CHAPTER 2

INTRODUCTION TO MONTE CARLO SIMULATION

2.1 INTRODUCTION

Monte Carlo simulation of radiation transport is indisputably the most accurate means of predicting dose distributions and other quantities of interest in radiation treatment of patients. The basic tenet of Monte Carlo is that it uses a relatively small, randomly selected sample of particles to predict the average behavior of all particles. Each particle in the sample is transported through various media in which it scatters, deposits energy and produces secondary particles, according to well known fundamental laws of nuclear physics. The secondary particles must also be transported similarly. Specific events that take place during the transport depend upon the type of incident particle, its energy, and the radiation interaction properties of the materials. The dose distribution is determined by accumulating (recording or scoring) contributions from energy deposition events of a very large number of particles to a grid of small voxels, or bins, comprising the medium (phantom or human body, for example). If one is interested in studying other properties of radiation beams, then the corresponding quantities must also be scored accordingly. For instance, to determine the characteristics of fluence incident on a plane at the isocenter, the energy, position, direction of particles (photons, electrons and positrons) at the time of crossing the plane must be recorded. Unfortunately, due to the inordinately long computation times, it has hither-to-forth been impractical for all but for investigational purposes. However, because of continuing advancements in computer technology,
algorithms, and variance reduction techniques, it now appears that we are rapidly approaching a time when we can begin exploiting the benefits of Monte Carlo methodology rapidly. Numerous investigators have proved that the Monte Carlo method is a very accurate calculation tool in simulation regions, where conventional photon and electron algorithms are adequate (Wang and Li 2002; Williamson et al 1991 and Anagnostopoulos et al 2003).

2.2 HISTORY OF MONTE CARLO SIMULATION

Perhaps the earliest documented use of random sampling to solve a mathematical problem was that of Compte de Buffon in 1772 (Compte de Buffon 1777). A century later people performed experiments, in which they threw a needle in a haphazard manner onto a board ruled with parallel straight lines and inferred the value of \( \pi \) from observations of the number of intersections between needle and lines (Hall 1873; Hammersly and Handscomb 1964). Laplace suggested in 1786 that \( \pi \) could be evaluated by random sampling (Marquis Pierre-Simon de Laplace 1786). Lord Kelvin appears to have used random sampling to aid in evaluating some time integrals of the kinetic energy that appear in the kinetic theory of gasses (Lord Kelvin 1901) and acknowledged his secretary for performing calculations for more than 5000 collisions (Wood 1986).

Fermi invented a form of the Monte Carlo method when he was studying the moderation of neutrons in Rome (Wood 1986). Though Fermi did not publish anything, he amazed his colleagues with his predictions of experimental results. After indulging himself, he would reveal that his "guesses" were really derived from the statistical sampling techniques that he performed in his head when he couldn't fall asleep.
During World War II at Los Alamos, Fermi joined many other eminent scientists to develop the first atomic bomb. It was here that Stan Ulam became impressed with electromechanical computers used for implosion studies. Ulam realized that statistical sampling techniques were considered impractical because they were long and tedious, but with the development of computers they could become practical. Ulam discussed his ideas with his colleagues among whom are John von Neumann and Nicholas Metropolis. Statistical sampling techniques reminded everyone of games of chance, where randomness would statistically become resolved in predictable probabilities. It was Nicholas Metropolis who noted that Ulam had an uncle who would borrow money from relatives because he “just had to go to Monte Carlo” and thus named the mathematical method “Monte Carlo” (Necia Grant Cooper 1989).

Meanwhile, a team of wartime scientists headed by John Mauchly was working to develop the first electronic computer at the University of Pennsylvania in Philadelphia. Mauchly realized that if Geiger counters in physics laboratories could count, then they could also do arithmetic and solve mathematical problems. When he saw a seemingly limitless array of women cranking out firing tables with desk calculators at the Ballistic Research Laboratory at Aberdeen, he proposed (Necia Grant Cooper 1989) that an electronic computer be built to deal with these calculations. The result was ENIAC (Electronic Numerical Integrator and Computer), the world's first computer, built for Aberdeen at the University of Pennsylvania. It had 18,000 double triode vacuum tubes in a system with 500,000 solder joints (Necia Grant Cooper 1989).
2.3 GENERAL PURPOSE MONTE CARLO CODES

The initial formulation of Monte Carlo simulations for the transport of high energy electrons was established by Berger. Calling this method as "condensed history theory", Berger combined the theoretical results of the previous generation of research into developing appropriate solutions of the Boltzmann transport equation with numerical algorithms for exploiting the power of computers to permit iterative, piece-wise solution of the transport equation in a computationally intensive but less appropriate fashion. The methods devised by Berger, with comparatively little modification provide the foundation of all present day available Monte Carlo electron transport simulation algorithms. Presently there are many general purpose Monte Carlo codes available in widespread use for the radiation transport calculations in high-energy physics and medicine. The main three popular Monte Carlo general-purpose codes are Berger and Seltzer's ETRAN based codes Monte Carlo N-Particle (MCNP), Integrated Tiger Series (ITS) and Electron Gamma Shower (EGS4). The general-purpose codes, which are familiar in medical radiation dosimetry, are MCNP and EGS4-its user codes are DOSXYZ and BEAM code. A brief description of these codes are given below, except MCNP. As the thesis uses MCNP code, a detailed description of the same will be discussed in section 2.5.

