordinates transport code[1] for neutron criticality and time-independent neutron leakage problems. One-dimensional (*r* only) spherical and two-dimensional (*r*-*z*) cylindrical geometries are discussed.

The purpose of this paper is to give a typical XCP- or XTD-Division employee with a background in particle transport the ability to create PARTISN input [geometry, source (fixed and criticality), materials] and understand PARTISN output. Some comparisons with the MCNP Monte Carlo code[2] are given because most people have greater familiarity with MCNP, but this familiarity is not required to understand this paper. Only the most basic functions and options are presented. Thus, many keywords and some options for certain keywords are omitted if they are not needed in a basic calculation.

The PARTISN version that this paper refers to is the most recent one available at Los Alamos, 6.52. The basics have not changed much over the years and most of this paper will apply to versions that are much older, provided that they are compatible with the NDI.

Setting up a PARTISN input file is discussed in Sec. II. The output file is discussed in Sec. III. An example input file for a cylinder is given in Sec. IV. Appendix A discusses the input of cross sections by hand for testing, and Appendix B discusses the input of boundary sources.

II. Input File

A PARTISN input file consists of a title control line, a title, five mandatory "blocks," and an optional sixth block. Each block is terminated with a single `t`. Many weird errors are caused by forgetting the block-terminating `t`.

Each block consists of cards or keywords that specify input as either integer flags, character flags, numeric arrays, or mixed character and numeric arrays. The word *keyword* rather than *card* is recommended for PARTISN input because, unlike MCNP cards, multiple PARTISN keywords may appear on the same input line. This paper adopts the notational

convention of Chap. 2 of the manual. If a keyword is given without brackets, the input is a single entry. If a keyword is given with a variable such as [IM], the entry is a single list or string of length IM. Spaces delimit items in a list or string. If a keyword is given with two numbers such as [IM;JM], the entry consists of JM strings of length IM; in the input file, strings are delimited by a semi-colon. If the second entry is a product, the entry order is important; for an example, see the `sourcf` array in Sec. II.F.2.

Special input operators are used to simplify long string input. See Table I.

Table I. The Most Useful Input Operators.

Operator[a]	Action
nR d	Repeat the data item d, n times
nY m	String repeat: Repeat the previous m strings, n times

[a] There is no space between the operator character and the number n.

Comments are entered after a forward slash (/), which can be at the beginning of a line or anywhere on a line. Input after column 80 is ignored.

PARTISN assumes no units; it is up to the user to enter values with units consistent with each other and with the cross-section library. When using the Nuclear Data Interface (NDI), which is assumed (see Sec. II.D), use the standard transport units: centimeters for length, grams for mass, atoms/barn·cm for atom density, shakes for time when dealing with particle velocity or time-dependent sources but seconds when dealing with steady-state source strengths, and MeV for energy in the rare cases when energies need to be input.

II.A. Title control and title

The first line of a PARTISN input deck is a fixed-field set of five integers. Only the first two are meaningful for this paper; the others don't need to be entered. The first integer tells how many title lines follow this title control line, and the second integer tells whether to suppress the output to the terminal (0/1 = no/yes). These integers must be entered as "I6", that is, six characters wide; in other words, enter integers ending in columns 6 and 12. The specified number of title lines then follow.

II.B. Block 1: Controls and Memory

PARTISN differs from MCNP and many other codes in that it needs to know how many items of certain types to expect. Thus, although the materials will be specified in block 4 and the geometry discretization in block 2, the total number of materials and geometric zones need to be input in block 1. Block 1 keywords are listed in Table II; all are required except `iquad`.

Table II. Block 1 Keywords.

Keyword	Meaning
`igeom`	Problem geometry; either SPHERE or R-Z (or SLAB; see Appendix A)
`isn`	S_N order (i.e., input N)
`ngroup`	Number of energy groups
`mt`	Number of materials
`nzone`	Number of zones (set equal to number of materials)
`niso`	Number of isotopes on the cross-section library; set to 0 if using NDI (assumed)
`im`	Total number of coarse mesh intervals in the radial direction
`it`	Total number of fine mesh intervals in the radial direction
`jm`[a]	Total number of coarse mesh intervals in the axial direction
`jt`[a]	Total number of fine mesh intervals in the axial direction
`iquad`[a]	For cylinders, enter 6 to use square Chebychev-Legendre S_N constants[3]

[a] Not input for `igeom=sphere`.

PARTISN has many options for assigning geometric coarse meshes to edit zones and materials to coarse meshes and edit zones. For simplicity, let the number of zones be equal to the number of materials. In this way, edit zones are synonymous with coarse meshes. PARTISN coarse meshes then correspond to MCNP cells. The rest of this paper adopts this convention.

In the past, `maxscm` and `maxlcm` were required input to manage memory. These keywords are generally no longer required (as of version 5.70), but if you get an error message stating that the LCM or SCM is too small, use the appropriate keyword with an appropriate value; in the error message, the code will state a necessary minimum.

What S_N order to use? Use the minimum ISN that gives the right answer; or, alternatively, use the maximum ISN that your computational time can tolerate. I generally use S_{32} for spheres and S_{16} for cylinders. In some applications, S_8 for spheres may be accurate enough.

II.C. Block 2: Geometry

The spatial dimensions and discretization are given in block 2. Required keywords are listed in Table III. `xmesh` entries specify spherical or cylindrical surfaces, according to the value of `igeom`; `ymesh` entries specify planar axial surfaces for cylinders. The order of radial specifications is inside to out and the order of axial specifications is bottom to top.

Table III. Block 2 Keywords.

Keyword	Meaning
`xmesh` [IM+1]	Radial coarse mesh edges (spherical or cylindrical radii); start with 0.
`xints` [IM]	Number of fine meshes in each radial coarse mesh
`ymesh`[a] [JM+1]	Axial coarse mesh edges
`yints`[a] [JM]	Number of fine meshes in each axial coarse mesh
`zones` [IM;JM]	Zone number (material number, in the scheme of this paper) assigned to each coarse mesh; 0 for void

[a] Not input for `igeom=sphere`.

What spatial discretization (meshing) to use? As with the angular meshing (`isn` of Sec. II.B), use the minimum number of meshes that gives the right answer; or, alternatively, use the maximum number of meshes that your computational time can tolerate. The spatial meshing is not independent of the angular meshing – if more angles are used, more meshes are needed. One indication that the angular and spatial discretizations are fine enough is if there are no negative fluxes; see the `nofxup` keyword in Sec. II.F.1.

For two-dimensional problems, there is no easy way to refine or coarsen the discretization in only one region of the problem. Thus, if one coarse mesh in a cylinder requires a fine discretization, all the coarse meshes directly above and below (those with the same radial index) and all those to the left and right (those with the same axial index) must have the same fine discretization. (Block AMR can be used for independent coarse mesh refinement,[3] but that capability is beyond the scope of this guide.)

In the simplified system suggested in this paper, the `zones` keyword is used to assign materials to coarse meshes. Materials are indexed according to their position in the `matls` list of block 4 (Sec. II.E); no `zones` entry can be greater than MT. The first IM numbers are the material numbers assigned to the coarse meshes in the radial direction (inside to out) for the axial coarse mesh at the bottom of the geometry. This string is terminated with a semi-colon. The next set of IM numbers are the material numbers assigned to the coarse meshes in the radial direction (inside to out) for the second axial coarse mesh from the bottom, and so on. The final set of IM numbers are the material numbers assigned to the coarse meshes in the radial direction (inside to out) for the axial coarse mesh at the top of the geometry. This final string is also terminated with a semi-colon. See the example in Sec. IV.

