CHAPTER 3

DESCRIPTION OF MCNP INPUT

Input to MCNP consists of several files, but the main one supplied by the user is the INP (the default name) file, which contains the input information necessary to describe the problem. Only a small subset of all available input cards will be needed in any particular problem. The input cards are summarized by card type on page 3–147. The word "card" is used throughout this manual to describe a single line of input up to 80 characters.

Maximum dimensions exist for some MCNP input items; they are summarized on page 3–151. The user can increase any of these maximum values by altering the code and recompiling.

All features of MCNP should be used with caution and knowledge. This is especially true of detectors and variance reduction schemes; you are encouraged to read the appropriate sections of Chapter 2 before using them.

The units used throughout MCNP are given in Chapter 1 on page 1–20.

I.   INP FILE

The INP file can have two forms, initiate-run and continue-run. Either can contain an optional message block that replaces or supplements the MCNP execution line information.

A.   Message Block

A user has the option to use a message block before the problem identification title card in the INP file. In computer environments where there are no execution line messages, the message block is the only means for giving MCNP an execution message. Less crucially, it is a convenient way to avoid retyping an often-repeated message. The message block starts with the string MESSAGE: and is limited to columns 1−80. Alphabetic characters can be upper, lower, or mixed case. The message block ends with a blank line delimiter before the title card.All cards before the blank line delimiter are continuation cards. A $ and & in the message block are end−of−line markers. The syntax and components of the message are the same as for the regular execution line message discussed on page 1–32. Any filename substitution, program module execution option or keyword entry on the execution line takes precedence over conflicting information in the message block. INP = *filename* is <u>not</u> a legitimate entry in the message block. The name INP can be changed on the execution line only.

B.   *Initiate-Run*

This form is used to set up a Monte Carlo problem (describe geometry, materials, tallies, etc.) and run if message block is present. The initiate-run file has the following form:

Message Block $\Big\}$ Optional
Blank Line Delimiter
Title Card
Cell Cards
  .
  .
  .
Blank Line Delimiter
Surface Cards
  .
  .
  .
Blank Line Delimiter
Data Cards
  .
  .
  .
Blank Line Terminator   Recommended
Anything Else   Optional

The first card in the file after the optional message block is the required problem title card. It is limited to one 80−column line and is used as a title in various places in the MCNP output.It can contain any information the user desires (or can even be blank) and often contains information describing the particular problem. Note that a blank card elsewhere is used as a delimiter or as a terminator. Alphabetic characters can be upper, lower, or mixed case.

With a valid set of data cards MCNP will run with or without the blank line terminator. With the terminator MCNP will stop reading the input file there even if additional lines are in the file. Some users like to keep additional material, such as alternative versions of the problem or textual information, associated with the input file itself. The terminator will prevent such additional lines from being read.

C.   *Continue−Run*

Continue-run is used to continue running histories in a problem that was terminated earlier−for example, to run the job 2 hours and then to run it an additional hour later. It can also be used to reconstruct the output of a previous run. A continue-run must contain C or CN in the MCNP execution line or message block to indicate a continue-run. It will start with the last dump unless C m is used to start with the mth dump.

In addition to the C or CN option on the MCNP execution line, two files can be important for this procedure: (1) the restart file (default name RUNTPE), and (2) an optional continue-run input file (default name INP).

The run file, generated by MCNP in the initiate-run sequence, contains the geometry, cross sections, problem parameters, tallies, and all other information necessary to restart the job. In addition the problem results at various stages of the run are recorded in a series of dumps. See the PRDMP card (page 3–128) for a discussion of the selection of dump times. As discussed below, the run may be restarted from any of these dumps.

The CN execution message option differs from the C option only in that the dumps produced during the continue-run are written immediately after the fixed data portion of the RUNTPE file rather than after the dump from which the continue-run started. The new dumps overwrite the old dumps, providing a way for the user to prevent unmanageable growth of RUNTPE files. RUNTPE growth also can be controlled by the NDMP entry on the PRDMP card.

The optional continue-run input file must have the word CONTINUE as the first entry on the first line (title card), or after the optional Message Block and its blank line delimiter. Alphabetic characters can be upper, lower, or mixed case. This file has the following form:

> Message Block
> Blank Line Delimiter   } Optional
> CONTINUE
> Data Cards
>   .
>   .
> Blank Line Terminator   Recommended
> Anything else   Optional

The data cards allowed in the continue-run input file are a subset of the data cards available for an initiate-run file. The allowed continue-run data cards are FQ, DD, NPS, CTME, IDUM, RDUM, PRDMP, LOST, DBCN, PRINT, KCODE, MPLOT, ZA, ZB, and ZC.

A very convenient feature is that if none of the above items is to be changed (and if the computing environment allows execution line messages), the continue-run input file is not required; only the run file RUNTPE and the C option on the MCNP execution line are necessary. For example, if you run a job for a minute but you want more particles run, execute with the C or CN message on the execute line, and the job will pick up where it stopped and continue until another time limit or particle cutoff is reached or until you stop it manually. This example assumes that a restart file called RUNTPE from the initial run is in your current directory.

The complete continue-run execution line option is C m or CN m, where m specifies which dump to pick up from the RUNTPE and to continue with. If m is not specified, the last dump is taken by default. If the initial run producing the RUNTPE was stopped because of particle cutoff (NPS card, page 3–126), NPS must be increased for a continue-run. The NPS card refers to total histories to be run, including preceding continue-runs and the initial run. CTME in a continue–run is the number of minutes <u>more</u> to run, not cumulative total time. To run more KCODE cycles, only the fourth entry KCT matters. Like NPS, KCT refers to total cycles to be run, including previous ones.

In a continue-run, a negative number entered on the NPS card produces a print output file at the time of the requested dump. No more histories will be run. This can be useful when the printed output has been lost or you want to alter the content of the output with the PRINT or FQ cards.

Be cautious if you use a FILES card in the initial run. See page 3–134.

D.   *Card Format*

All input lines are limited to 80 columns. Alphabetic characters can be upper, lower, or mixed case. Most input is entered in horizontal form; however, a vertical input format is allowed for data cards. A comment can be added to any input card. A $ (dollar sign) terminates data entry and anything that follows the $ is interpreted as a comment. Blank lines are used as delimiters and terminators. Data entries are separated by one or more blanks.

Comment cards can be used anywhere in the INP file after the problem title card and before the last blank terminator card. These cards must have a C anywhere in columns 1–5 followed by at least one blank. Comment cards are printed only with the input file listing and not anywhere else in the MCNP output file. The FCn input card is available for user comments and is printed as a heading for tally n (as a tally title, for example). The SCn card is available for user comments and is printed as a heading for source probability distribution n.

1.   Horizontal Input Format

Cell, surface, and data cards all must begin within the first five columns. The card name or number and particle designator is followed by data entries separated by one or more blanks. Blanks in the first five columns indicate a continuation of the data from the last named card. An & (ampersand) preceded by at least one blank ending a line indicates data will continue on the following card. Data on the continuation card can be in columns 1–80. Completely blank cards are reserved as delimiters between major sections of the input file. An individual entry must be entirely on one line. There can be only one card of any given type for a given particle designation (see page 3–7). Integers must be entered where integer input is required. Other numerical data can be entered as integer or floating point and will be read properly by MCNP. (In fact noninteger numerical data can be entered in any form acceptable to a FORTRAN E-edit descriptor.)

Four features incorporated in the code facilitate input card preparation:

1. nR means *repeat* the immediately preceding entry on the card n times. For example, 2 4R is the same as 2 2 2 2 2.

2. nI means *insert* n linear interpolates between the entries immediately preceding and following this feature. For example, 1.5 2I 3.0 on a card is the same as 1.5 2.0 2.5 3. In the construct X nI Y, if X and Y are integers, and if Y − X is an exact multiple of n+1, correct integer interpolates will be created. Otherwise only real interpolates will be created, but Y will be stored directly in all cases. In the above example, the 2.0 may not be exact, but in the example 1 4I 6 = 1 2 3 4 5 6, all interpolates are exact.

3. xM is a *multiply* feature and when used on an input card, it is replaced by the value of the previous entry on the card multiplied by the factor x. For example, 1 1 2M 2M 2M 2M 4M 2M 2M is equivalent to 1 1 2 4 8 16 64 128 256.

4. nJ can be used on an input card to *jump* over the entry where used and take the default value. As an example, the following two cards are identical in their effect:

DD	.1	1000
DD	J	1000

    J J J is also equivalent to 3J. You can jump to a particular entry on a card without having to explicitly specify prior items on the card. This feature is convenient if you know you want to use a default value but can't remember it. DBCN 7J 5082 is another example.

These features apply to both integer and floating point quantities. If n (an integer) is omitted in the constructs nR, nI, and nJ, then n is assumed to be 1. If x (integer or floating point) is omitted in xM, it is a fatal error. The rules for dealing with adjacent special input items are as follows:

1. nR must be preceded by a number or by an item created by R or M.

2. nI must be preceded by a number or by an item created by R or M, and must be followed by a number.

3. xM must be preceded by a number or by an item created by R or M.

4. nJ may be preceded by anything except I and may begin the card input list.

Examples:

1 3M 2R	=	1 3 3 3
1 3M I 4	=	1 3 3.5 4
1 3M 3M	=	1 3 9
1 2R 2I 2.5	=	1 1 1 1.5 2.0 2.5
1 R 2M	=	1 1 2
1 R R	=	1 1 1
1 2I 4 3M	=	1 2 3 4 12
1 2I 4 2I 10	=	1 2 3 4 6 8 10

> 3J 4R   is   illegal.
> 1 4I 3M   is   illegal.
> 1 4I J   is   illegal.

2.   <u>Vertical Input Format</u>

Column input is particularly useful for cell parameters and source distributions. Cell importances or volumes strung out on horizontal input lines are not very readable and often cause errors when users add or delete cells. In column format, all the cell parameters for one cell can be on a single line, labeled with the name of the cell. If a cell is deleted, the user deletes just one line of cell parameters instead of hunting for the data item that belongs to the cell in each of several multiline cell parameter cards. For source distributions, corresponding SI, SP, and SB values are side by side. Source options, other than defaults, are on the next line and must all be entered explicitly. The & continuation symbol is not needed, and if present, is ignored.

In column format, card names are put side by side on one input line and the data values are listed in columns under the card names. A # is put somewhere in columns 1−5 on the line with the card names. The card names must be all cell parameters, all surface parameters, or all something else. If a card name appears on a # card, there must not be a regular horizontal card by that name in the same input file. If there are more entries on data value lines than card names on the # line, the first data entry is a cell or surface number. If any cell names are entered, all must be entered. If cell names are entered, the cells don't have to be in the same order as they are in the cell cards block. If cell names are omitted, the default order is the order of the cells in the cell card block. The same rules apply to surface parameters, but because we presently have only one surface parameter (AREA), column input of surface parameters is less useful.

There can be more than one block of column data in an input file. Typically, there would be one block for cell parameters and one for each source distribution. If a lot of cell parameter options are being used, additional blocks of column data would be needed.

The entries in each column do not need to be precisely under the card name at the top of the column, but you might want the columns to be reasonably neat for readability. The column format is intended for input data that naturally fit into columns of equal length, but less tidy data are not prohibited. If a longer column is to the right of a shorter column, the shorter column must be filled with enough J entries to eliminate any ambiguity about which columns the data items are in.

Special syntax items (R, M, I, and J) are not as appropriate in column format as they are on horizontal lines, but they are not prohibited. They are, of course, interpreted vertically instead of horizontally. Multiple special syntax items, such as 9R, are not allowed if cell or surface names are present.

The form of a column input block is

$$
\begin{array}{ccccc}
\# & S_1 & S_2 & \dots & S_m \\
K_1 & D_{11} & D_{12} & \dots & D_{1m} \\
K_2 & D_{21} & D_{22} & \dots & D_{2m} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
K_n & D_{n1} & D_{n2} & \dots & D_{nm}
\end{array}
$$

1. The # is somewhere in columns 1−5.

2. Each line can be only 80 columns wide.

3. Each column, S_i through D_{li}, where l may be less than n, represents a regular input card.

4. The S_i must be valid MCNP card names. They must be all cell parameters, all surface parameters, or all something else.

5. D_{1i} through D_{ni} must be valid entries for an S_i card, except that $D_{l+1,i}$ through D_{ni} may be some J's possibly followed by some blanks.

6. If D_{ji} is nonblank, $D_{j,i-1}$ must also be nonblank. A J may be used if necessary to make $D_{j,i-1}$ nonblank.

7. The S_i must not appear anywhere else in the input file.

8. The K_j are optional integers. If any are nonblank, all must be nonblank.

9. If the S_i are cell parameter card names, the K_j, if present, must be valid cell names. The same is true with surface parameters.

10. If the K_j are present, the D_{ji} must not be multiple special syntax items, such as *9R*.

E.   *Particle Designators*

Several of the input cards require a particle designator to distinguish between input data for neutrons, for photons and for electrons. These cards are IMP, EXT, FCL, WWN, WWE, WWP, WWGE, DXT, DXC, F, F5X, F5Y, F5Z, PHYS, ELPT, ESPLT, CUT and PERT. The particle designator consists of the symbol : (colon) and the letter N, P or E immediately after the name of the card. At least one blank must follow the particle designator. For example, to enter neutron importances, use an IMP:N card; enter photon importances on an IMP:P card. To specify the same value for more than one kind of particle, a single card can be used instead of several. Example: IMP:E,P,N 1 1 0. With a tally card, the particle designator follows the card name including tally number. For example, *F5:N indicates a neutron point detector energy tally. In the heating tally case, both particle designators may appear. The syntax F6:N,P indicates the combined heating tally for both neutrons and photons.

F.   Default Values

Many MCNP input parameters have default values that are summarized on page 3–147. Therefore you do not always have to specify explicitly every input parameter every time if the defaults match your needs. If an input card is left out, the default values for all parameters on the card are used. However, if you want to change a particular default parameter on a card but that parameter is preceded by others, you have to specify the others or use the nJ jump feature to jump over the parameters for which you still want the defaults. CUT:P 3J −.10 is a convenient way to use the defaults for the first three parameters on the photon cutoff card but change the fourth.

G.   Input Error Messages

MCNP makes extensive checks (over 400) of the input file for user errors. A fatal error message is printed, both at the terminal and in the OUTP file, if the user violates a basic constraint of the input specification, and MCNP will terminate before running any particles. The first fatal error is real; subsequent error messages may or may not be real because of the nature of the first fatal message. The FATAL option on the MCNP execution line instructs MCNP to ignore fatal errors and run particles, but the user should be extremely cautious about doing this.

Most MCNP error messages are warnings and are not fatal. The user should not ignore these messages but should understand their significance before making important calculations.

In addition to FATAL and WARNING messages, MCNP issues BAD TROUBLE messages immediately before any impending catastrophe, such as a divide by zero, which would otherwise cause the program to "crash." MCNP terminates as soon as the BAD TROUBLE message is issued. User input errors in the INP file are the most common reason for issuing a BAD TROUBLE message. These error messages indicate what corrective action is required.

H.   Geometry Errors

There is one important kind of input error that MCNP will not detect while processing data from the INP file. MCNP cannot detect overlapping cells or gaps between cells until a particle track actually gets lost. Even then the precise nature of the error may remain unclear. However, there is much that you can and should do to check your geometry before starting a long computer run.

Use the geometry-plotting feature of MCNP to look at the system from several directions and at various scales. Be sure that what you see is what you intend. Any gaps or overlaps in the geometry will probably show up as dashed lines. The intersection of a surface with the plot plane is drawn as a dashed line if there is not exactly one cell on each side of the surface at each point. Dashed lines can also appear if the plot plane happens to coincide with a plane of the problem, if there are any cookie-cutter cells in the source, or if there are DXTRAN spheres in the problem.

Set up and run a short problem in which your system is flooded with particle tracks from an external source. The necessary changes in the INP file are as follows:

1. Add a VOID card to override some of the other specifications in the problem and make all the cells voids, turn heating tallies into flux tallies, and turn off any FM cards.

2. Add another cell and a large spherical surface to the problem such that the surface surrounds the system and the old outside world cell is split by the new surface into two cells: the space between the system and the new surface, which is the new cell, and the space outside the new surface, which is now the outside world cell. Be sure that the new cell has nonzero importance. Actually, it is best to make all nonzero importances equal. If the system is infinite in one or two dimensions, use one or more planes instead of a sphere.

3. Replace the source specifications by an inward directed surface source to flood the geometry with particles:

$$\text{SDEF} \quad \text{SUR=m} \quad \text{NRM} = -1$$

where m is the number of the new spherical surface added in Step 2. If the new surface is a plane, you must specify the portion to be used by means of POS and RAD or possibly X, Y, and Z source distributions.

Because there are no collisions, a short run will generate a great many tracks through your system. If there are any geometry errors, they should cause some of the particles to get lost.

When a particle first gets lost, whether in a special run with the VOID card or in a regular production run, the history is rerun to produce some special output on the OUTP file. Event-log printing is turned on during the rerun. The event log will show all surface crossings and will tell you the path the particle took to the bad spot in the geometry. When the particle again gets lost, a description of the situation at that point is printed. You can usually deduce the cause of the lost particle from this output. It is not possible to rerun lost particles in a multitasking run.

If the cause of the lost particle is still obscure, try plotting the geometry with the origin of the plot at the point where the particle got lost and with the horizontal axis of the plot plane along the direction the particle was moving. The cause of the trouble is likely to appear as a dashed line somewhere in the plot or as some discrepancy between the plot and your idea of what it should look like.

II. CELL CARDS

Form: j   m d   geom   params
   or: j   LIKE n BUT   list

j	= cell number; $1 \leq j \leq 99999$.
	If cell has transformation, $1 \leq j \leq 999$. See page 3–27.
m	= 0 if the cell is a void.
	= material number if the cell is not a void. This indicates that the cell is to contain material *m*, which is specified on the M*m* card. See page 3–109.
d	= absent if the cell is a void.
	= cell material density. A positive entry is interpreted as the atomic density in units of 10^{24} atoms/cm^3. A negative entry is interpreted as the mass density in units of g/cm^3.
geom	= specification of the geometry of the cell. It consists of signed surface numbers and Boolean operators that specify how the regions bounded by the surfaces are to be combined.
params	= optional specification of cell parameters by entries in the keyword = value form.
n	= name of another cell
list	= set of keyword = value specifications that define the attributes that differ between cell n and j.

In the geometry specification, a signed surface number stands for the region on the side of the surface where points have the indicated sense. The plus sign for positive sense is optional. The regions are combined by Boolean operators: intersection (no symbol—implicit, like multiplication in algebra); union, :; and complement, #. Parentheses can be used to control the order of the operations. Parentheses and operator symbols also function as delimiters. Where they are present, blank delimiters are not necessary. The default order of operations is complement first, intersection second, and union last. A number immediately after a complement operator, without parentheses, is interpreted as a cell number and is shorthand for the geometry specification of that cell number.

> Example: 3   0   -1   2   -4 $ definition of cell 3
>   #3 $ equivalent to next line
>   #(-1 2   -4)

For a simple cell (no union or complement operators), the geometry specification is just a blank-delimited list of the bounding surfaces and ambiguity surfaces of the cell with signs determined by the sense of the cell with respect to each surface. See the Geometry sections of Chapters 1, 2, and 4 for complete explanations of how to specify the geometry of cells in MCNP.

Cell parameters can be defined on cell cards instead of in the data card section of the INP file. A blank is equivalent to the equal sign. If a cell parameter is entered on any cell card, a cell-parameter

card with that name can not be present, nor can the mnemonic appear on any column-format input card. Some cell parameters can be specified on cell cards and a different subset on cell-parameter or column-format cards. The form is keyword=value, where the allowed keywords are IMP, VOL, PWT, EXT, FCL, WWN, DXC, NONU, PD, and TMP, with particle designators where necessary. Four cell parameter cards associated with the repeated structures capability are U, TRCL, LAT and FILL. Like any cell parameter card, these four cards can be placed in the data card section of the INP file. Our recommendation is that the mnemonic and entry for each cell be placed on the cell card line after the cell description. The entries on the TRCL card and the FILL card, in particular, can be quite long and involved and it seems to be conceptually simpler when they are placed on the cell card line.

The LIKE n BUT feature uses keywords for the cell material number and density. The mnemonics are MAT and RHO, respectively. These keywords only can be used following the LIKE n BUT construct. In a normal cell description, material number and density are still the second and third entries on the cell card.

TMP and WWN data can be entered on cell cards in two ways. The keyword=value form TMP1=value TMP2=value etc. can be used or a special syntax is available where the single keyword TMP is followed by all the temperatures of the cell in an order corresponding to the times on the THTME card. The form for the WWN card is analogous: WWN1:n=value or WWN:n followed by all the lower weight bounds for the energy intervals of the cell.

    Example: 10   16   −4.2   1   −2   3   IMP:N=4   IMP:P=8   EXT:N=−.4X

This says that cell 10 is to be filled with material 16 at a density of 4.2 g/cm^3. The cell consists of the intersections of the regions on the positive side of surface 1, the negative side of surface 2, and the positive side of surface 3. The neutron importance in cell 10 is 4 and the photon importance is 8. Neutrons in cell 10 are subject to an exponential transform in the minus X direction with stretching parameter 0.4.

Here are some precautions when you are preparing cell cards:

1.   Avoid excessively complicated cells. MCNP runs faster when the problem geometry is made up of many simpler cells rather than fewer more complicated cells.

2.   Avoid adding unneeded surfaces to the geometry description of a cell through poor use of the complement operator. The extra surfaces make the problem run slower and may destroy the necessary conditions for volume and area calculations. See page 4–15.

3.   Always use the geometry-plotting feature of MCNP to check the geometry of a problem. See Appendix B.

4.   Flood the system with particles from an outside source to find errors in the geometry. See page 3–8.

5. If you add or remove cells, change all your cell parameter cards accordingly. The difficulty of this can be reduced if you use vertical format for your cell parameter cards. See page 3–6. Alternatively, define the values of cell parameters on cell cards and eliminate cell parameter cards entirely.

A.   Shorthand Cell Specification

The LIKE n BUT feature is very useful in problems with a lot of repeated structures. Cell j inherits from cell n the values of all attributes that are not specified in the list. The cell card for cell n must be before the cell card for cell j in the INP file. Any card name that appears after the BUT is a cell parameter on a cell card and, therefore, must appear on cell cards only, not on any cards in the data block of the INP file.

    Example: 2   3 −3.7   −1   IMP:N=2   IMP:P=4
                 3   LIKE 2 BUT   TRCL=1   IMP:N=10

This says that cell 3 is the same as cell 2 in every respect except that cell 3 has a different location (TRCL=1) and a different neutron importance. The material in cell 3, the density and the definition are the same as cell 2 and the photon importance is the same.

III.   SURFACE CARDS

A.   Surfaces Defined by Equations

    Form: j   n   a   list
                j   =   surface number: $1 \le j \le 99999$, with asterisk for a reflecting surface or plus for a white boundary.
                       If surface defines a cell that is transformed with TRCL, $1 \le j \le 999$. See page 3–27.
               n   =   absent or 0 for no coordinate transformation.
                    =   *> 0*, specifies number of a TRn card.
                    =   *< 0*, specifies surface *j* is periodic with surface *n*.
               a   =   equation mnemonic from Table 3.1
           list   =   one to ten entries, as required.

The surface types, equations, mnemonics, and the order of the card entries are given in Table 3.1. To specify a surface by this method, find the surface in Table 3.1 and determine the coefficients for the equation (you may need to consult a book on analytical geometry). The information is entered on a surface card according to the above form. Under certain conditions a surface can be defined by specifying geometrical points, as discussed in sections B and C. Surfaces also can be produced by combinatorial–geometry–like macrobodies, described in section D.

A point (x,y,z) is defined as having positive sense with respect to a surface when the expression for that surface evaluated at (x,y,z) is positive. The expression for a surface is the left side of the equation for the surface in Table 3.1. With the sphere, cylinder, cone, and torus, this definition is identical to defining the sense to be positive outside the figure. With planes normal to axes (PX, PY, or PZ), the definition gives positive sense for points with x, y, or z values exceeding the intercept of the plane. For the P, SQ and GQ surfaces, the user supplies all of the coefficients for the expression and thus can determine the sense of the surface at will. This is different from the other cases where the sense, though arbitrary, is uniquely determined by the form of the expression. Therefore, in a surface transformation (see the TRn card on page 3–30) a PX, PY, or PZ surface will sometimes be replaced by a P surface just to prevent the sense of the surface from getting reversed.

If the surface number is preceded by an asterisk, a reflecting surface is defined. A particle track that hits a reflecting surface is reflected specularly. If the surface number is preceded by a plus, a white boundary is defined. Detectors and DXTRAN (next–event estimators) usually should not be used in problems that have reflecting surfaces or white boundaries. See page 2–92. Tallies in problems with reflecting surfaces will need to be normalized differently. See page 2–14.

A negative second entry n specifies that surface j is periodic with surface k. The following restrictions apply:

1. Surfaces j and k must be planes.
2. No surface transformation is allowed for the periodic planes.
3. The periodic cell(s) can be infinite or bounded by planes on the top and bottom that can be reflecting or white, but cannot be periodic.
4. Periodic planes can only bound other periodic planes or top and bottom planes.
5. A single zero–importance cell must be on one side of each periodic plane.
6. All periodic planes must have a common rotational vector normal to the geometry top and bottom.
7. Next–event estimators such as detectors and DXTRAN should not be used.

TABLE 3.1: MCNP Surface Cards

Mnemonic	Type	Description	Equation	Card Entires
P	Plane	General	$Ax + By + Cz - D = 0$	ABCD
PX		Normal to X–axis	$x - D = 0$	D
PY		Normal to Y–axis	$y - D = 0$	D
PZ		Normal to Z–axis	$z - D = 0$	D
SO	Sphere	Centered at Origin	$x^2 + y^2 + z^2 - R^2 = 0$	R
S		General	$(x-\bar{x})^2 + (y-\bar{y})^2 + (z-\bar{z})^2 - R^2 = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ R$
SX		Centered on X–axis	$(x-\bar{x})^2 + y^2 + z^2 - R^2 = 0$	$\bar{x}\ R$
SY		Centered on Y–axis	$x^2 + (y-\bar{y})^2 + z^2 - R^2 = 0$	$\bar{y}\ R$
SZ		Centered on Z–axis	$y^2 + y^2 + (z-\bar{z})^2 - R^2 = 0$	$\bar{z}\ R$
C/X	Cylinder	Parallel to X–axis	$(y-\bar{y})^2 + (z-\bar{z})^2 - R^2 = 0$	$\bar{y}\ \bar{z}\ R$
C/Y		Parallel to Y–axis	$(x-\bar{x})^2 + (z-\bar{z})^2 - R^2 = 0$	$\bar{x}\ \bar{z}\ R$
C/Z		Parallel to Z–axis	$(x-\bar{x})^2 + (y-\bar{y})^2 - R^2 = 0$	$\bar{x}\ \bar{y}\ R$
CX		On X–axis	$y^2 + z^2 - R^2 = 0$	R
CY		On Y–axis	$x^2 + z^2 - R^2 = 0$	R
CZ		On Z–axis	$x^2 + y^2 - R^2 = 0$	R
K/X	Cone	Parallel to X–axis	$\sqrt{(y-\bar{y})^2 + (z-\bar{z})^2} - t(x-\bar{x}) = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2 \pm 1$
K/Y		Parallel to Y–axis	$\sqrt{(x-\bar{x})^2 + (z-\bar{z})^2} - t(y-\bar{y}) = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2 \pm 1$
K/Z		Parallel to Z–axis	$\sqrt{(x-\bar{x})^2 + (y-\bar{y})^2} - t(z-\bar{z}) = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2 \pm 1$
KX		On X–axis	$\sqrt{y^2 + z^2} - t(x-\bar{x}) = 0$	$\bar{x}\ t^2 \pm 1$
KY		On Y–axis	$\sqrt{x^2 + z^2} - t(y-\bar{y}) = 0$	$\bar{y}\ t^2 \pm 1$
KZ		On Z–axis	$\sqrt{x^2 + y^2} - t(z-\bar{z}) = 0$	$\bar{z}\ t^2 \pm 1$
				± 1 used only for 1 sheet cone
SQ	Ellipsoid Hyperboloid Paraboloid	Axis not parallel to X–, Y–, or Z–axis	$A(x-\bar{x})^2 + B(y-\bar{y})^2 + C(z-\bar{z})^2 + 2D(x-\bar{x}) + 2E(y-\bar{y}) + 2F(z-\bar{z}) + G = 0$	A B C D E F G $\bar{x}\ \bar{y}\ \bar{z}$
GQ	Cylinder Cone Ellipsoid Hyperboloid Paraboloid	Axes not parallel to X–, Y–, or Z–axis	$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fzx + Gz + Hy + Jz + K = 0$	A B C D E F G H J K
TX	Elliptical or circular torus.		$(x-\bar{x})^2/B^2 + (\sqrt{(y-\bar{y})^2 + (z-\bar{z})^2} - A)^2/C^2 - 1 = 0$	$\bar{x}\ \bar{y}\ \bar{z}$ A B C
TY	Axis is Parallel to		$(y-\bar{y})^2/B^2 + (\sqrt{(x-\bar{x})^2 + (z-\bar{z})^2} - A)^2/C^2 - 1 = 0$	$\bar{x}\ \bar{y}\ \bar{z}$ A B C
TZ	X–,Y–, or Z– axis		$(z-\bar{z})^2/B^2 + (\sqrt{(x-\bar{x})^2 + (y-\bar{y})^2} - A)^2/(C^2 - 1) = 0$	$\bar{x}\ \bar{y}\ \bar{z}$ A B C
XYZP		Surfaces defined by points		See pages 3–16 and 3–18

Example 1:   j   PY   3

This describes a plane normal to the y–axis at $y = 3$ with positive sense for all points with $y > 3$.

Example 2:   j   K/Y   0   0   2   .25   1

This specifies a cone whose vertex is at $(x,y,z) = (0,0,2)$ and whose axis is parallel to the y–axis. The tangent t of the opening angle of the cone is 0.5 (note that t^2 is entered) and only the positive (right hand) sheet of the cone is used. Points outside the cone have a positive sense.

Example 3:   j   GQ   1   .25   .75   0   –.866
                              0   –12   –2   3.464   39

This is a cylinder of radius 1 cm whose axis is in a plane normal to the x–axis at $x = 6$, displaced 2 cm from the x–axis and rotated 30° about the x–axis off the y–axis toward the z–axis. The sense is positive for points outside the cylinder. Such a cylinder would be much easier to specify by first defining it in an auxiliary coordinate system where it is symmetric about a coordinate axis and then using the TRn input card (see page 3–30) to define the relation between the basic and auxiliary coordinate systems. The input would then be

$$\begin{array}{llll}
j & 7 & CX & 1 \\
\end{array}$$
$$*TR7 \quad 6 \quad 1 \quad -1.732 \quad 0 \quad 30 \quad 60$$

See Chapter 4 for additional examples of the TRn card.

The TX, TY, and TZ input cards represent elliptical tori (fourth degree surfaces) rotationally symmetric about axes parallel to the x, y, and z axes, respectively. A TY torus is illustrated in Figure 3.1a. Note that the input parameters $\bar{x}$ $\bar{y}$ $\bar{z}$ a b c specify the ellipse

$$\frac{s^2}{b^2} + \frac{(r-a)^2}{c^2} = 1$$

rotated about the s–axis in the (r,s) cylindrical coordinate system (Figure 3.1b) whose origin is at $\bar{x}$ $\bar{y}$ $\bar{z}$ in the x, y, z system. In the case of a TY torus,

$$s = (y - \bar{y})$$
$$\text{and} \qquad r = \sqrt{(x - \bar{x})^2 + (z - \bar{z})^2}$$

A torus is degenerate if $|a| < c$ where $0 < a < c$ produces the outer surface (Figure 3.1c), and $-c < a < 0$ produces the inner surface (Figure 3.1d).
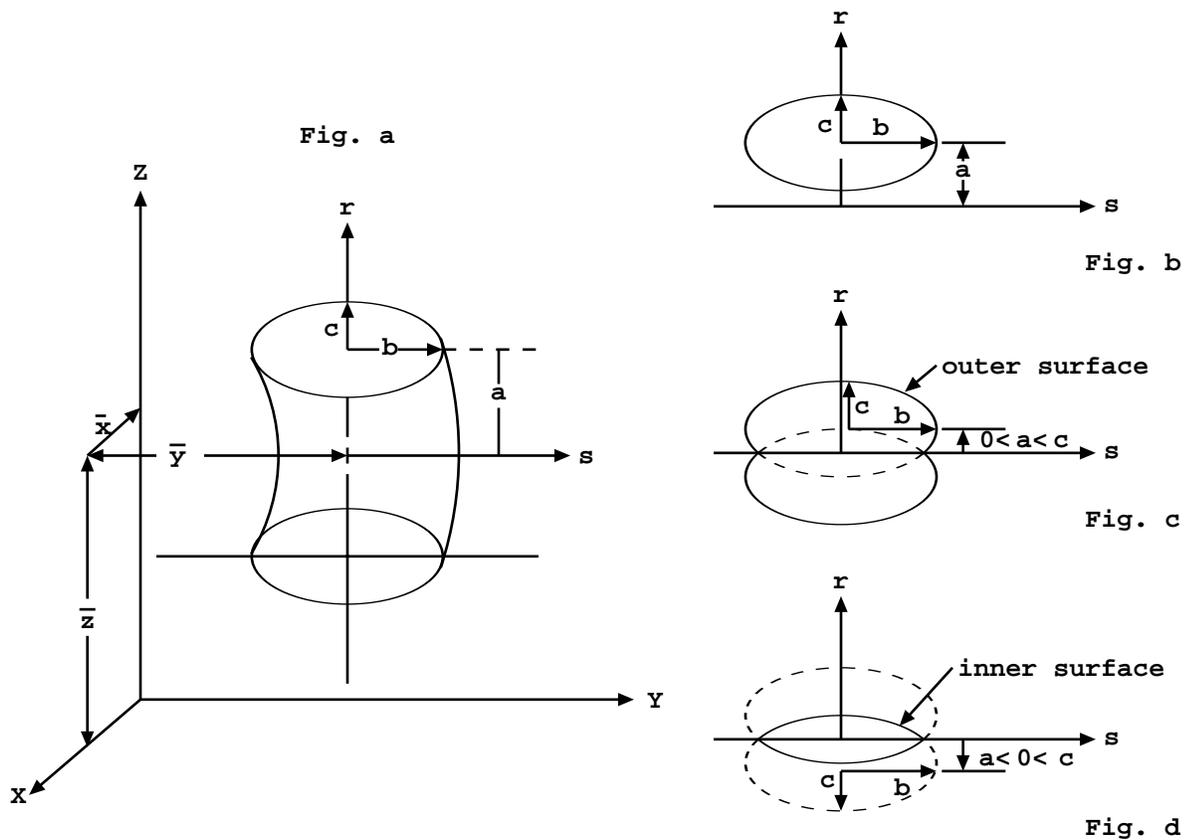
Figure 3-1. Torus

Coordinate transformations for tori are limited to those in which each axis of the auxiliary coordinate system is parallel to an axis of the main system.

B.   *Axisymmetric Surfaces Defined by Points*

Form: j   n   a   list

j   =   surface number: $1 \le j \le 99999$. If surface defines a cell that is transformed with TRCL, $1 \le j \le 999$. See page 3–27.

n   =   absent for no coordinate transformation, or number of TRn card.

a   =   the letter X, Y, or Z

list   =   one to three coordinate pairs.

Surface cards of type X, Y, and Z can be used to describe surfaces by coordinate points rather than by equation coefficients as in the previous section. The surfaces described by these cards must be

symmetric about the x–, y–, or z–axis, respectively, and, if the surface consists of more than one sheet, the specified coordinate points must all be on the same sheet.

Each of the coordinate pairs defines a geometrical point on the surface. On the Y card, for example, the entries may be

$$j \quad Y \quad y_1 \ r_1 \quad y_2 \ r_2$$

where $r_i = \sqrt{(x_i^2 + z_i^2)}$ and y_i is the coordinate of point i.

If one coordinate pair is used, a plane (PX, PY, or PZ) is defined.

If two coordinate pairs are used, a linear surface (PX, PY, PZ, CX, CY, CZ, KX, KY, or KZ) is defined.

If three coordinate pairs are used, a quadratic surface (PX, PY, PZ, SO, SX, SY, SZ, CX, CY, CZ, KX, KY, KZ, or SQ) is defined.

When a cone is specified by two points, a cone of only one sheet is generated.

The senses of these surfaces (except SQ) are determined by the code to be identical to the senses one would obtain by specifying the surface by equations. For SQ, the sense is defined so that points sufficiently far from the axis of symmetry have positive sense. Note that this is different from the equation-defined SQ, where the user could choose the sense freely.

Example 1: $j \quad$ X $\quad$ 7 5 $\quad$ 3 2 $\quad$ 4 3

This describes a surface symmetric about the x–axis, which passes through the three (x,r) points (7,5), (3,2), and (4,3). This surface is a hyperboloid of two sheets, converted in MCNP to its equivalent

$$j \quad SQ \quad -.083333333 \ \ 1 \ \ 1 \ 0 \ 0 \ 0 \ \ 68.52083 \ \ -26.5 \ 0 \ 0.$$

Example 2: $j \quad$ Y $\quad$ 1 2 $\quad$ 1 3 $\quad$ 3 4

This describes two parallel planes at $Y = 1$ and $Y = 3$ and is a fatal error because the requirement that all points be on the same sheet is not met.

Example 3: $j \quad$ Y $\quad$ 3 0 $\quad$ 4 1 $\quad$ 5 0

This describes a sphere of radius 1 with center at $(x,y,z) = (0,4,0)$.

Example 4: j   Z   1 0   2 1   3 4

This surface is rejected because the points are on two different sheets of the hyperboloid

$$x^2 + y^2 - 7z^2 + 20z - 13 = 0$$

However, the surface

j   Z   2 1   3 4   5 9.380832

which has the same surface equation as above is accepted because all coordinates lie on a single surface, the right sheet of the hyperboloid.