2.3.1 Electron Gamma Shower code

For various reasons, two completely different codes were written in the early-to-mid 1960's to simulate electromagnetic cascades. The first code was written by Zerby and Moran (1962) at Oak Ridge National Laboratory. The second was developed by Nagel (Nagel 1966) known as SHOWER1, the origin code of (Electron Gamma Shower) EGS. Then it was modified by Nicoli (SHOWER2) and then the Stanford Linear Accelerator Center (SLAC)
computation group under took the complete task, under the directions of WR Nelson, a programmer J. Ryder constructed SHOWER3 and it was tested in 1972. At the same time High Energy Physics Laboratory (HEPL) developed SHOWER, now referred as SHOWER4 by Richard Ford (Richard Ford and Walter Nelson 1978). Hofstadter working at SLAC found a couple of errors in SHOWER4 code while executing it on fast IBM-360/91 computers. Then Ford rewritten his code by taking support from Nelson and he renamed this code as Electron Gamma Shower (EGS1). The present available version is EGS4, which is released in 1985 and it has been maintained at SLAC. Some have referred to the EGS code as the de facto gold standard for clinical dosimetry. The latest version, EGS5 is also ready to release by SLAC.

The EGS4 (Nelson et al 1985) computer code system is a general purpose package for Monte Carlo simulations of the coupled transport of electrons and photons in any arbitrary geometry for particles with energies of few keV to several TeV. The advantage of the EGS system is that a structured set of subroutines handles all of the physics in the simulation in a manner which allows users to write their own geometry and scoring routines without actually touching the EGS system itself. The user is responsible for writing the routines which define the geometry using very simple but general interface (HOWFAR and HOWNEAR) and a scoring routine (AUSGAB) which is called under well-specified conditions which allow the user to score virtually any parameter of interest. The cross section data which acts an input to EGS4 is created by another code called PEGS4 (a processor to EGS4). Both the photons and charged particles are transported in random rather than in discrete steps. The dynamic range of charged particles energy goes from a few tens of keV up to a few thousands of GeV and for photons it lies between 1 keV to several hundred GeV. This code is extensively documented and benchmarked. The EGS4 code system take into account of all physics processes such as annihilation of positron at rest or in flight; inelastic Moller
and Bhabha scattering of electrons and positrons from atomic electrons; bremsstrahlung production by positron from interaction with the nucleus and atomic electrons; elastic multiple and single scattering of electrons and positron from nuclei and atomic electrons; pair production by photons; Compton scattering from bound atomic electrons; Rayleigh coherent scattering of photons from atoms; photoelectric interactions of photons with atomic electrons and relation of atom by production of fluorescent x-rays and auger electrons. The EGS4 is a package of main program and subroutines. The PEGS4 is a stand-alone data preprocessing code consisting of 12 subroutines and 85 functions. Some of the EGS4 user codes, which are used for medical radiation dosimetry purposes are DOSRZ, XYZDOS, FLURZ and DOSIMETER.

2.3.2 BEAMnrc code

The BEAMnrc (Rogers et al. 1995) is a Monte Carlo code dedicated for simulating radiotherapy beams from any radiotherapy source, including low energy x-rays, Co-60 units and both photon and electron accelerators. It was developed originally as part of OMEGA project, which was a collaborative work between Ionizing Radiation Standard Group- National Research Council Canada (NRCC) and Rock Maekies group at the University of Wisconsin in Madison to develop a full 3D electron beam treatment planning system based on Monte Carlo simulation to calculate the dose to the patient. BEAM is based on the PRESTA extension of the EGS4 Monte Carlo system for simulating radiation transport. Now, it is maintained by NRCC and it is built on EGSnrc code system. Many investigators used the BEAM to simulate Medical linear accelerator to study various beam characteristics of photons and electrons and benchmarked the beam data by comparing measured standard published data.
2.4 INTRODUCTION TO MONTE CARLO N-PARTICLE (MCNP) CODE

MCNP is a general-purpose, continuous energy, generalized-geometry, time-dependent, coupled neutron/photon/electron Monte Carlo transport code. It can be used in several transport modes such as neutron only, photon only, electron only, combined neutron/photon transport, where the photons are produced by neutron interactions, neutron/photon/electron, photon/electron, or electron/photon. The neutron energy regimes is from $10^{-11}$ MeV to 20 MeV and the photon and electron energy regimes are from 1 keV to 1000 MeV.