Frequently, people wonder whether there is an integrated graphics package like MCNP's that allows users to visualize or even create PARTISN geometries. The answer is no. However, there is a text "material map" in the output file that very usefully shows the geometry, though not to scale. See Sec. III and the example in Sec. IV.

II.D. Block 3: Nuclear Data

In the old days, users had to be intimate with cross-section libraries and had to tell PARTISN many details about the structure of the library to use so that the code could read it properly. This is still the case for many non-LANL users. Fortunately, using nuclear data has become dramatically easier at Los Alamos with the introduction of the Nuclear Data Interface (NDI). This paper assumes that NDI is the source of all nuclear data for PARTISN. Table IV on the next page shows the keywords needed in block 3 to use NDI. The actual data to use is specified in block 4 with the materials; see Sec. II.E.

Table IV. Block 3 Keywords for NDI.

Keyword	Meaning
`lib`	Name of cross-section data file; use NDILIB
`fissneut`	Type of neutron fission data to be used; 0/1 = prompt (default)/total. Similar to MCNP's `totnu`.
`libname`[a]	Needed for coupled neutron-photon problems (use[b] ACTI or MENDF6)
`glibname`[a]	Needed for coupled neutron-photon problems (use[b] ACTIG or MENDF6G)
`lng`[a]	Number of the last neutron group in a coupled neutron-photon library[b]

[a] There are also options for photon-only problems but these are not addressed here.
[b] For ACTI, use `ngroup=250` and `lng=130`; for MENDF6, use `ngroup=42` and `lng=30`.

In the past, fission spectrum (chi vector) information had to be given to the code in the input file, and there are still many options for doing so. When using NDI, this is not necessary. In fact, NDI is also better because much of the data is given as a chi matrix $\chi^{g'\to g}$ rather than just a vector χ^{g}.

One difference between PARTISN and MCNP to note: MCNP's default is to use total $\overline{\nu}$ for k_{eff} problems but prompt for fixed-source problems, and PARTISN's default is to use prompt $\overline{\nu}$ for both.

It is frequently useful to input a small cross section table for analytic benchmark comparisons or testing. The form for that is given in Appendix A.

II.E. Block 4: Materials

PARTISN offers many options for setting up materials, mixing them, and assigning them to zones. In this paper the simplest approach is used: define each material separately as it exists in the problem. For example, if there are 10 different mixtures of uranium and water, each representing a different solution concentration, then each of the 10 will be defined. This is essentially the MCNP material definition scheme. Materials are defined using the keywords shown in Table V.

Table V. Block 4 Keywords.

Keyword	Meaning
`matspec` [≤MT]	For each material, tells whether constituents are given in weight fraction (WTFRAC) (preferred), atom fraction (ATFRAC), or atom density (ATDENS). If there are fewer than MT entries, the last one fills the array.
`matls` [–;MT]	Material definitions; see text.
`assign` [3;MT]	Assignment of materials to zones; see text.

The form of the `matls` keyword is

$$\texttt{matls} = \begin{array}{llllllll} m^1 & Z_1^1 & \rho_1^1 & Z_2^1 & \rho_2^1 & \cdots & Z_{n^1}^1 & \rho_{n^1}^1 \quad ; \\ m^2 & Z_1^2 & \rho_1^2 & Z_2^2 & \rho_2^2 & \cdots & Z_{n^2}^2 & \rho_{n^2}^2 \quad ; \\ \vdots & & & & & & & \\ m^{MT} & Z_1^{MT} & \rho_1^{MT} & Z_2^{MT} & \rho_2^{MT} & \cdots & Z_{n^{MT}}^{MT} & \rho_{n^{MT}}^{MT} \quad ; \end{array}$$

In this notation, the superscript is the material index and the subscript is the index of the isotope in the material; n^i is the number of constituents (isotopes) of material i and m^i is a character name for material i. Z_j^i is the isotope identifier, including the cross-section table (more below), and ρ_j^i is the atom density, atom fraction, or weight fraction of isotope j in material i, according to the value of `matspec` for the material. Note the semi-colon delimiter for each string. Compare with the MCNP material definition in the same notation:

$$\begin{array}{llllllll} \texttt{m1} & Z_1^1 & \rho_1^1 & Z_2^1 & \rho_2^1 & \cdots & Z_{n^1}^1 & \rho_{n^1}^1 \\ \texttt{m2} & Z_1^2 & \rho_1^2 & Z_2^2 & \rho_2^2 & \cdots & Z_{n^2}^2 & \rho_{n^2}^2 \\ \vdots & & & & & & & \\ \texttt{m\{MT\}} & Z_1^{MT} & \rho_1^{MT} & Z_2^{MT} & \rho_2^{MT} & \cdots & Z_{n^{MT}}^{MT} & \rho_{n^{MT}}^{MT} \end{array}$$

Unlike in PARTISN, in MCNP the ρ_j^i are always atom fractions or weight fractions, the code being told which with a plus or minus sign, respectively.

The form of the `assign` keyword is

$$\texttt{assign} = \begin{array}{llll} z^1 & m^1 & \rho^1 & ; \\ z^2 & m^2 & \rho^2 & ; \\ \vdots & & & \\ z^{\mathrm{MT}} & m^{\mathrm{MT}} & \rho^{\mathrm{MT}} & ; \end{array}$$

In this notation, z^k is a character name for zone k, m^k is the character name (from the `matls` keyword) of the material assigned to zone k, and ρ^k is the mass density of the material in zone k if `matspec=atfrac` or `matspec=wtfrac` (in contrast to what was written in Rev. 0 of this report). If `matspec=atdens`, then the ρ_j^i on the `matls` keyword are atom densities and the ρ^k need not be entered on the `assign` keyword, or they may be entered as 1.0. It is the index k that is entered on the `zones` keyword to specify the material in block 2; therefore, it makes sense to use `zone`*k* (or `zone0`*k*, etc.) for z^k. A more descriptive material name can be used for m^i, but nothing is wrong with `m`*i* (or `m0`*i*, etc.). In MCNP, material assignments are done on the cell cards. (In Chap. 2 of the PARTISN manual, the ρ^k are "volume fractions," but that is a convention that is not used here. See Chap. 7 of the PARTISN manual, where ρ^k is the mass density.)

The cross-section table for each isotope is specified in its zaid Z_j^i, as in MCNP. The zaid format for NDI is `"descriptor.nnntt"`, which must be enclosed in quotation marks in the PARTISN input. The descriptor is the usual isotope indicator $(S \times 1{,}000{,}000) + (Z \times 1{,}000) + A$, where S is the isomeric state (0 is the ground state, 1 is the first excited state, etc.), Z is the atomic number, and A is the atomic mass number. Leading zeroes are not included. `nnn` is an integer that identifies the library to be used and `tt` is a character identifying the type of table.

Generally, MENDF5 is specified using `501nm`; MENDF6, `601nm`; and ACTI, `620nm`. Available cross sections may be found by looking in the gendir file associated with the NDI (search for the nuclides you want to use) or by using the Data Query Tool (DQT), <https://xweb.lanl.gov/projects/data/nuclear/dqt/>. This excellent though poorly documented tool allows you to find all of the available data for any nuclide as well as sort by energy group structure, etc., up to NDI version 2.0.9 (the most recent NDI version is 2.0.15). You can use isotopes from different evaluations or datasets as long as the energy group structures are the same.

II.F. Block 5: Solver Input

II.F.1. General controls

General controls, including switches that tell the type of calculation to do as well as iteration and acceleration options, are listed in Table VI. Parameters with default values do not need to be entered, except that the scattering order `isct` has a

Table VI. Block 5 General Keywords.