Example 5:   1 0   1   −2   3 $ cell 1

1   Y   −3 2   2 1
2   Y   2 3   3 3   4 2
3   Y   2 1   3 1   4 2

This final example defines a cell bounded by a cone, hyperboloid, and an ellipsoid. The three surfaces define the donut-like cell that is symmetric about the y–axis. A cross section of this cell is seen in Figure 3.2. To plot this view, type PX = 0 EX = 5. One surface goes through the points (-3,2) and (2,1). The second surface goes through (2,3), (3,3), and (4,2). The last surface is defined by the points (2,1), (3,1), and (4,2). These coordinate points are in the form *(y,r)*. Using these cards, MCNP indicates that surface 1 is a cone of one sheet, surface 2 is an ellipsoid, and surface 3 is a hyperboloid of one sheet. The equation coefficients for the standard surface equations are printed out for the various surfaces when the PRINT input card or execution option is used. For example, an SQ card defining surface 3 is

3   SQ 1 -1.5 1 0 0 0 −.625 0 2.5 0
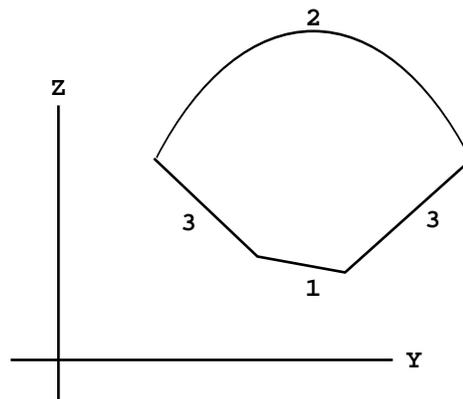
Figure 3-2.

C.   *General Plane Defined by Three Points*

Form: j   n   P $X_1\,Y_1\,Z_1$ $X_2\,Y_2\,Z_2$ $X_3\,Y_3\,Z_3$

j   =   surface number: $1 \leq j \leq 99999$ or ≤ 999 if repeated structure.

n   =   absent or 0 for no coordinate transformation.

= $> 0,$ specifies number of a TRn card.

= < 0, specifies surface j is periodic with surface n.

(X_i,Y_i,Z_i) =   coordinates of points to define the plane.

If there are four entries on a P card, they are assumed to be the general plane equation coefficients as in Table 3.1. If there are more than four entries, they give the coordinates of three points lying in the desired plane. The code converts them to the required surface coefficients to produce the plane

$$Ax + By + Cz - D \;=\; 0$$

The sense of the plane is determined by requiring the origin to have negative sense. If the plane passes through the origin $(D = 0)$, the point $(0, 0, \infty)$ has positive sense. If this fails $(D = C = 0)$, the point $(0, \infty, 0)$ has positive sense. If this fails $(D = C = B = 0)$, the point $(\infty, 0, 0)$ has positive sense. If this fails, the three points lie in a line and a fatal error is issued.

D.   *Surfaces Defined by Macrobodies*

Using a combinatorial–geometry–like macrobody capability is an alternative method of defining cells and surfaces. The combinatorial geometry bodies available are similar to those in the Integrated Tiger Series (ACCEPT) codes. The macrobodies can be mixed with the standard cells and surfaces. The macrobody surface is decomposed internally into surface equations and the facets are assigned individual numbers according to a predetermined sequence. The assigned numbers are the number selected by the user followed by a decimal point and 1, 2, ... The facets can be used for tallying, tally segmentation, other cell definitions, SDEF sources, etc. They cannot be used on the SSR/SSW cards, the surface flagging card, PTRAC, or MCTAL files.

The space inside a body has a negative sense with respect to the macrobody surface and all its facets. The space outside a body has a positive sense. The sense of a facet is the sense assigned to it by the macrobody "master" cell and the facet retains that assigned sense if it appears in other cell descriptions and must be properly annotated. See an example at the end of this section for an illustration.

The following geometry bodies are available and their complete descriptions follow.

BOX   Arbitrarily oriented orthogonal box
RPP   Rectangular ParallelePiped

CHAPTER 3
SURFACE CARDS

      SPH   Sphere
      RCC   Right Circular Cylinder
      RHP or HEX   Right Hexagonal Prism

BOX: Arbitrarily oriented orthogonal box (all corners are 90˚.)
      BOX   Vx Vy Vz A1x A1y A1z   A2x A2y A2z   A3x A3y A3z
            where Vx Vy Vz = x,y,z coordinates of corner
                  A1x A1y A1z = vector of 1st side
                  A2x A2y A2z = vector of 2nd side
                  A2x A3y A3z = vector of 3rd side

      Example: BOX –1 –1 –1 2 0 0 0 2 0 0 0 2
            a cube centered at the origin, 2 cm on a side, sides parallel to the major axes.

RPP: Rectangular ParallelePiped, surfaces normal to major axes, x,y,z values relative to origin.
      RPP   Xmin Xmax Ymin Ymax   Zmin Zmax

      Example: RPP   –1 1 –1 1 –1 1
            equivalent to BOX above.

SPH: Sphere. Equivalent to surface equation for general sphere.
      SPH   Vx Vy Vz R
            where Vx Vy Vz = x,y,z coordinates of center
                  R = radius

RCC: Right Circular Cylinder, can
      RCC   Vx Vy Vz Hx Hy Hz R
            where Vx Vy Vz = center of base
                  Hx Hy Hz = cylinder axis vector
                  R = radius

      Example: RCC 0 –5 0   0 10 0   4
            a 10-cm high can about the y-axis, base plane at y=–5 with radius of 4 cm.

RHP or HEX: Right Hexagonal Prism. Differs from ITS (ACCEPT) format.
      RHP   v1 v2 v3 h2 h2 h3 r1 r2 r3 s1 s2 s3 t1 t2 t3
            where v1 v2 v3 = x,y,z coordinates of the bottom of the hex
                  h1 h2 h3 = vector from the bottom to the top
                        for a z-hex with height h, h1,h2,h3 = 0 0 h
                  r1 r2 r3 = vector from the axis to the middle of the first facet
                        for a pitch 2p facet normal to y-axis, r1,r2,r3 = 0 p 0
                  s1 s2 s3 = vector to center of the 2nd facet

t1 t2 t3 = vector to center of the 3rd facet

Example: RHP   0 0 –4   0 0 8   0 2 0
           a hexagonal prism about the z-axis whose base plane is at z=–4 with a height
           of 8-cm and whose first facet is normal to the y-axis at y=2.

The facets of the bodies are sequentially numbered and can be used on other MCNP cards. BOX and RPP can be infinite in a dimension, in which case those two facets are skipped and the numbers of the remaining facets are decreased by two. RHP can be infinite in the axial dimension in which case facets 7 and 8 do not exist. The order of the facet numbering follows for each geometry body. Facet numbering can be displayed graphically with MBODY=OFF in the geometry plotter.

BOX: 1   plane normal to end of A1x A1y A1z
            2   plane normal to beginning of A1x A1y A1z
            3   plane normal to end of A2x A2y A2z
            4   plane normal to beginning of A2x A2y A2z
            5   plane normal to end of A3x A3y A3z
            6   plane normal to beginning of A3x A3y A3z

RPP: 1   Plane Xmax
            2   Plane Xmin
            3   Plane Ymax
            4   Plane Ymin
            5   Plane Zmax
            6   Plane Zmin

SPH: treated as a regular surface so no facet

RCC: 1   Cylindrical surface of radius R
            2   Plane normal to end of Hx Hy Hz
            3   Plane normal to beginning of Hx Hy Hz

RHP or HEX 1   Plane normal to end of r1 r2 r3
            2   Plane opposite facet 1
            3   Plane normal to end of s1 s2 s3
            4   Plane opposite facet 3
            5   Plane normal to end of t1 t2 t3
            6   Plane opposite facet 5
            7   Plane normal to end of h1 h2 h3
            8   Plane normal to beginning of h1 h2 h3

The following input file describes five cells and illustrates a combination of the various body and cell/surface descriptions. Surface numbers are in italics alongside the planes they define. Note that the cell and surface numbers do not have to start with 1 or be consecutive.

```
3  0   –1.2  –1.1  1.4  –1.5  –1.6  99
4  0    1.1  –2001.1  –5.3  –5.5  –5.6  –5.4
5  0   –5
1   0  –1
2  like 1 but trcl = (2 0 0)
9  0    (–5.1 : 1.3 : 2001.1 : –99 : 5.5 : 5.6) #5


5  rpp   –2  0  –2  0  –1  1
1  rpp    0  2   0  2  –1  1
99  py    –2
```

cell 9

1.3

5.1   cell 1   *1.1* cell 2

2001.1

5.3

5.2   cell 5   cell 3   cell 4

99
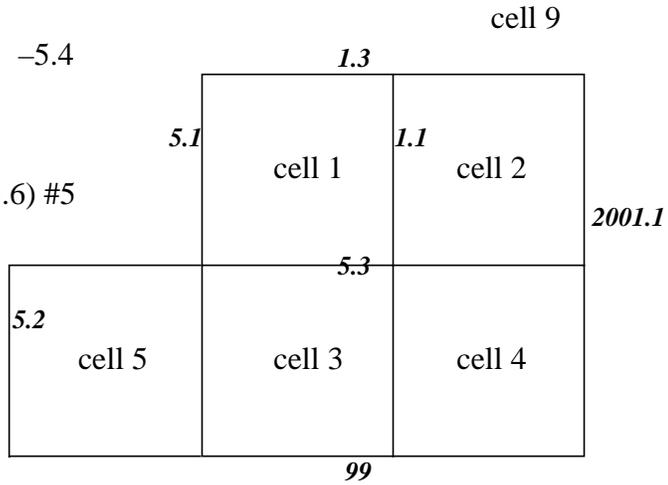
alternative descriptions of cell 3:

```
3  0     5.1  –1.1  –5.3  –5.5  –5.6  99
3  0     5.1  –1.1   1.4  –5.5  –5.6  –5.4
3  0   –1.2  –1.1  –5.3  –5.5  –5.6  –5.4
```

y

x
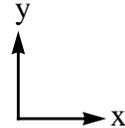
IV. DATA CARDS

All MCNP input cards other than those for cells and surfaces are entered after the blank card delimiter following the surface card block. The mnemonic must begin within the first five columns. These cards fall into the following categories:

Category	Page
(A) Problem type	3–23
(B) Geometry cards	3–23
(C) Variance reduction	3–32
(D) Source specification	3–49
(E) Tally specification	3–73
(F) Material and cross section specification	3–108
(G) Energy and thermal treatment	3–117
(H) Problem cutoffs	3–124
(I) User data arrays	3–127
(J) Peripheral cards	3–128

These card categories are described below. Only the cards listed on page 3–3 are allowed in a continue-run input file. No data card can be used more than once with the same number or particle type designations. For example, M1 and M2 are acceptable, as are CUT:N and CUT:P, but two M1 cards or two CUT:N cards are disallowed.

A.   *Problem Type (MODE) Card*

Form: MODE $x_1 \ldots x_i$

x_i = N for neutron transport
          P for photon transport
          E for electron transport

Default: If the MODE card is omitted, MODE N is assumed.

Use: A MODE card is required unless MODE=N. The entries are space delineated.

B.   *Geometry Cards*

Mnemonic	Card Type	Page
VOL	Cell volumes	3–23
AREA	Surface areas	3–24
U	Universes	3–26
TRCL	Cell transformations	3–27
LAT	Lattices	3–28
FILL	Fill card	3–29
TR	Coordinate transformation	3–30

1.   VOL   Cell Volume Card

Form: VOL $x_1 \; x_2 \; \ldots \; x_i$
  or: VOL   NO $x_1 \; x_2 \; \ldots \; x_i$

x_i = volume of cell *i* where *i*=1, 2, ... number of cells in the problem.
NO   = no volumes or areas are calculated.

Default: MCNP attempts to calculate the volume of all cells unless "NO" appears on the VOL card. If no value is entered for a cell on the VOL card, the calculated volume is used.

Use: Optional card used to input cell volumes.

With the VOL card, if the number of entries does not equal the number of cells in the problem, it is a fatal error. Use the nJ feature to skip over cells for which you do not want to enter values. The entry NO on the VOL card will bypass the volume calculation altogether. The x_i entries following NO are optional. If present, x_i entries are the volume values the code will use. For some problems the NO option saves considerable computer time.

Volumes or masses of cells are required for some tallies. MCNP calculates the volumes of all cells that are rotationally symmetric (generated by surfaces of revolution) about any axis, even a skew axis. It will also calculate the volumes of polyhedral cells. As a byproduct of the volume calculation, areas and masses are also calculated. These volumes, areas, and masses can be printed in the OUTP file by using the PRINT card. The user can enter values on the VOL card for the volume of any cell and these values, instead of the calculated values, will be used for tally purposes. If a cell volume required for a tally cannot be calculated and is not entered on the VOL or SDn cards, a fatal error message is printed.

The VOL card provides an alternative way to enter volumes required by tallies. Normally the SDn card would be used. The VOL card can be used only for cell volumes, whereas the SDn card can be used for cell and segment volumes or masses.

Volumes of cells or segments that cannot be calculated by MCNP or by the user can be obtained in a separate MCNP run using the ray-tracing technique described on page 2–183.

2.   AREA   Surface Area Card

> Form: AREA x_1 … x_i … x_n
>
> > x_i = area of surface i where i=1, 2, ... number of surfaces in the problem.

Default: MCNP attempts to calculate the area of all surfaces. If no value is entered for a surface on the AREA card, the calculated area, if any, is used.

Use: Optional card used to input surface areas.

This card is analogous to the VOL card. MCNP calculates the area of surfaces as a byproduct of the volume calculation. If the volume of all cells on either side of the surface can be calculated, the area of the surface will be calculated. Otherwise the area calculation will fail. A fatal error occurs if an area is required for tallying purposes and is not available either from the MCNP calculation or from an AREA or SDn card.

The AREA card provides an alternative way to enter areas required by tallies. Normally the SDn card would be used. The AREA card can be used only for areas of whole surfaces, whereas the SDn card can be used for areas of surface segments as well as whole surfaces.

3–6.   Repeated Structures Cards

The primary goal of the repeated-structures capability is to make it possible to describe only once the cells and surfaces of any structure that appears more than once in a geometry. The amount of input data the user has to provide and the amount of computer memory needed by problems that have a lot of geometrical repetition is reduced. Problems that would be impractical because they take an unreasonable amount of work to set up or they use too much memory can be run. One example of such a problem is a reactor core that has dozens of nearly identical fuel modules. Another example is a room containing some complicated nearly identical objects arranged in some not necessarily regular order. This feature reduces input and memory use but problems won't run any faster than with any other description. Examples of the use of repeated structures cards are in Chapter 4.

The repeated structures capability extends the concept of an MCNP cell. The user can specify that a cell is to be filled with something called a universe. A universe is either a lattice or an arbitrary collection of cells. A single universe, described only once, can be designated to fill each of any number of cells in the geometry. Some or all of the cells in a universe may themselves be filled with universes. Several concepts and cards combine in order to use this capability.

- Remember that cell parameters can be defined on cell cards.

- The "*LIKE m BUT*" feature is a shorthand making it possible to make one cell equivalent to another except for assorted attributes that can be specified with keyword=value entries. See page 3–11.

- The *universe* card, the U card, is used to specify to what universe the cell belongs.

- The *fill* card is used to specify with which universe a cell is to be filled.

- The *TRCL* card makes it possible to define only once the surfaces that bound several cells identical in size and shape but located at different places in the geometry. It follows the transformation rules established for the TR card. See page 3–30.

- The *lattice* card, the LAT card, is used to define an infinite array of hexahedra or hexagonal prisms. The order of specification of the surfaces of a lattice cell identifies which lattice element lies beyond each surface.

- A general source description can be defined in a repeated structures part of the geometry. Surface source surfaces must be regular MCNP surfaces, not surfaces associated with a repeated structures part of the geometry. No check is made that this requirement is met. The user must remember and this notification is your only warning.

- An importance in a cell that is in a universe is interpreted as a multiplier of the importance of the filled cell. Weight–window lower bounds are not multipliers. Mesh-based weight windows (MESH card) automatically address this problem.

CHAPTER 3
Geometry Cards

Chapter 4 contains several examples that illustrate the repeated structures input and logic. The reader is strongly encouraged to become familiar with these examples and to use them as teaching aids to help understand the card descriptions that follow.

3.   U   Universe Card

As mentioned earlier, a universe can be either a lattice or a collection of ordinary cells. A nonzero entry on the U card is the number of the universe that the corresponding cell belongs to. Lack of a U card or a zero entry means that the cell does not belong to any universe. Universe numbers are arbitrary integers chosen by the user. The FILL card, page 3–29, indicates that a cell is filled by all the cells having a corresponding integer entry on the U card. The cells of a universe may be finite or infinite, but they must fill all of the space inside any cell that the universe is specified to fill.

One way to think about the connection between a filled cell and the filling universe is that the filled cell is a "window" that looks into a second level, like a window in a wall provides a view of the outdoors. Cells in the second level can be infinite because they will be "ended" when they bump into or intersect the surfaces of the "window." The second level can have its own origin, in a primed coordinate system, unrelated to the upper level origin. However, if the filled cell and filling universe have all their surfaces in the same coordinate system, one TRCL card, explained on page 3–27, will define the coordinate system of both filled and filling cells. The first repeated structures example in Chapter 4 illustrates this fact.

A cell in a universe can be filled by another universe, in which case a third level is introduced. There is a maximum of 10 levels, more than most problems will need. To clarify some jargon about hierarchies, the highest to lowest level is in inverse order to the associated numerical value. The highest level is level zero, lower is level one, lower still is level two, etc.

Planar surfaces of a filled cell and those in a filling universe CAN be coincident. In other words, the cells of a universe can fit exactly into the filled cell. The following cell and surface cards illustrate this feature. They represent a $50 \times 20 \times 10$ –cm box filled with a lattice of $10 \times 10 \times 10$ – cm cubes, each of which is filled with a sphere.

A problem will run faster by preceding the U card entry with a minus sign for any cell that is not truncated by the boundary of any higher level cell. The minus sign indicates that calculating distances to boundary in higher level cells can be omitted. In the problem below, cell 3 has a negative universe number. It is a finite cell and is not truncated by any other cell. Cell 4 cannot have a negative universe number because it is an infinite region that is truncated by cell 2.

CAUTION: Use this capability AT YOUR OWN RISK. MCNP cannot detect errors in this feature as the logic that enables detection is omitted by the presence of the negative universe. Extremely wrong answers can be quietly calculated. Use this feature with EXTREME caution. Plot several views of the geometry or run with the VOID card (see page 3–8) to check for errors.

```
1   0    1 –2 –3 4 –5 6        fill=1
2   0    –7 1 –3 8      u=1   fill=2   lat=1
3   0    –11            u=–2
4   0    11             u=2
5   0    –1:2:3:–4:5:–6

1   px    0
2   px   50
3   py   10
4   py  –10
5   pz    5
6   pz   –5
7   px   10
8   py   0
10  py   10
11  s     5 5 0 4
```

Every cell in the problem is either part of the real world (universe level 0) or part of some universe, but the surfaces of a problem are less restricted. A single planar surface can be used to describe cells in more than one universe. Coincident surfaces can not be reflecting or periodic, source surfaces, or tally surfaces. Materials are normally put into the cells of the lowest level universe, not in the higher level but there is an exception in the case of a lattice.

The above example can be described with macrobodies as follows:

```
1   0    –20    fill=1
2   0    –30    u=1   fill=2   lat=1
3   0    –11    u=–2
4   0     11    u=2
5   0     20

20   rpp   0 50    –10  10    –5  5
30   rpp   0 10      0  10
11   s      5   5   0   4
```

4.   TRCL   Cell Transformation Card

The TRCL card makes it possible to describe just once the surfaces that bound several cells identical in size and shape but located at different places in the geometry. It is especially valuable when these cells are filled with the same universe. If the surfaces of these filled cells and the surfaces of the cells in the universe that fills them are all described in the same auxiliary coordinate system, a single transformation will completely define the interior of all these filled cells because the cells of the universe will inherit the transformation of the cells they fill. TRCL is intended to be

used with LIKE BUT, LAT, etc. With a regular cell description, it is suggested the TR on the surface cards be used.

The basic form of an entry is an integer that is interpreted as the number of a TR card that contains a transformation for all of the surfaces of the cell and is located in the data card section of the INP file. The absence of the TRCL card or zero means there is no transformation, the default. The actual transformation can be entered following the TRCL mnemonic, enclosed by parentheses. If the actual transformation is entered, all the rules applying to the TR card
(page 3–30) are valid. If the symbol *TRCL is used, the rotation matrix entries are angles in degrees instead of cosines, the same as the *TR card.

If a cell has a transformation, a set of new surfaces with unique names is generated from the original surfaces. The name of the generated surface is equal to the name of the original surface plus 1000 times the name of the cell. This formula gives generated names that are predictable and can be used on other cell cards and on tally cards. This method limits cell names and original surface names to no more than three digits, however. These generated surfaces are only the bounding surfaces of the transformed cell, not the surfaces of any universe that fills it. MCNP requires only one full description of each universe, no matter how many times that universe is referenced in the problem.

5.   LAT   Lattice Card

LAT=1 means the lattice is made of hexahedra, solids with six faces. LAT=2 means the lattice is made of hexagonal prisms, solids with eight faces. A nonzero entry on the LAT card means that the corresponding cell is the (0,0,0) element of a lattice. The cell description of a lattice cell has two main purposes. It is a standard MCNP cell description and the order of specification of the surfaces of the cell identifies which lattice element lies beyond each surface.

After you have designed your lattice, decide which element you want to be the (0,0,0) element and in which directions in the lattice you want the three lattice indices to increase. In the case of a hexagonal prism lattice you have two constraints: the first and second indices must increase across adjacent surfaces and the third index must increase in one or the other direction along the length of the prism. You will then enter the bounding surfaces of the (0,0,0) element on the cell card in the right order, in accordance with the following conventions. For a hexahedral lattice cell, beyond the first surface listed is the (1,0,0) element, beyond the second surface listed is the (-1,0,0) element, then the (0,1,0), (0,-1,0), (0,0,1) and (0,0,-1) lattice elements in that order. This method provides the order of arrangement of the lattice to the code so that when you specify element (7,9,3), the code knows which one that is. For a hexagonal prism lattice cell, on the opposite side of the first surface listed is element (1,0,0), opposite the second listed surface is
(-1,0,0), then (0,1,0), (0,-1,0), (-1,1,0), (1,-1,0), (0,0,1), and (0,0,-1). These last two surfaces must be the base surfaces of the prism. Example 7, page 4–34, illustrates a hexagonal prism lattice cell.

The hexahedra need not be rectangular and the hexagonal prisms need not be regular, but the lattices made out of them must fill space exactly. This means that opposite sides have to be identical and parallel. A hexahedral lattice cell may be infinite in one or two of its dimensions. A hexagonal prism lattice cell may be infinite in the direction along the length of the prism. The cross section must be convex (no butterflies). It does not matter whether the lattice is left-handed or right-handed. A lattice must be the only thing in its universe. The real world (universe level 0) itself can be a lattice. If a particle leaves the last cell of a real-world, limited-extent lattice (see the FILL card for how the extent of a lattice can be limited), it is killed (escapes).

6.   FILL   Fill Card

A nonzero entry on the FILL card indicates the number of the universe that fills the corresponding cell. The same number on the U card identifies the cells making up the filling universe. The FILL entry may optionally be followed by, in parentheses, either a transformation number or the transformation itself. This transformation is between the coordinate systems of the filled cell and the filling universe, with the universe considered to be in the auxiliary coordinate system. If no transformation is specified, the universe inherits the transformation, if any, of the filled cell. A ∗FILL may be used if the rotation matrix entries are angles in degrees rather than cosines. In the data card section of the INP file you cannot have both a FILL and a ∗FILL entry. If you want to enter some angles by degrees (∗FILL) and some angles by cosines (FILL), all FILL and ∗FILL data must be placed on the cell cards of the INP file.

If the filled cell is a lattice, the FILL specification can be either a single entry, as described above, or an array. If it is a single entry, every cell of the lattice is filled by the same universe. If it is an array, the portion of the lattice covered by the array is filled and the rest of the lattice does not exist. It is possible to fill various elements of the lattice with different universes, as shown below and in examples in Chapter 4, section III,

The array specification for a cell filled by a lattice has three dimension declarators followed by the array values themselves. The dimension declarators define the ranges of the three lattice indices. They are in the same form as in FORTRAN, but both lower and upper bounds must be explicitly stated with positive, negative, or zero integers, separated by a colon. The indices of each lattice element are determined by its location with respect to the (0,0,0) element. Reread the LAT card section, if needed, with particular emphasis on how the order of specification of the surfaces of the cell identifies the ordering of the lattice elements. The first two surfaces listed on the cell card define the direction the first lattice index must cover. The numerical range of the indices depends on where in the lattice the (0,0,0) element is located. For example, −5:5, 0:10, and −10:0 all define a range of 11 elements. The third and fourth surfaces listed in the cell description define the direction of the second lattice index.

The array values follow the dimension declarators. Each element in the array corresponds to an element in the lattice. Only those elements of the lattice that correspond to elements in the array

actually exist. The value of each array element is the number of the universe that is to fill the corresponding lattice. There are two values that can be used in the array that have special meanings. A zero in a real world (level zero) lattice means that the lattice element does not exist, making it possible, in effect, to specify a nonrectangular array. If the array value is the same as the number of the universe of the lattice, that element is not filled with any universe but with the material specified on the cell card for the lattice cell. A real world (level zero) lattice, by default, is universe zero and only can be universe zero. Therefore, using the universe number of the lattice as an array value to fill that element with the cell material is not possible. As with a single entry FILL specification, any value in the array optionally can be followed by, in parentheses, a transformation number or the transformation itself.

Example: FILL=0:2 1:2 0:1 4 4 2 0 4 0 4 3 3 0 4 0

Only eight elements of this lattice exist. Elements (0,1,0), (1,1,0), (1,2,0), (0,1,1) and (1,2,1) are filled with universe 4. Element (2,1,0) is filled with universe 2. Elements (1,1,1) and (2,1,1) are filled with universe 3.

7.   TRn   Coordinate Transformation Card

Form: TRn O_1 O_2 O_3 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 M

n	= number of the transformation: $1 < n < 999$. *TRn means that the B_i are angles in degrees rather than being the cosines of the angles.
O_1 O_2 O_3	= displacement vector of the transformation.
B_1 to B_9	= rotation matrix of the transformation.
M	= 1 (the default) means that the displacement vector is the location of the origin of the auxiliary coordinate system, defined in the main system.
	= -1 means that the displacement vector is the location of the origin of the main coordinate system, defined in the auxiliary system.

Default: TRn   0 0 0   1 0 0   0 1 0   0 0 1   1

The maximum number of transformations in a single problem is 999. A cone of one sheet can be rotated only from being on or parallel to one coordinate axis to being on or parallel to another coordinate axis (multiples of 90°). A cone of one sheet can have any origin displacement vector appropriate to the problem. A cone of two sheets can be transformed anywhere. A cone of two sheets with an ambiguity surface in the cell description to cut off one half (the cell looks like one sheet) can be transformed. The ambiguity surface must have the same transformation number as the cone of two sheets. Ambiguity surfaces are described on page 2–12.

The *B* matrix specifies the relationship between the directions of the axes of the two coordinate systems. B_i is the cosine of the angle (or the angle itself, in degrees in the range from 0 to 180, if the optional asterisk is used) between an axis of the main coordinate system *(x,y,z)* and an axis of the auxiliary coordinate system *x'y'z'* as follows:

Element	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
Axes	x,x'	y,x'	z,x'	x,y'	y,y'	z,y'	x,z'	y,z'	z,z'

The meanings of the B_i do not depend on M. It is usually not necessary to enter all of the elements of the *B* matrix. These patterns are acceptable:

1. All nine elements.

2. Two of the three vectors either way in the matrix (6 values). MCNP will create the third vector by cross product.

3. One vector each way in the matrix (5 values). The component in common must be less than 1. MCNP will fill out the matrix by the Eulerian angles scheme.

4. One vector (3 values). MCNP will create the other two vectors in some arbitrary way.

5. None. MCNP will create the identity matrix.

A vector consists of the three elements in either a row or a column in the matrix. In all cases MCNP cleans up any small nonorthogonality and normalizes the matrix. In this process, exact vectors like (1,0,0) are left unchanged. A warning message is issued if the nonorthogonality is more than about 0.001 radian.

Pattern #5 is appropriate when the transformation is a pure translation. Pattern #4 is appropriate when the auxiliary coordinate system is being used to describe a set of surfaces that are all surfaces of rotation about a common skew axis. Patterns 2 and 3 are about equally useful in more general cases. Pattern #1 is required if one of the systems is right handed and the other is left handed.

Coordinate transformations in MCNP are used to simplify the geometrical description of surfaces and to relate the coordinate system of a surface source problem to the coordinate system of the problem that wrote the surface source file. See the surface source SSR card on page 3–66. Periodic boundary surfaces cannot have surface transformations.

To use a transformation to simplify the description of a surface, choose an auxiliary coordinate system in which the description of the surface is easy, include a transformation number n on the surface card, and specify the transformation on a TRn card. See page 4–16 for an example showing how much easier it is to specify a skewed cylinder this way than as a GQ surface. Often a whole cluster of cells will have a common natural coordinate system. All of their surfaces can be described in that system, which can then be specified by a single TRn card.

Example: 17   4   PX   5
            TR4   7   .9   1.3   0   −1   0   0   0   1   −1   0   0

Surface 17 is set up in an auxiliary coordinate system that is related to the main coordinate system by transformation number 4. (Presumably there are many other surfaces in this problem that are using the same transformation, probably because they came from the input file of an earlier problem. Otherwise there would be no reason to use a transformation to set up a surface as simple as a plane perpendicular to a coordinate axis.) MCNP will produce coefficients in the main coordinate system as if surface 17 had been entered as

    17   P   0   −1   0   4.1

It will not produce

    17   PY   4.1

that is located at the same place in space, because this PY surface has the wrong sense. More examples of the transformation are in Chapter 4.

C.   *Variance Reduction*

The following cards define parameters for variance reduction cards.

Mnemonic	Card Type	Page
IMP	Cell importances	3–33
ESPLT	Energy splitting and roulette	3–34
PWT	Photon production weights	3–35
EXT	Exponential transform	3–36
VECT	Vector input	3–38
FCL	Forced collision	3–38
WWE	Weight window energies	3–40
WWN	Weight window bounds	3–40
WWP	Weight window parameter	3–41
WWG	Weight window generation	3–43
WWGE	Weight window generation energies	3–44
MESH	Superimposed importance mesh for mesh-based weight window generator	3–44
PD	Detector contribution to tally	3–47
DXC	DXTRAN cell contributions	3–48
BBREM	Bremsstrahlung biasing	3–48

Either an IMP or WWN card is required; most of the other cards are for optional variance reduction techniques.

Entries on a cell or surface parameter card correspond in order to the cell or surface cards that appear earlier in the INP file. To get to the particular cell(s) or surface(s) on a card, you must supply the appropriate default values on the cards as spacers (the nR repeat or nJ jump features may help). The number of entries on a cell or surface parameter card should always equal the number of cells or surfaces in the problem or a FATAL error will result.

Many of these cards require a knowledge of both the Monte Carlo method and the particular variance reduction technique being used. Chapter 2 and some of the references listed at the end of the manual may provide some of this knowledge.

1.   IMP   Cell Importance Cards

     Form: IMP:n x_1 x_2 ... x_i ... x_I

       n   = *N* for neutrons, *P* for photons, *E* for electrons. *N,P* or *P,E* or *N,P,E*
                  is allowed if importances are the same for different particle types.
      x_i = importance for cell *i*
      I   = number of cells in the problem

     Default: If an IMP:P card is omitted in a MODE N P problem, all photon cell importances are set to unity unless the neutron importance is 0. Then the photon importance is 0 also.

     Use: An IMP:n card is required with an entry for every cell unless a WWN weight window bound card is used.

The importance of a cell is used to terminate the particle's history if the importance is zero, for geometry splitting and Russian roulette as described on page 2–135 to help particles move to more important regions of the geometry, and in the weight cutoff game described on page 3–125. An importance in a cell that is in a universe is interpreted as a multiplier of the importance of the filled cell.

Neutrons, photons, and electrons can be split differently by having separate IMP:N, IMP:P, and IMP:E cards. It is a fatal error if the number of entries on any IMP:n card is not equal to the number of cells in the problem. The nJ feature is allowed and provides the default importance of zero. The nR repeat and nM multiply features are especially useful with this card.

     Example: IMP:N   1   2   2M   0   1   20R

The neutron importance of cell 1 is 1, cell 2 is 2, cell 3 is 4, cell 4 is 0, and cells 5 through 25 is 1. A track will be split 2 for 1 going from cell 2 into cell 3, each new track having half the weight of the original track before splitting. A track moving in the opposite direction will be terminated in about half (that is, probability=0.5) the cases but followed in the remaining cases with twice the weight.

Remember that both tracks and contributions to detectors and DXTRAN spheres are killed in cells of zero importance.

A track will neither be split nor rouletted when it enters a void cell even if the importance ratio of the adjacent cells would normally call for a split or roulette. However, the importance of the nonvoid cell it left is remembered and splitting or Russian roulette will be played when the particle next enters a nonvoid cell. As an example of the benefit of not splitting into a void, consider a long, void pipe surrounded by a material like concrete where the importances are decreasing radially away from the pipe. Considerable computer time can be wasted by tracks bouncing back and forth across the pipe and doing nothing but splitting then immediately undergoing roulette. Splitting into a void increases the time per history but has no counterbalancing effect on the expected history variance. Thus the *FOM* is reduced by the increased time per history.

If a superimposed weight window mesh is used, the IMP card is required but splitting/Russian roulette is not done at surfaces. Cell importances are only used for the weight cutoff game in zero–window meshes.

2. ESPLT   Energy Splitting and Roulette Card

    Form: ESPLT:n N_1 E_1 ... N_5 E_5

              n   = N for neutrons, P for photons, E for electrons.
              N_i = number of tracks into which a particle will be split.
              E_i = energy (MeV) at which particles are to undergo splitting.

    Default: Omission of this card means that energy splitting will not take place for those particles for which the card is omitted.

    Use: Optional; use energy-dependent weight windows instead.

The ESPLT card allows for splitting and Russian roulette in energy, as the IMP card allows for splitting and Russian roulette as a function of geometry. Energy splitting can result in low weight particles that are inadvertently killed by the weight cutoff game (CUT card). Because energy dependent weight windows perform the same function as the ESPLT card, are not limited to five energy groups, can have spatial dependence, and are more compatible with other variance reduction features, use of the ESPLT card is discouraged.

The entries on this card consist of pairs of energy-biasing parameters, N_i and E_i, with a maximum of five pairs allowed. N_i can be noninteger and also can be between 0 and 1, in which case Russian roulette on energy is played. For N_i between 0 and 1 the quantity becomes the survival probability in the roulette game.

If the particle's energy falls below E_i, the specified splitting or roulette always occurs. If the particle's energy increases above E_i, the inverse game is normally played. For example, suppose roulette is specified at 1 eV with survival probability 0.5; if a particle's energy increases above 1 eV, it is split 2 for 1.

A neutron's energy may increase by fission or from thermal up–scattering. There are cases when it may not be desirable to have the inverse splitting or roulette game played on energy increases (particularly in a fission-dominated problem). If $N_1 < 0$, then splitting or roulette will be played only for energy decreases and not for energy increases.

    Example: ESPLT:N   2 .1   2 .01   .25 .001

This example specifies a 2 for 1 split when the neutron energy falls below 0.1 MeV, another 2 for 1 split when the energy falls below 0.01 MeV, and Russian roulette when the energy falls below 0.001 MeV with a 25% chance of surviving.

3.   PWT   Photon Weight Card

    Form: PWT W_1 W_2 ... W_i ... W_I

    W_i = relative threshold weight of photons produced at neutron collisions in cell i
    I   = number of cells in the problem.

    Use: Recommended for MODE N P and MODE N P E problems without weight windows.

The PWT card is used in Mode N P or Mode N P E problems. Its purpose is to control the number and weight of neutron-induced photons produced at neutron collisions. Only prompt photons are produced from neutron collisions. Delayed gammas are neglected by MCNP. The PWT card application is further discussed on page 2–33.

For each cell with a positive W_i entry, only neutron-induced photons with weights greater than W_i*I_s/I_i are produced, where I_s and I_i are the neutron importances of the collision and source cells, respectively. Russian roulette is played to determine if a neutron-induced photon with a weight below this value survives.

For each cell with a negative W_i entry, only neutron-induced photons with weights greater than $-W_i*W_s*I_s/I_i$ are produced, where W_s is the starting weight of the neutron for the history being

followed, and I_s and I_i are the neutron importances of the collision and source cells, respectively. Russian roulette is played to determine if a neutron-induced photon with weight below this value survives.

If $W_i = 0$, exactly one photon will be generated at each neutron collision in cell i, provided that photon production is possible. If $W_i = -1.0E6$, photon production in cell i is turned off.

The PWT card controls the production of neutron-induced photons by comparing the total weight of photons produced with a relative threshold weight specified on the PWT card. This threshold weight is relative to the neutron cell importance and, if $W_i < 0$, to the source neutron weight. If more neutron-induced photons are desired, the absolute value of W_i should be lowered to reduce the weight and therefore increase the number of photons. If fewer neutron-induced photons are desired, the absolute value of W_i should be increased.

For problems using photon cell importances (IMP:P), rather than photon weight windows (WWNn:P), a good first guess for PWT card entries is either the default value, $W_i = -1$, or set W_i in every cell to the average source weight.

For problems with photon weight windows, the PWT card is ignored and the correct number of photons are produced to be born within the weight windows.

4.   EXT   Exponential Transform Card

> Form: EXT:n A_1 A_2 ... A_i ... A_I

>> n   = N for neutrons, P for photons, not available for electrons.
>> A_i = entry for cell i. Each entry A_i is of the form $A = QVm$, where Q describes the amount of stretching and Vm defines the stretching direction.
>> I   = number of cells in the problem.

> Default: No transform, $A_i = 0$.

> Use: Optional. Use cautiously. Weight windows strongly recommended.

The exponential transform should not be used in the same cell as forced collisions or without good weight control, such as the weight window. The transform works well only when the particle flux has an exponential distribution, such as in highly absorbing problems.

The exponential transform method stretches the path length between collisions in a preferred direction by adjusting the total cross section as follows:

$$\Sigma_t^* = \Sigma_t(1 - p\mu), \text{ where}$$

Σ_t^* = artificially adjusted total cross section;

Σ_t = true total cross section;

p = the stretching parameter; and

μ = cosine of angle between particle direction and stretching direction.

The stretching parameter p can be specified by the stretching entry Q three ways:

$Q = 0$; $p = 0$ exponential transform not used

$Q = p$; $0 < p < 1$ constant stretching parameter

$Q = S$; $p = \Sigma_a \Sigma_t$, where Σ_a is the capture cross section.

Letting $p = \Sigma_a/\Sigma_t$ can be used for implicit capture along a flight path, as described on page 2–36.

The stretching direction is defined by the *Vm* part of each A_i entry on the EXT card with three options.

1. Omit the *Vm* part of the A_i entry; that is, enter only the stretching entry $A_i = Q$ for a given cell. This causes the stretching to be in the particle direction $(\mu = 1)$, independent of the particle direction and is not recommended unless you want to do implicit capture along a flight path, in which case $A_i = Q = S$ and the distance to scatter rather than the distance to collision is sampled.