2.5 HISTORY OF MCNP CODE

MCNP was originally developed by the Monte Carlo Group, currently the Transport Methods Group, (Group XTM) in the Applied Theoretical and Computational Physics Division (X Division) at the Los Alamos National Laboratory. Group XTM improves MCNP (releasing a new version every two to three years), maintains it at Los Alamos and at other laboratories, where we have collaborators or sponsors and provides limited free consulting and support for MCNP users. MCNP is distributed to other users through the Radiation Safety Information Computational Center (RSICC) at Oak Ridge, Tennessee, and the OECD/NEA data bank in Paris, France.

MCNP has approximately 44,000 lines of FORTRAN and 1000 lines of C source coding, including comments and with the COMMON blocks listed only once and not in every subroutine. There are about 360 subroutines. There is only one source code; it is used for all systems. At Los Alamos, there
are about 250 active users. Worldwide, there are about 3000 active users at about 200 installations.

MCNP has not been successfully vectorized because the overhead required to set up and break apart vector queues at random decision points is greater than the savings from vectorizing the simple arithmetic between the decision points. MCNP (and any general Monte Carlo code) is little more than a collection of random decision points with some simple arithmetic in between. Because MCNP does not take advantage of vectorization, it is fairly inefficient on vectorized computers. In particular, many workstations run MCNP as fast as or faster than the Cray-YMP (Hendricks and Briesmeister 1992). MCNP has been made as system independent as possible to enhance its portability and has been written to comply with the ANSI FORTRAN 77 standard. With one source code, MCNP is maintained on many platforms.

MCNP takes advantage of parallel computer architectures. It is supported in multitasking mode on some mainframes and in multiprocessing mode on a cluster of workstations where the distributed processing uses the Parallel Virtual Machine (PVM) software from Oak Ridge.

John von Neumann was a consultant to both Aberdeen and Los Alamos. When he heard about ENIAC, he convinced the authorities at Aberdeen that he could provide a more exhaustive test of the computer than mere firing-table computations. In 1945 John von Neumann, Stan Frankel, and Nicholas Metropolis visited the Moore School of Electrical Engineering at the University of Pennsylvania to explore using ENIAC for thermonuclear weapon calculations with Edward Teller at Los Alamos (Necia Grant Cooper 1989). After the successful testing and dropping of the first atomic bomb, a few months later, work began in earnest to calculate a thermonuclear weapon. On March 11, 1947, John von Neumann sent a letter to Robert Richtmyer,
leader of the Theoretical Division at Los Alamos, proposing use of the statistical method to solve neutron diffusion and multiplication problems in fission devices (Necia Grant Cooper 1989). His letter was the first formulation of a Monte Carlo computation for an electronic computing machine. In 1947, while in Los Alamos, Fermi invented a mechanical device called FERMIAC to trace neutron movements through fissionable materials by the Monte Carlo Method.

By 1948 Stan Ulam was able to report to the Atomic Energy Commission that not only was the Monte Carlo method successfully used on problems pertaining to thermonuclear as well as fission devices, but also it was applied to cosmic ray showers and the study of partial differential equations (Necia Grant Cooper 1989). In the late 1940s and early 1950s, there was a surge of papers describing the Monte Carlo method and how it could solve problems in radiation or particle transport and other areas (Herman Kahn 1950, Householder et al 1951). Many of the methods described in these papers are still used in Monte Carlo today, including the method of generating random numbers (Lehmer 1951) used in MCNP. Much of the interest was based on continued development of computers such as the Los Alamos MANIAC (Mechanical Analyzer, Numerical Integrator, and Computer) in March, 1952.

The Atomic Energy Act of 1946 created the Atomic Energy Commission to succeed the Manhattan Project. In 1953, the United States embarked upon the “Atoms for Peace” program with the intent of developing nuclear energy for peaceful applications such as nuclear power generation. Meanwhile, computers were advancing rapidly. These factors led to greater interest in the Monte Carlo method. In 1954, the first comprehensive review of the Monte Carlo method was published by Herman Kahn (1954) and the first book was published by Cashwell and Everett (1959).
At Los Alamos, Monte Carlo computer codes developed along with computers. The first Monte Carlo code was the simple 19-step computing sheet in John von Neumann's letter to Richtmyer. But as computers became more sophisticated, so did the codes. At first, the codes were written in machine language and each code would solve a specific problem. In the early 1960s, better computers and the standardization of programming languages such as FORTRAN made possible more general codes. The first Los Alamos general-purpose particle transport Monte Carlo code was MCS (Johnston 1963) written in 1963. Scientists who were not necessarily experts in computers and Monte Carlo mathematical techniques now could take advantage of the Monte Carlo method for radiation transport. They could run the MCS code to solve modest problems without having to do either the programming or the mathematical analysis themselves. MCN could solve the problem of neutrons interacting with matter in a 3D geometry and used physics data stored in separate, highly-developed libraries (Cashwell et al 1972).