Keyword	Meaning
`ievt`	Calculation type; 0/1/2 = Inhomogeneous source/k_{eff} eigenvalue
`norm`	Source normalization; integral of source rate over space, angle, and energy = NORM for `ievt=0` and k_{eff} × NORM for `ievt=1`. `norm=0` or no entry means no normalization, which is customary for `ievt=0`.
`ith`	0/1 = forward (default)/adjoint
`isct`	Legendre order of scattering (default is 0, isotropic; *very dangerous!*)
`epsi`	Convergence precision (default is 1.E–4)
`oitm`	Maximum number of outer iterations (default is 20)
`iitl`	Maximum number of inner iterations per group at first (code computes default)
`iitm`	Maximum number of inner iterations allowed when near fission source convergence (code computes default)
`srcacc`	Source acceleration; dsa/tsa/no = Diffusion sythetic acceleration (default)/ Transport sythetic acceleration/no acceleration
`trcor`	Apply transport correction to estimate effect of higher-order scattering than `isct`; diag/no = Diagonal correction/no correction (default)
`nofxup`	Turn off negative flux fix-up; 0/1 = no (default)/yes
`ptconv`	Convergence of pointwise fluxes; 0/1/2 = none/transport fluxes (default)/transport fluxes + fission source. Use 0 to converge eigenvalue tightly, but not fluxes.[a]

[a] This was achieved using `kcalc=I` until version 5.93.

default value of 0, so `isct=`*L* must be used, where *L* is maybe 3 or 4 but no larger than the maximum scattering order in the cross-section set, which is generally 4 for Los Alamos tables (the DQT can be used to find the maximum scattering order).

If problems have trouble converging, things to try include increasing `oitm`, `iitm`, and/or `iitl`, and/or turning off source acceleration. Sometimes it is interesting to test a problem without the negative flux fix-up. If the results change a lot, there is too much fix-up, and you should refine the spatial mesh.

II.F.2. Source

For k_{eff} calculations (`ievt=1`), no sources need be entered. In this paper it is assumed that inhomogeneous volume sources are isotropic, so the number of moments to input (NMQ in the manual) is just 1, the zeroth moment. The `sourcf` keyword simplifies to the form shown in Table VII. In PARTISN, boundary sources are entered as incoming angular fluxes; the surface source keywords are also shown in Table VII. Internal surface sources are not allowed.

Table VII. Block 5 Source Keywords (Use One Option Only).

Keyword	Meaning
Option 1:	
`sourcf` [IT;JT*NGROUP]	Volumetric source in each fine mesh and each energy group.
Option 2:	
`sirite` [NGROUP;JT]	Isotropic incoming angular fluxes on outer radial surface.
`sibott` [NGROUP;IT][a]	Isotropic incoming angular fluxes on top axial surface.
`sitop` [NGROUP;IT][a]	Isotropic incoming angular fluxes on bottom axial surface.
Option 3:	
`sarite` [MM[b];NGROUP*JT]	Incoming angular fluxes on outer radial surface.
`sabott` [MM[b];NGROUP*IT][a]	Incoming angular fluxes on top axial surface.
`satop` [MM[b];NGROUP*IT][a]	Incoming angular fluxes on bottom axial surface.

[a] Not input for `igeom=sphere`.
[b] Here MM is the number of incoming angles (note different definition from the manual). MM = ISN/2 for one-dimensional spheres or slabs; MM = $(ISN)^2/2$ for two-dimensional cylinders with `iquad=6`.

Volume sources are the average source density in each fine mesh given in units of particles/cm^3·s. Boundary sources are the average incoming flux in each fine mesh surface on the boundary; they have units of particles/cm^2·s. These units assume steady-state problems. PARTISN uses the standard multigroup ordering of highest to lowest energy (more precisely, PARTISN uses whatever order the cross-section tables are in, which, for the NDI, is highest to lowest). For adjoint calculations, sources are entered in their normal (forward) order and the code reverses them internally.

It may seem complicated to enter a volumetric source in a two-dimensional problem using `sourcf`, but it is not, especially if a script or code actually does the job. Remember that a source density must be specified in each fine mesh in each energy group in IT-length strings. The first string contains the sources for the bottom axial fine meshes in the first (highest energy) group, the second string contains the sources for the second set of axial fine meshes in the first group, and so on until the JT'th string contains the sources for the top axial fine meshes in the first energy group. There are now IT × JT entries for the first energy group. The geometric sequence is repeated for the second and subsequent energy groups. Each IT-length string is terminated with a semi-colon, but it is a good idea to terminate regular intervals with a comment of some kind.

The `R` and `Y` operators from Table I are useful for this purpose. In the convention of this paper, each coarse mesh has one material in it. For many physical radioactive systems, a source with a given spectrum is associated with a material and distributed homogeneously within it. For such problems, all the fine meshes within a coarse mesh will have the same source. Let I_i and J_j represent the number of radial and axial fine meshes in radial and axial coarse meshes i and j, respectively. Let $S^g_{i,j}$ be the volumetric source density in energy group g for radial and axial coarse meshes i and j, respectively. Then the volumetric source in the entire problem is given by

```
sourcf =
/ group 1 source
```
I_1r $S_{1,1}^{g=1}$ I_2r $S_{2,1}^{g=1}$ ... I_{IM}r $S_{IM,1}^{g=1}$; J_1–1y 1 ; / bottom coarse mesh
I_1r $S_{1,2}^{g=1}$ I_2r $S_{2,2}^{g=1}$ ... I_{IM}r $S_{IM,2}^{g=1}$; J_2–1y 1 ; / second coarse mesh
⋮
I_1r $S_{1,JM}^{g=1}$ I_2r $S_{2,JM}^{g=1}$ ... I_{IM}r $S_{IM,JM}^{g=1}$; J_{JM}–1y 1 ; / top coarse mesh
⋮
```
/ group G source
```
I_1r $S_{1,1}^{g=G}$ I_2r $S_{2,1}^{g=G}$ ... I_{IM}r $S_{IM,1}^{g=G}$; J_1–1y 1 ; / bottom coarse mesh
I_1r $S_{1,2}^{g=G}$ I_2r $S_{2,2}^{g=G}$ ... I_{IM}r $S_{IM,2}^{g=G}$; J_2–1y 1 ; / second coarse mesh
⋮
I_1r $S_{1,JM}^{g=G}$ I_2r $S_{2,JM}^{g=G}$ ... I_{IM}r $S_{IM,JM}^{g=G}$; J_{JM}–1y 1 ; / top coarse mesh
```

Note the use of comments to make sense of it all. There are $I_1 + \cdots + I_{IM} = IT$ entries in each string, $1 + J_j - 1 = J_j$ strings in each axial coarse mesh grouping, and JM axial coarse mesh groupings. Be sure not to extend lines beyond column 80. Similar logic applies to the specification of boundary sources. See the specific example (for `sourcf`) in Sec. IV.

A caution on boundary sources is in order. A PARTISN isotropic boundary source does not mean the same thing as an MCNP isotropic boundary source. In MCNP boundary sources, *isotropic* refers to the incoming particle density, but in PARTISN boundary sources, *isotropic* refers to the incoming particle flux. Thus, with the `sirite`, `sibott`, or `sitop` keywords, the fluxes for all incoming ordinates are set equal to each other. This is equivalent (within the accuracy of the discrete ordinates approximation) to a cosine distribution of the boundary source in MCNP. To set up a boundary source in PARTISN that is isotropic in the density rather than the flux (i.e., a source that is isotropic in the MCNP sense), use the `sarite`, `sabott`, or `satop` keywords and set the incoming angular flux for ordinate $m$ to $c/\omega_m$, where $\omega_m$ is the appropriate cosine and $c$ is a normalization constant. For more on this issue, see Appendix B. Always check, in the system balance table (see Sec. III), that the total source is correct.

### *II.F.3. Boundary conditions*

For the purpose of this note, all problems will have vacuum boundaries on external surfaces and reflecting boundaries on internal symmetry surfaces (i.e., the center of a sphere and the axis of a cylinder). These are the code defaults, so no input is required. Note that a vacuum boundary can still have an incoming source on it.

### *II.F.4. Output controls*

There is a lot of useful information automatically printed to the output file (see Sec. III), but there are also options for output. Some of the most useful ones are listed in Table VIII. To print the fluxes in each coarse mesh or in specific fine meshes, use block 6 (Sec. II.G).

Table VIII. Block 5 Output Controls; All Default to 0.

| Keyword | Meaning |
|---|---|
| `balp` | Print coarse mesh balances in output file; 0/1 = no/yes |
| `xsectp` | Print macroscopic cross sections in output file; 0/1/2 = no/principal/all |
| `sourcp` | Print source in output file; 0/1/2/3 = no/as input/normalized/both |
| `fluxp` | Print flux moments in output file; 0/1/2 = no/isotropic/all |
| `angp`[a] | Print angular fluxes in output file; 0/1 = no/yes. This may lead to a huge output file. If the code thinks there will be too much output, it will refuse to write it. |
| `raflux` | Write binary angular flux file (raflxm or aaflxm); 0/1 = no/yes |
| `rmflux` | Write binary flux moments file (rmflux or amflux); 0/1 = no/yes |

[a] Not used for `igeom=r-z`.

The balance table is particularly useful if partial currents are desired; see Sec. III.
```