2. Specify the stretching direction as *Vm*, the line from the collision point to the point (x_m, y_m, z_m), where (x_m, y_m, z_m) is specified on the VECT card (see next section). The direction cosine μ is now the cosine of the angle between the particle direction and the line drawn from the collision point to point (x_m, y_m, z_m). The sign of A_i governs whether stretching is toward or away from (x_m, y_m, z_m).

3. Specify the stretching direction as $Vm = X$, Y, or Z, so the direction cosine μ is the cosine of the angle between the particle direction and the $X–$, $Y–$, or $Z–$axis, respectively. The sign of A_i governs whether stretching is toward or away from the $X–$, $Y–$, or $Z–$axis.

Example: EXT:N 0 0 .7V2 S −SV2 −.6V9 0 .5V9 SZ −.4X
      VECT V9 0 0 0 V2 1 1 1

The 10 entries are for the 10 cells in this problem. Path length stretching is not turned on for photons or for cells 1, 2, and 7. Following is a summary of path length stretching in the other cells.

cell	A_i	Q	V_m	stretching parameter	direction
3	.7V2	.7	V2	$p = .7$	toward point (1,1,1)
4	S	S		$p = \Sigma_a/\Sigma_t$	particle direction
5	−SV2	S	−V2	$p = \Sigma_a/\Sigma_t$	away from point (1,1,1)
6	−.6V9	.6	−V9	$p = .6$	away from origin
8	.5V9	.5	V9	$p = .5$	toward origin
9	SZ	S	Z	$p = \Sigma_a/\Sigma_t$	along +Z-axis
10	−.4X	.4	−X	$p = .4$	along −X-axis

5. <u>VECT</u>   Vector Input Card

Form: VECT   *Vm* $x_m\ y_m\ z_m$ ... V*n* $x_n\ y_n\ z_n$ ...

m,n =   any numbers to uniquely identify vectors *Vm, Vn* ...
$x_m\ y_m\ z_m$ =   coordinate triplets to define vector *Vm*.

Default: None.

Use: Optional.

The entries on the VECT card are quadruplets which define any number of vectors for either the exponential transform or user patches. See the EXT card (page 3–36) for a usage example.

6. <u>FCL</u>   Forced Collision Card

Form: FCL:*n* $x_1\ x_2$ ... x_i ... x_I

n = N for neutrons, P for photons, not available for electrons.
x_i = forced collision control for cell *i*. $-1 \le x_i \le 1$
I = number of cells in the problem.

Default: $x_i = 0$, no forced collisions.

Use: Optional. Exercise caution.

The FCL card controls the forcing of neutron or photon collisions in each cell. This is particularly useful for generating contributions to point detectors or DXTRAN spheres. The weight window game at surfaces is not played when entering forced collision cells.

If $x_i \neq 0$, all particles entering cell i are split into collided and uncollided parts with the appropriate weight adjustment (see page 2–147). If $|x_i| < 1$, Russian roulette is played on the collided parts with survival probability $|x_i|$ to keep the number of collided histories from getting too large. Fractional x_i entries are recommended if a number of forced collision cells are adjacent to each other.

If $x_i < 0$, the forced collision process applies only to particles entering the cell. After the forced collision, the weight cutoff is ignored and all subsequent collisions are handled in the usual analog manner. Weight windows are not ignored and are applied after contributions are made to detectors and DXTRAN spheres.

If $x_i > 0$, the forced collision process applies both to particles entering cell i and to the collided particles surviving the weight cutoff or weight window games. Particles will continue to be split into uncollided and (with probability $|x_i|$) collided parts until killed by either weight cutoff or weight windows.

Usage tips:

Let $x_i = 1$ or -1 unless a number of forced collision cells are adjacent to each other or the number of forced collision particles produced is higher than desired. Then fractional values are usually needed.

When cell–based weight window bounds bracket the typical weight entering the cell, choose $x_i > 0$. When cell–based weight window bounds bracket the weight typical of forced collision particles, choose $x_i < 0$. For mesh–based windows, $x_i > 0$ usually is recommended.

When using importances, use $x_i > 0$ because $x_i < 0$ turns off the weight cutoff game.

7–9. Weight Window Cards

Weight windows can be either cell–based or mesh–based. Mesh–based windows eliminate the need to subdivide geometries finely enough for importance functions.

Weight windows provide an alternative means to importances (IMP:n cards) and energy splitting (ESPLT:n cards) for specifying space and energy importance functions. They also can provide time–dependent importance functions. The advantages of weight windows are that they (1) provide an importance function in space and time or space and energy; (2) control particle weights; (3) are more compatible with other variance reduction features such as the exponential transform (EXT:n card); (4) can be applied at surface crossings, collisions, or both; (5) the severity of splitting or Russian roulette can be controlled; (6) can be turned off in selected space or energy regions; and (7) can be automatically generated by the weight window generator. The disadvantages are that (1) weight windows are not as straightforward as importances; and (2) when the source weight is

changed, the weight windows may have to be renormalized. You are strongly advised to read the section on weight windows in Chapter 2.

A cell–based weight-window lower bound in a cell that is in a universe is interpreted as a multiplier of the weight-window lower bound of the filled cell. Mesh–based windows are recommended in repeated structures.

7.   WWE   Weight Window Energies or Times

      Form: WWE:n E_1 E_2 ... E_i ... E_j; $j \leq 99$

        n = N for neutrons, P for photons, E for electrons
        E_i = upper energy or time bound of ith window
        E_{i-1} = lower energy or time bound of ith window
        E_0 = 0, by definition

      Default: If this card is omitted and the weight window is used, a single energy or time interval is established corresponding to the energy or time limits of the problem being run.

      Use: Optional. Use only with WWN card.

The WWE card defines the energy or time intervals for which weight window bounds will be specified on the WWN card. The minimum energy, which is not entered on the WWE card, is zero. The minimum time is $-\infty$. Whether energy or time is specified is determined by the 6th entry on the WWP card.

8.   WWN   Cell–Based Weight Window Bounds

      Form: WWNi:n w_{i1} w_{i2} ... w_{ij} ... w_{iJ}

        n = N for neutrons, P for photons, E for electrons
        w_{ij} = lower weight bound in cell j and energy or time interval $E_{i-1} < E < E_i$, $E_0 = 0$, as defined on the WWE card. If no WWE card, $i = 1$.
        J = number of cells in the problem.

      Default: None.

      Use: Weight windows (WWN and WWP cards) are required unless importances (IMP card) or mesh–based windows are used.

The WWN card specifies the lower weight bound of the space and energy dependent weight windows in cells. It must be used with the WWP card, and, if the weight windows are energy or

time dependent, with the WWE card. The IMP:n card should not be used if a WWN:n card, where n is the same particle type, is used.

If $w_{ij} < 0$, any particle entering cell j is killed. That is, negative entries correspond to zero importance. If negative entries are used for one energy group, they should be used for all the other energy groups in the same cell.

If $w_{ij} > 0$, particles entering or colliding in cell j are split or rouletted according to the options on the WWP card, described in the next section.

If $w_{ij} = 0$, the weight window game is turned off in cell j for energy bin i and the weight cutoff game is turned on with a 1–for–2 roulette limit. Sometimes it is useful to specify the weight cutoffs on the CUT card as the lowest permissible weights desired in the problem. Otherwise, too many particles entering cells with $w_{ij} = 0$ may be killed by the weight cutoff. Usually the 1–for–2 roulette limitation is sufficient to use the default weight cutoffs, but caution is needed and the problem output file should be examined carefully. The capability to turn the weight window game off in various energy and spatial regions is useful when these regions cannot be characterized by a single importance function or set of weight window bounds.

In terms of the weight window, particle weight bounds are always absolute and not relative; you have to explicitly account for weight changes from any other variance reduction techniques such as source biasing. You must specify one lower weight bound per cell per energy interval. There must be no holes in the specification; that is, if WWNi is specified, WWNj for $1 < j < i$ must also be specified.

Example 1:	WWE:N	E_1	E_2	E_3	
	WWN1:N	w_{11}	w_{12}	w_{13}	w_{14}
	WWN2:N	w_{21}	w_{22}	w_{23}	w_{24}
	WWN3:N	w_{31}	w_{32}	w_{33}	w_{34}

These cards define three energy or time intervals and the weight window bounds for a four-cell neutron problem.

Example 2:	WWN1:P	w_{11}	w_{12}	w_{13}

This card, without an accompanying WWE card, defines an energy or time independent photon weight window for a three-cell problem.

9. WWP   Weight Window Parameter Card

Form: WWP:n   WUPN WSURVN MXSPLN MWHERE SWITCHN MTIME

n = N for neutrons, P for photons, E for electrons

WUPN = If the particle weight goes above WUPN times the lower weight bound, the particle will be split.
Required: WUPN ≥ 2.

WSURVN = If the particle survives the Russian roulette game, its weight becomes MIN(WSURVN times the lower weight bound,WGT∗MXSPLN).
Required: 1 < WSURVN < WUPN.

MXSPLN = No particle will ever be split more than MXSPLN-for-one or be rouletted more harshly than one-in-MXSPLN.
MXSPLN=2 in zero window cells or meshes.
Required: MXSPLN > 1.

MWHERE = decides where to check a particle's weight.
−1 means check the weight at collisions only
0 means check the weight at surfaces and collisions
1 means check the weight at surfaces only

SWITCHN = decides where to get the lower weight window bounds.
< 0 means get them from an external WWINP file. Requires IMP card.
0 means get them from WWNi cards.
> 0 means set the lower weight window bounds equal to SWITCHN divided by the cell importances from the IMP card.

MTIME = 0 energy dependent windows (WWE card)
1 time dependent windows (WWE card)

Defaults: WUPN=5; WSURVN=0.6∗WUPN; MXSPLN=5; MWHERE=0; SWITCHN=0, MTIME=0

Use: Weight windows are required unless importances are used.

The WWP card contains parameters that control use of the weight window lower bounds specified on the WWN cards, the IMP cards, or an external file, depending on the value of SWITCHN. Having SWITCHN > 0 and also having WWNi cards is a fatal error. If SWITCHN is zero, the lower weight window bounds must be specified with the WWNi cards. If SWITCHN < 0, an external WWINP file with either cell- or mesh-based lower weight window bounds must exist and an IMP card is required. This file name can be changed on the MCNP execution line using "WWINP = filename." The different formats of the WWINP file will indicate to the code whether the weight windows are cell or mesh based. For mesh-based weight windows, the mesh geometry will also be read from the WWINP file. The WWINP file format is provided in Appendix J.

Using Existing Cell Importances to Specify the Lower Weight Bound

An energy-independent weight window can be specified using existing importances from the IMP card and setting the fifth entry (SWITCHN) on the WWP card to a positive constant C. If this option is selected, the lower weight bounds for the cells become C/I, where I is the cell importance. The remaining entries on the WWP card are entered as described above. A suggested value for C

is one in which source particles start within the weight window such as .25 times the source weight. If that is not possible, your window is probably too narrow or you need to respecify your source.

10–12. Weight Window Generation Cards

The weight window generator estimates the importances of the space-energy regions of phase space specified by the user. The space-energy weight window parameters are then calculated inversely proportional to the importances.

Recall that the cell–based generator estimates the <u>average</u> importance of a phase-space cell. If the cells are too large, the importance variation inside the cell will be large and the average importance will not represent the cell. Inadequate geometry specification also occurs with large importance differences between adjacent cells. Fortunately, the generator provides information about whether the geometry specification is adequate for sampling purposes. If geometries are inadequately subdivided for importances, mesh–based weight windows should be used.

The user is advised to become familiar with the section on the weight window in Chapter 2 before trying to use the weight window generator.

10. WWG   Weight Window Generation

   Form: WWG I_t I_c W_g J J J J I_E

   I_t = problem tally number (n of the Fn card). The particular tally bin for which the weight window generator is optimized is defined by the TFn card.
   I_c = invokes cell- or mesh-based weight window generator .
         > 0 means use the cell-based weight window generator with I_c as the reference cell (typically a source cell).
         0 means use the mesh-based weight window generator. (MESH card.)
   W_g = value of the generated lower weight window bound for cell I_c or for the reference mesh (see MESH card).
         0 means lower bound will be half the average source weight.
   J = unused
   I_E toggles energy- or time-dependent weight windows.
         0 means interpret WWGE card as energy bins.
         1 means interpret WWGE card as time bins.

   Default: No weight window values are generated.

   Use: Optional.

The WWG card causes the optimum importance function for tally I_t to be generated. For the cell-based weight window generator, the importance function is written on WWE and WWNi cards that

are printed, evaluated, and summarized in the OUTP file and are also printed on the weight window generator output file WWOUT. For the mesh-based weight window generator, the importance function and the mesh description are written only on the WWOUT file. (The format of the mesh-based WWOUT file is provided in Appendix J.) In either case, the generated weight window importance function easily can be used in subsequent runs using SWITCHN < 0 on the WWP card. For many problems, this importance function is superior to anything an experienced user can guess on an IMP card. To generate energy- or time-dependent weight-windows, use the WWGE card described below.

11. **WWGE   Weight Window Generation Energies or Times**

      Form: WWGE:n $E_1\ E_2\ ...\ Ei\ ...\ E_j;\ j \leq 15$

         n = N for neutrons, P for photons, E for electrons
         E_i = upper energy or time bound for weight window group to be
                    generated, $E_{i+1} > E_i$.

      Default: If this card is omitted and the weight window is used, a single energy or time
               interval will be established corresponding to the energy/time limits of the problem
               being run. If the card is present but has no entries, ten energy/time bins will be
               generated with energies/times of $E_i = 10^{i-8}$ MeV/shake and $j = 10$. Both the single
               time/energy and the energy/time–dependent windows are generated.

      Use: Optional.

If this card is present, time/energy-dependent weight windows are generated and written on the WWOUT file and, for cell-based weight windows, on the OUTP file. If $I_E = 1$ on the WWG card, time-dependent windows are generated. In addition, single-group energy- or time-<u>in</u>dependent weight windows are written on a separate output file, WWONE. Energy- and time-<u>in</u>dependent weight windows are useful for trouble-shooting the energy- and time-dependent weight windows on the WWOUT file. The WWONE file format is the same as that of the WWOUT file provided in Appendix J.

12. **MESH   Superimposed Importance Mesh for Mesh-Based Weight Window Generator**

      Form: MESH   mesh variable=specification

      Use: Required if mesh-based weight windows are used or generated.

The equal sign is optional. Keywords can be entered in any order. Special input features I, M, and R can be used except with GEOM. Table 3.2 summarizes the superimposed mesh variables and lists their defaults. The default geometry is rectangular and the default ORIGIN point is (0,0,0). For a cylindrical mesh, the default cylindrical axis is parallel to the MCNP geometry z axis and the half-

plane defining θ=0 is the MCNP geometry positive *x* axis. The reference point must always be specified.

TABLE 3.2:   Superimposed Mesh Variables

Variable	Meaning	Default
GEOM	Mesh geometry; either Cartesian ("xyz" or "rec") or cylindrical ("rzt" or "cyl").	xyz
REF	*x*, *y*, and *z* coordinates of the reference point	None (variable must be present)
ORIGIN	*x*, *y*, and *z* coordinates in MCNP cell geometry of the origin (bottom center for cylindrical or bottom, left, behind for rectangular) of the superimposed mesh	0., 0., 0.
AXS	vector giving the direction of the axis of the cylindrical mesh	0., 0., 1.
VEC	vector defining, along with AXS, the plane for θ= 0	1., 0., 0.
IMESH	locations of the coarse meshes in the *x* direction for rectangular geometry or in the *r* direction for cylindrical geometry	None
IINTS	number of fine meshes within corresponding coarse meshes in the *x* direction for rectangular geometry or in the *r* direction for cylindrical geometry	10 in each coarse mesh
JMESH	locations of the coarse meshes in the *y* direction for rectangular geometry or in the *z* direction for cylindrical geometry	None
JINTS	number of fine meshes within corresponding coarse meshes in the *y* direction for rectangular geometry or in the *z* direction for cylindrical geometry	10 in each coarse mesh
KMESH	locations of the coarse meshes in the *z* direction for rectangular geometry or in the θ direction for cylindrical geometry	None
KINTS	number of fine meshes within corresponding coarse meshes in the *z* direction for rectangular geometry or in the θ direction for cylindrical geometry	10 in each coarse mesh

The location of the *n*'th coarse mesh in the *u* direction (ru_n in what follows) is given in terms of the most positive surface in the *u* direction. For a rectangular mesh, the coarse mesh locations rx_n, ry_n, and rz_n are given as planes perpendicular to the *x*, *y*, and *z* axes, respectively, in the MCNP cell coordinate system; thus, the ORIGIN point (x_0, y_0, z_0) is the most negative point of the mesh. For a cylindrical mesh, $(r_0, z_0, θ_0) = (0., 0., 0.)$, corresponds to the bottom center point, which is the cylindrical ORIGIN point in MCNP cell geometry. The coarse mesh locations must increase monotonically (beginning with the ORIGIN point for a rectangular mesh).

The fine meshes are evenly distributed within the *n*'th coarse mesh in the *u* direction. The mesh in which the reference point lies becomes the reference mesh for the mesh-based weight window

generator; this reference mesh is analogous to the reference cell used by the cell-based weight window generator.

For a cylindrical mesh, the AXS and VEC vectors need not be orthogonal but they must not be parallel; the one half-plane that contains them and the ORIGIN point will define $\theta = 0$. The AXS vector will remain fixed. The length of the AXS or VEC vectors must not be zero. The θ coarse mesh locations are given in revolutions and the last one must be 1.

At least two coarse meshes per coordinate direction must be specified using IMESH, JMESH, and KMESH keywords, but the code uses a default value of 10 fine meshes per coarse mesh if IINTS, JINTS, or KINTS keywords are omitted . If IINTS, JINTS, or KINTS keywords are present, the number of entries must match the number of entries on the IMESH, JMESH, and KMESH keywords, respectively. Entries on the IINTS, JINTS, and KINTS keywords must be greater than zero. A reference point must be specified using the REF keyword.

A second method of providing a superimposed mesh is to use one that already exists, either written on the WWOUT file or on the WWONE file. To implement this method, use the WWG card with $I_c=0$ in conjunction with the MESH card where the only keyword is REF. The reference point must be within the superimposed mesh and must be provided because there is no reference point in either WWOUT or WWONE. If the mesh-based weight window generator is invoked by this method, MCNP expects to read a file called WWINP. WWOUT or WWONE can be renamed in the local filespace or the files can be equivalenced on the execution line using "WWINP=filename."

It is not necessary to use mesh-based weight windows from the WWINP file in order to use the mesh from that file. Furthermore, previously generated mesh-based weight windows can be used (WWP card with SWITCHN < 0 and WWINP file in mesh format) while the mesh-based weight window generator is simultaneously generating weight windows for a different mesh (input on the MESH card). However, it is not possible to read mesh-based weight windows from one file but a weight-window generation mesh from a different file.

The superimposed mesh should fully cover the problem geometry; i.e., the outer boundaries of the mesh should lie outside the outer boundaries of the geometry, rather than being coincident with them. This requirement guarantees that particles remain within the weight window mesh. A line or surface source should not be made coincident with a mesh surface. A point source should never be coincident with the intersection of mesh surfaces. In particular, a line or point source should never lie on the axis of a cylindrical mesh. These guidelines also apply to the WWG reference point specified using the REF keyword.

If a particle does escape the weight-window generation mesh, the code prints a warning message giving the coordinate direction and surface number (in that direction) from which the particle escaped; for example, "warning. particle escaped wwg mesh in z direction" (the mesh index number appears with NPS on the next line). The code prints the total number of particles escaping

the mesh (if any) after the tally fluctuation charts in the standard output file. Similarly, if a track starts outside the mesh, the code prints a warning message giving the coordinate direction that was missed and which side of the mesh the particle started on; for example, "warning. track started outside wwg mesh: x too great." The code prints the total number of particles starting outside the mesh (if any) after the tally fluctuation charts in the standard output file.

$I_c = 0$ on the WWG card with no MESH card is a fatal error. If AXS or VEC keywords are present and the mesh is rectangular, a warning message is printed and the keyword is ignored. If there are fatal errors and the FATAL option is on, weight-window generation is disabled.

    Example: GEOM=cyl REF=1e–6 1e–7 0   ORIGIN=1 2 3
                         IMESH 2.55 66.34
                         IINTS   2   15 $ 2 fine bins from 0 to 2.55, 15 from 2.55 to 66.34
                         JMESH 33.1 42.1 53.4 139.7
                         JINTS   6   3   4   13
                         KMESH .5   1
                         KINTS   5   5

    Example: GEOM=rec REF=1e–6 1e–7 0   ORIGIN=–66.34   –38.11   –60
                         IMESH –16.5 3.8 53.66
                         IINTS   10   3   8 $ 10 fine bins from –66.34 to –16.5, etc

13. PDn   Detector Contribution Card

    Form: PDn P_1 P_2 ... P_i ... P_I

      n = tally number
      P_i = probability of contribution to detector n from cell ***i***
      I = number of cells in the problem.

    Default: $P_i = 1$.

    Use: Optional. Consider also using the DD card, page 3–103.

The PDn card reduces the number of contributions to detector tallies from selected cells that are relatively unimportant to a given detector, thus saving computing time. At each collision in cell i, the detector tallies are made with probability P_i $(0 \leq P_i \leq 1)$. The tally is then increased by the factor $1/P_i$ to obtain unbiased results for all cells except those where $P_i = 0$. This enables you to increase the running speed by setting $P_i < 1$ for cells many mean free paths from the detectors. It also selectively eliminates detector contributions from cells by setting the P_i's to zero.

A default set of probabilities can be established for all tallies by use of a PD0 (zero) card. These default values will be overridden for a specific tally n by values entered on a PDn card.

14. DXC   DXTRAN Contribution Card

    Form: DXCm:n P_1 P_2 ... P_i ... P_I

          m = which DXTRAN sphere the DXC card applies to. If 0 or absent,
              the DXC card applies to all the DXTRAN spheres in the problem.
          n = N for neutrons, P for photons, not available for electrons.
          P_i = probability of contribution to DXTRAN spheres from cell i
          I = number of cells in the problem

    Default: $m = 0$, $P_i = 1$.

    Use: Optional. Consider also using the DD card, page 3–103.

This card is analogous to the above PDn card but is used for contributions to DXTRAN spheres.

15. BBREM   Bremsstrahlung Biasing Card

    Form: BBREM b_1 b_2 b_3 ... b_{49} m_1 m_2 ... m_n

        b_1 = any positive value (currently unused).
        $b_2 ... b_{49}$ = bias factors for the bremsstrahlung energy spectrum.
        $m_1 ... m_n$ = list of materials for which the biasing is invoked.

    Default: None.

    Use: Optional.

The bremsstrahlung process generates many low-energy photons, but the higher-energy photons are often of more interest. One way to generate more high-energy photon tracks is to bias each sampling of a bremsstrahlung photon toward a larger fraction of the available electron energy. For example, a bias such as

$$\text{BBREM}\ \ 1.\ \ 1.\ \ 46I\ \ 10.\ \ \ 888\ \ 999$$

would create a gradually increasing enhancement (from the lowest to the highest fraction of the electron energy available to a given event) of the probability that the sampled bremsstrahlung photon will carry a particular fraction of the electron energy. This biasing would apply to each instance of the sampling of a bremsstrahlung photon in materials 888 and 999. The sampling in other materials would remain unbiased. The bias factors are normalized by the code in a manner that depends both on material and on electron energy, so that although the ratios of the photon weight adjustments among the different groups are known, the actual number of photons produced in any group is not easily predictable. For the el03 treatment, there are more than 49 relative photon energy ratios so the lower energy bins have a linear interpolation between b_1 and b_2 for their values.

In most problems the above prescription will increase the total number of bremsstrahlung photons produced because there will be more photon tracks generated at higher energies. The secondary electrons created by these photons will tend to have higher energies as well, and will therefore be able to create more bremsstrahlung tracks than they would at lower energies. This increase in the population of the electron-photon cascade will make the problem run more slowly. The benefits of better sampling of the high-energy domain must be balanced against this increase in run time.

For a more detailed discussion of the bremsstrahlung energy biasing scheme, see Chapter 2.

D.   *Source Specification*

Every MCNP problem has one of four sources: general source (SDEF card), surface source (SSR card), criticality source (KCODE card), or user-supplied source (default if SDEF, SSR, and KCODE are all missing). All can use source distribution functions, specified on SIn, SPn, SBn, and DSn cards.

The following cards are used to specify the source.

Mnemonic	Card Type	Page
SDEF	General source	3–50
SIn	Source information	3–57
SPn	Source probability	3–58
SBn	Source bias	3–58
DSn	Dependent source	3–62
SCn	Source comment	3–63
SSW	Surface source write	3–65
SSR	Surface source read	3–66
KCODE	Criticality source	3–71
KSRC	Source points	3–71
ACODE	Alpha eigenvalue source	3–72

The MODE card also serves as part of the source specification in some cases by implying the type of particle to be started from the source.

The source has to define the values of the following MCNP variables for each particle it produces:

ERG	the energy of the particle (MeV). See ∗ below
TME	the time when the particle started (shakes)
UUU, VVV, WWW	the direction of the flight of the particle
XXX, YYY, ZZZ	the position of the particle

IPT	the type of the particle
WGT	the statistical weight of the particle
ICL	the cell where the particle started
JSU	the surface where the particle started, or zero if the starting point is not on any surface

Additional variables may have to be defined if there are point detectors or DXTRAN spheres in the problem.

*ERG has a different meaning in a special case. If there is a negative IGM on the MGOPT card, which indicates a special electron–photon multigroup problem, ERG on the SDEF card is interpreted as an energy group number, an integer.

1. SDEF   General Source Card

    Form: SDEF   source variable = specification ...

    Use: Required for problems using the general source. Optional for problems using the criticality source.

The equal signs are optional. The source variables are not quite the same as MCNP variables that the source must set. Many are intermediate quantities that control the sampling of the final variables. All have default values. The specification of a source variable has one of these three forms:

1. explicit value,

2. a distribution number prefixed by a D, or

3. the name of another variable prefixed by an F, followed by a distribution number prefixed by a D. Var = Dn means that the value of source variable var is sampled from distribution n. Var Fvar′ Dn means that var is sampled from distribution n that depends on the variable var′. Only one level of dependence is allowed. Each distribution may be used for only one source variable.

The above scheme translates into three levels of source description. The first level exists when a source variable has an explicit or default value (for example, a single energy) or a default distribution (for example, an isotropic angular distribution). The second level occurs when a source variable is given by a probability distribution. This level requires the SI and/or SP cards. The third level occurs when a variable depends on another variable. This level requires the DS card.

MCNP samples the source variables in an order set up according to the needs of the particular problem. Each dependent variable must be sampled after the variable it depends on has been sampled. If the value of one variable influences the default value of another variable or the way it

is sampled, as SUR influences DIR, they may have to be sampled in the right order. The scheme used in MCNP to set up the order of sampling is complicated and may not always work. If it fails, a message will be printed. The fix in such instances may be to use explicit values or distributions instead of depending on defaults.

Table 3.3 summarizes the source variables and lists their defaults.

TABLE 3.3:   Source Variables

Variable	Meaning	Default
CEL	Cell	Determined from XXX,YYY,ZZZ and possibly UUU,VVV,WWW
SUR	Surface	Zero (means cell source)
ERG	Energy (MeV)	14 MeV
TME	Time (shakes)	0
DIR	μ, the cosine of the angle between VEC and UUU,VVV,WWW (Azimuthal angle is always sampled uniformly in 0° to 360°)	Volume case: μ is sampled uniformly in -1 to 1 (isotropic) Surface case: $p(\mu) = 2\mu$ in 0 to 1 (cosine distribution)
VEC	Reference vector for DIR	Volume case: required unless isotropic Surface case: vector normal to the surface with sign determined by NRM
NRM	Sign of the surface normal	$+1$
POS	Reference point for position sampling	0,0,0
RAD	Radial distance of the position from POS or AXS	0
EXT	Cell case: distance from POS along AXS Surface case: Cosine of angle from AXS	0
AXS	Reference vector for EXT and RAD	No direction
X	x-coordinate of position	No X
Y	y-coordinate of position	No Y
Z	z-coordinate of position	No Z
CCC	Cookie-cutter cell	No cookie-cutter cell

ARA	Area of surface (required only for direct contributions to point detectors from plane surface source.)	None
WGT	Particle weight	1
EFF	Rejection efficiency criterion for position sampling	.01
PAR	Particle type source will emit	1=neutron if MODE N or N P or N P E 2=photon if MODE P or P E 3=electron if MODE E

The specification of WGT, EFF and PAR must be only an explicit value. A distribution is not allowed. The allowed value for PAR is 1 for neutron, 2 for photon, or 3 for electron. The default is the lowest of these three that corresponds to an actual or default entry on the MODE card. Only one kind of particle is allowed in an SDEF source.

Most of the source variables are scalars. VEC, POS, and AXS are vectors. Where a value of a source variable is required, as on SDEF, SI, or DS cards, usually a single number is appropriate, but with VEC, POS, and AXS, the value must actually be a triplet of numbers, the x, y, and z components of the vector.

The source variables SUR, POS, RAD, EXT, AXS, X, Y, Z, and CCC are used in various combinations to determine the coordinates (x,y,z) of the starting positions of the source particles. With them you can specify three different kinds of volume distributions and three different kinds of distributions on surfaces. Degenerate versions of those distributions provide line and point sources. More elaborate distributions can be approximated by combining several simple distributions, using the S option of the SIn and DSn cards.

The three volume distributions are cartesian, spherical, and cylindrical. The value of the variable SUR is zero for a volume distribution. A volume distribution can be used in combination with the CEL variable to sample uniformly throughout the interior of a cell. A cartesian, spherical, or cylindrical region that completely contains a cell is specified and is sampled uniformly in volume. If the sampled point is found to be inside the cell, it is accepted. Otherwise it is rejected and another point is sampled. If you use this technique, you must make sure that the sampling region really does contain every part of the cell because MCNP has no way of checking for this. Cookie-cutter rejection, described below, can be used instead of or in combination with CEL rejection.

A cartesian volume distribution is specified with the variables X, Y, and Z. A degenerate case of the cartesian distribution, in which the three variables are constants, defines a point source. A single point source can be specified by giving values to the three variables right on the SDEF card. If there are several source points in the problem, it would usually be easier to use a degenerate spherical

distribution for each point. Other degenerate cases of the cartesian distribution are a line source and a rectangular plane source. A cartesian distribution is an efficient shape for the CEL rejection technique when the cell is approximately rectangular. It is much better than a cylindrical distribution when the cell is a long thin slab. It is, however, limited in that its faces can only be perpendicular to the coordinate axes.

A spherical volume distribution is specified with the variables POS and RAD. X, Y, Z, and AXS must not be specified or it will be taken to be a cartesian or cylindrical distribution. The sampled value of the vector POS defines the center of the sphere. The sampled value of RAD defines the distance from the center of the sphere to the position of the particle. The position is then sampled uniformly on the surface of the sphere of radius RAD. Uniform sampling in volume is obtained if the distribution of RAD is a power law with $a = 2$, which is the default in this case. A common use of the spherical volume distribution is to sample uniformly in the volume between two spherical surfaces. The two radii are specified on the SIn card for RAD and the effect of a SPn −21 2 card is obtained by default (see page 3–58). If RAD is not specified, the default is zero. This is useful because it specifies a point source at the position POS. A distribution for POS, with an L on the SIn card, is the easiest way to specify a set of point sources in a problem.

A cylindrical volume distribution is specified with the variables POS, AXS, RAD, and EXT. The axis of the cylinder passes through the point POS in the direction AXS. The position of the particles is sampled uniformly on a circle whose radius is the sampled value of RAD, centered on the axis of the cylinder. The circle lies in a plane perpendicular to AXS at a distance from POS which is the sampled value of EXT. A common use of the cylindrical distribution is to sample uniformly in volume within a cylindrical shell. The distances of the ends of the cylinder from POS are entered on the SIn card for EXT and the inner and outer radii are entered on the SIn card for RAD. Uniform sampling between the two values of EXT and power law sampling between the two values of RAD, with $a = 1$ which gives sampling uniform in volume, are provided by default. A useful degenerate case is EXT=0, which provides a source with circular symmetry on a plane.

Warning: Never position any kind of degenerate volume distribution in such a way that it lies on one of the defined surfaces of the problem geometry. Even a bounding surface that extends into the interior of a cell can cause trouble. If possible, use one of the surface distributions instead. Otherwise, move to a position just a little way off of the surface. It will not make any detectable difference in the answers, and it will prevent particles from getting lost.

The value of the variable SUR is nonzero for a distribution on a surface. If X, Y, and Z are specified, their sampled values determine the position. You must in this case make sure that the point really is on the surface because MCNP does not check. If X, Y, and Z are not specified, the position is sampled on the surface SUR. The shape of the surface, which can be either a spheroid, sphere, or plane, determines the way the position is sampled. Sampling with CEL rejection is not available, but cookie-cutter rejection can be used to do anything that CEL rejection would do. Cylindrical surface sources must be specified as degenerate volume sources.

If the value of SUR is the name of a spheroidal surface, the position of the particle is sampled uniformly in area on the surface. A spheroid is an ellipse revolved around one of its axes. A spheroid for this purpose must have its axis parallel to one of the coordinate axes. There is presently no provision for easy nonuniform or biased sampling on a spheroidal surface. A distribution of cookie-cutter cells could be used to produce a crude nonuniform distribution of position.

If the value of SUR is the name of a spherical surface, the position of the particle is sampled on that surface. If the vector AXS is not specified, the position is sampled uniformly in area on the surface. If AXS is specified, the sampled value of EXT is used for the cosine of the angle between the direction AXS and the vector from the center of the sphere to the position point. The azimuthal angle is sampled uniformly in the range from $0°$ to 360°. A nonuniform distribution of position, in polar angle only, is available through a nonuniform distribution of EXT. A biased distribution of EXT can be used to start more particles from the side of the sphere nearest the tallying regions of the geometry. The exponential distribution function (−31; see page 3–61) is usually most appropriate for this.

If the value of SUR is the name of a plane surface, the position is sampled on that plane. The sampled value of POS must be a point on the plane. You must make sure that POS really is on the plane because MCNP, for the sake of speed, does not check it. The sampled position of the particle is at a distance from POS equal to the sampled value of RAD. The position is sampled uniformly on the circle of radius RAD centered on POS. Uniform sampling in area is obtained if the distribution of RAD is a power law with $a = 1$, which is the default in this case.

Cookie-cutter rejection is available for both cell and surface sources. If CCC is present, the position sampled by the above procedures is accepted if it is within cell CCC and is resampled if it is not, exactly like CEL rejection in the cell source case. You must be careful not to specify a cookie-cutter cell such that MCNP mistakes it for a real cell. There should be no trouble if the cookie-cutter cells are bounded by surfaces used for no other purpose in the problem and if the cookie-cutter cell cards are at the end of the list of cell cards. Don't make a cookie-cutter cell more complicated than it has to be. For a surface source, the only thing that matters is the intersection of the cookie-cutter cell with the source surface. An infinitely long cell of uniform cross section, bounded by planes and cylinders, is usually adequate for a plane surface source.

Warning: The combination of either CEL or CCC rejection with biased sampling of the position is nearly always an unfair game. If you use this combination, you must make sure that it really is a fair game because MCNP is not able to detect the error.

The source variables SUR, VEC, NRM, and DIR are used to determine the initial direction of flight of the source particles. The direction of flight is sampled with respect to the reference vector VEC, which, of course, can itself be sampled from a distribution. The polar angle is the sampled value of the variable DIR. The azimuthal angle is sampled uniformly in the range from $0°$ to 360°. If VEC and DIR are not specified for a volume distribution of position (SUR=0), an isotropic distribution

of direction is produced by default. If VEC is not specified for a distribution on a surface (SUR ≠0), the vector normal to the surface, with the sign determined by the sign of NRM, is used by default. If DIR is not specified for a distribution on a surface, the cosine distribution $p(DIR)=2*DIR$, 0<DIR<1, is used by default. A biased distribution of DIR can be used to make more source particles start in a direction towards the tallying regions of the geometry. The exponential distribution function (−31; see page 3–61) is usually most appropriate for this.

Discrete values of DIR are allowed. DIR=1 gives a monodirectional source in the direction of VEC. This is sometimes useful as an approximation to an actual source that is at a large distance from the geometry of the problem. In most cases discrete values of DIR will prevent direct contributions to point detectors from being scored. The direct contribution will be scored only if the source is on a plane surface, is sampled uniformly in area within a circle (using RAD sampled from SP −21 1), VEC is perpendicular to the surface (the default), and DIR=1. A cookie-cutter cell is allowed and a value of ARA is necessary. Discrete values of DIR with DXTRAN are generally wrong because $p(\mu) = .5$ is assumed. See page 2–152.

The efficiency criterion EFF applies to both CCC and CEL rejection. If in any source cell or cookie-cutter cell the acceptance rate is too low, the problem is terminated for inefficiency. The criterion for termination is MAX(number of successes, 10) < EFF * number of tries. The default value of EFF, 0.01, lets a problem get by at rather low efficiency, but for the rare problem in which low source efficiency is unavoidable, you may need to specify a lower value for EFF.

This section discusses a source in a geometry that we will call a repeated structure and is one that includes FILL, U, or LAT cards. We strongly emphasize that print table 110 should be studied carefully to ensure that the proper source path and position are being sampled when repeated structures are used in a source description.

Warning: Defining a source on a lattice cell bounding plane or on a "window" cell plane coincident with a lattice surface generally does not work and is not recommended.

The only part of the MCNP source specification that is different when the source is in a repeated structure part of the geometry is the use of the CEL parameter on the SDEF card. CEL must have a value that is a path from level **0** to level **n**, which is not necessarily the bottom:

$$c_0 : c_1 : ... : c_n$$

c_i is a cell in the universe that fills cell c_{i-1}, or is zero, or is Dm for a distribution of cells in the repeated structure case. Dm is not valid for a lattice. c_i can have a minus sign and is discussed more below. Dm cannot have a minus sign. If $c_i = 0$, the cell at that level is searched for. Recall that level n is not necessarily the bottom level in the problem. If c_i is one specific element in a lattice, it is indicated as: ... : c_i (j1 j2 j3) : ...

The coordinate system for **p**osition and **d**irection **s**ampling (pds) is the coordinate system of the first negative or zero c_i in the source path. Each entry in the source path represents a geometry level, where level zero is the first source path entry, level one the second, etc., and level zero is <u>above</u> level one, level two is <u>below</u> level one. The pds level is the level associated with the pds cell or pds coordinate system. All levels above the pds level must be included in the source path. Levels below the pds level need not be specified, and when given, may include one or more zero entries. The default pds level is the last entry in the source path when the path has no negative or zero entry.

Position rejection is done in cells at all levels where $c_i \neq 0$, but if any c_i has a negative universe number on its cell card and is at or above the pds level, higher level cells are not checked.

The following chart illustrates the idea of the pds level.