In 1973, MCN was merged with MCG (Cashwell et al 1973) a Monte Carlo gamma code that treated higher energy photons, to form MCNG, a coupled neutron-gamma code. In 1977, MCNG was merged with MCP (Cashwell 1973), a Monte Carlo Photon code with detailed physics treatment down to 1 keV, to accurately model neutron-photon interactions. The code has been known as MCNP ever since. Though at first MCNP stood for Monte Carlo Neutron Photon, now it stands for Monte Carlo N-Particle. Other major advances in the 70's included the present generalized tally structure, automatic calculation of volumes, and a Monte Carlo eigenvalue algorithm to determine $k_{\text{eff}}$ for nuclear criticality (KCODE).

In 1983 MCNP3 was released, entirely rewritten in ANSI standard FORTRAN 77. MCNP3 was the first MCNP version internationally
distributed through the Radiation Shielding and Information Center at Oak Ridge, Tennessee. Other 1980s versions of MCNP were MCNP3A (1986) and MCNP3B (1988), that included tally plotting graphics (MC PLOT), the present generalized source, surface sources, repeated structures/lattice geometries, and multigroup/adjoint transport.

MCNP4 was released in 1990 and was the first UNIX version of the code. It accommodated N-particle transport and multitasking on parallel computer architectures. MCNP4 added electron transport (patterned after the Integrated TIGER Series (ITS) continuous-slowing-down approximation physics) (Halblieb and Mehlhorn 1984) the pulse height tally (F8), a thick-target bremsstrahlung approximation for photon transport, enabled detectors and DXTRAN with the S(α, β) thermal treatment, provided greater random number control, and allowed plotting of tally results while the code was running.

MCNP4A, released in 1993, featured enhanced statistical analysis, distributed processor multitasking for running in parallel on a cluster of scientific workstations, new photon libraries, ENDF/B-VI capabilities, color X-Windows graphics, dynamic memory allocation, expanded criticality output, periodic boundaries, plotting of particle tracks via SABRINA, improved tallies in repeated structures, and many smaller improvements.

MCNP4B, released in 1997, features differential operator perturbations, enhanced photon physics equivalent to ITS 3.0, PVM load balance and fault tolerance, cross section plotting, postscript file plotting, 64-bit workstation upgrades, PC X-windows, inclusion of LAHET HMCNP, lattice universe mapping, enhanced neutron lifetimes, coincident-surface lattice capability, and many smaller features and improvements. MCNP4C
released in 2000 features an unresolved resonance treatment, macrobodies, superimposed importance mesh, perturbation enhancements, electron physics enhancements, an alpha eigenvalue, plotter upgrades, cumulative tallies, parallel enhancements and other small features and improvements. Due to non-availability of the code, we limit our study with MCNP4B only.

2.6 MCNP CROSS SECTION DATA

Nuclear cross-section data tables are essential part of the MCNP code. The data tables are produced by analyzing experimentally measured cross sections and combining those data with the predictions of nuclear model calculations. For most materials, there are many cross section sets available because of multiple sources of evaluated data and different parameters used in processing the data. The primary sources of nuclear cross section data for MCNP are evaluations from Evaluated Nuclear Data File (ENDF) (Kinsey 1979), Lawrence Livermore National Laboratory’s Evaluated Nuclear Data Library (ENDL) (Howerton et al 1979) and Activation Library (ACTL) (Gardner and Howerton 1978) and evaluations from the Applied Nuclear Science (T-2) (Foster and Arthur 1981; Arthur and Young 1980). The present existing nuclear data tables are classified into 8 categories. They are (1) Continuous energy neutron interaction data, (2) Discrete reaction neutron interaction data, (3) Photon interaction data, (4) Neutron dosimetry data, (5) Neutron S(α, β) thermal data, (6) Multi group neutron, coupled neutron/photon, and charged masquerading as neutrons, (7) Multi group photon, and (8) Electron data. Each of the nuclear data tables are identified by a ZAID whose general form is ZZZAAA.nnX, where ZZZ represents the atomic weight, AAA atomic number and nn is an evaluation identifier. For non-isotope elements AAA=000.
2.6.1 Neutron interaction data

Mode N problems are continuous (ZZZAAA.nnC) or discrete (ZZZAAA.nnD). Neutron interaction energy table is required for each isotope or element in the problem. MCNP does not access the evaluated data directly and must be processed into ACE format by using complex processing codes such as NJOY (MacFarlane et al 1982) in the case of ENDF/B and MCPOINT (Howerton et al 1983) for ENDL. In recent years the primary evaluated source of neutron interaction data for MCNP has been the ENDF/B system. Recently evaluated neutron interaction data also extracted from ENDL and Nuclear Theory Applications Group at Los Alamos. Cross-section for all reactions given in the evaluated data are specified on the data table. The neutron interaction table contains the cross-section, the total photon production cross-section, angular distributions of scattered neutrons for all reactions emitting neutrons, and heating numbers in the tabulated form, same as neutron interaction cross-section data. Other miscellaneous information on the neutron interaction tables includes the atomic weight ratio of target nucleus, the Q-values of each reaction and $\gamma$ data for fissionable isotopes.