The `fluxp` keyword prints the average scalar flux (and higher moments, if requested) in each fine mesh and energy group. The `angp` keyword prints the average angular flux on each fine mesh surface and in each energy group, but only for spheres. These quantities have units of flux. The `fluxp` keyword may be useful for constructing edits in post-processing, but for serious manipulations of angular fluxes or flux moments, `raflux=1` or `rmflux=1` is necessary.

II.G. Block 6: Edit Input

Block 6 is optional. It is frequently the case that the desired outputs (such as k_{eff} or partial currents) are obtained automatically in the output file (see Sec. III). Furthermore, when PARTISN is used as the transport code in a script-based system that automatically writes input files and reads output files, it is just as easy to have the script read the binary flux and cross-section files and construct its own edits.

Nevertheless, it is sometimes handy to have the code print fluxes by coarse mesh or at specific points, or to have the code compute and print reaction-rate edits. To do so, use the keywords of Table IX. Zone masses may also be printed; the explanations of the flags given in Table IX are simplifications based on the conventions in this paper.

Table IX. Block 6 Keywords.

Keyword	Meaning
Option 1[a]:	
`zned`	Set to 1 to do edits by coarse mesh (in the system of this paper).
Option 2[a]:	
`pted`	Set to 1 to do edits by fine mesh.
`points` [≤IT*JT]	Fine mesh indices[b] at which fluxes are desired (default is all points)
`byvolp`	0/1 = print volume-average fluxes (default)/multiply fluxes by fine-mesh volumes
Use with either option:	
`ajed`	0/1 = forward (default)/adjoint binary flux file from which to read fluxes
`igrped`	0/1/3 = Print energy group totals/Print by groups/Print both groups and totals
`massed`	Print zone masses; 0/1/2/3 = no/by material (default)/by coarse mesh/both
For fluxes[c]:	
`rsfe` [NGROUP]	Response function; use 1. for each group to get scalar fluxes
For reaction rates[c]:	
`resdnt`	Set to 1 to use resident coarse-mesh macroscopic cross sections.
`edxs`	Optional. List of numbers denoting edit positions of desired cross sections. 2, 3, and 4 are generally $\nu\Sigma_f$, Σ_t, and Σ_a, respectively. Default is all.

[a] Option 1 and option 2 can be used together.
[b] The index I of the fine mesh that is the i'th in the radial direction and the j'th in the axial direction is $I = i + \mathrm{IT} \times (j-1)$.
[c] Fluxes and reaction rates can be computed together.

Note that if edits of adjoint fluxes are desired (normally if `ith=1` in block 5), `ajed` must be set to 1 in block 6. This is because the modules, especially the edit module, are run somewhat independently. It is an easy detail to forget.

Further discussion of edit outputs appears in Sec. III.

III. Output

Normally, the results of interest from a PARTISN calculation include either 1) k_{eff} or 2) particle leakages, certain fluxes, and/or partial currents. In some cases fluxes and/or edits (average fluxes or reaction rates) may be desired in a k_{eff} calculation.

PARTISN does a very thorough job of writing user-input values as well as defaulted values to the output file. Another useful output is the S_N quadrature weights and cosines. These are listed under the heading "key start sn constants". The energy group structure appears with cross-section information under the heading "energy structure".

Information about the cross-section tables used in the problem appears under the heading "ndi table information". In PARTISN version 6.52, the `maxord` parameter that is printed is actually based on 1 being isotropic; this is inconsistent with the convention for the `isct` keyword (Table VI), the Data Query Tool (Sec. II.E), and the block 3 `maxord` keyword (see Appendix A), in which 0 is isotropic.

PARTISN also has a text-based graphical geometry display that is very helpful for checking material assignments and spatial discretizations. It appears under the heading "key start material map". The mass edit appears under the heading "mass edit".

Before looking for answers, it is important to ascertain that the problem is converged. Examine the output under "iteration monitor", or do a Unix grep or search the output file for the phrase "all convergence criteria met". If it doesn't appear, the problem is not converged and the printed results are suspect. There might be some helpful advice ("too many outers" or "too many inners", for example). In addition, beware of false convergence, which can sometimes be diagnosed through particle balances that are too high compared to the `epsi` parameter. Particle balances appear in many places, but one of the easiest to use is under the heading "integral summary information" and is titled "integral-particle bal-i".

In a k_{eff} problem (`ievt=1`), k_{eff} is listed under "integral summary information" with the title "summary integral-k-eff". A grep for "l-k-e" will find it.

The particle leakage by energy group and surface, as well as the totals, is found under the heading "system balance tables". The units of this table are particles, with the same time units as the source (i.e., generally s^{-1} for steady-state problems). This table is printed automatically. The leakage L^g is

$$L^g = \int_\Gamma dS \int_{\hat{\Omega}\cdot\hat{\mathbf{n}}>0} d\hat{\Omega}\, \hat{\Omega}\cdot\hat{\mathbf{n}}\, \psi^g(\vec{\mathbf{r}},\hat{\Omega}), \tag{1}$$

where $\psi^g(\vec{\mathbf{r}},\hat{\Omega})$ is the angular flux of particles in group g at spatial position $\vec{\mathbf{r}}$ traveling in direction $\hat{\Omega}$, Γ represents the exterior surface of the system, and $\hat{\mathbf{n}}$ is the outward unit normal vector on Γ at position $\vec{\mathbf{r}}$. The total leakage is the sum of the leakages over the energy groups. Of course, in PARTISN's evaluation of L^g, the discrete-ordinates angular quadrature is used. In the system balance tables, the leakage does not include any boundary source. To compare, in MCNP the total leakage is obtained from the problem summary table as the total weight escaped but can also be obtained as a function of energy using surface crossing (F1) tallies on exterior surfaces.