CEL Source Path	Cell of pds Level	pds Level
8:7:6:5	5	3
8:−7:6	7	1
8:7:−6:0:4:0	6	2
8:−(1 0 0):6(0 0 0):0	7	1
8:7(1 0 0):6(0 0 0):0	Will be determined	3

Lattice cell elements that are defined using the expanded FILL card (see page 3–29) can be uniformly sampled automatically. This feature is applied to lattice cell entries in the source path that lack an explicit lattice index <u>AND</u> that are *at or above* the pds level. Lattice cells not defined by the expanded FILL card <u>must</u> include an explicit lattice index when at or above the pds level. Rejection of automatically sampled lattice elements depends on the entry after the lattice cell number in the source path.

Assume the following cell cards:

    7   0   surfaces   lat=1   u=1   fill=0:2 0:0 0:0   1   2   3
    cells 8 and 9 belong to universe 2
    cells 10 and 11 belong to universe 3

Cell 7 is a lattice with three existing elements: (0 0 0) is filled by itself [u=1], (1 0 0) is filled by cells 8 and 9 [u=2], and (2 0 0) is filled by cells 10 and 11 [u=3]. The following combinations show what elements are accepted and what are rejected.

CEL Source Path	Accepted	Rejected
7	All elements	None
7:0	All elements	None
7:8	(1 0 0)	(0 0 0), (2 0 0)
7:10	(2 0 0)	(0 0 0), (1 0 0)

The sampling efficiency for cell 7 in the OUTP file will reflect the element rejections. Lattice cell entries that lack an explicit lattice index <u>AND</u> are *below* the pds level are not sampled. Instead, the appropriate lattice element is determined by the input source position.

Lattice element sampling is independent from position sampling. First a lattice element is chosen, then a position is chosen. If the sampled position is not in the sampled lattice element, the position is resampled until it <u>is</u> in the specified source path and in the lattice element chosen <u>or</u> until an efficiency error occurs. The lattice elements will <u>not</u> be resampled to accommodate the sampled position. Lattice element rejection is done only as described above.

Using the previous description of lattice cell 7, add that cell 6 is filled by cell 7. The source path becomes 6:7:0. Three elements of the lattice exist [fill=0:2 0:0 0:0] but element (0 0 0) now is cut off by cell 6. Lattice element (0 0 0) still will be sampled 1/3 of the time. The first time element (0 0 0) is sampled a fatal error will occur because the sampled position, no matter what it is, will be rejected because element (0 0 0) does not exist. <u>CAUTION</u>: Implement automatic lattice sampling carefully and ensure that all of the lattice elements specified on the expanded FILL card really do exist.

See Chapter 4 for an example of specifying a source in a lattice geometry.

    Example: SDEF   (no entries)

This card specifies a 14-MeV isotropic point source at position 0,0,0 at time 0 with weight 1 (all defaults).

2.   <u>SIn   Source Information Card</u>
3.   <u>SPn   Source Probabiy Card</u>
4.   <u>SBn   Source Bias Card</u>

    Form: SIn   option $I_1 \dots I_k$

      n =   distribution number ($n = 1,999$)
    option =   how the I_i's are to be interpreted. Allowed values are:
            omitted or H—bin boundaries for a histogram distribution,
               for scalar variables only. This is the default.
            L—discrete source variable values
            A—points where a probability density distribution is defined
            S—distribution numbers
    $I_1 \dots I_k$ =   source variable values or distribution numbers

    Default: SIn   H $I_i \dots I_k$

Form: SPn   option $P_1 ... P_k$
    or: SPn f a b

n	= distribution number ($n = 1,999$)
option	= how the P_i are to be interpreted. Allowed values are:

        omitted—same as D for an H or L distribution. Probability
           density for an A distribution on SI card.
        D—bin probabilities for an H or L distribution on SI card.
           This is the default.
        C—cumulative bin probabilities for an H or L distribution
           on SI card.
        V—for cell distributions only. Probability is proportional
           to cell volume (times P_i if the P_i are present).

$P_i ... P_k$	= source variable probabilities
f	= designator (negative number) for a built-in function
a,b	= parameters for the built-in function (see Table 3.4)

Default: SPn   D $P_1 ...P_k$

Form: SBn   option $B_1 ... B_k$
    or: SBn f a b

n,	option, f, a, and b are the same as for the SPn card, except that the only values allowed for f are −21 and −31
	$B_i ... B_k$ = source variable biased probabilities

Default: SBn   D $B_1 ... B_k$

The first form of the SP card, where the first entry is positive or nonnumeric, indicates that it and its SI card define a probability distribution function. The entries on the SI card are either values of the source variable or, when the S option is used, distribution numbers. The entries on the SP card are probabilities that correspond to the entries on the SI card.

When the H option is used, the numerical entries on the SI card are bin boundaries and must be monotonically increasing. The first numerical entry on the SP card must be zero and the following entries are bin probabilities or cumulative bin probabilities, depending on whether the D or C option is used. The probabilities need not be normalized. The variable is sampled by first sampling a bin according to the bin probabilities and then sampling uniformly within the chosen bin.

When the A option is used, the entries on the SI card are values of the source variable at which the probability density is defined. The entries must be monotonically increasing, and the lowest and highest values define the range of the variable. The numerical entries on the SP card are values of the probability density corresponding to the values of the variable on the SI card. They need not be normalized. In the sampling process, the probability density is linearly interpolated between the

specified values. The first and last entries on the SP card will typically be zero, but nonzero values are also allowed.

When the L option is used, the numerical entries on the SI card are discrete values of the source variable, such as cell numbers or the energies of photon spectrum lines. The entries on the SP card are either probabilities of those discrete values or cumulative probabilities, depending on whether the D or C option is used. The entries on the SI card need not be monotonically increasing.

The S option allows sampling among distributions, one of which is chosen for further sampling. This feature makes it unnecessary to fold distributions together and is essential if some of the distributions are discrete and others are linearly interpolated. The distributions listed on an SI card with the S option can themselves also have the S option. MCNP can handle this structure to a depth of about 20, which should be far more than necessary for any practical problem. Each distribution number on the SI card can be prefixed with a D, or the D can be omitted. If a distribution number is zero, the default value for the variable is used. A distribution can appear in more than one place with an S option, but a distribution cannot be used for more than one source variable.

The V option on the SP card is a special case used only when the source variable is CEL. This option is useful when the cell volume is a factor in the probability of particle emission. If MCNP cannot calculate the volume of such a cell and the volume is not given on a VOL card, you have a fatal error.

The SB card is used to provide a probability distribution for sampling that is different from the true probability distribution on the SP card. Its purpose is to bias the sampling of its source variable to improve the convergence rate of the problem. The weight of each source particle is adjusted to compensate for the bias. All rules that apply to the first form of the SP card apply to the SB card.

The second form of the SP card, where the first entry is negative, indicates that a built-in analytic function is to be used to generate a continuous probability density function for the source variable. Built-in functions can be used only for scalar variables. See Table 3.3 for a description of these functions.

TABLE 3.4: **Built-In Functions for Source Probability and Bias Specification**

Source Variable	Function No. and Input Parameters	Description
ERG	−2 a	Maxwell fission spectrum
ERG	−3 a b	Watt fission spectrum
ERG	−4 a b	Gaussion fusion spectrum
ERG	−5 a	Evaporation spectrum
ERG	−6 a b	Muir velocity Gaussian fusion spectrum
ERG	−7 a b	Spare
DIR, RAD, or EXT	−21 a	Power law $p(x) = c|x|^a$
DIR or EXT	−31 a	Exponential: $p(\mu) = ce^{a\mu}$
TME	−41 a b	Gaussian distribution of time

$\underline{f = -2}$ Maxwell fission energy spectrum: $p(E) = C\,E^{1/2}\exp(-E/a)$, where a is a temperature in MeV.
Default: $a = 1.2895$ MeV

$\underline{f = -3}$ Watt fission energy spectrum: $p(E) = C\exp(-E/a)\sinh(bE)^{1/2}$.
Defaults: $a = 0.965$ MeV, $b = 2.29$ MeV^{-1}. See Appendix H for additional parameters appropriate to neutron-induced fission in various materials and for spontaneous fission.

$\underline{f = -4}$ Gaussian fusion energy spectrum: $p(E) = C\exp[-((E-b)/a)^2]$, where a is the width in MeV and b is the average energy in MeV. Width here is defined as the ΔE above b where the value of the exponential is equal to e^{-1}. If $a < 0$, it is interpreted as a temperature in MeV and b must also be negative. If $b = -1$, the D–T fusion energy is calculated and used for b. If $b = -2$, the D–D fusion energy is calculated and used for b. Note that a is not the "full-width-at-half-maximum," but is related to it by FWHM $= a\,(\ln 2)^{1/2}$.
Defaults: $a = -0.01$ MeV, $b = -1$ (DT fusion at 10 keV).

$\underline{f = -5}$ Evaporation energy spectrum: $p(E) = C\,E\exp(-E/a)$.
Default: $a = 1.2895$ MeV.

$\underline{f = -6}$ Muir velocity Gaussian fusion energy spectrum: $p(E) = C\exp - ((E^{1/2} - b^{1/2})/a)^2$, where a is the width in MeV$^{1/2}$, and b is the energy in MeV corresponding to the average speed. Width here is defined as the change in velocity above the average velocity $b^{1/2}$, where the value of the exponential is equal to e^{-1}. To get a spectrum somewhat comparable to $f = -4$, the width can be determined by $a = (b + a_4)^{1/2} - b^{1/2}$, where a_4 is the width used with the Gaussian fusion energy spectrum. If $a < 0$, it is interpreted as a temperature in MeV. If $b = -1$, the D–T fusion energy is calculated and used for b. If $b = -2$, the D–D fusion energy is calculated and used for b.
Defaults: $a = -0.01$ MeV, $b = -1$ (DT fusion at 10 keV).

_f = −7_ Spare energy spectrum. The basic framework for another energy spectrum is in place to make it easier for a user to add a spectrum of his own. The subroutines to change are SPROB, SPEC, SMPSRC, and possibly CALCPS.

_f = −21_ Power law: $p(x) = c|x|^a$.
The default depends on the variable. For DIR, $a = 1$. For RAD, $a = 2$, unless AXS is defined or JSU $\neq 0$, in which case $a = 1$. For EXT, $a = 0$.

_f = −31_ Exponential: $p(\mu) = ce^{a\backslash\mu}$.
Default: $a = 0$.

_f = −41_ Gaussian distribution of time: $p(t) = c \exp[-(1.6651092(t - b)/a)^2]$, where a is the width at half maximum in shakes and b is the mean in shakes.
Defaults: a=no default, $b = 0$.

The built-in functions can be used only for the variables shown in Table 3.3. Any of the built-in functions can be used on SP cards, but only −21 and −31 can be used on SB cards. If a function is used on an SB card, only that same function can be used on the corresponding SP card. The combination of a regular table on the SI and SP cards with a function on the SB card is not allowed.

A built-in function on an SP card can be biased or truncated or both by a table on SI and SB cards. The biasing affects only the probabilities of the bins, not the shape of the function within each bin. If it is biased, the function is approximated within each bin by n equally probable groups such that the product of n and the number of bins is as large as possible but not over 300. Unless the function is −21 or −31, the weight of the source particle is adjusted to compensate for truncation of the function by the entries on the SI card.

Special defaults are available for distributions that use built-in functions.

1. If SB _f_ is present and SP _f_ is not, an SP _f_ with default input parameters is, in effect, provided by MCNP.

2. If only an SI card is present for RAD or EXT, an SP −21 with default input parameters is, in effect, provided.

3. If only SP −21 or SP −31 is present for DIR or EXT, an SI 0 1, for −21, or SI −1 1, for −31, is, in effect, provided.

4. If SI _x_ and SP −21 are present for RAD, the SI is treated as if it were SI 0 _x_.

5. If SI _x_ and SP −21 or SP −31 are present for EXT, the SI is treated as if it were SI −_x_ _x_.

CHAPTER 3
Source Specification

5. DSn   Dependent Source Distribution Card

> Form: DSn   option $J_1 \ldots J_k$
>   or: DSn   T $I_1 J_1 \ldots I_k J_k$
>   or: DSn   Q $V_1 S_1 \ldots V_k S_k$

$\quad$ n = distribution number ($n = 1,999$)

$\quad$ option = how the J_i are to be interpreted. Allowed values are:
$\qquad$ blank or H—source variable values in a continuous distribution, for scalar
$\qquad\qquad$ variables only
$\qquad$ L—discrete source variable values
$\qquad$ S—distribution numbers

$\quad$ T = values of the dependent variable follow values of the independent variable,
$\qquad$ which must be a discrete scalar variable

$\quad$ I_i = values of the independent variable

$\quad$ J_i = values of the dependent variable

$\quad$ Q = distribution numbers follow values of the independent variable, which must be
$\qquad$ a scalar variable

$\quad$ V_i = monotonically increasing set of values of the independent variable

$\quad$ S_i = distribution numbers for the dependent variable

> Default: DSn   H $J_1 \ldots J_k$

The DS card is used instead of the SI card for a variable that depends on another source variable, as indicated on the SDEF card. No SP or SB card is used. MCNP first determines the value of the independent variable as usual by sampling the probability function of the independent variable. Then the value of the dependent variable is determined according to the form of the DS card.

The first form of the DS card has several possibilities. If the SI card of the independent variable has a histogram distribution of n bins ($n + 1$ entries) and the DS card has the blank or H option, the DS card must have $n + 1$ entries to specify n bins. The first entry need not be zero. If the sampled value of the independent variable is $I_i + [f(I_{i+1} - I_i)]$, then the value of the dependent variable is $J_i + [f(J_{i+1} - J_i)]$, where the terms in f are used only for continuous distributions. The interpolation factor f always exists whether or not it is needed for the independent distribution.

If the L or S option is used on the DS card, n entries are required to specify n discrete values. It is not necessary for the distributions of the independent and dependent variables to be both discrete or both continuous. All combinations work correctly.

When the T option is used on a DS card, the sampled value of the independent variable is sought among the I_i, and if a match is found, the independent variable gets the value J_i. If no match is found, the dependent variable gets its default value. The purpose of the T option is to shorten the input when a dependent variable should usually get the default value.

When the Q option is used on a DS card, the V_i define a set of bins for the independent variable. The sampled value of the independent variable is compared with the V_i, starting with V_1, and if the sampled value is less than or equal to V_i, the distribution Si is sampled for the value of the dependent variable. The value of V_k must be greater than or equal to any possible value of the independent variable. If a distribution number S_i is zero, the default value for the variable is used. The Q option is the only form of the DS card that can be used when the distribution of the independent variable is a built-in function.

6.   SCn   Source Comment Card

   Form: SCn   comment

        n = distribution number (n=1,999)

   The comment is printed as part of the header of distribution n in the source distribution table and in the source distribution frequency table. The & continuation symbol is considered as part of the comment, not as a continuation command.

   Default: No comment.

Examples of the General Source

   Example 1:   SDEF   ERG=D1   POS=$x\ y\ z$ WGT=w
                 SI1   H E_1 E_2 ... E_k
                 SP1   D 0 P_2 ... P_k
                 SB1   D 0 B_2 ... B_k

This is a point isotropic source at x,y,z with a biased continuous-energy distribution and average source particle weight w. The starting cell is not specified. MCNP will determine it from the values of x, y, and z.

   Example 2:   SDEF   SUR=m AXS=$i\ j\ k$ EXT=D6
                 SB6   −31 1.5

This is a source on surface m. The presence of AXS and EXT implies that surface m is a sphere because AXS and EXT are not otherwise used together for sources on a surface. By default, the particles are emitted in a cosine distribution. They are emitted outward if the positive normal to the sphere is outward, which it is for all the spherical surface types but might not be if the sphere is specified as type SQ. The position on the surface is biased toward the direction i,j,k by an exponential bias (specified by −31) such that the maximum and minimum source particle weights are $e^{1.5} = 4.48$ and $e^{-1.5} = 0.223$. By default, MCNP provides the effect of two cards: SI6 −1 1 and SP6 −31 0.

Example 3:   SDEF   SUR=*m*   NRM=−1   DIR=D1   WGT=*w*
               SB1   −21 2

This is an inward-directed source on spherical surface *m*, assuming the positive normal of the surface is directed outward. If $w = \pi\, r^2$, where *r* is the radius of sphere *m*, this source in conjunction with a VOID card, a VOL card, and type 2 and type 4 tallies, is suitable for estimating the areas of surfaces and the volumes of cells. See page 2–183. By default, MCNP provides the effect of two cards: SI1 0 1 and SP1 −21 1. The directional bias by the SB1 card causes higher track density toward the center of the sphere, where presumably the cells of greatest interest lie, than it would be if the unbiased cosine distribution were used. This bias, incidentally, provides a zero-variance estimate of the (known) volume of the sphere *m*.

Example 4:   SDEF   ERG=D1   POS=*x y z*   CEL=m   RAD=D2
                       EXT=D3   AXS=*i j k*
               SP1   −3
               SI2 $r_1\, r_2$
               SI3   *l*

This source is distributed uniformly in volume throughout cell *m*, which presumably approximates a cylinder. The cell is enclosed by a sampling volume centered at *x,y,z*. The axis of the sampling volume is the line through *x,y,z* in the direction *i,j,k*. The inner and outer radii of the sampling volume are r_1 and r_2, and it extends along *i,j,k* for a distance ±1 from *x,y,z*. The user has to make sure that the sampling volume totally encloses cell *m*. The energies of the source particles are sampled from the Watt fission spectrum using the default values of the two parameters, making it a Cranberg spectrum. By default, MCNP interprets SI3 *l* as if it were actually SI3 −*l* +*l* and provides the effect of two cards: SP2 −21 1 and SP3 −21 0.

Example 5:   SDEF   SUR=*m*   POS=*x y z*   RAD=D1   DIR=1   CCC=*n*
               SI1   *r*

This is a monodirectional source emitted from surface *m* in the direction of the positive normal to the surface. The presence of POS and RAD implies that surface *m* is a plane because POS and RAD are not otherwise used together for sources on a surface. The position is sampled uniformly in area on the surface within radius *r* of point *x,y,z*. The user must make sure that point *x,y,z* actually lies on surface *m*. The sampled position is rejected and resampled if it is not within cookie-cutter cell *n*. The starting cell is found from the position and the direction of the particle. By default, MCNP interprets SI1 *r* as if it were actually SI1 0 *r* and provides the effect of card SP1 −21 1.

Example 6:   SDEF   POS D1   ERG FPOS D2
               SI1   L 5 3.3 6   75 3.3 6
               SP1   .3 .7
               DS2   S 3 4

        SI3   H 2 10 14
        SP3   D 0 1 2
        SI4   −3 *a* *b*

This is a point isotropic source in two locations, shown by two *x,y,z*'s on the SI1 card. The code will determine the starting cell. With probability .3 the first location will be picked and with probability .7 the second location will be chosen. Each location has a different energy spectrum, pointed to by the DS2 card. All other needed source variables will use their default values.

7. <u>SSW   Surface Source Write Card</u>

    Form: SSW S_1 S_2 (C_1 · · · C_k) S_3 S_n keyword=values

    The = signs are optional.
    S_i = problem surface number, with the appropriate sense of inward or outward
            particle direction, for which particle–crossing information is to be written to the
            surface source file WSSA. Macrobody surfaces are not allowed.
    C_i = problem cell number. A positive entry denotes an other–side cell. A negative
            entry specifies a just–left cell.
    SYM   m   symmetry option flag
            m=0, no symmetry assumed.
            m=1, spherical symmetry assumed. The list of problem surface numbers must
                contain only one surface and it must be a sphere.
            m=2, write particles to a surface bidirectionally. Otherwise, only particles going out
                of a positive surface and into a negative surface are recorded.
    PTY $n_1 n_2$ · · ·   tracks to record
            absent=record all tracks. This is the default.
            n_i=N, record neutron tracks
            n_i=P, record photon tracks
            n_i=E, record electron tracks
    CEL C_1 C_2 · · · C_n
            list of names of all the cells from which KCODE fission source neutrons are to be
            written, active cycles only.

    Default: SYM=0   PTY absent=record all particle types

    Use: Optional, as needed.

This card is used to write a surface source file or KCODE fission volume source file for use in a subsequent MCNP calculation. Care must be taken to include enough geometry beyond the specified surfaces to account for albedo effects. The card allows a list, in parentheses, of one or more cell names, positive or negative, after any of the surface names. If the list of cells is absent, any track that crosses the surface in the correct direction will be recorded. If the list is present, a

track will be recorded if it crosses the surface in the correct direction and is either entering a cell in the list whose entry is positive or leaving a cell in the list whose entry is negative.

If the SYM=1 option is used, the geometry inside the surface must be spherically symmetric and the materials must be symmetric. In very few cases will the SYM=1 option apply. The user must determine whether SYM=1 is appropriate for the problem. If the SYM=1 option is used, fewer words per particle need to be written to the surface source file and certain biasing options become available when reading the surface source file. In a KCODE calculation, particles are written only for active cycles.

Example: SSW   4   −7   19 (45 −46) 16 −83 (49) · · ·

A track that crosses surface 19 in the correct direction will be recorded only if it is either entering cell 45 or leaving cell 46. A track that crosses surface 83 in the correct direction will be recorded only if it is entering cell 49. A track that crosses surface 4 or 7 or 16 in the correct direction will be recorded regardless of what cells it happens to be leaving or entering.

Fission volume sources from a KCODE calculation can be written from active cycles only. The fission neutrons and prompt photons can then be transported in a subsequent calculation using the SSR surface source read fixed–source capability. In a KCODE criticality calculation the fission neutron sources and prompt photons produced from fission during each cycle are written to the WSSA surface source file if the SSW card has the CEL keyword followed by the names of all the cells from which fission source neutrons are to be written. Particles crossing specified surfaces can also be written by specifying S_i. The SYM=1 option (spherically symmetric surface source) cannot be used if CEL is specified.

Example: SSW   1   2 (3 4)   CEL 8 9

A track that crosses surface 2 in the correct direction will be recorded only if it enters cell 3 or 4. A track crossing surface 1 in the correct direction always will be recorded. And particles created from fission events in cells 8 and 9 will be recorded.

During execution, surface source information is written to the scratch file WXXA. Upon normal completion, WXXA becomes WSSA. If the run terminates abnormally, the WXXA file will appear instead of WSSA and must be saved along with the RUNTPE file. The job must be continued for at least one more history. At the subsequent normal termination, WXXA disappears and the correct surface source file WSSA is properly written.

8.   SSR   Surface Source Read Card

Form: SSR   keyword=values   keyword=values
The = signs are optional.

OLD $S_1 S_2 ... S_n$
list of problem surface numbers, a subset of the surfaces on the SSW card that
created the file WSSA, now called RSSA. Macrobody surfaces are not allowed.

CEL $C_1 C_2 ... C_n$
like OLD but for cells in which KCODE fission neutrons or photons were
written

NEW $S_{a1} S_{a2} ... S_{an} \quad S_{b1} S_{b2} ... S_{bn}$
problem surface numbers upon which the surface source is to start particles in
this run. The *n* entries may be repeated to start the surface source in *a,b,...*
transformed locations.

COL m collision option flag
$m = -1$ start from the surface source file only those particles that came directly
from the source without a collision
$m = 1$ start from the surface source file only those particles that had collisions
before crossing the recording surface
$m = 0$ start particles without regard to collisions

WGT x Each particle weight is multiplied by the constant x as it is accepted for
transport.

TR n transformation number. Track positions and velocities are transformed from
the auxiliary coordinate system (the coordinate system of the problem that wrote
the surface source file) into the coordinate system of the current problem, using
the transformation on the TRn card, which must be present in the INP file of the
current problem.

TR Dn Distribution number for a set of SIn, SPn, and SBn cards. If the surface
source is transformed into several locations, the SIn card lists the transformation
numbers and the SPn and SBn cards give the probabilities and bias of each
transformation.

PSC c a nonnegative constant that is used in an approximation to the PSC evaluation
for the probability of the surface source emitting a particle into a specified angle
relative to the surface normal.

The following four keywords are used only with spherically symmetric surface sources, that is,
sources generated with SYM=1 on the SSW card.

AXS $u \; v \; w$ Direction cosines that define an axis through the center of the surface
sphere in the auxiliary (original) coordinate system. This is the reference vector
for EXT.

EXT Dn n is the number of a distribution (SIn, SPn, and SBn cards) that will bias the
sampling of the cosine of the angle between the direction AXS and the vector
from the center of the sphere to the starting point on the sphere surface.

POA c Particles with a polar angle cosine relative to the source surface normal that
falls between 1 and c will be accepted for transport. All others are disregarded

and no weight adjustment is made.

BCW *r zb ze* $0 < zb < ze$ All particles with acceptable polar angles relative to the surface normal are started so that they will pass through a cylindrical window of radius *r*, starting at *zb* from the center of the source sphere, and ending at *ze* from the center. The axis of the cylinder is parallel to the z-axis of the auxiliary (original) coordinate system and contains the center of the source sphere. The weight of each source particle is adjusted to compensate for this biasing of position and direction.

Defaults: OLD   accept all surfaces in the original run.
            CEL   accept all cells in the original run.
            NEW   the surfaces in the OLD list.
            COL $m = 0$
            WGT   1
            TR   no transformation
            AXS   no axis
            EXT   no position bias
            POA $c = 0$
            BCW   no cylindrical window
            PSC   no default value

Use: Required for surface source problems.

The particle type is determined primarily by the type of the particle on the RSSA file, but particles incompatible with the problem mode are rejected without weight adjustment.

The number of particle histories reported in the output file for an SSR calculation is related to the number written to the WSSA file, so that proper normalization is preserved. However, a user may specify a different value on the NPS card than that used in the initial SSW calculation. If this NPS value is smaller than that of the initial calculation, an appropriate ratio of tracks will be rejected. If this NPS value is larger than that of the initial calculation, an appropriate duplication of tracks will be sampled. For example, if the SSW calculation used an NPS of 100 and the SSR calculation uses an NPS of 200, then every track is duplicated, each with a different random number seed and each with half the original weight. It is important to note that a larger value of NPS on the SSR calculation will indeed lower the tally errors until the weight variance contained on the RSSA file dominates (thus the reason to maximize the number of tracks on the RSSA file). Since the NPS value can readjust particle weights as described above, some variance reduction parameters (e.g., weight-window bounds) may need to be renormalized for SSR applications. See further discussion below.

An exact treatment of point detectors or DXTRAN spheres with a surface source is not possible because the $p(cos\theta)$ values required for the source contribution are not readily available. (See the

description of detector tallies in Chapter 2.) To use detectors or DXTRAN with a surface source, an approximate $p(cos\theta)$ must be specified on the SSR card. The most common azimuthally symmetric approximation for an angular emission probability density function is given by

$$p(cos\theta) = C_n(cos\theta)^n \qquad n \geq 0$$

The PSC=value entered is n, the power to which $p(cos\theta)$ is raised. C_n is a normalization constant calculated in MCNP and θ is the angle between the direction vector to the point detector and the surface normal at the point where the particle is to be started. Because surface crossings are recorded in only one direction specified on the SSW card, the limits on μ are always between 1 and 0. A PSC entry of zero specifies an isotropic angular distribution on the surface. An entry of 1 specifies a cosine angular distribution that produces an isotropic angular flux on the surface. In the case of a 1D spherical surface source of radius R, a cosine distribution is adequate if the point detector or DXTRAN sphere is more than 4R away from the source.

WARNING: Remember that the value entered is only an approximation. If the point detector or DXTRAN sphere is close to the source sphere and the approximation is poor, the answers will be **WRONG**.

Fission neutrons and photons written to the surface source file in a KCODE calculation can be used as a volume–distributed source in a subsequent calculation. A NONU card should be used so that fission neutrons and photons are not counted twice. Generally a TOTNU card also should be used. Total $\bar{\upsilon}$ is the default for KCODE sources but prompt $\bar{\upsilon}$ is the default for non–KCODE sources. Delayed gammas are ignored in MCNP. The keyword CEL specifies which fission cells to accept of those from the KCODE calculation that wrote the RSSA file.

The following observations and comments are relevant for using some variance reduction techniques with SSW/SSR calculations. This discussion applies to any technique that requires the input of normalized weight parameters (e.g., weight-window bounds, negative entries on the DD card, etc.).

1.) In general, weight window bounds generated in a SSW calculation are not useful in the SSR calculation, unless the tally identified on the WWG card of the SSW calculation is the same as that desired for the SSR calculation AND plenty of tracks contributed to that tally during the SSW calculation.

2.) A window generated in an SSR calculation will likely have to be renormalized in subsequent runs that use those windows, unless the value on the NPS card remains unchanged. If the value on the NPS card is changed, the WGT keyword on the SSR card can be used to renormalize the source weights to ensure weights are within the window in the source region. Whenever the WGT keyword is used in this fashion, tallies must be properly normalized by using this value on the SD card or the inverse of this value as a multiplier on the FM card.

Example 1:   Original run: SSW   1   2   3
              Current run: SSR   OLD 3 2 NEW 6 7   12 13 TR D5 COL 1
                             SI5   L 4 5
                             SP5   .4 .6
                       SB5 .3 .7

Particles starting on surface 1 in the original run will not be started in the current run because 1 is absent from the list of OLD surface numbers. Particles recorded on surface 2 in the original run will be started on surfaces 7 and 13 and particles recorded on surface 3 in the original run will be started on surfaces 6 and 12, as prescribed by the mapping from the OLD to the NEW surface numbers. The COL keyword causes only particles that crossed surfaces 2 and 3 in the original problem after having undergone collisions to be started in the current problem. The TR entry indicates that distribution function 5 describes the required surface transformations. According to the SI5 card, surfaces 6 and 7 are related to surfaces 3 and 2, respectively, by transformation TR4; surfaces 12 and 13 are related to 3 and 2 by TR5. The physical probability of starting on surfaces 6 and 7 is 40% according to the SP5 card, and the physical probability of starting on surfaces 12 and 13 is 60%. The SB5 card causes the particles from surfaces 3 and 2 to be started on surfaces 6 and 7 30% of the time with weight multiplier 4/3 and to be started on surfaces 12 and 13 70% of the time with weight multiplier 6/7.

Example 2:   Original run: SSW   3 SYM 1
              Current run: SSR   AXS 0 0 1   EXT D99
                             SI99   −1 .5   1
                             SP99   C .75 1
                             SB99   .5   .5

All particles written to surface 3 in the original problem will be started on surface 3 in the new problem, which must be exactly the same because no OLD, NEW, COL, or TR keywords are present. Because this is a spherically symmetric problem, indicated by the SYM 1 flag in the original run, the position on the sphere can be biased. It is biased in the z-direction with a cone bias described by distribution 99.

9.   KCODE   Criticality Source Card

    Form: KCODE   NSRCK RKK IKZ KCT MSRK KNRM MRKP KC8 ALPHA

              NSRCK = number of source histories per cycle
              RKK   = initial guess for k_{eff}
              IKZ   = number of cycles to be skipped before beginning tally accumulation
              KCT   = number of cycles to be done
              MSRK   = number of source points to allocate storage for
              KNRM   = normalize tallies by 0=weight / 1=histories
              MRKP   = maximum number of cycle values on MCTAL or RUNTPE
              KC8   = summary and tally information averaged over:

                            0 = all cycles
                            1 = active cycles only.
                ALPHA =   imposed value of alpha.

   Defaults: NSRCK=1000; RKK=1.0; IKZ=30; KCT=IKZ+100; MSRK=4500 or 2*
                NSRCK; KNRM=0; MRKP=6500;KC8=1; ALPHA=none.

   Use: This card is required for criticality calculations.

The KCODE card specifies the MCNP criticality source that is used for determining k_{eff}. The criticality source uses total fission nubar values unless overridden by a TOTNU NO card and applies only to neutron problems. In a MODE N,P problem, secondary photon production from neutrons is turned off during inactive cycles. SSW particles are not written during inactive cycles. If alpha is positive, use default implicit capture. See Chapter 1 for further information.

The NSRCK entry is the nominal source size for each cycle. The IKZ entry is the number of cycles to skip before beginning tally accumulation (this is important if the initial source guess is poor). The KCT entry specifies the number of cycles to be done before the problem ends. A zero entry means never terminate on the number of cycles but terminate on time. The MSRK is the maximum number of source points for which storage will be allocated. If an SRCTP file with a larger value of MSRK is read for the initial source, the larger value is used.

Fission sites for each cycle are those points generated by the previous cycle. For the initial cycle, fission sites can come from an SRCTP file from a similar geometry, from a KSRC card, or from a volume distribution specified by an SDEF card.

If in the first cycle the source being generated overruns the current source, the initial guess (RKK) is probably too low. The code then proceeds to print a comment, continues without writing a new source, calculates k'_{eff} , reads the initial source back in, and begins the problem using k'_{eff} instead of RKK. If the generated source again overruns the current source after the first cycle, the job terminates and either a better initial guess (RKK) or more source space (MSRK) should be specified on the next try.

KC8=0 causes tallies and summary table information to be for both active and inactive cycles and should not be used. KC8=0 also results in a strange MCTAL file normalization.

10. KSRC   Source Points for KCODE Calculation

   Form: KSRC $x_1\ y_1\ z_1$ $x_2\ y_2\ z_2$ ...

         x_i, y_i, z_i = location of initial source points

   Default: None. If this card is absent, an SRCTP source file or SDEF card must be supplied
             to provide initial source points for a criticality calculation.

Use: Optional card for use with criticality calculations.

This card contains up to NSRCK (*x,y,z*) triplets that are locations of initial source points for a KCODE calculation. At least one point must be in a cell containing fissile material and points must be away from cell boundaries. It is not necessary to input all NSRCK coordinate points. MCNP will start approximately (NSRCK/number of points) particles at each point. Usually one point in each fissile region is adequate, because MCNP will quickly calculate and use the new fission source distribution. The energy of each particle in the initial source is sampled from a Watt fission spectrum hardwired into MCNP, with $a = 0.965$ MeV and $b = 2.29$ MeV^{-1}.

An SRCTP file from a previous criticality calculation can be used instead of a KSRC card. If the current problem has a lot in common with the previous problem, using the SRCTP file may save some computer time. Even if the problems are quite different, the SRCTP file may still be usable if some of the points in the SRCTP file are in cells containing fissile material in the current problem. Points in void or zero importance cells will be deleted. The number of particles actually started at each point will be such as to produce approximately NSRCK initial source particles.

An SDEF card also can be used to sample initial source points in fissile material regions. The SDEF card parameters applicable to volume sampling can be used: CEL, POS, RAD, EXT, AXS, X, Y, Z; and CCC, ERG, and EFF. If a uniform volume distribution is chosen, the early values of k_{eff} will likely be low because too many particles are put near where they can escape, just the opposite of the usual situation with the KSRC card. Do not change the default value of WGT for a KCODE calculation.

11.   ACODE   Alpha Eigenvalue Source Card

Form: ACODE   NSRCK RKK IKZ KCT MSRK KNRM KALPHA KALSAV
                  KALREG MRKP ALMIN

NSRCK = number of source histories per cycle
RKK   = initial guess for k_{eff}
IKZ   = number of cycles to be skipped before beginning tally accumulation
KCT   = number of cycles to be done
MSRK   = number of source points to allocate storage for
KNRM   = normalize tallies by 0=weight / 1=histories
KALPHA = alpha estimator: 1/2/3 = col/abs/trk
                4 = differential operator perturbation estimator
KALSAV = cycle to start averaging alphas
KALREG = cycle to start ln-ln regression and reduce number of internal
                settle cycles
MRKP   = maximum number of cycle values on MCTAL or RUNTPE
ALPHA = initial guess of alpha
ALMIN =   minimum floor value of alpha.

Defaults: NSRCK=1000; RKK=1.0; IKZ=30; KCT=IKZ+100; MSRK=4500 or 2*
NSRCK; KNRM=0; KALPHA=1; KALSAV=automatic;
KALREG=KALSAV+2; MRKP=6500; ALPHA=0; ALMIN=0

The alpha eigenvalue is $N = N_o e^{\alpha t}$, where N is the neutron population at some time t that builds up from some initial population N_o. Alpha is of interest for comparison to benchmark calculations and comparison to deterministic transport codes such as DANTSYS. The alpha capability allows positive and negative alpha searches and a fixed positive or negative alpha value to be used in a k_{eff} eigenvalue calculation. Negative values of alpha can result in a time creation (n,2n) delta scattering reaction. Positive alpha is treated as time absorption. We recommend the use of default implicit capture for positive alpha problems.

12. Subroutines SOURCE and SRCDX

If SDEF, SSR, or KCODE cards are not present in the INP file, a user supplied source is assumed and is implemented by calling subroutine SOURCE, which the user must provide. Chapter 4 has examples of a SOURCE subroutine and discusses the SRCDX subroutine. The parameters that must be specified within the subroutine are listed and defined on page 3–49. Prior to calling subroutine SOURCE, isotropic direction cosines u,v,w (UUU,VVV,WWW) are calculated and need not be specified if you want an isotropic distribution.

The SIn, SPn, and SBn cards also can be used with the SOURCE subroutine, although modifications to other parts of MCNP may be required for proper initialization and to set up storage. A random number generator RANG( ) is available for use by SOURCE for generating random numbers between 0 and 1. Up to 50 numerical entries can be entered on each of the IDUM and RDUM cards for use by SOURCE. The IDUM entries must be integers and the RDUM entries floating point numbers.

If you are using a detector or DXTRAN and your source has an anisotropic angular distribution, you will need to supply an SRCDX subroutine to specify PSCs for each detector or DXTRAN sphere (see Chapters 2 and 4).

There are unused variables stored in the particle bank that are reserved for the user called SPARE(M), M=1,MSPARE, where MSPARE=3. Depending on the application, you may need to reset them to 0 in SOURCE for each history; MCNP does not reset them.

E.   Tally Specification

The tally cards are used to specify what type of information the user wants to gain from the Monte Carlo calculation; that is, current across a surface, flux at a point, heating in a region, etc. This

information is requested by the user by using a combination of the following cards. To obtain tally results, only the Fn card is required; the other tally cards provide various optional features.

Mnemonic	Card Type	Page
Fna	Tally type	3–75
FCn	Tally comment	3–84
En	Tally energies	3–84
Tn	Tally times	3–85
Cn	Tally cosines	3–86
FQn	Tally print hierarchy	3–87
FMn	Tally multiplier	3–88
DEn/DFn	Dose energy/Dose function	3–92
EMn	Energy multiplier	3–93
TMn	Time multiplier	3–94
CMn	Cosine multiplier	3–94
CFn	Cell flagging	3–95
SFn	Surface flagging	3–96
FSn	Tally segment	3–96
SDn	Segmented volume/area	3–98
FUn	TALLYX input	3–99
TFn	Tally fluctuation print	3–101
DD	Detector and DXTRAN diagnostics	3–103
DXT	DXTRAN	3–104
FTn	Special treatments	3–106

The n is a user-chosen tally number < 999; choices of n are given in the following section. When a choice of n is made for a particular tally type, any other input card used with that tally (such as En for energy bins) is given the same value of n by the user.