2.6.2 Photon interaction data

Photon interaction tables are required for all photon problems. The form ZAID is ZZZ000.nnP. There are two photon libraries, one is $nn=01$ and another is $nn=02$. For ZAID=ZZZ000.01P photon library, the photon interaction tables for $Z=84, 85, 88, 89, 91$ and $93$ are based on the Storm and Israel (1967) data from 1 keV to 15 MeV. For all other elements for $Z=1-94$, the photon data is based on the evaluated data of ENDF (Hubbell et al 1975) from 1 keV to 100 MeV. The ZAID=ZZZAAA.02P data library is the super set of ZAID=ZZZAAA.01P library with pair production thresholds added for the Storm and Israel data. Data above 15 MeV for the Storm-Israel data and
above 100 MeV for ENDF data is extracted from EPDL (Cullen et al. 1989) and go up to 100 MeV. But it is impractical to run above 1 GeV because electron data only go to 1 GeV. For each nuclide the photon interaction libraries contains an energy grid (logarithms of energies) and are followed by the tables of incoherent and coherent form factors tabulated as function of momentum transfer. The next tables are logarithms of the incoherent scattering, coherent scattering, photo-electric, and pair production cross sections followed by the photon heating numbers. The directions and energies of scattered photons are obtained from well known sampling, Thomson and Klein-Nishina formula. The energy of incoherently scattered photon is calculated from the sampled scattering angles.

2.6.3 Electron interaction data

Electron interaction data tables are required for all electron transport problems and for photon problems in which thick-target bremsstrahlung model is used. The form of ZAID for electron data is ZZZ000.nnE and it has two libraries with nn=01 and nn=03. The electron library contains data on an element by element basis for atomic numbers Z=1-94. The library data contain energies for tabulation, bremsstrahlung energy distributions, K-edge energies, Auger electron production energies, parameters for the evaluation of the Goudsmith-Saunderson theory for angular detections based on the Riley cross-section calculation, and Mott correction factors to the Rutherford cross-section also used in the Goudsmith-Saunderson theory. The e103 data base, which is derived from ITS 3.0 code system (Halbleib et al. 1992), also includes the atomic data of Carlson used in the density effect calculation. Internally, calculated data are electron stopping powers and ranges, K-ray production probabilities, knock-on probabilities, bremsstrahlung angular distributions, and Landui Blunck-Leisegang theory of energy-loss fluctuations.
2.7 MCNP GEOMETRY

The geometry features of MCNP (Briesmeister 1997) deals with the 3D configuration of user defined materials in geometric cells bounded by first and second-degree surfaces and forth degree elliptical tori. The cells are defined from the surfaces by the use of interceptions, unions and compliment operators and surfaces are defined by supplying coefficients of surface equations recognized by the MCNP, in Cartesian coordinate system. It gives the user the added flexibility of defining geometrical regions from all the above mentioned surfaces than is available in most combinatorial geometry codes. The geometry plotting capability in MCNP helps the user in finding the geometry errors. The cells are defined in cell cards and each is described by a cell number, material number, material density followed by a list of operators and signed surfaces bounding the cell. The card is used to describe a single line of input up to 80 characters.

The general form of the cell card is as follows:

```
j  m  d  geom. params
or
j  LIKE n  BUT List
```

where

- **j** = Cell number, \( 1 \leq j \leq 9999 \)
  - If cell has transformation, \( 1 \leq j \leq 999 \).
- **m** = 0, if the cell is void.
  - = material number, if the cell is not void.
- **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\(^3 \) and negative entry 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.

params = optional specification of cell parameters by entries in the keyboard value form.

n = name of another cell.

List = set of keywords = value specification that define the attribute that differ between cell and j.

In cell cards the surfaces are assigned positive (the region above or to the right) or negative (the region below or to the left) sense. Each surface divides all the space into two regions, one with positive sense with respect to the surface and another with negative sense with respect to the surface. The positive or negative senses are represented by assigning the positive or negative value to the surfaces in the cell cards.

The general form of surface cards defined equations is as follows.

\[ j \; n \; a \; \text{List} \]

where,

\( j \) = surface number: \( 1 \leq j \leq 9999 \), with asterisk for a reflecting surface or plus for a white boundary.

If surface defines a cell that is transformed with TRCL,

\[ 1 \leq j \leq 999. \]

\( n \) = 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 recognized by MCNP.

List = one to ten entries, as required.

Using a combinatorial-geometry known as macrobody capability, the cells and surfaces are also defined. The macrobodies can be mixed with the standard cells and surfaces. The macrobody surface is decomposed internally into surface equations and the facts are assigned individual numbers.
according to a predetermined sequence. The space inside a body has a negative sense with respect to the macrobody surface and all its facts. The space outside a body has a positive sense.