The coarse mesh balance tables, under the heading "key start coarse mesh balances", are printed if `balp=1` in block 5. These tables also have units of particles with the same time units as the source. The partial currents in the positive and negative directions on the left and bottom surfaces of each coarse mesh are listed. The partial currents $J_\pm^g$ across surface S are calculated using a generalization of Eq. (1),

$$J_\pm^g = \int_S dS \int_{\substack{\hat{\Omega}\cdot\hat{\mathbf{n}}>0 \\ \hat{\Omega}\cdot\hat{\mathbf{n}}<0}} d\hat{\Omega}\, \left|\hat{\Omega}\cdot\hat{\mathbf{n}}\right| \psi^g(\vec{\mathbf{r}},\hat{\Omega}). \tag{2}$$

The sphere's center or the cylinder's axis is considered the leftmost radial surface. In the coarse mesh balance tables, the partial currents also include boundary sources. There are IM+1 coarse meshes listed in the radial direction; the last contains the net radial leakage, which includes the boundary source (if any). For two-dimensional problems, there are JM+1 coarse meshes listed in the axial direction; the first contains the net leakage through the bottom and the last contains the net leakage through the top. Partial currents are useful for transmission calculations; the transmission through a region (which need not be a single coarse mesh) is defined as the ratio of the partial current exiting to that entering. Normally, coordinate directions are used in the definition; for example, we may speak of the axial or radial transmission. Partial currents are obtained from MCNP using surface-crossing (F1) tallies with inward- and outward-directed cosine bins.

The coarse mesh balance tables also give useful information about other particle activity in each coarse mesh.

Flux and reaction rate edits requested in block 6 appear under the heading "key start edit output". If `rsfe=1` . for each group (Table IX), then zone edits (`zned=1`) produce the volume-integrated scalar flux in each zone or coarse mesh n as

$$\phi_n^g = \int_{V_n} dV \int_{4\pi} d\hat{\Omega}\, \psi^g(\vec{r},\hat{\Omega}), \tag{3}$$

where V_n is the volume of the mesh. This corresponds to a track-length (F4) cell tally in MCNP, except that MCNP normally gives the cell-average flux $\bar{\phi}_n^g \equiv \phi_n^g / V_n$; PARTISN results (by zone) must be divided by coarse mesh volumes to get averages. Including `resdnt=1`, volume-integrated reaction rates in each zone or coarse mesh are calculated as

$$R_{x,n}^g = \int_{V_n} dV \int_{4\pi} d\hat{\Omega}\, \Sigma_x^g(\vec{r})\psi^g(\vec{r},\hat{\Omega}) = \Sigma_{x,n}^g \phi_n^g, \tag{4}$$

where x indicates a specific reaction that is specified by integer index using the `edxs` keyword (lack of an `edxs` keyword produces reaction rates for all edit reactions available). The second equality is obtained using the conventions of this paper. The $R_{x,n}^g$ correspond to energy-dependent reaction rates computed in MCNP using track-length tallies with reaction multipliers (FM cards), which again are normally volume averages.

In k_{eff} problems, the total $\nu\Sigma_f$ reaction rate (normally `edxs=2`) should be equal to `norm` × k_{eff}. Thus, use `norm=1.` to normalize the PARTISN fluxes in the same way that MCNP fluxes are normalized.

Whether point edits (`pted=1`) produce volume-average or volume-integrated scalar fluxes in fine meshes is controlled by the `byvolp` keyword. If `byvolp=0`, the point results are averages and correspond to an MCNP point-detector (F5) tally if the fine meshes are small enough and certain other conditions are satisfied. If scalar fluxes are printed to the output file using `fluxp=1` in block 5, they are the average flux in each fine mesh (the same as would be obtained from point edits using `byvolp=0`) and appear under the heading "key start fluxes".

Extreme caution must be used if the flux is desired at a point external to a scattering material in a multidimensional problem; for example, in the simulation of a point detector stood off some distance from a radiating cylinder. Discrete-ordinates ray effects will ordinarily destroy the accuracy of the computed flux at that point (although k_{eff} and the total leakage will generally be accurate).

Note that Eqs. (1) and (2) compute currents, not fluxes. There is no easy way to compute surface fluxes using PARTISN, as there is in MCNP using the surface flux (F2) tally.

The definition of the *source multiplication* M_S used in PARTISN for an inhomogeneous source problem is

$$M_S \equiv \frac{S_{inh} + S_{fiss}}{S_{inh}}, \tag{5}$$

where S_{inh} is the inhomogeneous source rate and S_{fiss} is the induced fission source rate. M_S is only printed in the iteration monitor, but it can be recovered from the "integral-source-i" and "integral-fission-i" values in the integral summary information, which are also the total "source" and "fission source" values from the system balance tables. The multiplication definition of Eq. (5) is a standard one.[4] In a time-dependent problem, S_{inh} would represent the number of neutrons in an instantaneous burst and S_{fiss} would be the total number of subsequent fission neutrons born.[4]

The definition of the *net multiplication* M_{net} printed in the MCNP problem summary is

$$M_{net} \equiv \frac{S_{inh} + (S_{fiss} - L_{fiss}) + (S_{n,xn} - L_{n,xn})}{S_{inh}}, \tag{6}$$

where L_{fiss} is the loss rate due to fission and $S_{n,xn}$ and $L_{n,xn}$ are the source rate and loss rate due to nonfission multiplication reactions. The loss rate due to fission is difficult to obtain from PARTISN because the data contains only the product $\nu\sigma_f$, and separating (n,xn) reactions in the cross-section data is also extremely difficult. Thus, comparing M_{net} between PARTISN and MCNP is not feasible. However, M_S can easily be computed from values in the MCNP problem summary (use the "neutron creation half" only, and note that prompt and delayed fissions are listed separately). The multiplication definition of Eq. (6) is *not* standard; there are other definitions of the "net multiplication."

Sometimes a quantity of interest is the *leakage multiplication* M_{lkg}, defined as

$$M_{lkg} \equiv \frac{L_{lkg}}{S_{inh}}, \tag{7}$$

where L_{lkg} is the loss rate due to leakage, or, in other words, the leakage rate. The leakage rate and the inhomogeneous source

rate are easily found in both the PARTISN and MCNP outputs as described previously in this section, so M_{lkg} can easily be compared. The multiplication definition of Eq. (7) is a standard one.[5]

Yet another definition of the subcritical multiplication, of course, is the standard[4]

$$M_k \equiv \frac{1}{1-k_{eff}}. \tag{8}$$

The k_{eff} eigenvalue can be obtained from both the PARTISN and MCNP outputs. It is frequently interesting to compare M_k with M_S of Eq. (5). If the inhomogeneous source in the M_S calculation approximates the fundamental eigenmode in the M_k calculation, the two values will be similar.

IV. Example Input for a Cylinder

An example input file that combines a neutron leakage and a k_{eff} calculation for a cylinder is listed here. The combined calculation is done with PARTISN's "stacked run" feature, in which a complete input file is terminated with the line "`]eof`" and there follow only the blocks necessary for a subsequent calculation. In this case, the geometry, cross sections, materials, and material assignments are the same, so blocks 2, 3, and 4 are not repeated. However, in the second problem, blocks 1 and 5 are given in their entirety; no user-input variables are passed to the second problem. In addition, different edits are requested, and block 6 appears twice. The outputs from both calculations would appear in a single output file.

This problem represents a cylinder of plutonium metal (90 wgt% ^{239}Pu, 10 wgt% ^{240}Pu; 16 g/cm^3) in a cylindrical lead (natural lead; 11.4 g/cm^3) pig. The plutonium radius and height are 8 and 4 cm, respectively, for a mass of 12.868 kg. The lead pig is 2 cm thick on the bottom and the cylindrical side and 1 cm thick on the top. There is a 2.5-cm void gap between the top of the plutonium and the lid of the pig; otherwise, the plutonium fits snugly inside the pig.