Much of the information on these cards is used to describe tally "bins," subdivisions of the tally space into discrete and contiguous increments such as cosine, energy, or time. Usually when the user subdivides a tally into bins, MCNP can also provide the total tally summed over appropriate bins (such as over energy bins). Absence of any bin specification card results in one unbounded bin rather than one bin with a default bound. No information is printed about the limits on the unbounded bin.

If there are reflecting surfaces or periodic boundaries in the problem, the user may have to normalize the tallies in some special way (this can be done by setting the weight of the source particles or by using the FMn card).

Printed with each tally bin is the relative error of the tally corresponding to one standard deviation. These errors *cannot* be believed reliable (hence neither can the tally itself) unless the error is fairly

low. Results with errors greater than 50% are useless, results between 20% and 50% can be believed to within a factor of a few, results between 10% and 20% are questionable, results less than 10% are generally (but not always) reliable except for detectors, and detector results are generally reliable below 5%. One bin of every tally is designated for the tally fluctuation charts at the end of the output file. This bin is also used for the weight window generator. It also is subject to ten statistical checks for tally convergence, including calculation of the variance of the variance (VOV). The VOV can be printed for all bins in a tally by using the DBCN card.

1. Fna   Tally Cards

Seven basic neutron tally types, six basic photon tally types, and four basic electron tally types are available in MCNP as standard tallies. All are normalized to be per source particle unless changed by the user with a TALLYX subroutine or normed by weight in a criticality (KCODE) calculation.

Mnemonic	Tally Description	Fn units	*Fn units
F1:N or F1:P or F1:E	Current integrated over a surface	particles	MeV
F2:N or F2:P or F2:E	Flux averaged over a surface	particles/cm^2	MeV/cm^2
F4:N or F4:P or F4:E	Flux averaged over a cell	particles/cm^2	MeV/cm^2
F5a:N or F5a:P	Flux at a point or ring detector	particles/cm^2	MeV/cm^2
F6:N or F6:N,P or F6:P	Energy deposition averaged over a cell	MeV/g	jerks/g
F7:N	Fission energy deposition averaged over a cell	MeV/g	jerks/g
F8:P or F8:E or F8:P,E	Energy distribution of pulses created in a detector	pulses	MeV
+F8:E	Charge deposition	charge	N/A

The tallies are identified by tally type and particle type as follows. Tallies are given the numbers 1, 2, 4, 5, 6, 7, 8, or increments of 10 thereof, and are given the particle designator :N, :P, or :E (or :N,P only in the case of tally type 6 or :P,E only in the case of tally type 8). Thus you may have as many of any basic tally as you need, each with different energy bins or flagging or anything else. F4:N, F14:N, F104:N, and F234:N are all legitimate neutron cell flux tallies; they could all be for the same cell(s) but with different energy or multiplier bins, for example. Similarly F5:P, F15:P, and *F305:P are all photon point detector tallies. Having both an F1:N card and an F1:P card in the same INP file is not allowed. The tally number may not exceed three digits.

Tally types 1, 2, 4, and 5 are normally weight tallies (particles in the above table); however, if the Fn card is flagged with an asterisk (for example, *F1:N), energy times weight will be tallied. The

asterisk flagging can also be used on tally types 6 and 7 to change the units from MeV/g to jerks/g (1 jerk = 1 GJ = 10^9 J). The asterisk on a tally type 8 converts from a pulse height tally to an energy deposition tally. All of the units are shown in the above table.

Tally type 8 can also be flagged with a plus (+) to convert it from an energy deposition tally (flagged with an asterisk) to a charge deposition tally. The tally is the negative particle weight for electrons and the positive weight for positrons. The +F8 tally can be checked against an F1:E type surface tally. See page 3–80 for an example.

Only the F2 surface flux tally requires the surface area. The area calculated is the total area of the surface that may bound several cells, not a portion of the surface that bounds only a particular cell. If you need only the segment of a surface, you might segment the full surface with the FSn card (see page 3–96) and use the SDn card (see page 3–98) to enter the appropriate values. You can also redefine the geometry as another solution to the problem. The detector total is restricted to 20. The tally total is limited to 100. Note that a single type 5 tally may create more than one detector.

    1.   Surface and Cell Tallies (tally types 1, 2, 4, 6, and 7)

Simple Form: Fn:pl $S_1 \dots S_k$
General Form: Fn:pl $S_1 (S_2 \dots S_3) (S_4 \dots S_5) S_6 S_7 \dots$

    n = tally number.
    pl = N or P or N,P or E
    S_i = problem number of surface or cell for tallying, or T.

Only surfaces bounding cells and listed in the cell card description can be used on F1 and F2 tallies. Tally 6 does not allow E. Tally 7 allows N only.

In the simple form above, MCNP creates k surface or cell bins for the requested tally, listing the results separately for each surface or cell. In the more general form, a bin is created for each surface or cell listed separately and for each collection of surfaces or cells enclosed within a set of parentheses. Entries within parentheses also can appear separately or in other combinations. Parentheses indicate that the tally is for the union of the items within the parentheses. For unnormalized tallies (tally type 1), the union of tallies is a sum, but for normalized tallies (types 2, 4, 6, and 7), the union results in an average. See page 3–80 for an explanation of the repeated structure and lattice tally format.

The symbol T entered on surface or cell Fn cards is shorthand for a region that is the union of all of the other entries on the card. A tally is made for the individual entries on the Fn card plus the union of all the entries.

If a tally label of the surfaces or cells in the output requires more than eleven characters, including spaces, MCNP defines an alphabetical or numerical designator for printing purposes. The

designator [for example, G is (1 2 3 4 5 6)] is printed with the tally output. This labeling scheme is usually required for tallies over the union of a long list of surfaces or cells.

Example 1:   F2:N 1 3 6 T

This card specifies four neutron flux tallies, one across each of the surfaces 1, 3, and 6 and one which is the average of the flux across all three of the surfaces.

Example 2:   F1:P (1 2) (3 4 5) 6

This card provides three photon current tallies, one for the sum over surfaces 1 and 2; one for the sum over surfaces 3, 4, and 5; and one for surface 6 alone.

Example 3:   F371:N (1 2 3) (1 4) *T*

This card provides three neutron current tallies, one for the sum over surfaces 1, 2, and 3; one for the sum over surfaces 1 and 4; and one for the sum over surfaces 1, 2, 3, and 4. The point of this example is that the *T* bin is not confused by the repetition of surface 1.

Another case for study is in the DEMO example in Chapter 5.

2.   Detector Tallies (tally type 5)

Form for point detectors: Fn:pl X Y Z $\pm R_o$

n =   tally number.
pl   =   *N* for neutrons or *P* for photons,
X Y Z =   location of the detector point.
$\pm R_o$ =   radius of the sphere of exclusion:
            in centimeters, if R_o is entered as positive,
            in mean free paths, if entered as negative. (Negative entry illegal in a void.)

Form for ring detectors: Fna:pl a_o r $\pm R_o$

n	=	tally number.
a	=	the letter X, Y, or Z.
pl	=	N for neutrons or P for photons.
a_o	=	distance along axis "a" where the ring plane intersects the axis.
r	=	radius of the ring in centimeters.
$\pm R_o$	=	same meaning as for point detectors, but describes a sphere about the point selected on the ring.

Default: None.

Use: You are encouraged to read about detectors in Chapter 2 before using them because they are very susceptible to unreliable results if used improperly. Remember that contributions to a detector are not made through a region of zero importance. Ring (rather than point) detectors should be used in all problems with axial symmetry. A detector located right on a surface probably will cause trouble. Detectors and DXTRAN can be used in problems with the $S(\alpha,\beta)$ thermal treatment, but the $S(\alpha,\beta)$ contributions are approximate (see page 2–53. Detectors used with reflecting, white, or periodic surfaces give wrong answers (see page 2–92). Consider using the PDn and DDn cards.

For more than one detector with the same *n* or *na* designation, sets of the above input parameters (quadruplets for Fn or triplets for Fna) are simply continued on the same Fn or Fna card. If more than one detector of the same type (an F5:N and an F15:N, for example) are at the same location, the time-consuming contribution calculation upon collision is made only once and not independently for each detector, according to the rules in Chapter 2. Thus it is inexpensive to add more than one detector (each with a different response function, for example) at the same location as another.

The printout for detectors is normally in two parts: (1) the total of all contributions to the detector (as a function of any defined bins such as energy) and (2) the direct (or uncollided) contribution to the detector from the source. The direct contribution is always included in the total of all contributions. Adding the symbol ND at the end of a type 5 detector tally card inhibits the separate printing of the direct contribution for that tally. In coupled neutron/photon problems, the direct contribution in photon tallies is from photons created at neutron collisions.

Rules of Thumb for R_o: R_o should be about 1/8 to 1/2 mean free path for particles of average energy at the sphere and zero in a void. Supplying R_o in terms of mean free path will increase the variance and is not recommended unless you have no idea how to specify it in centimeters. R_o must not encompass more than one material. MCNP cannot check this and the consequences may be wrong answers.

3.   Pulse Height Tally (tally type 8)

Simple Form: Fn:pl $S_1 ... S_k$
General Form: Fn:pl $S_1 (S_2 ... S_3) (S_4 ... S_5) S_6 S_7 ...$

n = tally number.
pl = P, E or P,E
S_i = problem number of cell for tallying, or T.

The F8 tally provides the energy distribution of pulses created in a detector by radiation and is called a pulse height tally. See page 3–80 for an explanation of the repeated structure and lattice tally format. The F8 card is used to list the cell bins, just like an F4 tally. The union of tallies produces a tally sum, not an average. Cell, user, and energy bin cards are allowed. Flagging, segment, multiplier, time, and cosine bins are not allowed. The energy bins accumulate the energy deposited in a cell by all the tracks of a history rather than the energy of the scoring tracks. Both photons and electrons will be tallied if present, even if only E or only P is on the F8 card. In other words, the F8:P, F8:E, and F8:P,E are all equivalent tallies. An asterisk on the F8 card converts the tally from a pulse height tally to an energy deposition tally. A plus on the F8 card converts the tally from a pulse height tally to a charge deposition tally in units of electron charge. Energy binning is not recommended.

If the F8 card has an asterisk or plus and there is no energy binning (E8 card), variance reduction of all kinds is allowed. In this special case, the total energy deposition or charge is correct even though the tallies in energy bins would not be correct. However, there still can be problems as described on page 2–83.

Care must be taken when selecting energy bins for a pulse height tally. It is recommended that a zero bin and an epsilon bin be included such as

E8   0   1E-5   1E-3   1E-1   ...

The zero bin will catch nonanalog knock–on electron negative scores. The epsilon (1E-5) bin will catch scores from particles that travel through the cell without depositing energy. See page 2–83.

The pulse height tally is a radical departure from other MCNP tallies. All other tallies are estimates of macroscopic variables, such as flux, whose values are determined by very large numbers of microscopic events. The pulse height tally records the energy or charge deposited in a cell by each source particle and its secondary particles. For other tallies it is not necessary to model microscopic events realistically as long as the expectation values of macroscopic variables were correct. For the pulse height tally, microscopic events must be modeled much more realistically.

The departures from microscopic realism in MCNP are everywhere. The number, energies, and directions of the secondary neutrons and photons from a neutron collision are sampled without any

correlation between the particles and with no regard for the conservation of energy. Modeling the fluctuations in the number of fission neutrons is limited to choosing between the integer next larger and the integer next smaller than the average number of fission neutrons. The fluctuations in the energy loss rate of an electron are not correlated with the production of knock-ons and x rays. The variance reduction schemes in MCNP create unrealistic histories that nevertheless give correct results for macroscopic tallies.

Problems that give correct pulse height tallies are <u>severely</u> limited. The only possible variance reduction scheme is biasing of the source itself. **<u>CAUTION</u>**: The pulse height tally does not work well with neutrons (and is not allowed) because of the nonanalog nature of neutron transport that departs from microscopic realism at every turn. One can have a neutron source in a MODE N,P or N,P,E problem, but only the photons and electrons can be tallied on the F8 card. The F8 tally can be used effectively in photon problems. Electron problems may give correct results as long as the tally cells are thick enough for the errors in the energy loss rate to average out. MCNP tries to detect conditions in a problem that would invalidate pulse height tallies, but it is not able to catch all of them. The user must ascertain that his problem does not violate the necessary conditions for obtaining correct answers.

Scoring the pulse height tally is done at the end of each history. A temporary account of energy is kept for each pulse height tally cell bin during the history. At the source, the source cell is credited with the energy times the weight of the source particle. When a particle crosses a surface, the energy times the weight of the particle is subtracted from the account of the cell that it is leaving and is added to the account of the cell that it is entering. The energy here is the kinetic energy of the particle plus 1.022016 MeV if it is a positron. At the end of the history, the account in each tally cell is divided by the source weight. The resulting energy determines which energy bin the score is put in. The value of the score is the source weight, or, if there is an asterisk on the F8 card, the source weight times the energy in the account. If there is a plus on the F8 card, the tally is the negative particle weight for electrons and the positive weight for positrons. The +F8 charge deposition tally can be checked against an electron F1:E surface tally with the FT ELC option if the volume of the +F8 is exactly enclosed by the surfaces on the F1:E card. For example, if cell 1 is enclosed by spherical surface 2, then the following tallies give the same result provided the two F1 current tally bins (in – out) are properly subtracted.

$$+F8:E \quad \quad and \quad \quad F1:E \ \ 2$$
$$FT1 \ \ ELC \ \ 1$$
$$C1 \ \ 0 \ \ 1$$

Note that the meaning of the energy bins of a pulse height tally is entirely different from the meaning of the energy bins of the other tallies in MCNP. The normal meaning of energy bins is the energy of a scoring track. The meaning of the energy bins of a pulse height tally is the energy deposited in a cell bin by all the tracks of a history.

4.   Surface, Cell, and Pulse Height Tallies for Repeated Structures and Lattices
     (tally types 1, 2, 4, 6, 7, and 8)

Simple Form: Fn:pl $S_1 \dots S_k$
General Form: Fn:pl $S_1 (S_2 \dots S_3) ((S_4 S_5) < (C_1 C_2 [I_1 \dots I_2]) < (C_3 C_4 C_5)) \dots$

n = tally number.
pl = N or P or N,P or E.
S_i = problem number of a surface or cell for tallying, U=#, or T.
C_i = problem number of a cell filled with a universe, or U=#.
= problem number of a universe used on a fill card.
I_i = index data for a lattice cell element, with three possible formats:

I_1	Indicating the I_{1th} lattice element of cell C_2, as defined by the FILL array.
$I_1 : I_2 \quad I_3 : I_4 \quad I_5 : I_6$	Range of one or more lattice elements. Use the same format as on the FILL card.
$I_1 \ I_2 \ I_3, \ I_4 \ I_5 \ I_6$	Indicating lattice element (I_1, I_2, I_3), (I_4, I_5, I_6), etc See LAT and FILL cards for indices explanation.

In the simple form, MCNP creates k surface or cell bins for the requested tally, listing the results separately for each surface or cell. In the more general form, a bin is created for each surface or cell listed separately and for each collection of surfaces or cells enclosed within a set of parentheses. A tally bin can involve a single tally level or multiple tally levels. Tallies involving repeated structure and lattice geometries can use either form.

If a tally label of the surface or cells in a given bin exceeds eleven characters, including spaces, an alphabetical or numerical designator is defined for printing purposes. The designator [G is (1 2 3 4 5 6)], for example, would be printed with the tally output. This labeling scheme is usually required for tallies over the union of a long list of surfaces or cells or with repeated structure tallies.

Some operators and nomenclature need to be introduced before the explanation of repeated structures and lattice tallies. The left arrow or less than symbol < is used to identify surfaces or cells within levels of repeated structures. See page 3–26 for explanation of geometry levels. A tally bin that includes one or more left arrows implies multiple levels, called a chain. Multiple entries enclosed by parentheses at any level of a tally chain indicates the union of the items. Brackets [ ] immediately following a filled lattice cell identify one or more elements of that lattice. The input tally bin chain involving multiple levels <u>MUST</u> be enclosed by an outer set of parentheses.

F4:N   (5 < 4 < 2 [1 0 0])

This example could specify an F4 tally in cell 5 when it is in cell 4, when cell 4 is in cell 2, which is a lattice, and only in lattice element [1,0,0]. While any cell (lattice, filled, or simple) can be

entered as a tally cell (e.g., S_1 through S_5), only cells filled with a universe can be used in higher levels (e.g., C_1 through C_5.) See General Form, page 3–80.

Multiple bin format: In addition to multiple levels, multiple entries can be used in each level of the tally chain resulting in multiple output bins. Within the parentheses required around the tally bin chain, other sets of parentheses can be used to indicate the union of cells as in a simple tally description, resulting in fewer output tally bins.

$$((S_4\ S_5) < (C_1\ C_2\ [I_1\ ...\ I_2]) < (C_3\ C_4\ C_5))$$

This example results in one output tally bin and will be the union of the tally in S_4 plus S_5 that fill C_1 or C_2 [elements I_1 ... I_2] and when C_1 or C_2 fills cells C_3, C_4, or C_5. Removing the first and third inner parentheses:

$$(S_4\ S_5 < (C_1\ C\ _2\ [I_1\ ...\ I_2]) < C_3\ C_4\ C_5)$$

results in the creation of 2*1*3=6 bins as follows:

$$(S_4 < (C_1\ C_2\ [I_1\ ...\ I_2]) < C_3),\quad (S_5 < (C_1\ C_2\ [I_1\ ...\ I_2]) < C_3),$$
$$(S_4 < (C_1\ C_2\ [I_1\ ...\ I_2]) < C_4),\quad (S_5 < (C_1\ C_2\ [I_1\ ...\ I_2]) < C_4),$$
$$(S_4 < (C_1\ C_2\ [I_1\ ...\ I_2]) < C_5),\quad (S_5 < (C_1\ C_2\ [I_1\ ...\ I_2]) < C_5),$$

The repeated structure/lattice input tally bin format with levels that have multiple entries automatically creates multiple output tally bins. The total number of bins generated is the product of the number of entries at each level. If parentheses enclose all entries at a level, the number of entries at that level is one and results in the union of all those entries. See Chapter 4, page 4–46, for a caution when tallying a union of overlapping regions. For unnormalized tallies (type 1, 8), the union is a sum. For normalized tallies (type 2, 4, 6, 7), the union is an average. A symbol T on the tally line creates an additional tally bin that is the union or total of all the other tally bins.

Brackets: Brackets [ ] enclose index data for lattice cell elements. Brackets make it possible to tally on a cell or surface only when it is within the specified lattice elements. The brackets must immediately follow a filled lattice cell. Listing a lattice cell without brackets will produce a tally when the tally cell or surface is in any element of the lattice, provided the tally cell or surface fills an entry at all other levels in the chain. The use of brackets is limited to levels after the first < in the tally specification.

To tally within lattice elements of a real world (level zero) lattice cell, use the special syntax that follows. Cell 3 contains material 1 and is bounded by four surfaces. The F4 card specifies a tally only in lattice element (0,0,0). This syntax is required because brackets can only follow a <.

```
3    1   −1.0    −1   2   3   4   lat=1
                  .

F4:N      (3 < 3 [0 0 0])
```

Universe format: The universe format, U=#, is a shorthand method of including all cells and lattice elements <u>filled</u> by universe #. This format can be used in any level of the tally chain. The following example illustrates valid shorthand U=# descriptions in the left column. The right column shows the tally after the shorthand has been expanded. Cells 4 and 5 are filled with universe 1.

	<u>shorthand</u>	<u>expanded</u>
F4:N	u=1	4 5
	(u=1)	(4 5)
	(u=1 < 2 < 3)	(4 5 < 2 < 3)
	((u = 1) < 2 < 3)	((4 5) < 2 < 3)
	(1 < u = 1 < 2 < 3)	(1 < 4 5 < 2 < 3)
	(1 < (u = 1) < 2 < 3)	(1 < (4 5) < 2 < 3)

In complex geometries, the U=# format should be used sparingly, especially with the multiple bin format. If 100 cells are filled by universe 1 and 10 cells are filled by universe 2, then the tally

$$F4:N \quad (u = 1 < u = 2)$$

will create 1000 output tally bins. However,

$$F4:N \quad ((u = 1) < (u = 2))$$

will create only one output tally bin.

Use of SDn card: When making tallies in repeated structure and lattice geometries, often a volume or area is required and MCNP will be unable to calculate it. Possibly the geometry causes the calculation to fail. A universe can be repeated a different number of times in different cells and the code has no way to determine this. There are two distinct options for entries on the SDn card relating to repeated structures and they cannot be mixed within a single tally.

The first option is to enter a value for each <u>first level</u> entry on the related F card. If the entry on the F card is the union of cells, the SD card value will be the volume of the union of the cells. The following examples illustrate Fn card tally descriptions in the left column. The right column shows the SDn card entries.

F4:N	(1 < 4 5 6 < 7 8)	SD4	V_1
	(1 2 3 < 4 5 6 < 7 8)		$V_1 \, V_2 \, V_3$
	(1 2 3 < (4 5 6) < (7 8))		$V_1 \, V_2 \, V_3$
	((1 2 3) < 4 5 6 < 7 8)		V_{123L}

V_i = volume of cell i and V_{123} = volume of the union of cells 1, 2, and 3. Even though the first line creates six tally bins, only one SD value is entered. This divisor is applied to all bins generated by the input tally bin. You do not need to know the number of bins generated by each input tally bin

in order to use the SD card. The last line is the union of cells 1, 2, and 3 and only one divisor is entered on the SD card.

The second option is to enter a value for each <u>bin</u> generated by the Fn card.

F4:N	(1 < 4 5 6 < 7 8)	SD4 $\quad V_1^1 \; V_1^2 \; V_1^3 \; V_1^4 \; V_1^5 \; V_1^6$
	(1 2 3 < 4 5 6 < 7 8)	$V_1^1 \; V_2^2 \; V_3^3 \; V_1^4 \; V_2^5 \; V_3^6 \; V_1^7 \; ... \; V_1^{16} \; V_2^{17} \; V_3^{18}$
	(1 2 3 < (4 5 6) < (7 8))	$V_1 \; V_2 \; V_3$
	((1 2 3) < 4 5 6 < 7 8)	$V_{123}^1 \; V_{123}^2 \; V_{123}^3 \; V_{123}^4 \; V_{123}^5 \; V_{123}^6$

V_1^j = volume of cell i for bin j and V_{123}^j = volume of the union of cells 1, 2, and 3 for bin j. If cell i is repeated the same number of times in all six bins generated by the first line above, then all six SD values for this input bin will be the same ($V_1^1 = V_1^2 = V_1^3...$). However, if cell 1 is repeated a different number of times in each bin, then different SD values should be entered. The volume is multiplied by the number of times it is repeated. In these cases, the total cell 1 volume for each generated bin will not be calculated. The bin generation order is explained previously in the Fn card section. For the first line above, the bin order is (1<4<7), (1<5<7), (1<6<7), (1<4<8), (1<5<8), and (1<6<8). The second line above generated 18 tally bins, and 18 SD values are required in the proper sequence. This option requires the knowledge of both the number <u>and</u> sequence of bins generated by each input tally bin.

2. <u>FCn   Tally Comment Card</u>

    Form: FCn   any desired information

    Default: No comment.

    Use: Recommended for modified tally.

Anything entered after FCn will appear as the title heading of tally Fn. This card is particularly useful when tallies are modified in some way, so later readers of the output will be warned of modified or nonstandard tallies. The FCn card can be continued only by blanks in columns 1–5 on suceeding lines. The & continuation symbol is considered as part of the comment, not as a continuation command.

3. <u>En   Tally Energy Card</u>

    Form: En $E_1 \, ... \, E_k$

            n =   tally number.
            E_i =   upper bound (in MeV) of the ith energy bin for tally n.

Default: If the En card is absent, there will be one bin over all energies unless this default has been changed by an E0 card.

Use: Required if EMn card is used.

The entries on the En card must be entered in the order of increasing magnitude. If a particle has an energy greater than the last entry, it will not be tallied, but you will be warned that this has happened. If the last entry is greater than the upper energy limit E_{max} specified on the PHYS card, the last bin will be lowered to E_{max}. If there are several bins above E_{max}, the extra bins are eliminated.

An E0 (zero) card can be used to set up a default energy bin structure for all tallies. A specific En card will override the default structure for tally n.

MCNP automatically provides the total over all specified energy bins, but the total can be inhibited for a given tally by putting the symbol NT at the end of the En card for that tally. The symbol C at the end of the line causes the bin values to be cumulative and the last energy bin is also the total over all energy.

Example: E11 .1 1 20

This will separate an F11 current tally into four energy bins: (1) from the energy cutoff to 0.1 MeV, (2) from 0.1 to 1.0 MeV, (3) from 1.0 to 20.0 MeV, and (4) a total over all energy.

4. <u>Tn   Tally Time Card</u>

Form: Tn $T_1 ... T_k$

n = tally number.
T_i = upper bound (in shakes) of the ith time bin for tally n.

Default: If the Tn card is absent, there will be one bin over all times unless this default has been changed by a T0 card.

Use: Required if TMn card is used. Consider FQn card.

The times on the Tn card must be entered in order of increasing magnitude. If a particle has a time greater than the last entry on the Tn card, it will not be tallied, but you will be warned that this has happened. The last time bin entry should always be less than or equal to the time cutoff (see CUT card) except for point detectors. If time bins greater than the time cutoff are entered for tallies other than point detectors, the first bin limit over the cutoff will be lowered to the cutoff. All higher bins will be eliminated. For point detector tallies, time bins can exceed the time cutoff so that particles will score at detectors remote from the main body of the system. Setting the time cutoff lower than

the last time bin will inhibit unproductive transport of slow neutrons in the system and will increase the efficiency of the problem.

A T0 (zero) card can be used to set up a default time bin structure for all tallies. A specific Tn card will override the default structure for tally *n*.

MCNP automatically provides the total over all specified time bins, but the total can be inhibited for a given tally by putting the symbol NT at the end of the Tn card for that tally. The symbol C at the end of the line causes the bin values to be cumulative and the last time bin is also the total over all time.

> Example: T2   −1   1   1.0+37   NT

This will separate an F2 flux surface tally into three time bins: (1) from −∞ to −1.0 shake, (2) from −1.0 shake to 1.0 shake, and (3) from 1.0 shake to 1.0e37 shakes, effectively infinity. No total bin will be printed in this example.

5.   Cn   Cosine Card (tally type 1 only)

> Form: Cn C_1 ... C_k
>
> > n = tally number.
> > C_i = upper cosine limit of the ith angular bin for surface current tally *n*.
> > > $C_1 > -1$. $C_k = 1$.
>
> Default: If the Cn card is absent, there will be one bin over all angles unless this default has been changed by a C0 card.
>
> Use: Tally type 1. Required if CMn card is used. Consider FQn card.

The angular limits described by the Cn card are defined with respect to the positive normal to the surface at the particle point of entry. An FTn card with an FRV U V W option can be used to make the cosine bins relative to the vector u,v,w. The positive normal to the surface is always in the direction of a cell that has positive sense with respect to that surface. The cosines must be entered in increasing order, beginning with the cosine of the largest angle less than 180° to the normal and ending with the normal (cos=1). A lower bound of −1 is set in the code and should not be entered on the card. The last entry must always be 1.

A C0 (zero) card can be used to set up a default angular bin structure for all tallies. A specific Cn card will override the default structure for tally n. Note that the selection of a single cosine bin for an F1 tally gives the total and not the net current crossing a surface.

MCNP does not automatically provide the total over all specified cosine bins, but the total can be generated for a given tally by putting the symbol *T* at the end of the Cn card for that tally. The symbol C at the end of the line causes the bin values to be cumulative and the last cosine bin is also the total over all cosine bins.

Example: C1   −.866   −.5   0   .5   .866   1

This will tally currents within the angular limits (1) 180° to 150°, (2) 150° to 120°, (3) 120° to 90°, (4) 90° to 60°, (5) 60° to 30°, and (6) 30° to 0° with respect to the positive normal. No total will be provided.

As an example of the relation between a surface normal and sense for the C1 card, consider a source at the origin of a coordinate system and a plane (PY) intersecting the +*y* axis. An entry of 0 and 1 on the C1 card will tally all source particles transmitted through the plane in the 0 to 1 cosine bin (0° to 90°) and all particles scattered back across the plane in the −1 to 0 cosine bin (90° to 180°). A plane (PY) intersecting the −*y* axis will result in a tally of all source particles transmitted through the second plane in the −1 to 0 bin (90° to 180°) and all particles scattered back across the plane in the 0 to 1 bin (0° to 90°). Note that the positive normal direction for both planes is the same, the +*y* axis.

6.   FQn   Print Hierarchy Card

Form: FQn a_1 a_2 ... a_8

n = tally number
a_i = F—cell, surface, or detector
        D—direct or flagged
        U—user
        S—segment
        M—multiplier
        C—cosine
        E—energy
        T—time

Default: Order as given above.

Use: Recommended where appropriate.

The a_i's are the letters representing all eight possible types of tally bins. This card can be used to change the order in which the output is printed for the tallies. For a given tally, the default order is changed by entering a different ordering of the letters, space delimited. An example of this card is in the DEMO example in Chapter 5.

A subset of the letters can be used, in which case MCNP places them at the end of the FQn card and precedes them with the unspecified letters in the default order. The first letter is for the outermost loop of the nest in the tally printout coding. The last two sets of bins make a table: the next to last set goes vertically; the last set of bins goes horizontally in the table.

Note that the default order is a table in E and T; any other bins in a tally will be listed vertically down the output page. Thus if you have a tally with only cell, user, and energy bins, the print for that tally will be a hard-to-read vertical list. Specifying U E as the only entries or last two entries on the FQn card will produce the same output, but in an easy-to-read table.

An FQ0 (zero) card can be used to change the default order for all tallies. A specific FQn card will then override that order for tally number n.

An example of this card is in the DEMO example in Chapter 5.

<u>7.</u>   <u>FMn   Tally Multiplier Card</u>

Form: FMn   (bin set 1)   (bin set 2)   ...   T

$$
\begin{aligned}
n &= \text{tally number} \\
\text{(bin set i)} &= \text{((multiplier set 1) (multiplier set 2) ... (attenuator set))} \\
T &= \text{absent for no total over bins} \\
&= \text{present for total over all bins} \\
C &= \text{cumulative tally bins}
\end{aligned}
$$

attenuator set   = C −1 m_1 px_1 m_2 px_2 ...
multiplier set i = C m (reaction list 1) (reaction list 2) ...
special multiplier set i = C −k

$$
\begin{aligned}
C &= \text{multiplicative constant} \\
-1 &= \text{flag indicating attenuator rather than multiplier set} \\
m &= \text{material number identified on an Mm card} \\
px &= \text{density times thickness of attenuating material;} \\
&\quad \text{atom density if positive, mass density if negative} \\
k &= \text{special multiplier option;} \\
\text{(reaction list } i) &= \text{sums and products of ENDF or} \\
&\quad \text{special reaction numbers, described below.}
\end{aligned}
$$

Parentheses:
  1.   If a given multiplier set contains only one reaction list, the parentheses surrounding the reaction list can be omitted. Parentheses within a reaction list are forbidden.

2. If a given bin set consists of more than a single multiplier or attenuator set, each multiplier or attenuator set must be surrounded by parentheses, and the combination must also be surrounded by parentheses.

3. If the FMn card consists only of a single bin set, and that bin set consists only of a single multiplier or attenuator bin, surrounding parentheses can be omitted.

Default: If the *C* entry is negative (for type 4 tally only), *C* is replaced by |*C*| times the atom density of the cell where the tally is made.

Use: Optional. Use the attenuators only when they are thin. Use only the multiplicative constant for tally types 6 and 7. Disallowed for tally type 8.

The FMn card is used to calculate any quantity of the form

$$C \int \varphi(E) R_m(E) dE \, ,$$

where $\varphi(E)$ is the energy-dependent fluence (particles/cm^2) and *R(E)* is an operator of additive and/or multiplicative response functions from the MCNP cross-section libraries or specially designated quantities. Note that some MCNP cross–section library reaction numbers are different from ENDF/B reaction numbers. See below. The constant *C* is any arbitrary scalar quantity that can be used for normalization. The material number *m* must appear on an Mm card, but need not be used in a geometrical cell of the problem.

A reaction list consists of one or more reaction numbers delimited by spaces and/or colons. A space between reaction numbers means multiply the reactions. A colon means add the reactions. The hierarchy of operation is multiply first and then add. One bin is created for each reaction list. Thus, if R_1, R_2, and R_3 are three reaction numbers, the form $R_1 R_2 : R_3$ represents one reaction list (one bin) calling for reaction R_3 to be added to the product of reactions R_1 and R_2. No parentheses are allowed within the reaction list. The product of R_1 with the sum of R_2 and R_3 would be represented by the form $R_1 R_2 : R_1 R_3$ rather than by the form $R_1 (R_2 : R_3)$. The latter form would produce two bins with quite a different meaning (see Examples 1 and 2 below).

The reaction cross sections are *microscopic* (with units of barns) and not macroscopic. Thus, if the constant *C* is the atomic density (in atoms per barn · cm), the results will include the normalization "per cm^3." The examples in Chapter 4 illustrate the normalization.

Any number of ENDF/B or special reactions can be used in a multiplier set as long as they are present in the MCNP cross-section libraries, or in special libraries of dosimetry data. If neither a material nor any reactions are given, the tally is multiplied by the constant *C*.

A multiplier set that has only two entries, *C* −k, has special meaning. If k= −1, the tally is multiplied by 1/weight and the tally is the number of tracks (or collisions for the F5 tally.) If k= −2, the tally is multiplied by 1/velocity and the tally is the neutron population integrated over

time, or the prompt removal lifetime. See Chapter 2 page 2–169, Chapter 4 example 3 page 4–39 and the KCODE problem in Chapter 5.

In addition to most of the approximately one hundred standard ENDF reaction numbers available (for example, $R = 1, 2, 16$, representing σ_{tot}, σ_{el}, $\sigma_{n,2n}$), the following nonstandard special R numbers may be used:

Neutrons: −1 total cross section without thermal
           −2 absorption cross section
           −3 elastic cross section without thermal
           −4 average heating number (MeV/collision)
           −5 gamma-ray production cross section, barns
           −6 total fission cross section
           −7 fission ν
           −8 fission Q (MeV/fission)

Photons: −1 incoherent scattering cross section
           −2 coherent scattering cross section
           −3 photoelectric cross section
           −4 pair production cross section
           −5 total cross section
           −6 photon heating number

Multigroup: −1 total cross section
            −2 fission cross section
            −3 nubar data
            −4 fission chi data
            −5 absorption cross section
            −6 stopping powers
            −7 momentum transfers

A list of many of the ENDF reaction numbers can be found in Appendix G. The total and elastic cross sections, $R = 1$ and $R = 2$, are adjusted for temperature dependence. All other reactions are interpolated directly from the library data tables. Note that for tritium production, the R number differs from one nuclide to another. Note also that tally types 6 and 7 already include reactions, so the FMn card makes little sense for $n = 6$ or 7. Only the constant-multiplier feature should be used for these tally types, generally. Photon production reactions can be specified according to the MTRP prescription in Table F.6 in Appendix F.

An attenuator set of the form C 1 m px, where m is the material number and px is the product of density and thickness, allows the tally to be modified by the factor $e^{-\sigma_{tot}px}$ representing an exponential line-of-sight attenuator. This capability makes it possible to have attenuators without

actually modeling them in the problem geometry. **Caution**: The assumption is made that the attenuator is thin, so that simple exponential attenuation without buildup from scattering is valid.

The attenuator set can include more than one layer:

$$C \ -1 \ m_1 \ px_1 \ m_2 \ px_2$$

in which case the factor is $e^{-\sigma_1 px_1 - \sigma_2 px_2}$. The attenuator set can also be part of a bin set, for example,

$$((C_1 \ m_1 \ R_1) \ (C_2 \ m_2 \ R_2) \ (C_3 \ -1 \ m_3 \ px_3))$$

in which case the attenuation factor is applied to every bin created by the multiplier sets. Note that both the inner and the outer parentheses are required for this application.

Tallies are posted in all multiplier bins for each score. MCNP does not automatically provide the total over all specified multiplier bins for a particular tally. The total is available for a tally, however, by putting the symbol T at the end of the FMn card for that tally.

In perturbed problems (see PERT card, page 3–142), the perturbation keyword RXN can affect the cross sections used with the FM card tally multipliers. If a tally in a cell is dependent on a cross section that is perturbed, then $R_{ij'} \neq 0$ and a correction is made to the $R_{1j'} = 0$ case (see page 2-XII.A.??.) For this required $R_{1j'}$ correction to be made, the user must ensure that the R reactions on the FM card are the same as the RXN reactions on the PERT card AND that the FM card multiplicative constant C is negative, indicating multiplication by the atom density to get macroscopic cross sections. For example, if R = –6 for fission on the FM card, you should not use RXN=18 for fission on the PERT card. If $C > 0$, the cross sections are not macroscopic, it is assumed that there is no tally dependence on a perturbed cross section, $R_{1j'} = 0$, and no correction is made. The same $R_{ij'} \neq 0$ correction is automatically made for the F6 tally and the KCODE k_{eff} calculation, and for an F7 tally if the perturbation reaction is fission because these three tallies all have implicit associated FM cards

Example 1:   FMn C m $R_1 R_2 : R_1 R_3$

Example 2:   FMn C m $R_1 (R_2 : R_3)$

These two examples reiterate that parentheses cannot be used for algebraic hierarchy within a reaction list. The first example produces a single bin with the product of reaction R_1 with the sum of reactions R_2 and R_3. The second case creates two bins, the first of which is reaction R_1 alone; the second is the sum of R_2 and R_3, without reference to R_1.

Example 3:   F2:N   1   2   3   4
                    FM2 (C_1) (C_2) (C_3) (C_4) T

Example 4:   F12:N   1   2   3   4
                    FM12 C_1

Example 5:   F22:N   (1 2 3)   4 T
                    FM22 (C_1) (C_2) (C_3) (C_4)

These three examples illustrate the syntax when only the constant-multiplier feature is used. All parentheses are required in these examples. Tally 2 creates 20 bins: the flux across each of surfaces 1, 2, 3, and 4 with each multiplied by each constant C_1, C_2, C_3, C_4, and the sum of the four constants. Tally 12 creates 4 bins: the flux across each of surfaces 1, 2, 3, and 4 with each multiplied by the constant C_1. Tally 22 creates 12 bins: the flux across surface 1 plus surface 2 plus surface 3, the flux across surface 4, and the flux across all four surfaces with each multiplied by each constant C_1, C_2, C_3, and C_4. An FQn card with an entry of F M or M F would print these bins of the tallies in an easy-to-read table rather than strung out vertically down the output page.