2.8 DATA CARD

All MCNP input cards entered after cell cards and following surface card book, with blank card delimiter are known as data cards. These data cards fall into the following categories. No data card can be used more than once with same number or particle type designations. Only some of the important following data cards are discussed briefly in the following sections.

1. Problem type
2. Geometry cards
3. Variance reduction
4. Source specification
5. Tally specification
6. Material and cross section specification
7. Problem cutoffs
8. Energy and thermal treatment
9. User data arrays
10. Peripheral cards

2.8.1 Problem type

The type of particle to be transported mentioned in the user supplied input file, is described on the problem type card.

Form: Mode $x_1, \ldots, x_i$

Where, $x_i = N$ for neutron transport
$P$ for photon transport
$E$ for electron transport

Default: If the MODE card is omitted MODE=$N$ is assumed.
2.8.2 Geometry cards

The various data cards related to geometry cards and their functions are as follows.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Card type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. VOL</td>
<td>Cell volumes</td>
</tr>
<tr>
<td>2. AREA</td>
<td>Surface area</td>
</tr>
<tr>
<td>3. U</td>
<td>Universes</td>
</tr>
<tr>
<td>4. TRCL</td>
<td>Cell transformation</td>
</tr>
<tr>
<td>5. LAT</td>
<td>Lattices</td>
</tr>
<tr>
<td>6. FILL</td>
<td>Fill cards</td>
</tr>
<tr>
<td>7. TR</td>
<td>Coordinate transformation</td>
</tr>
</tbody>
</table>

2.8.2.1 VOL-Cell volume card

Form: VOL x1 x2.....xi

Or       VOL NO

Where, \(x_i = \text{volume of cell } i\), where \(i = 1, 2, ..., \) number of cells in the problem.

\(\text{NO} = \text{no volumes or areas are calculated.}\)

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, then calculated volumes which are mentioned by the user on the VOL card are used.

With VOL card, if the number of entries does not equal to the number of cells in the problem, it is a fatal error. Volumes or masses of cells are required for some tallies. MCNP calculates the volumes of all cells that are rotationally symmetric about any axis, even a skew axis. As a by product
it also calculates areas and masses of all the cells. If a cell volume required for a tally cannot be calculated and is not entered on VOL cards, a fatal error will occur.

2.8.2.2 Repeated structures

The primary goal of the repeated structural capability is to make it possible to describe only once the cells and surfaces of any structures that appears more than once in the geometry. This feature reduces the input and memory use but the problem won’t run away faster than with any other description and extends the concept of MCNP cell. The user can specify the cell, is to be filled with something called universe, which is either a lattice or an arbitrary collection of cells. A single universe, described only once, can be used to fill 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.

- The “LIKE m BUT” feature is shorthand making it is possible to make one cell equivalent to another except for assorted attributes that can be specified with key word = value entries.
- The universe U card is used to specify to what universe the cell belongs.
- The fill card is used to specify with which universe a cell 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 place of the geometry.
- The lattice LAT card is used to define an infinity array of hexagonal or hexahedral prisms.
- The TRn card is used to define the auxiliary co-ordinate system is main co-ordinate system while defining some typical geometry cells.

2.8.3 Estimation of Monte Carlo errors

In MCNP simulation studies the tallies are normalized to be per starting particles and are printed in the output along with a second number, designated by R, which is known as the estimated relative error defined to be one estimated standard deviation of the mean Sx divided by the estimated mean x. For a well-behaved tally, R will be proportional to 1/√(N), where N is the number of histories. For poorly behaved tally, R may increase as the number of histories increases. The R is used to define the confidence intervals about the estimated mean allowing one to make statement about what true result is. As per Central Limit Theory as N approaches to infinity there is a 68% of chance that the true result lies in the range x(1±R) and a 95% chance of range x(1±2R). The confident intervals statements refer only to the precision of the Monte Carlo calculations itself and not to the accuracy of the results compared to the physical value.

For all tallies except the point detector tally, the R should be less than 0.10 to produce reliable confidence intervals and for point detector tallies it should be less than 0.05. If an important but unlikely particle path, in phase space has not been sampled in a problem, the Monte Carlo results will not correct expected values and hence stated confidence interval may not be correct. While the setting problem, it is must that one should not exclude any regions of phase space and should sample all regions of the problem adequately. Despite one’s best effort, an important path may not be sampled.
often enough, causing confidence intervals statement to be incorrect. To inform the user about this behavior, MCNP calculates a Figure Of Merit (FOM) for one tally bin of each tally as a function of the number of histories and print its values in Tally Fluctuation Chart (TFC). The FOM is defined as

\[
\text{FOM} = \frac{1}{R^2 T} \quad (2.1)
\]

Where, T is the computer times in minutes.

The more efficient a Monte Carlo calculation is the larger the FOM value. For a well-behaved tally, the FOM value should be fairly constant. A sharp decrease in the FOM indicates that a seldom-sampled particle path has significantly affected the tally and the relative error estimate.