For the leakage calculation, the spontaneous fission neutron source density in each fine mesh and energy group was obtained in the following way. First, the source density in the plutonium was computed to be 1020 n/s·(g ^{240}Pu) × 16 g Pu/cm^3 × 0.10 g ^{240}Pu/g Pu = 1632 n/cm^3·s. The spectrum was obtained in a spreadsheet using the midpoints of the standard 30-group bin structure and the Watt fission spectrum with a = 0.799 MeV, b = 4.903 MeV^{-1}, as given in Appendix H of the MCNP manual. The `sourcf` entries are the product of the spectrum and the source density. The total inhomogeneous source rate, as given in the system and coarse mesh balance tables, is correctly 1.31256 × 10^6 n/s.

The output material map showing the geometry, number of fine meshes per coarse mesh, and material assignments for this problem are shown in Figure 1.

To change to an adjoint input file, it would only be necessary to change the `ith` parameter in two places (block 5 in each problem) and the `ajed` parameter in two places (block 6 in each problem). (Of course, the inhomogeneous source in block 5 for the source problem would have to change to a meaningful adjoint source.)

```
     1     0
plutonium cylinder -- forward input file, leakage.
/ * * * * block 1 * * * * *
 igeom=r-z isn= 16 iquad=6 ngroup=  30
 niso=  0 mt= 2 nzone= 2
 im=  2 it=   250
 jm=  5 jt=   238
 t
/ * * * * block 2 * * * * *
 xmesh= 0.  8.  10.
 ymesh= 0.
        2.
        4.
        6.
        8.5
        9.5
 xints=    200    50
 yints=     50
            50
            50
            63
            25
 zones=   2  2 ;
          1  2 ;
```

```
          1  2 ;
          0  2 ;
          2  2 ;
 t
/ * * * * block 3 * * * *
 lib=ndilib fissneut=1
 t
/ * * * * block 4 * * * *
 matspec=wtfrac;
 matls= pu  "94239.690nm"   0.90
            "94240.601nm"   0.10 ;
        pig "82000.602nm"   1.   ;
 assign= zone01   pu    16.  ;
         zone02   pig   11.4 ;
 t
/ * * * * block 5 * * * *
 ievt=0 ith=0 isct=3 epsi= 1.00E-06
 balp=1 xsectp=2
 sourcf=
  / group  1 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 1.146E-03  50r 0.000E+00 ; 49y 1 ;
      200r 1.146E-03  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group  2 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 6.224E-03  50r 0.000E+00 ; 49y 1 ;
      200r 6.224E-03  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group  3 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 2.588E-02  50r 0.000E+00 ; 49y 1 ;
      200r 2.588E-02  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group  4 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 1.318E-01  50r 0.000E+00 ; 49y 1 ;
      200r 1.318E-01  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group  5 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 8.764E-01  50r 0.000E+00 ; 49y 1 ;
      200r 8.764E-01  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group  6 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 4.723E+00  50r 0.000E+00 ; 49y 1 ;
      200r 4.723E+00  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group  7 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 2.415E+01  50r 0.000E+00 ; 49y 1 ;
      200r 2.415E+01  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group  8 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 7.417E+01  50r 0.000E+00 ; 49y 1 ;
      200r 7.417E+01  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group  9 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 1.146E+02  50r 0.000E+00 ; 49y 1 ;
      200r 1.146E+02  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group 10 source
```

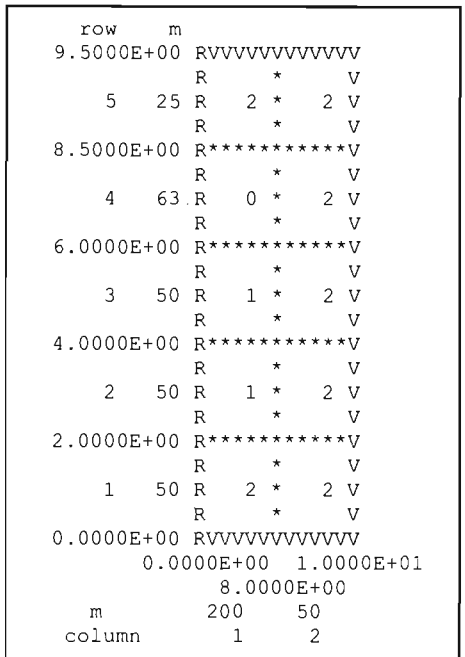

Figure 1. Material map, from the output file. "V" means vacuum boundary and "R" means reflective; m is the number of fine meshes per coarse mesh.

```
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 1.530E+02  50r 0.000E+00 ; 49y 1 ;
     200r 1.530E+02  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 11 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 1.833E+02  50r 0.000E+00 ; 49y 1 ;
     200r 1.833E+02  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 12 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 2.074E+02  50r 0.000E+00 ; 49y 1 ;
     200r 2.074E+02  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 13 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 2.090E+02  50r 0.000E+00 ; 49y 1 ;
     200r 2.090E+02  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 14 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 1.877E+02  50r 0.000E+00 ; 49y 1 ;
     200r 1.877E+02  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 15 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 1.584E+02  50r 0.000E+00 ; 49y 1 ;
     200r 1.584E+02  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 16 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 1.205E+02  50r 0.000E+00 ; 49y 1 ;
     200r 1.205E+02  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 17 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 7.572E+01  50r 0.000E+00 ; 49y 1 ;
     200r 7.572E+01  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 18 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 4.648E+01  50r 0.000E+00 ; 49y 1 ;
     200r 4.648E+01  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 19 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 2.831E+01  50r 0.000E+00 ; 49y 1 ;
     200r 2.831E+01  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 20 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 1.720E+01  50r 0.000E+00 ; 49y 1 ;
     200r 1.720E+01  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 21 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 1.044E+01  50r 0.000E+00 ; 49y 1 ;
     200r 1.044E+01  50r 0.000E+00 ; 49y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
     200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
/ group 22 source
     200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
     200r 6.335E+00  50r 0.000E+00 ; 49y 1 ;
```

```
      200r 6.335E+00  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group 23 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 3.842E+00  50r 0.000E+00 ; 49y 1 ;
      200r 3.842E+00  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group 24 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 2.330E+00  50r 0.000E+00 ; 49y 1 ;
      200r 2.330E+00  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group 25 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 1.414E+00  50r 0.000E+00 ; 49y 1 ;
      200r 1.414E+00  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group 26 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 8.576E-01  50r 0.000E+00 ; 49y 1 ;
      200r 8.576E-01  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group 27 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 5.204E-01  50r 0.000E+00 ; 49y 1 ;
      200r 5.204E-01  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group 28 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 3.159E-01  50r 0.000E+00 ; 49y 1 ;
      200r 3.159E-01  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group 29 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 1.913E-01  50r 0.000E+00 ; 49y 1 ;
      200r 1.913E-01  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
  / group 30 source
      200r 0.000E+00  50r 0.000E+00 ; 49y 1 ;
      200r 9.916E-02  50r 0.000E+00 ; 49y 1 ;
      200r 9.916E-02  50r 0.000E+00 ; 49y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 62y 1 ;
      200r 0.000E+00  50r 0.000E+00 ; 24y 1 ;
 t
/ * * * * block 6 * * * *
 ajed=0 zned=1 igrped=3 massed=1
 pted=1 points=45100 byvolp=0
 rsfe=  30r 1.
 t
]eof
     1     0
plutonium cylinder -- forward input file, keff.
/ * * * * block 1 * * * *
 igeom=r-z isn= 16 ngroup=  30
 niso=  0 mt= 2 nzone= 2
 im=  2 it=   250
 jm=  5 jt=   238
 iquad=6
 t
/ * * * * block 5 * * * *
 ievt=1 ith=0 isct=3 epsi= 1.00E-06 norm=1.
 balp=1
 t
/ * * * * block 6 * * * *
 ajed=0 zned=1 igrped=3 massed=2
 resdnt=1 edxs=2 ; 3 ; 4 ;
```

```
rsfe=  30r 1.
t
```

V. Summary and Conclusions

PARTISN is an extremely versatile multigroup discrete-ordinates code with many features. Many, many options have not been included in this brief guide. In this paper the focus has been on running k_{eff} and steady-state leakage problems in one-dimensional spheres and two-dimensional cylinders using the most basic options.