Several more examples of the FMn card are in Chapter 4. The DEMO example in Chapter 5 also illustrates the general form of the card.

Using MCNP tallies, there are two ways to obtain the energy deposited in a material in terms of rads (1 rad = 100 ergs/g). When the actual material of interest is present in the MCNP model, the simplest way is to use the heating tally with units MeV/g in conjunction with C=1.602E−08 on the companion FMn card, where C=(1.602E−06 ergs/MeV)/(100 ergs/g). When the material is not present in the model, rads can be obtained from type 1, 2, 4, and 5 tallies by using an FMn card where C is equal to the factor above times $N_o \eta \times 10^{-24} / A$, where N_o is Avogadro's number and η and A are the number of atoms/molecule and the atomic weight, respectively, of the material of interest. This value of C equals ρ_a / ρ_g as discussed on page 2–82. The implicit assumption when the material is not present is that it does not affect the radiation transport significantly. In the reaction list on the FM card, you must enter −4 1 for neutron heating and −5 −6 for photon heating. See page 2–82 and 4–38 for examples. For both F4 and F6, if a heating number from the data library is negative, it is set to zero by the code.

8.   <u>DEn   Dose Energy Card</u>
      <u>DFn   Dose Function Card</u>

Form: DEn   A E_1 ... E_k
               DFn   B F_1 ... F_k

         n = tally number.
         E_i = an energy (in MeV).
         F_i = the corresponding value of the dose function.

A	=	LOG or LIN interpolation method for energy table.
B	=	LOG or LIN interpolation method for dose function table.

Defaults: If A or B is missing, LOG is chosen for that table.

Use: Tally comment recommended.

This feature allows you to enter a pointwise response function (such as flux-to-dose conversion factors) as a function of energy to modify a regular tally. Both cards must have the same number of numerical entries and they must be monotonically increasing in energy. Particle energies outside the energy range defined on these cards use either the highest or lowest value.

By default MCNP uses log-log interpolation between the points rather than a histogram function as is done for the EMn card. The energy points specified on the DEn card do not have to equal the tally energy bins specified with the En card for the Fn tally. Unlike EMn card use, there can be many points on the DEn and DFn cards, but the response can be tallied in only a few energy bins such as one unbounded energy bin.

If n is zero on these two cards, the function will be applied to all tallies that do not have DEn and DFn cards specifically associated with them.

LIN or LOG can be chosen independently for either table. Thus any combination of interpolation (log-log, linear-linear, linear-log, or log-linear) is possible. The default log-log interpolation is appropriate for the ANSI/ANS flux-to-dose rate conversion factors (they are listed in Appendix H); kermas for air, water, and tissue; and energy absorption coefficients.

Example: DE5 E_1 E_2 E_3 E_4 ... E_k
          DF5   LIN F_1 F_2 F_3 F_4 ... F_k

This example will cause a point detector tally to be modified according to the dose function F(E) using logarithmic interpolation on the energy table and linear interpolation on the dose function table.

9. <u>EMn   Energy Multiplier Card</u>

Form: EMn M_1 ... M_k

n =   tally number.
M_i =   multiplier to be applied to the ith energy bin.

Default: None.

Use: Requires En card. Tally comment recommended.

This card can be used with any tally (specified by *n*) to scale the usual current, flux, etc. by a response function. There should be one entry for each energy entry on the corresponding En card. When a tally is being recorded within a certain energy bin, the regular contribution is multiplied by the entry on the EMn card corresponding to that bin. For example, a dose rate can be tallied with the appropriate response function entries. Tallies can also be changed to be per unit energy if the entries are $1/\Delta E$ for each bin. Note that this card modifies the tally by an energy-dependent function that has the form of a histogram and not a continuous function. It also requires the tally to have as many energy bins as there are histograms on the EMn card. If either of these two effects is not desired, see the DEn and DFn cards.

A set of energy multipliers can be specified on an EM0 (zero) card that will be used for all tallies for which there is not a specific EMn card.

10. **TMn   Time Multiplier Card**

> Form: TMn $M_1 ... M_k$
>
> > n =   tally number.
> > M_i =   multiplier to be applied to the ith time bin.
>
> Default: None.
>
> Use: Requires Tn card. Tally comment recommended.

This card is just like the EMn card except that the entries multiply time bins rather than energy bins. The Tn and TMn cards must have the same number of entries. Note that this card modifies the tally by a time-dependent function that has the form of a histogram and not a continuous function.

A set of time multipliers can be specified on a TM0 (zero) card that will be used for all tallies for which there is not a specific TMn card.

For example, if the entries are $1/\Delta T$, where ΔT is the width of the corresponding time bin, the tally will be changed to be per unit time with the units of $1/\Delta T$.

11. **CMn   Cosine Multiplier Card (tally type 1 only)**

> Form: CMn $M_1 ... M_k$
>
> > n =   tally number.
> > M_i =   multiplier to be applied to the ith cosine bin.
>
> Default: None.
>
> Use: Tally type 1. Requires Cn card. Tally comment recommended.

This card is just like the EMn and TMn cards except that the entries multiply cosine bins. The number of entries on the CMn card must be the same as on the Cn card. Note that this card modifies the tally by an angular-dependent function that has the form of a histogram and not a continuous function.

A set of cosine multipliers can be specified on a CM0 (zero) card that will be used for all type 1 tallies for which there is not a specific CMn card.

For example, if you want the directionally dependent F1 tally results to be per steradian, the ith entry on the CM1 card is

$$\frac{1}{2\pi(cos\,\theta_i - cos\,\theta_{i-1})}$$

where θ_o is 180°.

12. CFn   Cell-Flagging Card (tally types 1, 2, 4, 6, 7)

      Form: CFn $C_1 ... C_k$

              n = tally number.
              C_i = problem cell numbers whose tally contributions are to be flagged.

      Default: None.

      Use: Not with detectors or pulse height tallies. Consider FQn card.

Particle tracks can be "flagged" when they leave designated cells and the contributions of these flagged tracks to a tally are listed separately in addition to the normal total tally. This method can determine the tally contribution from tracks that have passed through an area of interest.

Cell flagging cannot be used for detector tallies. The same purpose can be accomplished with an FTn card with the ICD option.

The cell flag is turned on only upon leaving a cell. A source particle born in a flagged cell does not turn the flag on until it leaves the cell.

In MODE N P the flagged neutron tallies are those caused by neutrons leaving the flagged cell, but the flagged photon tallies can be caused by either a photon leaving a flagged cell or a neutron leaving a flagged cell and then leading to a photon which is tallied.

Example: F4:N   6   10   13
            CF4   3   4

In this example the flag is turned on when a neutron leaves cell 3 or 4. The print of Tally 4 is doubled. The first print is the total track length tally in cells 6, 10, and 13. The second print is the tally in these cells for only those neutrons that have left cell 3 or 4 at some time before making their contribution to the cell 6, 10, or 13 tally.

13. <u>SFn   Surface-Flagging Card (tally types 1, 2, 4, 6, 7)</u>

Form: SFn $S_1 ... S_k$

n =   tally number.
S_i =   problem surface numbers whose tally contributions are to be flagged.

Default: None.

Use: Not with detectors. Consider FQn card.

This feature is identical to cell flagging except that particles turn the flag on when they cross the specified surfaces. Thus a second tally print is given for only those particles that have crossed one or more of the surfaces specified on the SFn card.

Surface flagging cannot be used for detector tallies but an FTn card with the ICD option will do the same thing.

The situation for photon tallies in MODE N P is like that for the CFn card: a photon can be flagged either because it has crossed a flagged surface or because it was created by a neutron that crossed a flagged surface.

Both a CFn and an SFn card can be used for the same tally. The tally is flagged if the track leaves one or more of the specified cells or crosses one or more of the surfaces. Only one flagged output for a tally is produced from the combined CFn and SFn card use.

14. <u>FSn   Tally Segment Card (tally types 1, 2, 4, 6, 7)</u>

Form: FSn $S_1 ... S_k$

n =   tally number.
S_i =   signed problem number of a segmenting surface.

Default: No segmenting.

Use: Not with detectors. May require SDn card. Consider FQn card.

This card allows you to subdivide a cell or a surface into segments for tallying purposes, the advantage being that it is then not necessary to specify the problem geometry with extra cells just for tallying. The segmenting surfaces specified on the FSn card are listed with the regular problem surfaces, but they need not be part of the actual geometry and hence do not complicate the cell/ surface relationships.

If k surfaces are entered on the FSn card, $k + 1$ surface or volume segments are created. Tally n is subdivided into $k + 1$ segment bins according to the order and sense of the segmenting surfaces listed on the FSn card. If the symbol T is on the FSn card, there will be an additional total bin. The symbol C at the end of the FS card causes the bin values to be cumulative. Segmenting is done according to the following scheme:

$$Fn:N \quad S \quad (\text{or } C)$$
$$FSn \quad S_1 \dots S_k \ \text{T (optional)}$$

Tally n over surface S (or in cell C) will be subdivided into the following bins:

1. the portion with the same sense with respect to surface S_1 as the sign given to S_1,
2. the portion with the same sense with respect to surface S_2 as the sign given to S_2 but excluding that already scored in a previously listed segment,

   .

k   the portion with the same sense with respect to surface S_k as the sign given to S_k but excluding that already scored in a previously listed segment,
k+1   everything else,
k+2   entire surface or cell if T is present on FSn card.

If the symbol T is absent from the FSn card, the $(k+2)^{th}$ bin is missing and MCNP calculates the tally only for each segment (including the "everything else" segment). If multiple entries are on the Fn card, each cell or surface in the tally is segmented according to the above rules. For tally types 1 or 2, the segmenting surfaces divide a problem surface into segments for the current or flux tallies. The segmenting surfaces divide a problem cell into segments for tally types 4, 6, or 7. For normalized tallies, the segment areas (for type 2), volumes (for type 4), or masses (for types 6 and 7) may have to be provided. See the discussion under the SDn card.

Example 1:   F2:N   1
               FS2   −3 −4

This example subdivides surface 1 into three sections and calculates the neutron flux across each of them. There are three prints for the F2 tally: (1) the flux across that part of surface 1 that has negative sense with respect to surface 3, (2) the flux across that part of surface 1 that has negative sense with respect to surface 4 but that has not already been scored (and so must have positive sense with respect to surface 3), (3) everything else (that is, the flux across surface 1 with positive sense with respect to both surfaces 3 and 4).

CHAPTER 3
Tally Specification

It is possible to get a zero score in some tally segments if the segmenting surfaces and their senses are not properly specified. In Example 1 above, if all tallies that are positive with respect to surface 3 are also all positive with respect to surface 4, the second segment bin will have no scores.

Example 2: F2:N   1
            FS2   −3 4

The order and sense of the surfaces on the FS2 card are important. This example produces the same numbers as does Example 1 but changes the order of the printed flux. Bins two and three are interchanged.

Example 3: F1:N   1 2 *T*
            FS1   −3 *T*

This example produces three current tallies: (1) across surface 1, (2) across surface 2, and (3) the sum across surfaces 1 and 2. Each tally will be subdivided into three parts: (1) that with a negative sense with respect to surface 3, (2) that with a positive sense with respect to surface 3, and (3) a total independent of surface 3.

Several additional examples of the FSn card are in Chapter 4.

15. <u>SDn   Segment Divisor Card (tally types 1, 2, 4, 6, 7)</u>

Form: SDn $(D_{11}\ 8_2 ... D_{1m})\ (D_{21}\ D_{22} ... D_{2m})... (D_{k1}\ D_{k2} ... D_{km})$

n = tally number. n cannot be zero.
k = number of cells or surfaces on Fn card, including *T* if present.
m = number of segmenting bins on the FSn card, including the remainder segment, and the total segment if FSn has a *T*.
D_{ij} = area, volume, or mass of j_{th} segment of the i_{th} surface or cell bin for tally n.

The parentheses are optional.

Hierarchy for obtaining volume, area, or mass:
1. For cell or surface without segmenting (tally types 2, 4, 6, and 7):
   a. nonzero entry on SDn card,
   b. nonzero entry on VOL or AREA card,
   c. volume, area or mass calculated by MCNP,
   d. fatal error
2. For cell or surface with segmenting (tally types 2, 4, 6, and 7):
   a. nonzero entry on SDn card,
   b. volume, area or mass calculated by MCNP
   c. fatal error

3. For surface in a type 1 tally:
    a.   nonzero entry on SDn card,
    b.   no divisor.

Use: Not with detectors. May be required with FSn card. Can be used without FSn card.

For segmented cell volumes or surface areas defined by the FSn card that are not automatically calculated by MCNP, the user can provide volumes, areas, or masses on this segment divisor card to be used by tally *n*. This card is similar to the VOL and AREA cards but is used for specific tallies, whereas the other two are used for the entire problem geometry. For tally type 2 the entry is area, for tally type 4 the entry is volume, and for tally types 6 and 7 the entries are masses. Tally type 1 (the current tally) is not normally divided by anything, but with the SD1 card the user can introduce any desired divisor, for example, area to tally surface current density.

Example   F4:N   1 2 3 *T*
           SD4   1 1 1 1

Note that the SDn card can be used to define tally divisors even if the tally is not segmented. In this example the tally calculates the flux in the three cells plus the union of the three cells. The VOL card can be used to set the volume divisor of the three cells (to unity, for example), but it cannot do anything about the divisor for the union. Its divisor is the sum of the volumes (whether MCNP-calculated or user-entered) of the three cells. But the divisors for all four of the cell bins can be set to unity by means of the SDn card. These entries override entries on the VOL and AREA cards. See page 3–83 for use with repeated structure tallies.

16.   FUn   TALLYX Input Card

Form: FUn X_1 X_2 ... X_k
   or: FUn   *blank*

           n =   tally number.
           X_i =   input parameter establishing user bin *i*.

Default: If the FU card is absent, subroutine TALLYX is not called.

Use: Used with a user-supplied TALLYX subroutine or FTn card.

This card is used with a user-supplied tally modification subroutine TALLYX and some cases of the FTn card. If the FUn card has no input parameters, TALLYX will be called but no user bins will be created. The *k* entries on the FUn card serve three purposes: (1) each entry establishes a separate user tally bin for tally *n*, (2) each entry can be used as an input parameter for TALLYX to define the user bin it establishes, and (3) the entries appear in the output as labels for the user bins. IPTAL(LIPT+3,1,ITAL) is the pointer to the location in the TDS array of the word preceding the location of the data entries from the FUn card. Thus if the FUn card has the form shown above,

$$TDS(L+1) = X_1$$
$$TDS(L+2) = X_2$$
$$\vdots$$
$$TDS(L+k) = X_k$$

where   L   = IPTAL(LIPT+3,1,ITAL)

         k = IPTAL(LIPT+3,4,ITAL) − 1

                 = IPTAL(LIPT+3,3,ITAL) − 1

         n = JPTAL(LJPT+1,ITAL)

    ITAL   = program number of the tally

MCNP automatically provides the total over all specified user bins. The total can be inhibited for a tally by putting the symbol NT at the end of the FUn card for that tally as follows:

     FUn X_1 $X_2 \dots X_k$ NT

and there is one change in the preceding list of variables:

         k = IPTAL(LIPT+3,4,ITAL) − 1

                 = IPTAL(LIPT+3,3,ITAL)

The symbol C at the end of the FU card causes the bin values to be cumulative in which case

     IPTAL(LIPT+3,3,ITAL) = IPTAL(LIPT+3,4,ITAL)
     IPTAL(LIPT+3,6,ITAL) = 1.

The discussion of the IPTAL and JPTAL arrays in Appendix E and the following description of TALLYX may be useful.

SUBROUTINE TALLYX   User-supplied Subroutine

     Use: Called for tally n only if an FUn card is in the INP file.

TALLYX is called whenever a tally with an associated FUn card but no FTn card is scored. The locations of the calls to TALLYX are such that TALLYX is the very last thing to modify a score before it is posted in the tally. TALLYX calls can be initiated by more than one FUn card for different values of *n*; a branch must be constructed inside the subroutine based on which tally Fn is calling TALLYX, where *n* = JPTAL(LJPT+1,ITAL). TALLYX has the following form:

```
          SUBROUTINE TALLYX(T,IB)
  *CALL   CM
          User-supplied FORTRAN statements
```

```
            RETURN
            END
```

The quantity *T* (first argument of TALLYX) that is scored in a standard tally can be multiplied or
replaced by anything. The modified score *T* is then put into one of the *k* user bins established by
the FUn card. In TALLYX(T,IB) the second argument IB is defined to allow for more than one pass
through TALLYX per tally score. By default, IB=0, which means make one pass through the
MCNP coding where user bin tally scores are posted. If the user sets IB<0 in TALLYX, no score
will be made. If the user sets IB>0, passes through the user bin loop including TALLYX will be
made until IB is reset to zero. This scheme allows for tally modification and posting in more than
one user bin. The variable IBU is the variable designating the particular user bin established by the
FUn card. Its value is 1 before the first pass through the user bin loop. The indices of the current
user, segment, cosine, energy, and time bins (IBU, IBS, IBC, IBE, and IBT, respectively) and the
flag JBD that indicates flagged- or direct-versus-not are in Common for optional modification by
TALLYX. Note that the index of the multiplier bin is not available and cannot be modified. NTX
is a variable in blank Common. It is set equal to NX just before the CALL TALLYX in TALLYD
and TALLY. The variable NX is set to unity just before the start of the user bins loop and is
incremented after the CALL TALLYX, so NTX contains the number of the TALLYX call. An
example of using NTX to tally in every user bin before leaving the user bin loop follows:

```
            SUBROUTINE TALLYX(T,IB)
    *CALL  CM
            T = whatever
            IBU = NTX
            IB = 1
            IF(NTX.GE.IPTAL(LIPT+3,4,ITAL)-1) IB = 0
            RETURN
            END
```

If IBU is out of range, no score is made and a count of out-of-range scores is incremented. If
excessive loops through TALLYX are made, MCNP assumes IB has been incorrectly set and
terminates the job with a BAD TROUBLE error (excessive is greater than the product of the
numbers of bins of all kinds in the tally). Several examples of the FUn card and TALLYX are in
Chapter 4. The procedure for implementing a TALLYX subroutine is the same as for the user-
provided SOURCE subroutine.

17. TFn   Tally Fluctuation Card

     Form: TFn $I_1 ... I_8$

            n   = tally number. *n* cannot be zero.
            I_i = bin number for bin type *i*. $1 \leq I_i \leq last$
            last = IPTAL(LIPT+i,3,ITAL)

        = total number of bins in one of the eight bin types.

Default: 1 1 *last* *last* 1 *las* *last* *last*

1. first cell, surface, or detector on Fn card
2. total rather than flagged or uncollided flux
3. last user bin
4. last segment bin
5. first multiplier bin on FMn card
6. last cosine bin
7. last energy bin
8. last time bin.

Use: Whenever one or more tally bins are more important than the default bin. Particularly useful in conjunction with the weight window generator.

At the end of the output, one chart for each tally is printed to give an indication of tally fluctuations; that is, how well the tally has converged. The tally mean, relative error, variance of the variance, Pareto slope (see page 2–118), and figure of merit ($FOM = 1/(\sigma^2 t)$, where σ is the relative error printed with the tally and t is computer time in minutes) are printed as functions of the number of histories run. The *FOM* should be roughly constant. The TF card determines for which bin in tally n the fluctuations are printed. It also determines which tally bin is optimized by the weight window generator (WWE and WWG cards).

The TFn card allows you to change the default bin for a given tally and specify for which tally bin the chart and all the statistical analysis output will be printed. The eight entries on the card correspond (in order) to the list of bin indices for the eight dimensions of the tally bins array. The order is fixed and not affected by an FQn card.

The mean printed in a chart will correspond to some number in the regular tally print. If you have more than one surface listed on an F2 card, for example, the chart will be for the first surface only; charts can be obtained for all surfaces by having a separate tally for each surface.

You may find the *J* feature useful to jump over *last* entries. Remember that totals are calculated for energy, time, and user bins (unless inhibited by using NT), so that *last* for eight energy bins is 9. If one segmenting surface divides a cell or surface into two segments, *last* in that case is 2, unless *T* is used on the FS card, in which case *last* is 3. If there are no user bins or cosine bins, for example, *last* is 1 for each; *last* is never less than 1.

Example: Suppose an F2 tally has four surface entries, is segmented into two segments (the segment plus everything else) by one segmenting surface, and has eight energy bins. By default one chart will be produced for the first surface listed, for the part outside the segment, and totaled over energy. If we wish a chart for the fifth energy bin of the third surface in the first segment, we would use   TF2 3 2J 1 2J 5.

18. DDn   Detector Diagnostics Card

   Form: DDn k_1 m_1 k_2 m_2 ...

   n = 1 for neutron DXTRAN spheres
   = 2 for photon DXTRAN spheres
   = tally number for specific detector tally
   k_i = criterion for playing Russian roulette for detector i
   m_i = criterion for printing large contributions

   Defaults: If k_i is not specified on a DDn card, k_i on the DD card is used. If that is not
                specified, k_1 on the DD card is used. If that is not specified, $k_i = 0.1$ is used.
                A similar sequence of defaults defines m_i, with a final default of $m_i = 1000$.

   Use: Optional. Remember that Russian roulette will be played for detectors and
                DXTRAN unless specifically turned off by use of the DD card. Consider
                also using the PDn or DXC cards.

This card (1) using a Russian roulette game, can speed up calculations significantly by limiting
small contributions that are less than some fraction k of the average contribution per history to
detectors or DXTRAN spheres, and (2) can provide more information about the origin of large
contributions or the lack of a sufficient number of collisions close to the detector or DXTRAN
sphere. The information provided about large contributions can be useful for setting cell
importances or source-biasing parameters.

For a given detector or DXTRAN sphere, the Russian roulette criterion works as follows:

1. If k_i is positive, all contributions to the detector or sphere are made for the first 200
    histories. Then the average contribution per history is computed (and will be updated
    from time to time throughout the problem). Thereafter, any contribution to the detector
    or sphere larger than k_i times this average contribution will always be made, but any
    contribution smaller than k_i times the average will be subject to the Russian roulette
    game. (k_i is not allowed to be greater than 1.)

2. If k_i is negative, contributions larger than $|k_i|$ will always be made, and contributions
    smaller than $|k_i|$ will be subject to Russian roulette. This rule applies to all histories from
    the beginning of the problem, and the 200th history has no significance.

3. If k_i is zero, no Russian roulette game will be played for the detector or sphere.

Probably, $k = 0.5$ is suitable for most problems; the nonzero default value 0.1 means that the game
is *always played unless explicitly turned off by the user.*

The second entry, m_i, determines the condition for printing diagnostics for large contributions. If
the entry is zero, there is no diagnostic print. If the entry is positive, two possibilities exist.

1. If the corresponding k_i is positive or zero, no diagnostic prints will be made for the first 200 histories. Thereafter, the first 100 contributions larger than m_i times the average tally per history will be printed.

2. If the corresponding k_i is negative, the first 100 contributions larger than m_i times $|k_i|$ will be printed.

Remember that when k_i is positive the Russian roulette game is played on the basis of the estimated average contribution per history. Because the estimate improves from time to time, the game is based on different values for different histories. This can make debugging a problem more complicated, and the variance estimate does not quite obey the Central Limit Theorem. A procedure worth considering is to determine the average contribution per history in a preliminary run and then to use some fraction of the negative of this value in subsequent longer runs. The Russian roulette game is played without regard to particle time or energy; thus time and energy bins for which the ultimate tally is small may lose a disproportionate share of scores by the roulette game.

The DD card eliminates <u>tracks</u> with DXTRAN but only <u>contributions</u> with detectors.

Example: DXT:N

x_1	y_1	z_1	RI_1	RO_1
x_2	y_2	z_2	RI_2	RO_2
x_3	y_3	z_3	RI_3	RO_3

DXT:P x_4 y_4 z_4 RI_4 RO_4

F15X:P a_1 r_1 R_1
         a_2 r_2 R_2

DD   .2   100   .15   2000
DD1   −1.1E25   3000   J   J   J   3000
DD15   .4   10

Detector/sphere	k	m
sphere 1	−1.1E25	3000
sphere 2	.15	2000
sphere 3	.2	3000
sphere 4	.2	100
detector 1	.4	10
detector 2	.15	2000

Another example of the DD card and a description of its output is in Chapter 5. For a more detailed discussion of the Russian roulette game, see page 2–95 in Chapter 2.

19. <u>DXT   DXTRAN Card</u>

Form: DXT:n x_1 y_1 z_1 RI_1 RO_1 x_2 y_2 z_2 RI_2 RO_2 ... DWC_1 DWC_2 $DPWT$

n = N for neutrons, P for photons, not available for electrons.

$x_i\, y_i\, z_i$ =   coordinates of the point at the center of the ith pair of spheres

RI_i =   radius of the ith inner sphere in cm

RO_i =   radius of the ith outer sphere in cm

DWC_1 =   upper weight cutoff in the spheres

DWC_2 =   lower weight cutoff in the spheres

$DPWT$ =   minimum photon weight. Entered on DXT:N card only.

Defaults: Zero for DWC_1, DWC_2,   and $DPWT$.

Use: Optional. Consider using the DXC:N, DXC:P, or DD cards when using DXTRAN.

DXTRAN is used to improve the particle sample in the vicinity of a tally (see page 2–152). It should not be misconstrued as a tally itself, such as a detector; it is used in conjunction with tallies as a variance reduction technique. DXTRAN spheres must not overlap. The inner sphere should normally cover the tally region if possible. Specifying a tally cell or surface partly inside and partly outside a DXTRAN sphere usually will make the mean of the tally erratic and the variance huge.

The technique is most effective when the geometry inside the spheres is very simple and can be costly if the inside geometry is complicated, involving several surfaces. The inner sphere is intended to surround the region of interest. The outer sphere should surround neighboring regions that may scatter into the region of interest. In MCNP, the relative importance of the two regions is five. That is, the probability density for scattering toward the inner sphere region is five times as high as the probability density for scattering between the inner and outer spheres. The weight factor is 1/5 for particles scattered toward the inner sphere.

Rule of Thumb for RI and RO: The inner radius RI should be at least as large as the tally region, and RO–RI should be about one mean free path for particles of average energy at the spheres. DXTRAN can be used around detectors, but the combination may be very sensitive to reliable sampling.

There can be up to five sets of X Y Z RI RO on each DXT card. There is only one set of DWC_1 and DWC_2 entries for each particle type. This pair is entered after conclusion of the other data and (with DXT:N) before the one value of DPWT. The weight cutoffs apply to DXTRAN particle tracks inside the outer radii and have default values of zero. The DXTRAN photon weight cutoffs have no effect unless the simple physics is used, with one exception: upon leaving the sphere, track weights (regardless of what physics is used) are checked against the cutoffs of the CUT:P card. The DXTRAN weight cutoffs DWC and DWC_2 are ignored when mesh-based weight windows are used.

The minimum photon weight limit *DPWT* on the DXT:N card parallels almost exactly the minimum photon weight entries on the PWT card. One slight difference is that in Russian roulette

during photon production inside DXTRAN spheres, the factor for relating current cell importance to source cell importance is not applied. Thus, the user must have some knowledge of the weight distribution of the DXTRAN particles (from a short run with the DD card, for example) inside the DXTRAN sphere, so the lower weight limit for photon production may be intelligently specified. As in the case of the PWT entries, a negative entry will make the minimum photon weight relative to the source particle starting weight. The default value is zero, which means photon production will occur at each neutron DXTRAN particle collision in a material with nonzero photon production cross section inside the DXTRAN sphere.

DXTRAN can be used in a problem with the $S(\alpha,\beta)$ thermal treatment, but contributions to the DXTRAN spheres are approximate. DXTRAN should not be used with reflecting surfaces, white boundaries, or periodic boundaries (see page 2–92). DXTRAN is incompatible with a monodirectional source because direct contributions from the source are ignored.

If more than one set of DXTRAN spheres is used in the same problem, they can "talk" to each other in the sense that collisions of DXTRAN particles in one set of spheres cause contributions to another set of spheres. The contributions to the second set have, in general, extremely low weights but can be numerous with an associated large increase in computer time. In this case the DXTRAN weight cutoffs probably will be required to kill the very-low-weight particles. The DD card can give you an indication of the weight distribution of DXTRAN particles.

20. FTn   Special Treatments for Tallies

Form: FTn ID_1 $P_{1,1}$ $P_{1,2}$ $P_{1,3}$ ... ID_2 $P_{2,1}$ $P_{2,2}$ $P_{2,3}$ ...

n =   tally number.
ID_i =   the alphabetic keyword identifier for a special treatment.
FRV   fixed arbitrary reference direction for tally 1 cosine binning.
GEB   Gaussian energy broadening.
TMC   time convolution.
INC   identify the number of collisions.
ICD   identify the cell from which each detector score is made.
SCX   identify the sampled index of a specified source distribution.
SCD   identify which of the specified source distributions was used.
PTT   put different multigroup particle types in different user bins.
ELC   electron current tally.
$P_{i,j}$ =   parameters for that special treatment, either a number, a parenthesis or a colon.

Default: If the FT card is absent, there is no special treatment for tally n.

Use: Optional; as needed.

The syntax and meaning of the $P_{i,j}$ is different for each ID_i. A special treatment may cause a set of user bins or possibly a set of some other kind of bins to be created. The information in the $P_{i,j}$ allows the number and kind of those bins to be inferred easily. More than one special treatment can be specified by a given tally except for combinations of INC, ICD, SCX,SCD, PTT and ELC. Only one of these special treatments can be used by a tally at one time because all require user bins, making them mutually exclusive.

A description of the special treatments available follows with an explanation of the allowed parameters for each.

FRV V_1 V_2 V_3
The V_i are the xyz components of vector V, not necessarily normalized. If the FRV special treatment is in effect for a type 1 tally, the direction V is used in place of the vector normal to the surface as the reference direction for getting the cosine for binning.

GEB   a b c
The parameters specify the full width at half maximum of the observed energy broadening in a

physical radiation detector: fwhm $= a + b\sqrt{E + cE^2}$, where E is the energy of the particle. The

units of a, b, and c are MeV, $MeV^{1/2}$, and none, respectively. The energy actually scored is sampled from the Gaussian with that fwhm. See Chapter 2.

TMC   a b
All particles should be started at time zero. The tally scores are made as if the source was actually a square pulse starting at time a and ending at time b.

INC
No parameters follow the keyword but an FUn card is required. Its bin boundaries are the number of collisions that have occurred in the track since the creation of the current type of particle, whether at the source or at a collision where some other type of particle created it. If the INC special treatment is in effect, the call to TALLYX that the presence of the FUn card would normally trigger does not occur. Instead IBU is set by calling JBIN with the number of collisions as the argument.

ICD
No parameters follow the keyword but an FUn card is required. Its bins are the names of some or all of the cells in the problem. If the cell from which a detector score is about to be made is not in the list on the FUn card, the score is not made. TALLYX is not called. The selection of the user bin is done in TALLYD.

SCX   k

The parameter k is the name of one of the source distributions and is the k that appears on the SIk card. One user bin is created for each bin of source distribution k plus a total bin. The scores for tally n are then binned according to which bin of source distribution k the source particle came from. The score of the total bin is the score you would see for tally n without the special treatment, if source distribution k is not a dependent distribution.

CAUTION: For a dependent distribution, the score in the total bin is the subtotal portion of the score from dependent distribution k.

SCD

No parameters follow the keyword but an FUn card is required. Its bins are a list of source distribution numbers from SIk cards. The scores for tally n are then binned according to which distribution listed on the FUn card was sampled. This feature might be used to identify which of several source nuclides emitted the source particle. In this case, the source distributions listed on the FUn card would presumably be energy distributions. Each energy distribution is the correct energy distribution for some nuclide known to the user and the probability of that distribution being sampled from is proportional to the activity of that nuclide in the source. The user might want to include an FCn card that tells to what nuclide each energy distribution number corresponds.

CAUTION: If more than one of the source distributions listed on the FU card is used for a given history, only the *first* one used will score.

PTT

No parameters follow the keyword but an FUn card is required. Its bins are a list of atomic weights in units of MeV of particles masquerading as neutrons in a multigroup data library. The scores for tally n are then binned according to the particle type as differentiated from the masses in the multigroup data library. For example, .511 0 would be for electrons and photons masquerading as neutrons.

ELC   c

The single parameter c of ELC specifies how the charge on an electron is to affect the scoring of an F1 tally. Normally, an electron F1 tally gives particle current without regard for the charges of the particles. There are 3 possible values for c:

    c=1 to cause negative electrons to make negative scores
    c=2 to put positrons and negative electrons into separate user bins
    c=3 for the effect of both c=1 and c=2

If c=2 or 3, three user bins, positrons, electrons and total are created.

F.   *Material Specification*

The cards in this section specify the isotopic composition of the materials in the cells and which cross-section evaluations are to be used.

Mnemonic	Card Type	Page
Mm	Material	3–109
DRXS	Discrete reaction	3–110
TOTNU	Total fission $\bar{\upsilon}$	3–111
NONU	Fission turnoff	3–112
AWTAB	Atomic weight	3–113
XSn	Cross-section files	3–113
VOID	Negates materials	3–113
PIKMT	Photon–production bias	3–114
MGOPT	Multigroup card	3–115

1. <u>Mm   Material Card</u>

Form: Mm $ZAID_1$ $fraction_1$ $ZAID_2$ $fraction_2$ ... keyword=value ...

m corresponds to the material number on the cell cards
$ZAID_i$ = either a full ZZZAAA.nnX or partial ZZZAAA element or
               nuclide identifier for constituent $i,$ where ZZZ is the atomic
               number, AAA is the atomic mass, nn is the library identifier,
               and X is the class of data
$fraction_i$ = atomic fraction (or weight fraction if entered as a negative
               number) of constituent i in the material.
keyword = value, where = sign is optional. Keywords are:
    GAS = m flag for density–effect correction to electron stopping power.
               $m = 0$ calculation appropriate for material in the condensed
                  (solid or liquid) state used.
               $m = 1$ calculation appropriate for material in the gaseous state
                  used.
    ESTEP = n causes the number of electron substeps per energy step to
               be increased to n for the material. If n is smaller than the
               built–in default found for this material, the entry is ignored.
               Both the default value and the ESTEP value actually used
               are printed in Table 85.
      NLIB = id changes the default neutron table identifier to the string id.
               The neutron default is a blank string, which selects the
               first matching entry in XSDIR.
      PLIB = id changes the default photon table identifier to id.
      ELIB = id changes the default electron table identifier to id.
    COND = id sets conduction state of a material only for el03 evaluation.
               <0 nonconductor
               =0 (default) nonconductor if at least one nonconducting
                  component; otherwise a conductor
               >0 conductor if at least one conducting component.

Default: None for ZAID fraction; GAS=0; ESTEP internally set; NLIB, PLIB, and ELIB=first match in XSDIR; COND=0.

Use: Optional, but required if you want materials in cells.

Neutrons. For naturally occurring elements, AAA = 000. Thus, ZAID = 74182.55 represents the isotope ^{182}W and ZAID = 74000.55 represents the element tungsten. Natural elements not available from among those listed in Appendix G must be constructed on an Mm card by adding together the individual isotopes if they are available. If the density for cells with AAA = 000 is input in g/cm^3, MCNP will assume the atomic weight for the natural element. The ZZZ and AAA quantities are determined for neutrons by looking at the list of cross sections in Appendix G and finding the appropriate ZAID associated with an evaluation that you want.

Photons and electrons. If neutrons are not being run, the AAA can be set to 000. Cross sections are specified exactly like the neutron cross sections, but ZZZAAA.nnX equals ZZZ000. There is no distinction between isotope and element for photons and electrons. However, if the isotopic distribution for the element differs from the natural element, the atom density should be entered on the cell cards to ensure the correct atom density for these cells.

Nuclide Fraction. The nuclide fractions can be normalized to 1.0 or left unnormalized. For instance, if the material is H$_2$O the atom fractions for H and O can be entered as 0.667 and 0.333 or as 2 and 1, respectively. If the fractions are entered with negative signs they are assumed to be weight fractions. Weight fractions and atom fractions cannot be mixed on the same Mm card. There is no limit to the number of "nuclide fraction" entries or the total number of different cross-section tables allowed.

Default Library Hierarchy. When NLIB=id is included on an Mm card, the default neutron table identifier for that material is changed to id. Fully specifying a ZAID on that Mm card, ZZZAAA.nnX, overrides the NLIB=id default.

Example: M1   NLIB=50D   1001   2   8016.50C   1   6012   1

This material consists of three isotopes. Hydrogen (1001) and carbon (6012) are not fully specified and will use the default neutron table that has been defined by the NLIB entry to be 50D, the discrete reaction library. Oxygen (8016.50C) is fully specified and will use the continuous energy library. The same default override hierarchy applies to photon and electron specifications.

2.   DRXS   Discrete Reaction Cross-Section Card

Form: DRXS $ZAID_1$ $ZAID_2$ ... $ZAID_i$ ...
         or $blank$

$ZAID_i$ = Identifying number of the form ZZAAA.nn, where ZZ is the atomic number, AAA the mass number, and *nn* the neutron library identifier.

Default: Continuous-energy cross-section treatment if DRXS is absent.

Use: Optional. Applies only to neutron cross sections.

Nuclides listed on the optional DRXS card are given a discrete energy treatment instead of the regular fully continuous-energy cross-section treatment if the necessary discrete data are available. Check the list in Appendix G for availability. If the DRXS card is present but has no entries after the mnemonic, discrete cross sections will be used for every nuclide, if available.

All discrete reaction libraries are based on a 262 energy group structure. Groups below 1 eV make the discrete treatment appropriate for thermal neutron problems near room temperature. All discrete reaction libraries have photon production data given in expanded format.

It is not recommended that this card be used unless you are transporting neutrons in an energy region where resonances and hence self-shielding are of little importance. However, if the problem under consideration meets this criterion, using the DRXS card can reduce computer storage requirements and enhance timesharing.

Use of these discrete cross sections will not result in the calculation being what is commonly referred to as a multigroup Monte Carlo calculation because the only change is that the cross sections are represented in a histogram form rather than a continuous-energy form. The angular treatment used for scattering, energy sampling after scattering, etc., is performed using identical procedures and data as in the continuous-energy treatment. The user wanting to make a truly multigroup Monte Carlo calculation should use the MGOPT card multigroup capability.

3. TOTNU Total Fission Card

Form: TOTNU NO
              or *blank*

Default: If the TOTNU card is absent, prompt $\bar{\upsilon}$ is used for non-KCODE calculations and total $\bar{\upsilon}$ is used for KCODE calculations.

Use: All steady-state problems should use this card.

In a non-KCODE problem, prompt $\bar{\upsilon}$ is used for all fissionable nuclides for which prompt $\bar{\upsilon}$ values are available if the TOTNU card is absent. If a TOTNU card is present but has no entry after it, total $\bar{\upsilon}$, sampling both prompt and delayed $\bar{\upsilon}$, will be used for those fissionable nuclides for which

prompt and delayed values are available. A TOTNU card with NO as the entry is the same as if the card were absent, that is, prompt $\bar{\upsilon}$ is used.