The estimated relative error R is inversely proportional to \(\sqrt{N}\), where N is the number of particles in the histories. For a MCNP run, the computer time T consumed is proportional to N.

Thus,

\[
R = \frac{C}{\sqrt{N}} \quad (2.2)
\]

where, C is a positive constant.

The R can be reduced in two ways. One is by increasing computer consumption and another is by decreasing the C value. The constant C value depends on the tally choice and or sampling choices. The first approach is limited due to the limitation in the computation speed of the present computers. For second one, to decrease the C value MCNP has special variance reduction technique. In the case of tally choice, the fluence in a cell is estimated either by collision estimate or by track time estimate. The collision is obtained by tallying \(1/\sum t\) where \(\sum t\) macroscopic total cross section at each collision in the cell and the track length estimate is obtained by tallying
the distance, the particle moves inside the cell. As $\Sigma t$ gets very smaller, very few particles have the collision but give enormous tallies when they do, a high variance situation. In contrast, the track length estimates gets a tally from every particle that enters the cell. For this reason MCNP has track length tallies as standard tallies, where as collision tallies are standard.

The effect of sampling choice on the C value can explain by discussing the Analog and Non-Analog Monte Carlo models. The analog Monte Carlo model, uses natural probabilities, holds good when a significant particle contributes to the tally estimate and can be compared to detecting a significant fraction of particles in the physical situation. But there are many cases for which the fraction of particles detected very small, $10^{-6}$ for these cases the analog Monte Carlo fails due to unacceptable statistical uncertainties. In analog Monte Carlo model interested particle tracks, which contributes a large amount to the tally, are followed by rather uninteresting particle tracks. It estimates the same average values as the Monte Carlo model, while often making the variance of the estimate much smaller than the variance for analog estimate. It uses different non-analog technique and they are all meant to increase the odd that a particle scores. To ensure that the average score is the same in the non-analog model as in the analog model, the score is modified to remove the effect of biasing the natural odds. The variance reduction techniques (Forster 1994) are classified into four types ranging from the trivial to esoteric.

2.8.4 Variance reduction tools in MCNP

2.8.4.1 Truncation methods

These are simplest variance reduction methods. They speed up the calculations by truncating the parts of the phase, which don’t contribute
significantly to the solution. Geometry truncation (IMP card), energy cutoff and time cut off come under this classification. The importance of cell is used to terminate the particles history if the importance is insignificant. In energy cutoff truncation, particles are terminated when their energy falls below the energy cutoff. It should be used only when it is known that low energy particles are either zero or almost zero importance. But the low energy particle often produces high energy particles. Thus, even if a detector is not sensitive to low energy particles, the low energy particle may be important to the tally. In the case of time limit cutoff truncation, it terminates particles tracks when their time exceeds the time cutoff and thus decreases pre history. It should be used only in time dependent problems where last time bin will be earlier than the cutoff.

2.8.4.2 Population control methods

This method uses particle splitting and Russian roulette to control the number of samples taken in various regions of phase-space. In important regions, many samples of low weight are tracked and in unimportant regions few samples of high weight are tracked. A weight adjustment is made to ensure that the problem solution remains unbiased. Geometry splitting and Russian roulette, energy splitting-roulette, weight cut-off, and weight windows comes under this category. In geometry splitting with Russian roulette, as particles migrate in an important direction, they are split into a number of identical particles of lower weight according to the IMP ratio. As particles migrate from unimportant direction, Russian roulette is played and the particles are killed or followed with as per IMP ratios. In some cases, particles are most important in some energy ranges than other, in such cases the particles population can be built up rather than rapidly depleted, as would occur naturally with the high photo-electric absorption cross section, by using energy splitting. In some cases, the number of tracks increases with decreasing energy. In such cases, these numbers are reduced by playing Russian roulette game. In weight cut off, Russian roulette is played if a
particle's weight drops below a user specified weight cut off. The particle is either killed or its weight is increased to a user specified level. The weight window is a space energy dependant splitting and Russian roulette technique. For each space-energy phase space cell, the user supplies a lower weight bound. The upper weight bound is a user specified, multiple of the lower weight window. These weight bounds define a window of acceptable weights. If a particle is below the weight bound, Russian roulette is played and the particle's weight is either increased to a value within the window or the particle is terminated. If a particle is above the upper weight bound, it is split so that all the split particles are within window.