Among the most useful capabilities not addressed here are α-eigenvalue calculations, dimension and concentration criticality searches, various acceleration controls, time- and temperature-dependent calculations, and the ability to link to MCNP through lnk3dnt files. Materials may be set up and mixed in many more interesting and useful ways than presented here. The edit module is also much more flexible than is granted here.

The best place to look for answers to PARTISN questions, including information not appearing in this brief guide, is Chapter 2 of the manual, where all the options and keywords are listed and described. It is hoped that this paper will be a useful starting point for those trying to learn PARTISN.

Appendix A: How to Manually Input a Small Cross-Section Table

It is frequently desirable to input a set of simple, known cross sections. For this, use the block 3 keywords of Table X rather than those of Table IV (Sec. II.D). The fission spectrum χ^g is entered as a string in block 3, in which case it applies to all fissionable isotopes, or as a zone- (material-) dependent set of strings in block 5, as shown in Table XI. It should be noted that many benchmark problems are slabs; `igeom=slab` is a valid entry in block 1. Also in block 1, `niso` must be the number of isotopes in the table, not 0 as when using the NDI (Table II).

Table X. Block 3 Keywords for User-Input Cross Sections.

Keyword	Meaning
`lib`	Name of cross-section data file; use ODNINP
`maxord`	Highest Legendre order in the scattering tables (0 is isotropic).
`ihm`	Number of positions (entries) in each row[a] of the cross-section table.
`iht`	Position number of the total cross section; use `iht=3`.
`ihs`	Position number of the self-scatter cross section.
`ifido`	Format of the library; –1/0 = 18/12 characters for each number
`ititl`	Set to 1 and use a title line before each table for clarity.
`i2lp1`	Higher-order scattering cross sections contain the (2*l*+1) factor; 0/1=no (default)/yes.
`names` [NISO]	Character name for each of the input isotopes or materials.
`ntpi` [NISO]	Number of Legendre scattering orders for each isotope in the library (1 is isotropic). Default MAXORD+1.
`chivec` [NGROUP]	Fission fraction born into each group. Applied to all isotopes. Overridden by `chi` in block 5 if present.
`lng`	Number of the last neutron group in a coupled neutron-photon library

[a] "Row" does not have the common meaning of a table row that ends in a hard return. See text and example.

Table XI. Block 5 Keywords for User-Input Cross Sections.

Keyword	Meaning
`chi` [NGROUP;MT[a]]	Fission fraction born into each group, in each material. Overrides `chivec` in block 3 if present.

[a] Recall MT=NZONE in the conventions of this paper.

In this form, cross sections are entered as a set of tables after the `t` that terminates block 3. There are NISO × NTPI(*i*) tables, one for each isotope *i* and each scattering order for isotope *i*, each with a single title line (in the convention suggested in Table X). After the title line there are IHM × NGROUP numerical values, with either six or four on a line (for `ifido=0` or `-1`, respectively), with each value occupying 12 or 18 characters (again, for `ifido=0` or `-1`, respectively). In PARTISN

parlance, a *row* is IHM entries regardless of how many input file lines they require, and only the first row of a table necessarily begins in position 1.

The normal cross section ordering within a row is shown in Table XII. Note that the product $\nu\sigma_f^g$ is entered, not the separate components. Also note that the scattering matrix is entered as scattering *in* for each group.

Table XII. Cross Section Order in a Row.

Position in Row	Value
1	Absorption, σ_a^g
2	Neutrons per fission times fission cross section, $\nu\sigma_f^g$
3	Total, σ_t^g (iht=3)
IHS–*G*+I = 4	Upscattering begins (see Table XIII)
⋮	
IHS–1	Upscattering into g, $\sigma_s^{g+1\to g}$
IHS	Self scattering, $\sigma_s^{g\to g}$
IHS+1	Downscattering into g, $\sigma_s^{g-1\to g}$
⋮	
IHS+*G*–1 = IHM	Downscattering ends (see Table XIII)

Table XII shows the positions that would be entered if the full scattering matrix (including all upscattering) were used. If there is no upscattering, then the first scattering cross section is self-scattering and IHS = 4. Table XIII shows the table row positions and the cross sections to enter for an example case of a five-group set with full upscattering. Note that zero is entered as a placeholder to fill in those parts of the matrix that are undefined (e.g., downscattering into group 1). In the usual case in which upscattering occurs only among the lower groups, the full upscattering matrix need not be given; columns in Table XIII can shift down toward position 4 as unnecessary columns are removed.

Table XIII. Full Scattering Matrix for NGROUP = 5.

Position in Row	4	5	6	7	8 = IHS	9	10	11	12 = IHM
Group I	$\sigma_s^{5\to1}$	$\sigma_s^{4\to1}$	$\sigma_s^{3\to1}$	$\sigma_s^{2\to1}$	$\sigma_s^{1\to1}$	0.	0.	0.	0.
Group 2	0.	$\sigma_s^{5\to2}$	$\sigma_s^{4\to2}$	$\sigma_s^{3\to2}$	$\sigma_s^{2\to2}$	$\sigma_s^{1\to2}$	0.	0.	0.
Group 3	0.	0.	$\sigma_s^{5\to3}$	$\sigma_s^{4\to3}$	$\sigma_s^{3\to3}$	$\sigma_s^{2\to3}$	$\sigma_s^{1\to3}$	0.	0.
Group 4	0.	0.	0.	$\sigma_s^{5\to4}$	$\sigma_s^{4\to4}$	$\sigma_s^{3\to4}$	$\sigma_s^{2\to4}$	$\sigma_s^{1\to4}$	0.
Group 5	0.	0.	0.	0.	$\sigma_s^{5\to5}$	$\sigma_s^{4\to5}$	$\sigma_s^{3\to5}$	$\sigma_s^{2\to5}$	$\sigma_s^{1\to5}$

In higher-order scattering tables (i.e., P_n scattering, where $n > 0$), positions IHM+1, IHM+2, and IHM+3 are set to zero (merely placeholders) and the P_n scattering begins at position IHM+4.

A two-group example problem with P_1 scattering and no upscattering appears on the next page. It is an analytic k_{eff} benchmark problem in which k_{eff} = I, problem 71 of Ref. 6, using the macroscopic cross sections of Tables 49 and 50 of that reference. Reference 6 is explicit that anisotropic scattering cross sections are given without the (2*l*+1) factor.

The problem is a homogeneous, one-material slab. In the example, the same cross sections have been entered for two isotopes just as a demonstration. The cross sections for isotope 1 use the standard FORTRAN "1pe12.5" format. The cross sections for isotope 2 have two extra significant figures because the formatting dispenses with the "E" and uses a single digit (and a sign) for the exponent. An extra digit can still be gained by occupying the leading space in each position. If more precision is desired, use `ifido=-1`.