In a KCODE calculation, total $\bar{\upsilon}$, including both prompt and delayed $\bar{\upsilon}$ as available, is used for all fissionable nuclides if the TOTNU card is absent. If a TOTNU card is present but has no entry after it, total $\bar{\upsilon}$, using both prompt and delayed $\bar{\upsilon}$, is again used. A TOTNU card with NO as the entry causes prompt $\bar{\upsilon}$ to be used for all fissionable nuclides for which prompt values are available.

The nuclide list of Appendix G indicates data available for each fissionable nuclide. The MCNP neutron cross-section summary print from XACT will show whether prompt or total was used.

4.   NONU   Fission Turnoff Card

Form: NONU a_1 a_2 ... a_i ... a_{mxa}
         or   *blank*

a_i = 0   fission in cell *i* treated as capture; gammas produced
       = 1   fission in cell *i* treated as real; gammas produced
       = 2   fission in cell *i* treated as capture; gammas not produced
mxa =   number of cells in the problem

Default: If the NONU card is absent, fission is treated as real fission.

Use: Optional, as needed.

This card turns off fission in a cell. The fission is then treated as simple capture and is accounted for on the loss side of the problem summary as the "Loss to fission" entry. If the NONU card is not used, all cells are given their regular treatment of real fission, that is, the same as if all entries were one. If the NONU card is present but blank, all a_i's are assumed to be zero and fission in all cells is treated like capture. The NONU card cannot be added to a continue-run.

A value of 2 treats fission as capture and, in addition, no fission gamma rays are produced. This option should be used with KCODE fission source problems written to surface source files. Suppressing the creation of new fission neutrons and photons is important because they are already accounted for in the source.

Sometimes it is desirable to run a problem with a fixed source in a multiplying medium. For example, an operating reactor power distribution could be specified as a function of position in the core either by an SDEF source description or by writing the fission source from a KCODE calculation to a WSSA file with a CEL option on an SSW card. The non-KCODE calculation would be impossible to run because of the criticality of the system and because fission neutrons have already been accounted for. Using the NONU card in the non-KCODE mode allows this problem to run correctly by treating fission as simple capture.

5. <u>AWTAB   Atomic Weight Card</u>

    Form: AWTAB $ZAID_1$ AW_1 $ZAID_2$ AW_2 ...

               $ZAID_i$ = $ZAID$ used on the Mm material card excluding the X for class of data specification.

               AW_i =   atomic weight ratios.

    Default: If the AWTAB card is absent, the atomic weight ratios from the cross–section directory file XSDIR and cross–section tables are used.

    Use: Optional, as needed.

Entries on this card override the existing atomic weight ratios as contained in both the cross–section directory file XSDIR and the cross–section tables. The AWTAB card is needed when atomic weights are not available in an XSDIR file. Also, for fission products, ZAID=50120.35, the atomic weight of tin ($^{120}_{50}Sn$) will be used, so the following AWTAB card is needed:

<p style="text-align:center">AWTAB   50120.35   116.490609</p>

<u>WARNING</u>: Using atomic weight ratios different from the ones in the cross–section tables in a neutron problem can lead to negative neutron energies that will cause the problem to terminate prematurely.

6. <u>XSn   Cross-Section File Card</u>

               n =   1 to 999

    Use: Optional, as an alternative to the directory part of the XSDIR file.

The XSn card can be used to load cross–section evaluations not listed in the XSDIR file directory. You can use XSn cards in addition to the XSDIR file. Each XSn card describes one cross section table. The entries for the XSn card are identical to those in XSDIR except that the + is not used for continuation. A detailed description of the required entries is provided in Appendix F.

7. <u>VOID   Material Void Card</u>

    Form: VOID   no entries

        or: VOID C_1 C_2 ... C_i

               C_i =   cell number

    Default: None.

    Use: Debugging geometry and calculating volumes.

The first form is used when calculating volumes stochastically (see page 2–183) and when checking for geometry errors (see page 3–8). When the VOID card is blank, the material number and density is set to zero for all cells, FM cards are turned off, heating tallies are turned into flux tallies, and, if there is no NPS card, the effect of an NPS 100000 card is created. If there is a TALLYX subroutine, it may need to be changed, too.

The second form is used to selectively void cells instead of setting the material number and density to zero by hand on cell cards. It is a convenience if you want to check whether the presence of some object in your geometry makes any significant difference in the answers.

8.   PIKMT   Photon–Production Bias Card

Form: PIKMT Z_1 $IPIK_1$ $MT_{1,1}$ $PMT_{1,1}$ ... $MT_{1, IPIK_1}$ $PMT_{1, IPIK_1}$

          Z_n $IPIK_n$ $MT_{n,1}$ $PMT_{n,1}$ ... $MT_{n, IPIK_n}$ $PMT_{n, IPIK_n}$

Z_i =   the ZAID of the ith entry. Full or partial ZAIDs can be specified; that is, 29000 is equivalent to 29000.50.

$IPIK_i$ = the parameter that controls the biasing for $ZAID_i$.
   0 = no biasing for $ZAID_i$; photons from $ZAID_i$ are produced with the normal sampling technique.
   -1 = no photons are produced from $ZAID_i$.
   > 0 = there is biasing for $ZAID_i$. The value of $IPIK_i$ is the number of partial photon–production reactions to be sampled.

$MT_{i,j}$ and $PMT_{i,j}$ are only required for ZAIDs with $IPIK_i > 0$, where $IPIK_i$ pairs of entries of MTs and PMTs are necessary. The MTs are the identifiers for the partial photon–production reactions to be sampled. The PMTs control, to a certain extent, the frequency with which the specified MTs are sampled. The entries need not be normalized. For a ZAID with a positive value of IPIK, any reaction that is not identified with its MT on the PIKMT card will not be sampled.

Default: If the PIKMT card is absent, there is no biasing of neutron–induced photons. If PIKMT is present, any ZAID not listed has a default value of $IPIK_i = -1$.

Use: Optional; see caveats below.

For several classes of coupled neutron–photon calculations, the desired result is the intensity of a small subset of the entire photon energy spectrum. Two examples are discrete–energy (line) photons and the high–energy tail of a continuum spectrum. In such cases, it may be profitable to bias the spectrum of neutron–induced photons to produce only those that are of interest.

1.   WARNING: Use of the PIKMT card can cause nonzero probability events to be completely excluded and the biasing game may be not necessarily a fair one. While

neutron tallies will be unaffected (within statistics), the only reliable photon tallies will be those with energy bins immediately around the energies of the discrete photons produced.

2. Users need information about the MT identifiers of the reactions that produce discrete--energy photons. This information is available on the web.

3. The feature is also useful for biasing the neutron–induced photon spectrum to produce very high energy photons (for example, $E_\gamma \geq 10\ MeV$). Without biasing, these high–energy photons are produced very infrequently; therefore, it is difficult to extract reliable statistical information about them. An energy cutoff can be used to terminate a track when it falls below the energy range of interest. Los Alamos users interested in using the PIKMT card for this application should see X–5 regarding an internal code (NIPE) that is useful for optimizing such problems.

Example: PIKMT   26000.55 1   102001 1   7014 0
                               29000 2   3001 2   3002 1
                               8016 −1

This example results in normal sampling of all photon–production reactions for ^{14}N. All photons from neutron collisions with Fe are from the reaction with MT identifier 102001. Two photon–production reactions with Cu are allowed. Because of the PMT parameters the reaction with MT identifier 3001 is sampled twice as frequently relative to the reaction with MT identifier 3002 than otherwise would be the case. No photons are produced from ^{16}O or from any other isotopes in the problem that are not listed on the PIKMT card.

9.   <u>MGOPT   Multigroup Adjoint Transport Option</u>

    Form: MGOPT   MCAL IGM IPLT ISB ICW FNW RIM

           MCAL = F for forward problem
                    A for adjoint problem

           IGM   = the total number of energy groups for all kinds of particles in the problem. A negative total indicates a special electron–photon problem.

           IPLT   = indicator of how weight windows are to be used.
                    = 0 means that IMP values set cell importances. Weight windows, if any, are ignored for cell importance splitting and Russian roulette.
                    = 1 means that weight windows must be provided and are transformed into energy–dependent cell importances. A zero weight–window lower bound produces an importance equal to the lowest nonzero importance for that energy group.
                    = 2 means that weight–windows do what they normally do.

           ISB   = Controls adjoint biasing for adjoint problems only (MCAL=A).

		= 0 means collisions are biased by infinite–medium fluxes.
		= 1 means collisions are biased by functions derived from weight–windows, which must be supplied.
		= 2 means collisions are not biased.
	ICW	= name of the reference cell for generated weight windows.
		= 0 means weight windows are not generated.
		≠ 0 requires volumes be supplied or calculated for all cells of nonzero importance.
	FNW	= normalization value for generated weight windows. The value of the weight–window lower bound in the most important energy group in cell ICW is set to FNW.
	RIM	= compression limit for generated weight windows. Before generated weight windows are printed out, the weight windows in each group separately are checked to see that the ratio of the highest to the lowest is less than RIM. If not, they are compressed.

Default: IPLT=0, ISB=0, ICW=0, FNW=1, RIM=1000. MCAL and IGM must be specified.

Use: Required for multigroup calculation.

MCAL and IGM are required parameters. The others are optional. "J" is not an acceptable value for any of the parameters.

At this time, the standard MCNP multigroup neutron cross sections are given in 30 groups and photons are given in 12 groups. Thus, an existing continuous–energy input file can be converted to a multigroup input file simply by adding one of the following cards:

```
MGOPT   F  30    $MODE N
MGOPT   F  42    $MODE N P
MGOPT   F  12    $MODE P
```

A negative IGM value allows a single cross–section table to include data for more than one sort of particle. This feature applies currently to electron/photon multigroup calculations only. A problem with 50 electron groups followed by 30 photon groups in one table would have IGM=−80. Also all tables must have the same group structure. A negative IGM value will use the energy variable on the source or tally card as a group index unless it is associated with a distribution. For an energy distribution on the source card, there should be IGM increasing integer entries for each group on the SI card. On a tally energy card, if there are less than IGM entries, they will be taken as energies in MeV; otherwise, the bins will be according to group index. The particles can be separated in tallies by using the PTT option on the FTn tally card.

An input file for an adjoint problem can have both an IMP card and weight window cards (IPLT=0   ISB=1). The entries on the weight window cards are not weight windows in the normal sense but biasing functions. If IPLT=1 the values on a weight window card become energy–dependent cell importances. Until now, importances have been energy independent.

See Appendix G for a more complete discussion of multigroup libraries.

G.   *Energy and Thermal Treatment Specification*

The following cards control energy and other physics aspects of MCNP. All energies are in units of MeV and all times are in shakes

.

Mnemonic	Card Type	Page
PHYS	Energy physics cutoff	3–103
TMP	Free-gas thermal temperature	3–108
THTME	Thermal times	3–108
MTm	$S(\alpha,\beta)$ material	3–109

1.   PHYS   Energy Physics Cutoff Card

    a)   Neutrons

    Form: PHYS:N   EMAX   EMCNF   IUNR   DNB

    EMAX   = upper limit for neutron energy, MeV.
    EMCNF = energy boundary above which neutrons are treated with implicit
                   capture and below which they are treated with analog capture.
    IUNR   = 0/1 = on/off unresolved resonance range probability tables.
    DNB   = number of delayed neutrons produced from fission
                 –1/0/>0 = natural sampling/no delayed neutrons produced/DNB
                   delayed neutrons per fission.
                 DNB > 0 not allowed in KCODE calculation.

    Default: EMAX = very large; EMCNF = 0.0 MeV; IUNR = 0; DNB = –1

    Use: Optional.

EMAX is the upper limit for neutron energy. All neutron cross-section data above EMAX are expunged. If EMAX is not specified, there is no upper energy expunging of cross-section data to save computer storage space. The physics of MCNP is such that if a neutron energy is greater than the maximum energy in a table (typically 20 MeV), the cross section for the maximum energy is

used with no extrapolation. If a particle is born above EMAX, either by source or collision, it is rejected and the particle energy is resampled.

EMCNF controls the type of capture. Any neutron with energy greater than EMCNF will receive the implicit capture treatment; below EMCNF, it will receive analog capture. This parameter is analogous to EMCPF on the PHYS:P card and is useful in eliminating low-energy histories when using a thermal treatment. Substantial computer time may be saved in a region of low absorption (especially if the region is heterogeneous and bounded by a reflecting surface) simply by reducing the number of tracks. EMCNF should be set to operate when a neutron enters a thermal regime, typically a few kT. However, analog capture may undesirably kill important particles before they are tallied or before they participate in physics important to the problem.

If EMCNF = EMAX, analog capture is used regardless of the value of WC1 on the CUT card. If WC1 = 0, analog capture is used regardless of the value of EMCNF.

IUNR controls the treatment of cross sections in the unresolved energy range. The probability table treatment (IUNR=0) should be left on for better physics but can be turned off (IUNR=1) to measure the effect of the probability table treatment or to speed calculations when unresolved resonances are unimportant.

DNB controls the number of delayed neutrons produced from fission and can be used only when TOTNU is specified for fissionable nuclides for which delayed and prompt ν values are available. If DNB is not specified, the number of delayed neutrons produced per fission is determined from the ratio of delayed ν to total ν. The nuclide list of Appendix G indicates data available for each fissionable nuclide.

    b)   <u>Photons</u>

    Form: PHYS:P   EMCPF IDES NOCOH

                EMCPF = upper energy limit for detailed photon physics treatment, MeV.
                IDES   = 0   photons will produce electrons in MODE E problems or bremsstrahlung photons with the thick target bremsstrahlung model.
                       = 1   photons will not produce electrons as above.
                NOCOH = 0   coherent scattering occurs.
                       = 1   coherent scattering will not occur.

    Default: EMCPF = 100 MeV;   IDES = 0;   NOCOH = 0.

    Use: Optional.

Photons with energy greater than EMCPF will be tracked using the simple physics treatment. If WC1 = 0 on the CUT:P card, analog capture is used in the energy region above EMCPF. Otherwise

capture is simulated by weight reduction with Russian roulette on weight cutoff. Photons with energy less than EMCPF will be treated with the more detailed physics that always includes analog capture. For a detailed discussion of the simple and detailed photon physics treatments, see Chapter 2.

The simple physics treatment, intended primarily for higher energy photons, considers the following physical processes: photoelectric effect without fluorescence, Compton scattering from free electrons without the use of form factors, and pair production. The highly forward peaked coherent Thomson scattering is ignored.

In the detailed physics treatment, photoelectric absorption can result in fluorescent emission, the Thomson and Klein-Nishina differential cross sections are modified by appropriate form factors taking electron binding effects into account, and coherent scattering is included.

To turn off the production of secondary electrons generated by photons, the switch IDES can be set, either on the PHYS:P or on the PHYS:E card. If either of these cards sets IDES = 1, photons will <u>NOT</u> produce electrons, even if IDES = 0 is set on the other. In a photon-only problem, turning off secondary electrons causes the thick-target bremsstrahlung model to be bypassed. This option should be exercised only with great care because it alters the physics of the electron-photon cascade and will give erroneously low photon results when bremsstrahlung and electron transport are significant.

NOCOH is a switch to allow coherent scattering to be turned off for photons with energies below EMCPF. Thus, coherent scattering can be suppressed within the detailed physics treatment without losing the other advantages of the detailed model. When NOCOH = 1, the cross section for coherent scattering will be set to zero. This approximation can be useful in problems with bad point detector variances.

    <u>c)</u>   <u>Electrons</u>

Form: PHYS:E EMAX IDES IPHOT IBAD ISTRG BNUM XNUM RNOK ENUM NUMB

EMAX	=	upper limit for electron energy in MeV.
IDES	=	0/1 = photons will/will not produce electrons.
IPHOT	=	0/1 = electrons will/will not produce photons.
IBAD	= 0	full bremsstrahlung tabular angular distribution.
	= 1	simple bremsstrahlung angular distribution approximation.
ISTRG	= 0	sampled straggling for electron energy loss.
	= 1	expected-value straggling for electron energy loss.
BNUM	< 0	only applicable for el03 evaluation. See below for details.
	= 0	bremsstrahlung photons will not be produced
	> 0	produce BNUM times the analog number of bremsstrahlung

			photons. Radiative energy loss uses the bremsstrahlung energy of the first sampled photon.
XNUM	>	0	produce XNUM times the analog number of electron-induced x–rays.
	=	0	x-ray photons will not be produced by electrons.
RNOK	>	0	produce RNOK times the analog number of knock-on electrons.
	=	0	knock-on electrons will not be produced.
ENUM	>	0	produce ENUM times the analog number of photon-induced secondary electrons.
	=	0	photon-induced secondary electrons will not be produced.
NUMB	>	0	produce bremsstrahlung on each substep
	=	0	nominal bremsstrahlung production

Defaults: EMAX = 100 MeV; IDES, IPHOT, IBAD, ISTRG = 0;
BNUM, XNUM, RNOK, ENUM = 1., NUMB = 0

Use: Optional.

EMAX is the upper electron energy limit in MeV. Electron cross sections and related data are generated on a logarithmic energy grid from EMAX down to an energy at least as low as the global energy cutoff for electrons. Setting the value of EMAX too high results in longer processing times and larger storage requirements for electron data. EMAX should be set to the highest electron energy encountered in your problem.

IDES is a switch to turn off electron production by photons. The default (IDES = 0) is for photons to create electrons in all photon-electron problems and for photons to produce bremsstrahlung photons using the thick-target bremsstrahlung approximation in photon problems run without electrons. In either case the electron default cross section library will be read, which requires considerable processing time. Electron transport is also very slow. However, the neglect of electron transport and bremsstrahlung production will cause erroneously low photon results when these effects are important. IDES = 1 turns off electron production, but it does not turn off the pair production--produced annihilation photons. See ENUM.

IPHOT is a switch to turn off photon production by electrons. Because photon transport is fast relative to electron transport and is usually required for an accurate physical model, the default (IPHOT = 0, which leaves photon production on) is recommended.

IBAD is a switch to turn on the simple approximate bremsstrahlung angular distribution treatment and turn off the full, more detailed model. The electron transport random walk can be done with either the simple or full treatment, but photon contributions to detectors and DXTRAN can use only the simple treatment. The full detailed physics model is more accurate and just as fast as the simple approximate treatment for the electron transport random walk, and is therefore the default

(IBAD = 0) even though it is inconsistent with the way bremsstrahlung photons contribute to detectors and DXTRAN spheres. Setting IBAD = 1 causes the simple treatment to be used for detectors and DXTRAN and the electron random walk, which is self-consistent.

ISTRG is a switch to control the electron continuous-energy slowing down treatment. If ISTRG = 1, the expected value for each collision is used; if ISTRG = 0 (default), the more realistic sampled value is used. The option of using the expected value is useful for some comparisons to deterministic electron transport calculations.

BNUM, XNUM, RNOK, and ENUM are biasing parameters for specific classes of electron or photon production processes. For each parameter the default is 1.0, which invokes an analog treatment for the associated process. Other values allow biasing of the sampling of the processes. The processes associated with the four parameters follow.

BNUM is used to control the sampling of bremsstrahlung photons produced along electron substeps. The default value (BNUM = 1) results in the analog number of bremsstrahlung tracks being sampled. If BNUM > 0, the number of bremsstrahlung photons produced is BNUM times the number that would be produced in the analog case. If the number of tracks is increased, an appropriate weight reduction is made; if the biasing reduces the number of tracks, the weight is increased. If BNUM = 0, the production of bremsstrahlung photons is turned off. In the el1 treatment, BNUM > 0 produces BNUM times the number of analog identical photons with appropriately modified weights. In the el03 treatment, BNUM > 0 produces BNUM times the number of analog photons, each sampled independently for energy and angle with appropriately modified weights. Such a scheme is similar to the one used in ITS3.0 and recommended by Bielajew, et. al. (A. F. Bielajew, R. Mohan, and C. S. Chui, "Improved Bremsstrahlung Photon Angular Sampling in the EGS4 Code System," Nov. 1989, PIR-0203.) In either case radiative energy loss uses the bremsstrahlung energy of the first sampled photon. BNUM < 0 (only for el03) produces |BNUM| times the number of analog photons, each sampled independently for energy and angle with appropriately modified weights. However, the radiative energy loss uses the average energy of all the bremsstrahlung photons sampled. Such a scheme conserves energy more closely but becomes more like a continuous slowing down approximation energy loss model.

XNUM is used to control the sampling of x-ray photons produced along electron substeps. The default value (XNUM = 1) results in the analog number of tracks being sampled. If XNUM > 0, the number of photons produced is XNUM times the number that would be produced in the analog case, and an appropriate weight adjustment is made. If XNUM = 0, the production of x-ray photons by electrons is turned off.

RNOK is used to control the number of knock-on electrons produced in electron interactions. The default value (RNOK = 1) results in the analog number of tracks being sampled. If RNOK > 0, the number of knock-on electrons produced is RNOK times the analog number, and an appropriate weight adjustment is made. If RNOK = 0, the production of knock-on electrons is turned off.

ENUM is used to control the generation of photon-induced secondary electrons. The default value (ENUM = 1) results in an analog treatment. If ENUM > 0, ENUM times the analog number of secondaries will be produced, and an appropriate weight adjustment is made. If ENUM = 0, the generation of secondary electrons by photons will be turned off. ENUM = 0 differs from IDES = 1. If ENUM = 0, pair production is totally turned off. If IDES = 1, the pair production–produced annihilation photons are still produced.

NUMB generates bremsstrahlung on each electron substep. Only a real event, one that has been sampled to have a bremsstrahlung interaction, causes energy loss. The weights of the bremsstrahlung photons are multiplied by the probability of interaction in a substep. If two or more photons are produced in a real event, the weight of the second or more photons is the unadjusted value because there is no Poisson sampling, except for real events.

In any of these biasing schemes, increasing the population of photons also increases the population of electrons because the additional photon tracks create photoelectrons, Compton recoil electrons, pair production electrons, etc. Similarly, increasing the number of electrons will propagate an increase in the population of subsequent generations of the cascade. Because electron transport is slow, a judicious use of ENUM < 1 may often be appropriate. When BNUM is set by the user, ENUM=1/BNUM in the el03 treatment unless the user sets ENUM. When NUMB>0, ENUM=1% by default.

The use of the switches, or of zero values for the biasing parameters, to turn off various processes goes beyond biasing, and actually changes the physics of the simulation. Therefore such actions should be taken with extreme care. These options are provided primarily for purposes of debugging, code development, and special-purpose studies of the cascade transport process.

2.   TMP   Free-Gas Thermal Temperature Card

      Form: TMPn T_{1n} T_{2n} ... T_{in} ... T_{In}

                n =   index of time on the THTME card.
                T_{in} =   temperature of ith cell at time n, in MeV.
                I =   number of cells in the problem.

      Default: 2.53×10^{-8} MeV, room temperature.

      Use: Optional. Required when THTME card is used. Needed for low-energy neutron transport at other than room temperature. A fatal error occurs if a zero temperature is specified for a nonvoid cell.

The TMP cards provide MCNP the time-dependent thermal cell temperatures that are necessary for the free-gas thermal treatment of low-energy neutron transport described on page 2–28. This treatment becomes important when the neutron energy is less than about 4 times the temperature

of heavy nuclei or less than about 400 times the temperature of light nuclei. Thus the TMP cards should be used when parts of the problem are not at room temperature and neutrons are transported with energies within a factor of 400 from the thermal temperature.

Thermal temperatures are entered as a function of time with a maximum of 99 time entries allowed. These times are entered on a thermal time (THTME) card. The thermal temperatures at time t_{1n} are listed, cell by cell, on the TMP1 card; the cell thermal temperatures at time t_{2n} are listed on the TMP2 card, etc. A linear interpolation is used to determine the cell thermal temperatures at times between two entries. Time values before t_{1n} or after t_{In} use the thermal temperatures at the nearest time entry.

We use kT to denote the thermal temperature of a cell and use units of MeV. The following formulas can be used to provide the values of kT for temperatures in degrees Kelvin, Celsius, Rankine, and Fahrenheit.

$$
\begin{aligned}
kT(\text{MeV}) \quad &= \quad 8.617 \times 10^{-11} T \text{ where } T \text{ is in degrees } K \\
&= \quad 8.617 \times 10^{-11} (T + 273.15) \text{ where } T \text{ is in degrees } C \\
&= \quad 4.787 \times 10^{-11} T \text{ where } T \text{ is in degrees } R \\
&= \quad 4.787 \times 10^{-11} (T + 459.67) \text{ where } T \text{ is in degrees } F
\end{aligned}
$$

3.   <u>THTME   Thermal Times Card</u>

   Form: THTME $t_1\ t_2 \dots t_n \dots t_N$

   $t_n\ =$ time in shakes at which thermal temperatures are specified on the TMP card.

   $N\ =$ total number of thermal times specified.

   Default: Zero; temperature is not time dependent.

   Use: Optional. Use with TMP card.

The THTME card specifies the times at which the thermal temperatures on the TMPn cards are provided. The temperatures on the TMP1 card are at time t_1 on the THTME card, the temperatures on the TMP2 card are at time t_2 on the THTME card, etc. The times must be monotonically increasing: $t_n < t_{n+1}$. For each entry on the THTME card there must be a TMPn card.

4.   <u>MTm $S(\alpha,\beta)$ Material Card</u>

   Form: MTm $X_1\ X_2 \dots$

   $X_i\ =$ $S(\alpha,\beta)$ identifier corresponding to a particular component on the Mm card.

   Default: None.

   Use: Optional, as needed.

For any material defined on an Mn card, a particular component of that material (represented by a ZAID number) can be associated through an MTm card with an $S(\alpha,\beta)$ data set if that data set exists. The $S(\alpha,\beta)$ data for that ZAID are used in every cell in which that material is specified. For a particular ZAID in a material, the free-gas treatment can be used down to the energy where $S(\alpha,\beta)$ data are available. At that point, the $S(\alpha,\beta)$ treatment automatically overrides the free-gas treatment (that is, there is no mixing of the two treatments for the same ZAID in the same material at a given energy). Typically the free-gas model is used for a particular ZAID of a material down to 4 eV and then the $S(\alpha,\beta)$ treatment will take over. In general, $S(\alpha,\beta)$ effects are most significant below 2 eV.

The $S(\alpha,\beta)$ treatment is invoked by identifiers on MTm cards. The *m* refers to the material *m* defined on a regular Mm card. The appearance of an MTm card will cause the loading of the corresponding $S(\alpha,\beta)$ data from the thermal data file. The currently available $S(\alpha,\beta)$ identifiers for the MTm card are listed in Table G.1 of Appendix G. $S(\alpha,\beta)$ contributions to detectors or DXTRAN spheres are approximate.

Examples:	M1	1001 2	8016 1	$ light water
	MT1	LWTR.07		
	M14	1001 2	6012 1	$ polyethylene
	MT14	POLY.03		
	M8	6012 1		$ graphite
	MT8	GRPH.01		

H.   *Problem Cutoff Cards*

The following cards can be used in an initiate-run or a continue-run input file to specify parameters for some of the ways to terminate tracks in MCNP.

Mnemonic	Card Type	Page
CUT	Cutoffs	3–124
ELPT	Cell–by–cell energy cutoff	3–126
NPS	History cutoff	3–126
CTME	Computer time cutoff	3–127

1.   CUT   Cutoffs Card

Form: CUT:n   *T   E   WC1   WC2   SWTM*

> *n*   = *N* for neutrons, *P* for photons, *E* for electrons.
> *T*   = time cutoff in shakes, 1 shake=10^{-8} sec.
> *E*   = lower energy cutoff in MeV.
> *WC1* and *WC2* = weight cutoffs.

SWTM = minimum source weight.

Use: Optional, as needed.

Neutron default: *T*=very large, *E*=0.0 MeV, *WC1* = −0.50, *WC2* = −0.25,
SWTM=minimum source weight if the general source is used.

If a neutron's time becomes greater than *T*, its transport is stopped and it is killed. Even though MCNP is time dependent, neutron decay is not considered. Any neutron with energy lower than *E* is killed.

If a neutron's weight *WGT* falls below *WC2* times the ratio *R* of the source cell importance to the current cell importance, then with probability *WGT/(WC1 * R)*, the neutron survives and is assigned *WGT* = *WC1* * *R*. If negative values are entered for the weight cutoffs, the values

$$|WC1| * W_s \text{ and } |WC2| * W_s$$

will be used for *WC1* and *WC2*, respectively, where W_s is the minimum weight assigned to a source neutron from an MCNP general source. These negative entries are recommended for most problems. If only *WC1* is specified, then *WC2* = 0.5 * *WC1*. See page 2–139 for a discussion of weight cutoffs.

In a coupled neutron/photon problem, photons are generated before the neutron weight cutoff game is played.

If *WC1* is set to zero, capture is treated explicitly by analog rather than implicitly by reducing the neutrons's weight according to the capture probability. If *EMCNF* = E_{max} on the PHYS card, analog capture is used regardless of the value of *WC1* except for neutrons leaving a DXTRAN sphere.

SWTM (source weight minimum) can be used to make the weight cutoffs relative to the minimum starting weight of a source particle for user source as is done automatically for the general source. The entry will in general be the minimum starting weight of all source particles, including the effects of energy and direction biasing. The entry is also effective for the general source as well. Then *SWTM* is multiplied by the W entry on the SDEF card but is unaffected by any directional or energy biasing. This entry is ignored for a KCODE calculation.

Photon default: *T*=neutron cutoff, *E*=0.001 MeV, *WC1* = −0.50, *WC2* = −0.25,
SWTM=minimum source weight if the general source is used.
If there are pulse height tallies, *WC1* = *WC2* = 0.

The CUT:P weight cutoffs are analogous to the CUT:N card except that they are used only for energies above the EMCPF entry on the PHYS:P card (see page 3–118). If WC1=0, analog capture is specified for photons of energy greater than EMCPF, just as it is for neutrons. For energies below EMCPF, analog capture is the only choice with one exception: photons leaving a DXTRAN

sphere. Their weight is always checked against the CUT:P weight cutoff upon exiting. If only *WC1* is specified, then *WC2* = 0.5 * *WC1*.

In a coupled neutron/photon problem, the photon weight cutoffs are the same as the neutron weight cutoffs unless overridden by a CUT:P card. Again, the photon weight cutoffs have no effect at energies below EMPCF (except with DXTRAN as noted above).

MCNP allows only analog capture below 0.001 MeV. Because the photoelectric cross section is virtually 100% of the total cross section below that energy for all isotopes, tracks will be quickly captured and terminated.

Electron default: *T*=neutron cutoff, *E*=0.001 MeV, *WC1* = 0, *WC2* = 0,
                    SWTM=minimum source weight if the general source is used.

The CUT:E weight cutoff entries have the same meaning as the neutron entries have.

2.   ELPT   Cell–by–cell Energy Cutoff

   Form: ELPT:n x_1 x_2 ... x_i ... x_I

            n =   N for neutrons, P for photons, E for electrons.
            x_i =   lower energy cutoff of cell i
            I =   number of cells in the problem.

A separate lower energy cutoff can be specified for each cell in the problem. The higher of either the value on the ELPT:n card or the global value E on the CUT:n card applies.

3.   NPS   History Cutoff Card

   Form: NPS   N

            N =   number of particle histories.

   Default: None.

   Use: As needed to terminate the calculation. In a criticality calculation, the NPS
              card has no meaning and a warning error message is issued if it is used.

The single entry *N* on this card is used to terminate the Monte Carlo calculation after *N* histories have been transported—unless the calculation is terminated earlier for some other reason such as computer time cutoff.

In a continue-run, NPS is the total number of particles including runs before the continue-run; it is cumulative. However, a negative NPS entry means to print an output file at the time of the last history run and then stop.

In a surface source problem, either more or less than all of the particle histories on the RSSA surface source file will be run, depending on the value N entered on the NPS card. If $N < NP1$, where $NP1$ is the number of original histories, Russian roulette with weight adjustment will be played with each history in the file, using a survival probability of $N/NP1$. If $N > NP1$, the histories will be split $N/NP1$ to 1, and the fractional part is taken care of by sampling. This can be done equally well for nonspherical sources by cell importance splitting. With a spherical source, each multiple occurrence of the history is sampled for a different starting location on the source sphere, possibly improving the spatial statistics of the results. In either case, the use of the NPS card will not provide additional information about the original source distributions or the transport to the recording surface crossing.

4.　CTME　　Computer Time Cutoff Card

Form:　　CTME　*x*

　　　　x　＝　maximum amount of computer time (in minutes) to be spent in the
　　　　　　　　Monte Carlo calculation.

Default:　None.

Use:　　As needed.

For a continue-run job the time on the CTME card is the time relative to the start of the continue-run; it is not cumulative.

Five normal ways to terminate an MCNP calculation are the NPS card, the CTME card, the job time limit, the end of a surface source file, and the number of cycles on a KCODE card. If more than one is in effect, the one encountered first will control. MCNP checks the computer time remaining in a running problem and will terminate the job itself, leaving enough time to wrap up and terminate gracefully.

I.　　User Data Arrays

Two arrays, IDUM and RDUM, are in MCNP variable COMMON and are available to the user. They are included in the dumps on the RUNTPE file and can therefore be used for any purpose, including accumulating information over the entire course of a problem through several continue-runs. Each array is dimensioned 50, and they can be filled by cards in the input file. IDUM is an integer array and RDUM is a floating point array.

CHAPTER 3
Peripheral Cards

1. IDUM   Integer Array Card

    Form: IDUM $\quad I_1 \dots I_n, \quad 1 \le n \le 50$

    Default: All array values zero.

    Use: Useful only in user-modified versions of MCNP.

Entries (up to 50) fill the IDUM array with integer numbers. If floating point numbers are entered, they will be truncated and converted to integers.

2. RDUM   Floating Point Array Card

    Form: RDUM $\quad R_1 \dots R_n, \quad 1 \le n \le 50$

    Default: All array values zero.

    Use: Useful only in user-modified versions of MCNP.

Entries (up to 50) fill the RDUM array with floating point numbers.

J.   Peripheral Cards

The following cards offer a variety of conveniences:

Mnemonic	Card Type	Page
PRDMP	Print and dump cycle	3–128
LOST	Lost particle	3–130
DBCN	Debug information	3–130
FILES	Create user files	3–134
PRINT	Printing control	3–135
MPLOT	Plot tally while problem is running	3–137
PTRAC	Particle track output card	3–138
PERT	Perturbation Card	3–142

1. PRDMP   Print and Dump Cycle Card

    Form: PRDMP   NDP NDM MCT NDMP DMMP

                NDP = increment for printing tallies
                NDM = increment for dumping to RUNTPE file
                MCT = flag to write MCTAL file and for OUTP comparisons

NDMP = maximum number of dumps on RUNTPE file

DMMP		Sequential MCNP TFC entries every	Multiprocessing MCNP TFC entries and rendezvous every
	< 0	1000 particles	1000 particles
=	0	1000 particles	10 during the run (see discussion below)
	> 0	DMMP particles	DMMP particles

Default: Print only after the calculation has successfully ended. Dump every 15 minutes and at the end of the problem. Do not write a MCTAL file. Write all dumps to the RUNTPE file. DMMP=0 (see table above).

Use: Recommended, especially for complex problems.

The PRDMP card allows the user to control the interval at which tallies are printed to the OUTP file and information is dumped to the RUNTPE file. Positive entries mean that after every NDP histories the summary and tallies are printed to the output file, and after every NDM histories a dump is written to the run file. A negative entry changes the unit from histories to minutes of computer time.In a criticality calculation, positive entries for NDP and NDM on the PRDMP card are interpreted as the number of cycles rather than the number of particles started. Printing and dumping are done only at the ends of cycles.

If the third entry MCT on the PRDMP card is nonzero, a MCTAL file is written at the problem end. The MCTAL file is an ASCII file of tallies that can be subsequently plotted with the MCNP MCPLOT option (see description elsewhere). The MCTAL file is also a convenient way to store tally information in a format that is stable for use in the user's own auxiliary programs. For example, if the user is on a system that cannot use the MCNP MCPLOT option, the MCTAL file can be manipulated into whatever format is required by the user's own local plotting algorithms. If MCT=−1, references to code name, version number, problem ID, figure of merit, and anything else having to do with running time are omitted from MCTAL and OUTP so that tracking runs (identical random walks) yield identical MCTAL and OUTP files. MCT=−2 turns off additional prints in OUTP to assist in comparing multitasking output.

The PRDMP card also allows the user to control the size of the RUNTPE file by specifying the maximum number of dumps, NDMP, to be written. The RUNTPE file will contain the last NDMPs that were written. For example, if NDMP = 4, after dump 20 is written only dumps 17, 18, 19, and 20 will be on the RUNTPE file. In all cases, the fixed data and cross section data at the front of the RUNTPE file are preserved.

The fifth entry DMMP has several possible meanings. For sequential MCNP, a value of DMMP~≤ 0 results in TFC entries every 1000 particles initially. This value doubles to 2000 after

20 TFC entries. A positive value of DMMP produces TFC entries every DMMP particles initially. For distributed memory multiprocessing, DMMP < 0 produces TFC entries and task rendezvous every 1000 particles initially, the same as does the sequential version. DMMP=0, the default value, produces ten TFC entries and task rendezvous, rounded to the nearest 1000 particles, based on other cutoffs such as NPS, CTME, etc. This selection optimizes speedup in conjunction with TFC entries. If detectors/DXTRAN are used with default Russian roulette criteria (DD card default), the DMMP=0 entry is changed by MCNP to < 0, ensuring tracking with the sequential version (i.e., TFC entries and rendezvous every 1000 particles). As with the sequential version, DMMP > 0 produces TFC entries and task rendezvous every DMMP particles, even with detectors/DXTRAN with default Russian roulette criteria. Setting DMMP to a large positive number minimizes communication time and maximizes speedup. However, the TFC may not have many entries, possibly only one, if DMMP=NPS.

2.   LOST   Lost Particle Card

> Form: LOST   LOST(1)   LOST(2)
>
> LOST(1) =   number of particles which can be lost before the job
>   terminates with BAD TROUBLE
> LOST(2) =   maximum number of debug prints that will be made for
>   lost particles

> Defaults: 10 lost particles and 10 debug prints.

> Use: Discouraged. Losing more than 10 particles is rarely justifiable.

The word "lost" means that a particle gets to an ill-defined section of the geometry and does not know where to go next. This card should be used cautiously: you should know why the particles are being lost, and the number lost should be statistically insignificant out of the total sample. Even if only one of many particles gets lost, there could be something seriously wrong with the geometry specification. Geometry plots in the area where the particles are being lost can be extremely useful in isolating the reason that particles are being lost. See page 3–8.

