CHAPTER 5: GEOMETRY IN MCNP

Geometry of MCNP is defined by SURFACES, specified by 1st, 2nd or 4th degree polynomials.

Volume CELLS are defined by the intersection, union and complements of REGIONS bounded by the surfaces. A right-handed Cartesians coordinate system is used.

The code begins by determining the intersection of a ray along the direction of the source particles with the surfaces bounding the source's cell. The code determines the minimum positive distance (along direction of travel) to surface. If this distance is greater than the distance to next collision (sampled using the cross section of material present in the cell) and if there is no detector along the track requiring deterministic transport (DXTRAN), the particle leaves the current cell. At the surface intersection, the code determines the next cell the particle will enter, by checking the SENSE of the intersection point for each surface listed for the cell; i.e. the point has to lie in the correct side of all surfaces defining a cell, otherwise the next cell is checked.

Note that if the entire particle path is within the cell, particle track continues by finding the attributes of the next flight. The DXTRAN procedure is explained in Section 9.3.

MCNP also uses the concept of SEGMENT in geometry definition. A segment is a portion of a cell or a surface used for tallying purposes. The FS card is used to divide a cell or a surface into segments. The SD (segment divider card) is used to supply tally-specific information (volume, area or mass).

5.1 Surfaces

MCNP has built-in defined surfaces, defined by mnemonics, such as C/Z for a cylinder parallel to the *z*-axis., see your manual. A surface can be defined in two ways:

1. Provide the appropriate coefficients for built-in surfaces.
2. Supply known points on the surface (useful for setting up geometry from a blueprint). Such surfaces must be unique, real and continuos. They must be either skew planes or surface rotationally symmetric about the *x*, *y* and *z* axes.

5.1.1 Surface Areas

MCNP attempts to calculate the area of all surfaces, unless a surface area is entered in the AREA data card or an SD card. Surface area is sometimes needed for tallying purposes. If the area is not calculated by the code, a fatal error will occur, and the area must be specified explicitly. The SD card can be used for areas of surface segments as well as whole surfaces, while the AREA card can be sued only for area of whole surfaces.

5.1.2 Boundary Surfaces

Specular Reflection

A surface can be designated as a being a specular (mirror) reflector if its number on the surface card is preceded by an asterisk (*). But such surfaces should be used with caution, since they are nor realistic and can lead to misleading answers (do not use them with detector or DXTRAN).

Isotropic Reflection

A surface can be designed as a white boundary, causing isotropic reflection, by preceding its number in the surface card with a plus, +. Such surfaces are useful only in dealing with an infinite scatter, and should not be used with detectors or
DXTRAN.

Periodic Boundaries

These are used to simulate an infinite lattice, say infinite number of fuel rods. It is defined in the surface card by giving the surface number at which the particle should reenter the lattice having left the surface under consideration, e.g. 1 -2 states that a particle leaving the lattice at surface 1 reenters at surface 2. See the limitations listed in the manual on the use of reflected surfaces.

5.2 Cells

Each cell is defined by a CELL card . A cell is given a numbered label and must contain a single material defined by a material number and material density (in g/c3). The cell boundaries are defined by the surrounding surfaces using logical Boolean operators. Note that a zero material number defines a void. If the void is internal, the particle path is simply stretched to cross the void region. If the void is external, the particle track is terminated. An external void usually surrounds the problem's space to kill escaping particles.

MCNP uses the SENSE concept defined for a point x',y',z' as having a +ve sense with respect to a surface $f(x,y,z)$ if $f(x',y',z') > 0$, and vice versa.

A cell can be defined simply by the INTERSECTION operator (simply a blank space between two surface numbers on the cell card, provided that all points in the cell must have the same sense with respect to given bounding surfaces. Therefore, they can be no concave corners in a cell defined only by a intersections.

The UNION operator, a colon (:), allows concave corners in cells and also cells that are completely disjoint. Spaces on either side of union operator are irrelevant. Intersection operations are performed first, followed by unions, but parentheses can be used to clarify operations or force a certain order.

Note that intersection between two regions defines only points that belong to both regions at the same time, while a union defines all points that exist in both regions. For surfaces, say *A* and *B*, *A -B* defines a cell whose points have a +ve sense wrt to surface A and *at the same time* have a -ve sense wrt surface B; while *-A*: *B* defines *everything* in the universe with -ve sense wrt to *A* and a +ve sense wrt *B*. A cell may contain both operations, e.g. *-A B* (*C*:*D*) defines a cell in which the intersection of -*A* and *B* is intersected with the union of *C* and *D*.

The complement operator, #, stands for *not in* and is used to simplify cell specification; #*n* means that the current cell is the complement of cell *n*, while $(...) defines the complement of the portion of the cell description in the parentheses. The form #(*n*) is not allowed. Caution must be used with this operator as it can lead to some confusion, or an unnecessary increase in calculation of intersection of particle trajectory with surfaces. Repeated structure definitions are also allowed, consult the manual.

Attention should be paid to the fact that a unique cell must be found for each particle position and that all points in the simulation space must exist within some cell. A dummy surface may be used to avoid ambiguity, also called ambiguity surface, see manual. The VOID card is useful in debugging geometry and calculating volume. Keep in mid, however, that MCNP cannot detect overlapping cells or gaps between cells until a particle track actually gets lost. The geometry-plotting feature of the code can be helpful.

5.2.1 Cell Volumes

The code calculate the cell volume for most goemtries. However, the VOL or SD cards can be used to enter user-supplied cell volumes.

5.3 Problems

In given your answer, justify each input card and explain how it performs the intended task. Hint: The solution to these problems may be in the MCNP manual.

1. For three concentric boxes, write the MCNP cell cards required to define a cell in the inner-most box and cells in the space contained between the boxes. Keep in mind the problem of convex corners.
2. Coordinate transformation is used in MCNP to describe tilted objects. Use this feature to write the surface cards for a cylindrical can whose axis in the *yz* plan and is tilted 30 degrees from the *y*-axis. The can's centre is at (0,10,15) in the (*x*,*y*,*x*) coordinate system.
3. Use the repeat feature of MCNP to write the geometry specification cards for a CANDU channel consisting of 37 fuel elements. The dimensions of the channel as follows: pressure tube 103.4 mm ID, calandria tube ID 129 mm, fuel element 13 mm OD, fuel pellet OD 12 mm. For the purpose of this exercise, consider the channel to be infinite in the *z*-direction.