Since material macroscopic cross sections are given, the atom density of the material must be unity in block 4. In contrast to Table V in Sec. II.E, here `matspec=atdens` is used. In this format, the isotopic constituents on the `matls` keyword are given in atom densities and the code computes the atom density of the material as the sum of the atom densities of the constituents and it is *not* entered on the `assign` keyword. Of course, microscopic cross sections may also be given.

When cross sections are entered this way, definitely use `xsectp=2` in block 5 to print the macroscopic cross sections and compare them with what was intended.

```
     1     0
sood URRa-2-1-SL, problem 71; tables 49, 50, and 51
/ * * * * block 1 * * * *
 igeom=slab ngroup=2 isn=196 niso=2 mt=1 nzone=1 im=1 it=950
 t
/
/ * * * * block 2 (geometry) * * * *
 xmesh=0. 18.9918 xints= 950 zones= 1
 t
/
/ * * * * block 3 (cross sections) * * * *
 lib=odninp
 maxord=1 ihm=5 iht=3 ihs=4 ifido=0 ititl=1 ntpi=2 ; 2 ; i2lp1=0
 chivec= 1. 0. ;
 names=i1 ; i2 ;
 t
 iso 1, siga      nusigf        sigt   self-scat dn-scat(in)|rep for g>1
 2.05300E-03 2.62100E-03 6.56960E-01 6.25680E-01 0.00000E+00 7.64200E-02
 1.26580E-01 2.52025E+00 2.44383E+00 2.92270E-02
 isotope 1, P1 scattering
          0.          0.          0. 2.74590E-01 0.00000E+00          0.
          0.          0. 8.33180E-01 7.57370E-03
 iso 2, siga      nusigf        sigt   self-scat dn-scat(in)|rep for g>1
 2.0530000-3 2.6210000-3 6.5696000-1 6.2568000-1 0.0000000+0 7.6420000-2
 1.2658000-1 2.5202500+0 2.4438300+0 2.9227000-2
 isotope 2, P1 scattering
          0.          0.          0. 2.7459000-1 0.0000000+0          0.
          0.          0. 8.3318000-1 7.5737000-3
/
/ * * * * block 4 (mixing) * * * *
 matspec= atdens ;
 matls= m1 i1 0.4     / these atom densities are arbitrary since the
           i2 0.6 ;   / cross sections are identical, but must sum to 1.
 assign= zone01 m1 ;
 t
/
/ * * * * block 5 (solver) * * * *
 ith=0 ievt=1 isct=1 norm=1.
 balp=1 fluxp=1 xsectp=2
 epsi=1.e-12 oitm=99
 t
/
/ * * * * block 6 (edit) * * * *
 zned=1 edxs=2 ; 3 ; 4 ; resdnt=1 igrped=3
 t
```

Appendix B: Details of Boundary Source Input

On Table VII in Sec. II.F, MM represents the number of incoming angles on a boundary (the total number of angles in the quadrature set is MM × 2). When using the `sarite`, `sabott`, or `satop` keywords to enter boundary sources, the order of the MM incoming fluxes is the order in which the ordinates appear in the "key start sn constants" section of the output.

For one-dimensional spheres (and slabs), MM = ISN/2 ordinates are listed in the "key start sn constants" output, and "mu" is the cosine of the angle between the inward normal and the ingoing ordinate. Inputting sources as $\{c/\mu_m, m = 1,\ldots,MM\}$, where c is a normalization constant, will give a source that is isotropic in the particle density. The `sirite` keyword will give a source that is isotropic in the flux (all incoming fluxes equal).

For two-dimensional cylinders, only half of the required MM = $(ISN)^2/2$ ordinates are listed in the "key start sn constants" output because the ordinates listing is for only one octant and there are four in a cylinder (there are two in a sphere). The second half of the required ordinates for the input of boundary sources is in the same order as those listed. On the radial surface (`sarite`), "mu" is the cosine of the angle between the inward normal and the ingoing ordinate. As for spheres, inputting sources as $\{c/\mu_m, m = 1,\ldots,MM\}$ will give a source on the radial boundary that is isotropic in the particle density. The `sirite` keyword will give a source that is isotropic in the flux (all incoming fluxes equal).

On the top and bottom surfaces of a cylinder (`satop` and `sabott`), "eta" is the cosine of the angle between the inward normal and the ingoing ordinate, in contrast to how it appears things should be based on Fig. 8.2 in the PARTISN manual. This is because the "top" is actually in the y direction, as depicted in Fig. 2.2 of the manual. Thus, inputting sources as $\{c/\eta_m, m = 1,\ldots,MM\}$ will give a source on the top or bottom boundary that is isotropic in the particle density. The `sitop` and `sibott` keywords will give a source that is isotropic in the flux (all incoming fluxes equal).

References

1. Ray E. Alcouffe, Randal S. Baker, Jon A. Dahl, Scott A. Turner, and Robert C. Ward, "PARTISN: A Time-Dependent, Parallel Neutral Particle Transport Code System," LA–UR–08-7258, Version 6.52, Los Alamos National Laboratory (November 2008). PARTISN version 6.52 was released in April 2011.

2. X-5 Monte Carlo Team, "MCNP–A General Monte Carlo N-Particle Transport Code, Version 5," Chap. 2, Vol. I, LA–UR–03–1987, Los Alamos National Laboratory (April 24, 2003).

3. Randy S. Baker, CCS-2, Los Alamos National Laboratory, private communication (November 2008).

4. Doug Reilly, Norbert Ensslin, Hastings Smith, Jr., and Sarah Kreiner, "Passive Nondestructive Assay of Nuclear Materials," Chap. 12, Los Alamos National Laboratory, NUREG/CR-5550 (March 1991).

5. Doug Reilly, Norbert Ensslin, Hastings Smith, Jr., and Sarah Kreiner, "Passive Nondestructive Assay of Nuclear Materials," Chap. 14, Los Alamos National Laboratory, NUREG/CR-5550 (March 1991).

6. Avneet Sood, R. Arthur Forster, and D. Kent Parsons, "Analytical Benchmark Test Set For Criticality Code Verification," *Progress in Nuclear Energy*, **42**, *1*, 55-106 (2003).

JAF:jaf

Distribution:
R. C. Little, XCP-DO, MS B259, rcl@lanl.gov
A. Sood, XCP-7, MS A143, sooda@lanl.gov
J. K. Mattingly, Sandia National Laboratory, MS 0791, jkmatti@sandia.gov
J. E. Koster, GS-WME, MS C937, jkoster@lanl.gov
D. A. Pimentel, HPC-1, MS F663, davidp@lanl.gov
R. S. Baker, CCS-2, MS D409, rsb@lanl.gov
J. A. Dahl, CCS-2, MS D409, dahl@lanl.gov
E. J. Fichtl, CCS-2, MS D409, efichtl@lanl.gov
S. A. Turner, CCS-2, MS D409, turners@lanl.gov
B. D. Rooney, N-2, MS B228, rooney@lanl.gov
S. C. Frankle, XTD-4, MS T082, frankles@lanl.gov
D. K. Parsons, XCP-5, MS F663, dkp@lanl.gov
M. C. White, XCP-5, MS F663, morgan@lanl.gov
G. P. Estes, XCP-7, MS F663, gpe@lanl.gov
J. P. Lestone, XCP-7, MS F663, lestone@lanl.gov
E. F. Shores, XCP-7, MS F663, eshores@lanl.gov
C. J. Solomon, XCP-7, MS F663, csolomon@lanl.gov
B. A. Temple, XCP-7, MS F663, temple@lanl.gov
J. A. Favorite, XCP-7, MS F663, fave@lanl.gov
X-Archive, xarchive@lanl.gov
XCP-DO File
XCP-7 File