3.   DBCN   Debug Information Card

> Form: DBCN X_1 X_2 X_3 ... X_{20}
>
> X_1 =   the starting pseudorandom number. Default $=(5^{19})^{152917}$;
> X_2 =   debug print interval;
> X_3 and X_4 = history number limits for event log printing;
> X_5 =   maximum number of events in the event log to print per history.
>   Default $= 600$;
> X_6 =   unused.
> X_7 =   1 produces a detailed print from the volume and surface area

		calculations;
X_8	=	number of the history whose starting pseudorandom number is to be used to start the first history of this problem;
X_9	=	closeness of coincident repeated structures surfaces. Default = 1.E-4;
X_{10}	=	seconds between time interrupts. Default = 100 seconds;
X_{11}	=	1 causes collision lines to print in lost particle event log;
X_{12}	=	expected number of random numbers;
X_{13}	=	random number stride. Default = 152917;
X_{14}	=	random number multiplier. Default = 5^{19};
X_{15}	=	1 prints the shifted confidence interval and the variance of the variance for all tally bins;
X_{16}	=	scale the score grid for the accumulation of the empirical $f(x)$ in print tables 161 and 162;
X_{17}	=	0 default angular treatment for partial substeps to generation sites of secondary particles;
		> 0 alternate angular treatment for secondary generation;
		< 0 MCNP4A treatment of electron angles at secondary generation sites;
X_{18}	=	0 default "MCNP–style" energy indexing algorithm;
		1 "ITS–style" energy indexing algorithm;
X_{20}	=	track previous version.

Use: Optional.

The entries on this card are used primarily for debugging problems and the code itself. The first 12 can be changed in a continue run which is useful for diagnosing troubles that occur late in a long-running problem.

1. X_1 is the random number used for starting the transport of the first particle history in a run. See also entry X_8, which for repeating particle histories, is the preferred method of changing the pseudorandom number sequence. See the **caution** after the last DBCN item listed below.

2. X_2 is used to print out information about every X_2th particle. The information consists of: (a) the particle history number, (b) the total number of neutron, photon, and electron collisions, (c) the total number of random numbers generated, and (d) the random number at the beginning of the history. This information is printed at the beginning of the history and is preceded by the letters DBCN in the output to aid in a pattern search.

3. and 4. Event log printing is done for histories X_3 through X_4, inclusively. The information includes a step-by-step account of each history, such as where and how a particle is born, which surface it crosses and which cell it enters, what happens to it in a cell, etc. See X_{11}.

5. X_5 is the maximum number of events the event log will print per history. The default is 600.

6. Unused.

7. $X_7 = 1$ will cause a detailed print from the volume and surface area calculations and is useful only to MCNP code developers.

8. The X_8th entry causes the starting random number of the problem to be the random number that would normally be picked for the X_8th history. If a surface source is used, the X_8th surface source history will be taken from the RSSA file at the problem start. The purpose of this entry is to let the X_8th history be the first history of a problem for debugging purposes or to select a random number sequence different from that in an identical problem to compare statistical convergence. See the **caution** after the last DBCN item listed below.

9. X_9 defines the distance allowed between coincident repeated structures surfaces for them still to be considered coincident. The default is 1.E–4. A value of 1.E–30 reproduces the earlier treatment where coincident repeated structures surfaces was not allowed. X_9 should not have to be changed unless geometries have dimensions greater than 1.E5 or unless surfaces at different levels are intended to be closer than 2.E–4.

10. X_{10} is the seconds between time interrupts for checking if a history has run too long or is in an infinite loop. The default is 100 seconds. If in two consecutive time interrupts the random walk is in the same history, MCNP assumes that something is wrong and stops the job. If histories should legitimately take longer than X_{10} seconds the job can be continued with a larger value for X_{10} specified on the DBCN card in the continue-run INP file. This entry also affects the time increment MCNP reserves for itself to terminate a job before the job time limit is reached. The increment for interactive jobs is $2X_{10}$ or 1% of the time limit, whichever is greater.

11. $X_{11} = 1$ causes collision lines to print in the lost particle event log.

12. X_{12} is the expected number of random numbers for this calculation. Entering X_{12} will cause the last line of the output file to print X_{12} and the actual number of random numbers used so that a quick comparison can be made to see if two problems tracked each other.

13. X_{13} is the random number stride, S. The default is $S = X_{13} = 152917$. Each source history starts with a random number S numbers up the pseudorandom number sequence from the random number of the previous history. If any history requires more than S random numbers, the number of times S was exceeded is printed in the problem summary of the OUTP file. The maximum number of random numbers required for a history is always printed in the problem summary. Exceeding the random number stride will cause a correlation between histories and should be avoided because variances may be underestimated. However, if the stride is too large, the period of the random number sequence, $2^{46} \approx 7.04\,E13$, will be exceeded.

S should be chosen so that $NPS*S < 2^{46}$. Exceeding the period will underestimate variances, particularly if S is a power of 2.

14. X_{14} is the random number multiplier. The default is $X_{14} = 5^{19} = 19073486328125$, which is adequate for all known problems. If a new entry is such that the sum of its left and right 24-bit halves is not less than 2^{24} then the input value is rejected. If X_{14} is even it is rejected because the random number sequence rapidly converges to zero.

15. A nonzero X_{15}th entry causes the shifted confidence interval and the variance of the variance (VOV) to be calculated and printed for all tally bins. An extra line of tally output is created for each tally that contains nonzero information. The shifted confidence interval center is followed by the estimated VOV. If the tally mean and relative error (RE) are all zeros, the VOV line is not printed because it is all zero also. Changing X_{15} from nonzero to zero in a CONTINUE run will cause the VOV information not to be printed. X_{15} cannot be changed from zero to nonzero in a CONTINUE run.

16. MCNP uses a logarithmically spaced history score grid in print table 161 for $f(x)$, producing a straight line for $f(x)$ on a log–log plot for $1/x^n$ behavior, covering 60 decades of unnormalized tally magnitudes from $1E-30$ to $1E30$. This range can be multiplied by the X_{16}th entry when the range is not sufficient. A negative entry means that negative history scores will be accrued in the score grid $f(-x)$ and the absolute value of X_{16} will be used as the score grid multiplier. Positive history scores will then be lumped into the lowest bin with this option. This scaling can be done only in the original problem, not in a CONTINUE run.

17. If 0, the default angular treatment for partial substeps to generation sites of secondary particles is invoked. This treatment accounts for the probability of the delta function first, then interpolates in the cosine of the deflection angle. It does not preserve the plane in which the deflection angle will lie at the end of the full substep.
If > 0, an angular treatment for secondary generation is invoked as follows. The cosine of the electron angle is interpolated and the end–of–substep plane is preserved, but the changing probability of the delta function along the substep is ignored. This option is preserved for further testing of angular algorithms because results have been known to be sensitive to these details.
If < 0, the MCNP4A treatment of electron angles at secondary generation sites is invoked. Used with dbcn(20)=0, comparisons to the earlier treatment can be made.

18. If 0, the default "MCNP–style" energy indexing algorithm is used, also called the "bin–centered" treatment.
If $\neq 0$, the "ITS–style" energy algorithm is used, also called the "nearest group boundary" treatment. Allows us to match ITS results as closely as possible.

19. A nonzero X_{20}th entry causes MCNP to track the previous version of the code, except in the few cases when bugs are too hard to duplicate with this option. Because bug corrections, new features, and enhanced physics must be undone, $X_{20} \neq 0$ should be used only for debugging purposes.

CAUTION: When trying to duplicate a particle history by setting the starting random number with either X_1 or X_8, the random number sequence may be altered by a default Russian Roulette game on contributions to detectors or DXTRAN spheres. If a problem has detectors or DXTRAN, the only ways to reproduce histories with X_1 or X_8 are: (a) turn off the Russian Roulette game on the DD card by setting k = 0; (b) play the roulette game with a fixed criterion by setting k < 0 on the DD card; or (c) reproduce a history with NPS < 200.

4.   FILES   File Creation Card

      Form: FILES   unit no.   filename   access   form   record length

unit no.	=	1 to 99
filename	=	name of the file
access	=	sequential or direct
form	=	formatted or unformatted
record length	=	record length in a direct access file

      Default: None; none; sequential; formatted if sequential, unformatted if direct; not required if sequential, no default if direct.

      Use: When a user-modified version of MCNP needs files whose characteristics may vary from run to run. Not legal in a continue-run.

If this card is present, the first two entries are required and must not conflict with existing MCNP units and files. The words "sequential," "direct," "formatted," and "unformatted" can be abbreviated. If more than one file is on the FILES card, the defaults are not much help but the abbreviations will keep it brief. The maximum number of files allowed is six, unless the dimension of the KUFIL array in Fixed Common is increased.

     Example: FILES 21   ANDY S F 0 22   MIKE D U 512

If the filename is DUMN1 or DUMN2, the user can optionally use the execution line message to designate a file whose name might be different from run to run, for instance in a continue-run.

     Example: FILES   17 DUMN1
                   MCNP INP=TEST3   DUMN1=POST3

Caution: The names of any user files in a continue-run will be the same as in the initial run. The names are not automatically sequenced if a file of the same name already exists; therefore, a second output file from a continue-run will clobber an existing file of the same name. If you are using the FILES card for an input file and do a continue-run, you will have to provide the coding for keeping track of the record number and then positioning the correct starting location on the file when you continue or MCNP will start reading the file at the beginning.

5.   PRINT   Output Print Tables

    Form: PRINT x

                x =   no entry gives the full output print
                x = x_1 x_2 ... prints basic output plus the tables specified by the table numbers x_1, x_2, ...
                x = $-x_1$ $-x_2$ ... prints full output except the tables specified by x_1, x_2, ...

    Default: No PRINT card in the INP file or no PRINT option on the execution line will result in a reduced output print.

    Use: Optional.

The following output will be printed automatically, as applicable:
- a listing of the input file,
- the problem summary of particle creation and loss,
- KCODE cycle summaries,
- tallies,
- tally fluctuation charts, and
- the tables listed below marked basic and default.

You will always get the information indicated by the first five bullets above and the tables labelled "basic" below. They *cannot* be turned off. Tables marked "default" will be printed automatically but they *can* be turned off with the PRINT card.

To get all optional print tables applicable to your problem, indicated in the table below as blank type, use the PRINT card in the INP file or the PRINT option on the execute line. The execute line takes precedence over the input card. Absence of a PRINT card or a PRINT option produces only the tables marked "basic," "default," and "shorten." Entries are allowed only on the PRINT card, not following the PRINT option. Entries on the PRINT card can be in any order.

The PRINT card entries are table numbers of optional and default tables, and control turning the table off or on. If all the entries are positive, you will get the "basic" tables plus the tables requested on the PRINT card. If any entry is negative, you will get all tables applicable to your problem *except* those turned off by the negative entries.

The table number appears in the upper righthand corner of the table, providing a convenient pattern when scanning the output file with an editor. The pattern is PRINT TABLE n, where n is preceded always by one space and is a two- or three-digit number. The table numbers and titles and type are summarized in the table below. Tables that can not be controlled by the PRINT card are marked as type "basic." Tables that are automatically printed but *can* be turned off are marked as type "default." Tables with no type (blank) can be turned off and on with the PRINT card or option.

Tables 160, 161, and 162 are different from the other tables. If you turn off table 160, tables 161 and 162 will not appear either. If table 160 is printed, they will all be printed. They are all automatically printed if there is no PRINT card or if there is a blank PRINT card. If a PRINT card has a positive entry, tables 160, 161, and 162 will not appear, unless table 160 is explicitly requested. If the entry is negative, they will appear, unless table 160 is explicitly turned off.

Table 175 can not be turned off completely, but the output can be greatly shortened to every 100 cycles plus the last five cycles. PRINT −175 and PRINT 110 both will produce the short version of Table 175.

Table 128, the repeated structure universe map, is special. If table 128 is not turned on in an initial run, it CANNOT be turned on in a subsequent continue–run because the (often large) storage arrays have not been set up. Table 128 is the only print table that affects storage. The information in the other tables is always stored, whether or not it is printed. A warning will be printed in a repeated structures problem if you do not request the universe map/lattice activity table in the original run.

The PRINT control can be used in a continue–run to recover all or any applicable print tables, even if they were not requested in the original run. A continue file with NPS −1 and PRINT will create the output file for the initial run starting with the Problem Summary (located after table 110). Table 128 can never be printed if it was not requested in the original run.

Table Number	Type	Table Description
10		Source coefficients and distribution
20		Weight window information
30		Tally description
35		Coincident detectors
40		Material composition
50		Cell volumes and masses, surface areas
60	basic	Cell importances
62	basic	Forced collision and exponential transform
70		Surface coefficients
72	basic	Cell temperatures
85		Electron range and straggling tables multigroup: flux values for biasing adjoint calcs
86		Electron bremsstrahlung and secondary production
90		KCODE source data
98		Physical constants and compile options
100	basic	Cross section tables
102		Assignment of $S(\alpha,\beta)$ data to nuclides
110		First 50 starting histories

120		Analysis of the quality of your importance function
126	basic	Particle activity in each cell
128		Universe map
130		Neutron/photon/electron weight balance
140		Neutron/photon nuclide activity
150		DXTRAN diagnostics
160	default	TFC bin tally analysis
161	default	$f(x)$ tally density plot
162	default	Cumulative $f(x)$ and tally density plot
170		Source distribution frequency tables, surface source
175	shorten	Estimated k_{eff} results by cycle
178		Estimated k_{eff} results by batch size
180		Weight window generator bookkeeping summary controlled by WWG(7), not print card
190	basic	Weight window generator summary
198		Weight windows from multigroup fluxes
200	basic	Weight window generated windows

Example: PRINT   110   40   150

The output file will contain the "basic" tables plus tables 40, 110, and 150, not 160, 161, 162 (the "default" tables), and the shortened version of 175.

Example: PRINT   170   −70   −110

The output file will contain all the "basic" tables, all the "default" tables, the long version of table 175, and all the optional tables except tables 70, 110, and 170 applicable to your problem.

6.   <u>MPLOT   Plot tally while problem is running</u>

Form: MPLOT   MCPLOT keyword=parameter

Default: None.

Use: Optional.

This card specifies a plot of intermediate tally results that is to be produced periodically during the run. The entries are MCPLOT commands for one picture. The = sign is optional. During the run, as determined by the FREQ n entry, MCRUN will call MCPLOT to display the current status of one or more of the tallies in the problem. If a FREQ n command is not included on the MPLOT card, n will be set to 5000. The following commands can not appear on the MPLOT card: RMCTAL, RUNTPE, DUMP, and END. All of the commands on the MPLOT card are executed for each displayed picture, so coplots of more than one bin or tally are possible. No output is sent

to COMOUT. MCPLOT will not take plot requests from the terminal and returns to MCRUN after each plot is displayed. See Appendix B for a complete list of MCPLOT commands available.

Another way to plot intermediate tally results is to use the TTY interrupt <ctrl–c>IMCPLOT or <ctrl–c>IM that allows interactive plotting during the run. At the end of the history that is running when the interrupt occurs, MCRUN will call MCPLOT, which will take plot requests from the terminal. No output is sent to the COMOUT file. The following commands can not be used: RMCTAL, RUNTPE, DUMP and END.

7.   PTRAC   Particle Track Output Card

    Form: PTRAC   keyword=parameter(s)   keyword=parameter(s)

    Default: See Table 3.5.

    Use: Optional.

This card generates an output file, default name PTRAC, of user–filtered particle events. The name PTRAC can be changed on the execution line or within the message block. Using this card without any keywords causes all particle events to be written to the PTRAC file. CAUTION: an extremely large file likely will be created unless NPS is small. Use of one or more keywords listed in Table 3.5 will reduce significantly the PTRAC file size. In Table 3.5 the keywords are arranged into three categories: output control keywords, event filter keywords, and history filter keywords. The output control keywords provide user control of the PTRAC file and I/O. The event filter keywords filter particle events on an event–by–event basis. That is, if the history meets the filter criteria, all filtered events for that history are written to file PTRAC. The PTRAC card keywords can be entered in any order and, in most cases, the corresponding parameter values can appear in any order (exceptions noted below.) The PTRAC card is not legal in a continue–run input file because a change in the PTRAC input would require a readjustment in dynamically allocated storage.

When multiple keywords are entered on the PTRAC card, the filter criteria for each keyword must be satisfied to obtain an output event. For example:

        PTRAC   FILTER=8,9,erg   EVENT=sur   NPS=1,50   TYPE=e   CELL=3,4

will write only surface crossing events for 8–9 MeV electrons generated by histories 1–50 that have entered cells 3 or 4.

TABLE 3.5: PTRAC Keywords, Parameter Values, and Defaults

Keyword	Parameter Values	Default	Entries
OUTPUT CONTROL KEYWORDS			
BUFFER	Integer > 0	100	1
FILE	asc, bin	bin	1
MAX	Integer ≠ 0	10000	1
MEPH	Integer > 0	*	1
WRITE	pos, all	pos	1
EVENT FILTER KEYWORDS			
EVENT	src, bnk, sur, col, ter	*	1–5
FILTER	Real, Integer, Mnemonic	*	2–72
TYPE	n, p, e	*	1–3
HISTORY FILTER KEYWORDS			
NPS	Integer > 0	*	1–2
CELL	Integer > 0	*	Unlimited
SURFACE	Integer > 0	*	Unlimited
TALLY	Integer ≠ 0	*	Unlimited
VALUE	Real, Integer	*	Unlimited

BUFFER   Determines the amount of storage available for filtered events. A small value
         results in increased I/O and a decrease in required memory, whereas a large
         value minimizes I/O and increases memory requirements.

FILE   Controls file type. One of the following values can be entered:
         asc—generates an ASCII output file.
         bin—generates a binary output file. This is the default.

MAX   Sets the maximum number of events to write to the PTRAC file. A negative
         value terminates MCNP when this value is reached.

MEPH   Determines the maximum number of events per history to write to the PTRAC
         Default: write all events.

WRITE   Controls what particle parameters are written to the PTRAC file.
         pos—only x, y, z location with related cell and material numbers.
         all—additionally, u, v, w direction cosines, energy, weight, and time.
         If the size of the PTRAC file is a concern and the additional parameters are not
         needed, the default value of "pos" is recommended.

EVENT   Specifies the type of events written to PTRAC. One or more of the
         following parameter values can be entered:
         src—initial source events
         bnk—bank events

    sur—surface events
    col—collision events
    ter—termination events
    The bank events include secondary sources, e.g., photons produced by
    neutrons, as well as particles created by variance reduction techniques, e.g.,
    DXTRAN and energy splitting. See page I-5 for a complete list.

FILTER Specifies additional MCNP variables for filtering. The parameter
values consist of one or two numerical entries and a variable mnemonic that
corresponds to a variable in the PBLCOM common block. See Table 3.6 for
available mnemonics. A single numerical entry requires an exact value.
EXAMPLE: FILTER=2,icl   writes only those events that occur in cell 2.
Two numerical entries represent a range.
EXAMPLE: FILTER=0,10,x   writes only those events in which
the particle's x–coordinate is between 0 and 10 cm. When a range is
specified, the first entry must be less than or equal to the second. Multiple
sets of numerical entries and mnemonics are also allowed.
EXAMPLE: FILTER=0.0,10.0,x 0,1,u 1.0,2,erg   writes only those
events in which the particle's x–coordinate is between 0 and 10 cm
<u>and</u> the particle's x–axis cosine in between 0 and 1 <u>and</u> the particle's
energy is between 1 and 2 MeV.
Default: No additional filtering.

TYPE   Filters events based on particle type. One or more of the following parameter
values can be entered:
n—neutron events;   p—photon events;   e—electron events
EXAMPLE: TYPE=p,e   writes only photon and electron events.
Default: Events for all particle types are written.

NPS   Sets the range of particle histories for which events will be output. A single
value produces filtered events only for the specified history.
EXAMPLE: NPS=10   writes events only for particle number 10.
Two entries indicate a range and will produce filtered events for all histories
within that range. The first entry must be less than or equal to the second.
EXAMPLE: NPS=10,20 writes events for particles 10 through 20.
Default: Events for all histories.

CELL, SURFACE, TALLY   The cell, surface, or tally numbers entered after these
keywords are used for history filtering. If any track of the history enters listed
cells or crosses listed surfaces or contributes to the TFC bin of listed tallies,
all filtered events for the history are written to the PTRAC file. See page 3–101
for specification of the TFC bin.
EXAMPLE: CELL=1,2   writes all filtered events for those histories that
enter cell 1 or 2.

EXAMPLE: TALLY=4   writes all filtered events for those histories that contribute to tally 4 (see VALUE keyword for filter criteria.) The number of entries following CELL, SURFACE, and TALLY is unlimited. A negative TALLY entry indicates that the corresponding VALUE entry is a multiplier rather than an absolute value.
Default: No history filtering.

VALUE   Specifies the tally cutoff above which history events will be written. The number of entries must match those of the TALLY keyword.
EXAMPLE: Tally=4 VALUE=2.0   writes all filtered events of any history that contributes 2.0 or more to the TFC bin of tally 4. A negative TALLY value indicates that the corresponding VALUE entry is a multiplier.
EXAMPLE: TALLY=–4   VALUE=2.0   writes all filtered events of any history that contributes more than $2.0*T_a$ to tally 4, where T_a is the average tally of the TFC bin. The values for T_a are updated every DMMP histories. Typically, DMMP=1000. See the PRDMP card, page 3–128. Filtering based on the T_a values will occur only when they become nonzero. Thus, when using a multiplier, PTRAC events may not be written for several thousand particles, or at all, if scores are seldom or never made to the TFC bin of the specified tally. In such cases, it is best to enter an absolute value.
EXAMPLE: TALLY=4 VALUE=0.0   writes all filtered events of every history that scores to tally 4.
Default: A multiplier of 10.0 for each tally associated with the TALLY keyword .

TABLE 3.6: Mnemonic Values for the FILTER Keyword

Mnemonic	MCNP Variable	Description
X	XXX	X–coordinate of particle position (cm)
Y	YYY	Y–coordinate of particle position (cm)
Z	ZZZ	Z–coordinate of particle position (cm)
U	UUU	Particle X–axis direction cosine
V	VVV	Particle Y–axis direction cosine
W	WWW	Particle Z–axis direction cosine
ERG	ERG	Particle energy (MeV)
WGT	WGT	Particle weight
TME	TME	Time at the particle position (shakes)
VEL	VEL	Speed of the particle (cm/shake)
IMP1	FIML(1)	Neutron cell importance
IMP2	FIML(2)	Photon cell importance
IMP3	FIML(3)	Electron cell importance
SPARE1	SPARE(1)	Spare banked variable
SPARE2	SPARE(2)	Spare banked variable
SPARE3	SPARE(3)	Spare banked variable
ICL	ICL	Problem number of current cell
JSU	JSU	Problem number of current surface
IDX	IDX	Number of current DXTRAN sphere
NCP	NCP	Count of collisions for current branch
LEV	LEV	Geometry level of particle location
III	III	1st lattice index of particle location
JJJ	JJJ	2nd lattice index of particle location
KKK	KKK	3rd lattice index of particle location

8. <u>PERTn</u>   <u>Perturbation Card</u>

    Form: PERTn:pl   keyword=parameter(s)   keyword=parameter(s)

    n   = unique, arbitrary perturbation number.
    pl   = N, P, or N,P. Not available for electrons.
    keyword = See Table 3.7.

    Default: Some keywords are required. See Table 3.7.

    Use: Optional.

This card allows perturbations in cell material density, composition, or reaction cross-section data. The perturbation analysis uses the first and second order differential operator technique described

in Chapter 2, page 2–191. Using this technique, the perturbation estimates are made without actually changing the input material specifications. Multiple perturbations can be applied in the same run, each specified by a separate PERT card. There is no limit to the number of perturbations because dynamic memory is used for perturbation storage. The entire tally output is repeated for each perturbation, giving the estimated differential change in the tally or this change can be added to the unperturbed tally (see the METHOD keyword). For this reason, the number of tallies and perturbations should be kept to a minimum. A track length estimate of perturbations to k_{eff} is automatically estimated and printed for KCODE problems. The CELL keyword that identifies one or more perturbed problem cells is required. Also, either the MAT or RHO keyword must be specified.

TABLE 3.7: PERT Keywords, Parameter Values, and Defaults

Keyword	Parameter Values	Default	Entries
BASIC KEYWORDS			
CELL	Integer > 0	Required	Unlimited
MAT	Integer > 0	*	1
RHO	Real, integer	*	1
ADVANCED KEYWORDS			
METHOD	±1, 2, 3	1	1
ERG	Real, Integer > 0	All Energies	2
RXN	Integer	1	Unlimited

CELL   Indicates which cells are perturbed. At least one entry is required, and there is no limit to the number of entries. A comma or space delimiter is required between entries:
          CELL=1,2,3,4
          CELL=1 10i 12

MAT   Specifies the perturbation material number, which must have a corresponding M card. Composition changes can only be made through the use of this keyword. If the RHO keyword is omitted, the MAT keyword is required. Note in the CAUTIONS below that certain composition changes are prohibited.

RHO   Specifies the perturbed density of the cells listed after the CELL keyword. A positive entry indicates units of atoms/cm^3 and a negative entry indicates units of g/cm^3. If the MAT keyword is omitted, the RHO keyword is required.

METHOD   Specifies the number of terms to include in the perturbation estimate.
          1 — include first and second order (default)
          2 — include only first order

3 — include only second order

A positive entry produces perturbation tallies that give the estimated **<u>differential</u>** change in the unperturbed tally (default). A negative entry **<u>adds</u>** this change to the unperturbed tally. The ability to produce first and second order terms separately enables the user to determine the significance of including the second-order estimator for subsequent runs. If the second-order results are a significant fraction (20-30%) of the total, then higher order terms are necessary to accurately predict the change in the unperturbed tally. In such cases, the magnitude of the perturbation should be reduced to satisfy this condition. Typically, this technique is accurate to within a few percent for up to 30% changes in the unperturbed tally. It is *strongly* recommended that the magnitude of the second order term be determined before the user continues with this capability.

ERG The two entries specify an energy range in which the perturbation is applied. The default range includes all energies. This keyword is usually used with the RXN keyword to perturb a specific cross-section over a particular energy range.

RXN Entries must be ENDF/B reaction types that identify one or more specific reaction cross-sections to perturb. A list of available ENDF/B reaction types is given in Table I, Appendix G. This keyword allows the user to perturb a single reaction cross-section of a single nuclide in a material, all reaction types of a single nuclide, a single reaction for all nuclides in a material, and a set of cross-sections for all nuclides in a material. The default reaction is the total cross section (RXN=1 for neutrons and multigroup, RXN=-5 for photons.) Relevant nonstandard special R numbers on page 3–89 can be used. Those that cannot be used are −4, −5, −7, and −8 for neutrons; −6 for photons; and −3, −4, −6, and −7 for multigroup problems. If these irrelevant R numbers are used, the following fatal error will be printed: "fatal error. reaction # illegal in perturbation #."

RXN=2 elastic cross section
RXN=−2 absorption cross section

RXN reaction numbers must be consistent with FM card reaction numbers (see page 3–89) if the perturbation affects the tally cross section. RXN=−6 is most efficient for fission, although MT=18, MT=19, or MT=−2 (multigroup) also work for k_{eff} and F7 tallies.

CAUTIONS

1. There is no limit to the number of perturbations, but they should be kept to a minimum as each perturbation can degrade performance by 10-20%.

2. It is not possible to take a region originally specified as void and put in a material with the perturbation technique. However, you can specify a region as containing a material and use the PERT card to make it void by setting RHO=0.

3. It is not possible to introduce a new nuclide into a material composition. However, you can set up the problem with a mixture of all nuclides of interest and use PERT cards to remove one or more (see the examples below).

4. The track length estimate of k_{eff} in KCODE criticality calculations assumes the fundamental eigenvector (fission distribution) is unchanged in the perturbed configuration.

5. Use caution in selecting the multiplicative constant and reaction number on FM cards used with F4 tallies in perturbation problems. The track length correction term $R_{1j'}$ is made only if the multiplicative constant on the FM card is negative (indicating macroscopic cross sections with multiplication by the atom density of the cell). If the multiplicative constant on the FM card is positive, it is assumed that any FM card cross sections are independent of the perturbed cross sections. If there is a reaction (RXN) specified on the PERT card, the track length correction term R_{1j} is set only if the exact same reaction is specified on the FM card. For example, an entry of RXN=2 on the PERT card is not equivalent to the special elastic reaction −3 on the FM card (should either enter 2 and 2 or −3 and −3).

6. DXTRAN, F5 point detector tallies, and F8 pulse height tallies are not compatible with the PERT card. DXTRAN will give a fatal error; F5 and F8 will give zero perturbations.

7. Large perturbations require higher than second order terms to avoid inaccurate tallies. Refer to the METHOD keyword for a more complete discussion.

Examples of the PERT Card

Example 1:   PERT1:n,p   CELL=1   RHO=0.03

This perturbation specifies a density change to 0.03 atoms/cm^3 in cell 1. This change is applied to both neutron and photon interactions.

Example 2:   3   1   −1   −1 2 −3 4 −5 6 $ mat 1 at 1 g/cm^3
                12   1   −1   −7 8 −9 10 −11 12 $ mat 1 at 1 g/cm^3
                    …

C M1 material is semiheavy water
M1 1001 .334 1002 .333 8016 .333
C M8 material is heavy water
M8   1002 .667 8016 .333
PERT2:n   CELL=3,12 MAT=8 RHO=−1.2

This perturbation changes the material composition of cells 3 and 12 from material 1 to material 8. The MAT keyword on the PERT card specifies the perturbation material. The material density was also changed from 1.0 to 1.2 g/cm^3 to change from water to heavy water.

Example 3:   PERT3:n,p CELL=1 10i 12 RHO=0 METHOD=−1

This perturbation makes cells 1 through 12 void for both neutrons and photons. The estimated changes will be added to the unperturbed tallies.

Example 4:   60 13 −2.34 105 −106 −74 73 $ mat 13 at 2.34 g/cm^3
              …
              M13 1001 −.2 8016 −.2 13027 −.2 26000 −.2 29000 −.2
              M15 1001 −.2 8016 −.2 13027 −.2 26000 −.2 29000 −.4
              PERT1:p CELL=60 MAT=15 RHO=−2.808 RXN=51 9i 61,91
                   ERG=1,20
              PERT2:p CELL=60 RHO=−4.68 RXN=2

This example illustrates sensitivity analysis. The first PERT card generates estimated changes in tallies caused by a 100% increase in the Cu (n,n') cross section (ENDF/B reaction types 51–61 and 91) above 1 MeV. To effect a 100% increase, double the composition fraction (−.2 to −.4) and multiply the ratio of this increase by the original cell density
(RHO=[1.2/1.0] ∗ −2.34 = −2.808 g/cm^3, where the composition fraction for material 13 is 1.0 and that for material 15 is 1.2.) A change must be made to RHO to maintain the other nuclides in their original amounts. Otherwise, after MCNP normalizes the M15 card, it would be as follows, which is different from the composition of the original material M13:

M15 1001 −.167 8016 −.167 13027 −.167 26000 −.167 29000 −.333

The second PERT card (PERT2:p) gives the estimated tally change for a 100% increase in the elastic (RXN=2) cross section of material 13. RHO=−2.34 ∗ 2 = −4.68 g/cm^3

Example 5:   M4 6000.60C .5 6000.50C .5
              M6 6000.60C 1
              M8   6000.50C 1
              PERT1:n CELL=3 MAT=6 METHOD=−1
              PERT2:n CELL=3 MAT=8 METHOD=−1

The perturbation capability can be used to determine the difference between one cross–section evaluation and another. The difference between these perturbation tallies will give an estimate of the effect of using different cross section evaluations.

> Example 6:   1 1 0.05 −1 2 −3 \$ mat 1 at 0.05 x 10^{24} atoms/cm^3
> …
> M1   1001 .1 8016 .2   92235 .7
> M9   1001 .1 8016 .22 92235 .7
> F14:n   1
> FM14   (−1 1 −6 −7 \$ keff estimator for cell 1
> PERT1:n   CELL=1 MAT=9 RHO=0.051 METHOD=1
> PERT2:n   CELL=1 MAT=9 RHO=0.051 METHOD=−1

These perturbations involve a 10% increase in the oxygen atom fraction of material 1 (RHO=0.05 x [1.02/1.0] = 0.051). The effect of this perturbation on tally 14, which is a track length estimate of k_{eff}, will be provided as a differential change (PERT1) as well as with this change added to the unperturbed estimate of k_{eff} (PERT2). Note: if the RHO keyword is omitted from the PERT cards, the ^{235}U composition will be perturbed, which can produce invalid results (see Caution #4.)

> Example 7:   1 1 −1.5 −1 2 −3 4 −5 6 \$ mat 1 at 1.5 g/cm^3
> …
> M1   1001 −.4333 6000 −.2000 8016 −.3667 \$ half water
>   \$ half plastic
> M2   1001 −.6666   8016 −.3334 \$ water
> M3   1001 −.2000 6000 −.4000 8016 −.4000 \$ plastic
> PERT1:n   CELL=1 MAT=2 RHO=−1.0 METHOD=−1
> PERT2:n   CELL=1 MAT=3 RHO=−2.0 METHOD=−1

This example demonstrates how to make significant composition changes (e.g., changing a region from water to plastic.) The unperturbed material is made from a combination of the two desired materials, typically half of each. PERT1 gives the predicted tally as if cell 1 were filled with water and PERT2 gives the predicted tally as if cell 1 were filled with plastic. The difference between these perturbation tallies is an estimate of the effect of changing cell 1 from water to plastic.

V.   *SUMMARY OF MCNP INPUT FILE*

A.   *Input Cards*

The following table lists the various input cards and when they are required. Two kinds of defaults are involved in the following table: (1) if a particular entry on a given card has a default value, that

value is listed in the appropriate location on the card, and (2) the omission of a card from the input file sometimes has a default meaning, and if so, the default description is preceded by an asterisk.

TABLE 3.8:   Summary of MCNP Input Cards

Use	Card and Defaults	

Use	General Categories	Page
optional	Message block plus blank terminator	3–1
required	Problem title card	3–2
required	Cell cards plus blank terminator	3–10
required	Surface cards plus blank terminator	3–12
required	Data cards plus blank terminator	3–22
optional	C Comment card	3–4

	Problem type card	page 3–23
(a)	MODE N	

(a) Required for all but MODE N

	Geometry cards	page 3–23
optional	VOL 0	
optional	AREA 0	
optional	U 0	
optional	TRCL 0	
optional	LAT 0	
optional	FILL 0	
optional	TRn none	

	Variance reduction cards	page 3–32
required	IMP required unless weight windows used	
optional	ESPLT *no energy splitting or roulette	
optional	PWT −1 MODE N P or N P E only	
optional	EXT 0	
optional	VECT none	
optional	FCL 0	
optional	WWE none	
required	WWN required unless importances used	
optional	WWP 5 3 5 0 0 0	
optional	WWG none	
optional	WWGE single energy or time interval	

TABLE 3.8: Summary of MCNP Input Cards

optional	MESH	none
optional	PDn	1
optional	DXC	1
optional	BBREM	none electron photon transport only

	Source specification cards		page 3–49
optional	SDEF	ERG=14 TME=0 POS=0,0,0 WGT=1	
optional	SIn	H I_i ... I_k	
optional	SPn	D P_i ... P_k	
optional	SBn	D B_i ... B_k	
optional	DSn	H J_i ... J_k	
optional	SCn	none	
optional	SSW	SYM 0	
optional	SSR	OLD NEW COL m=0	
(b)	KCODE	1000 1 30 130 MAX(4500,2∗NSRCK) 0	
		6500 1 none	
(c)	KSRC	none	
(b)	ACODE	1000 1 30 130 MAX(4500,2∗NSRCK) 0	
		1 automatic KALSAV+2 6500 0 0	

(b) neutron criticality problems only

(c) KCODE or ACODE only

	Tally specification cards		page 3–73
optional	Fn*a*	$R_o = 0$ for $n = 5$	
optional	FCn	none	
optional	En	very large	
optional	Tn	very large	
optional	Cn	1	
optional	FQn	F D U S M C E T	
optional	FMn	1	
optional	DEn/DFn	none	
optional	EMn	1	
optional	TMn	1	
optional	CMn	1	
optional	CFn	none	
optional	SFn	none	
optional	FSn	none	
optional	SDn	0	

TABLE 3.8:   Summary of MCNP Input Cards

optional	FUn	(Requires SUBROUTINE TALLYX)
optional	TFn	1 1 last last 1 last last last
optional	DD	0.1 1000
optional	DXT	– – – – – 0 0 0
optional	FTn	none

Material specification cards   page 3–108

optional	Mm	no ZAID default; 0; set internally; first match in XSDIR; .01p; .01e
(d)	DRXS	*fully continuous
(d)	TOTNU	*prompt $\bar{\nu}$ for non-KCODE; total $\bar{\nu}$ for KCODE
(d)	NONU	*fission treated as real fission
optional	AWTAB	*atomic weights from cross-section tables
optional	XSn	none
optional	VOID	none
optional	PIKMT	*no photon–production biasing
optional	MGOPT	*fully continuous

(d) neutron problems only

Energy and Thermal cards   page 3–117

optional	PHYS:N	*very large 0 0
optional	PHYS:P	*100 0 0
optional	PHYS:E	*100 0 0 0 0 1 1 1 1
(e)	TMP	2.53 x 10^{-8}
(e)	THTME	0
(e)	MTm	none

(e) neutron problems only

Problem cutoffs   page 3–124

optional	CUT:N	very large 0 −0.5 −0.25 *SWTM*
optional	CUT:P	very large .001 −0.5 −0.25 *SWTM*
optional	CUT:E	very large .001 0 0 *SWTM*
optional	EPLT	cut card energy cutoff
optional	NPS	none
optional	CTME	none

TABLE 3.8:   Summary of MCNP Input Cards

	User arrays		page 3–127
optional	IDUM	0	
optional	RDUM	0	

	Peripheral cards	page 3–128
optional	PRDMP	end −15 0 all 10 rendezvous points
optional	LOST	10 10
optional	DBCN	$(15^{19})^{152917}$ 0 0 0 600 0 0 0 1.E–4 100 0 0
		152917 5^{19} 0 0 0 0 0 0
optional	FILES	none none sequential formatted –
optional	PRINT	*short output
optional	MPLOT	none
optional	PTRAC	none
optional	PERT	none

*This describes the effect of not using this particular card.

B.   *Storage Limitations*

Table 3.9 summarizes some of the more important limitations that have to be considered when setting up a problem. It may be necessary to modify MCNP to change one or more of these restrictions for a particular problem.

TABLE 3.9:   Storage Limitations

Entries in the description of a cell	*1000 after processing
Total number of tallies	NTALMX = 100
Detectors	MXDT = 20
Neutron DXTRAN spheres	MXDX = 5
Photon DXTRAN spheres	MXDX = 5
NSPLT or PSPLT card entries	*10
Entries on IDUM card	*50
Entries on RDUM card	*50

*Set as a dimension in an array