2.8.4.3 Modified sampling methods

This method alters the statistical sampling of a problem to increase the number of tallies per particle. For any Monte Carlo event, it is possible from any arbitrary distribution rather than the physical probability as long as particle weights are then adjusted to compensate. Thus with modified sampling methods, sampling is done from distributions that send particles in desired directions or into other desired regions of phase space as time or energy, or change the location or type of collisions. The exponential transform, implicit capture, forced collisions, source biasing and neutron induced photon production biasing, comes under this category. The exponential transform samples the distance to collision from a non-analog probability density functions. Although, many impressive results are claimed for the exponential transform, it should be remembered that these results are usually obtained for one-dimensional geometries and quite often for energy independent problems. Sarkar and Prasad (1979) have done a purely analytical analysis for the optimum transform parameters for an infinite slab and one energy group. The exponential transform allows particle walks to move in preferred directions by artificially reducing the macroscopic cross-section in the preferred direction and increasing the cross-section in the
opposite directions. Implicit capture is a splitting process, where the particle is split into absorbed weight and surviving weight. In some cases, it is often useful in certain cells to increase the number of collisions that produce large detector contributions or large weight DXTRAN particles. In such cases, forced collision variance reduction method is used that increases sampling of collisions in a specified cell. The source variable biasing is used for biasing the MCNP sources in any or all of the source variables specified. MCNP's source biasing allows the production of more source particles, with suitably reduced weights, in the more important regimes of each variable. Source biasing samples from a non-analog probability density function. The source direction can be biased from a continuous exponential function or using cones of fixed size and starting a fixed fraction of particles within each cone. The user can bias a particle in any arbitrary direction or combination of directions. The source biasing is the only variance reduction scheme allowed with F8 tallies having energy binning.

2.8.4.4 Partially-deterministic methods

These are the most complicated class of variance reduction methods. They circumvent the normal random walk process by using deterministic-like techniques, such as next event estimators or by controlling of random number sequence. The point detectors, DXTRAN and correlated sampling come under this method. The point detectors reduce the number of contributions to detector tallies from selected cells that are relatively unimportant to a given detector. DXTRAN typically is used when a small region is being inadequately sampled because particles have a very small probability of scattering towards that region. Upon collision outside the sphere, DXTRAN creates a special DXTRAN particle and deterministically scatters it toward the DXTRAN sphere and deterministically transport it, without collision, to the surface of the DXTRAN sphere. Correlated sampling estimates the change in a quality resulting from a small perturbation of any type in the problem. This technique enables the evaluation of small quantities
that would otherwise be marked by the statistical errors of uncorrelated calculation.

2.8.5 Source Specification Cards

The generalized user-input source capability allows the user to define a wide variety of source conditions without modifying the main code. There are four different source definition cards available in MCNP. They are General Source Card (SDEF), Surface Source Card (SSR), Criticality Source Card (KCODE) and user-supplied source card. Any one of these four source definition card can be used in MCNP input file. Independent probability distributions can be specified for source variables of energy, time, position, direction, and other parameters such as starting cell or surface. These probability functions are specified on the Source Information Cards (SIn), Source Probability Card (SPn), Source Biasing Card (SBn) and Dependent Source distribution Card (DSn). Information about the geometrical extent of the source can also be given. In addition, source variable may depend on the other source variables thus extending the built-in source capabilities of the code. The user can bias all input distributions.

In addition to input probability distributions for source variables, MCNP has certain in-built functions such as Watt, Maxwellian, and Gaussian spectra; Gaussian for time; and isotropic, cosine and mono-directional for direction for fission and fusion studies. MCNP provides the user, three methods to define an initial criticality source to estimate $k_{\text{eff}}$, the ratio of neutrons produced in successive generations in fissile systems.

2.8.5.1 SDEF General Source Card

Form: SDEF source variable=specification....

Use: Required for problems using the general source.
Optional for problems using the criticality.
The specifications of source variables can be any one of the three forms. In the first form, the source variable has an explicit or default value or default distribution. In second form, source variable is given by a probability distribution. In third form, the source variable is dependant on another variable. MCNP samples the source variable in an order set up according to the needs of the particular problem. Each dependant variable must be samples after the variable it depends on has been sampled.

### 2.8.5.2 SSR Surface Source Card

Form: \( \text{SSR} \) \( \text{keyword}=\text{values} \)

The '=' signs are optional.

A surface source allows particles crossing a surface in one problem to be used as the source for subsequent problem. The decoupling of a calculation into several parts allows detailed design of certain geometrical regions without having to return the entire problem from the beginning each time. The surface source has a fission volume source option that starts particles from fission sites, where they were written in a previous run.

### 2.8.6 Tally cards

The tally cards are used to specify what type of information the user would like to get from Monte Carlo calculations such as particle current across a surface, particle flux at a point and energy deposition etc. Currents can be tallied as a function of direction across set of surfaces, surface segments, or sum of surfaces in the problem. Charge can be tallied for electrons and positrons. Fluxes across any set of surfaces, surface segment, sum of cells also available. Similarly, the fluxes at designated detectors (point
or ring) are standard tallies. Heating and fission tallies give the energy deposition in specified cells. A pulse height tally provides the energy, distribution of functions created in a detector by radiation. In addition, particles may be flagged when they cross-specified surfaces or enter designated cells, and the contribution of these flagged tallies are listed in Table 2.1.

Table 2.1 Contribution of flagged tallies

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Tally Description</th>
<th>Fn units</th>
<th>*Fn units</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1:N or F1:P or F1:E</td>
<td>Current integrated over a surface</td>
<td>particles</td>
<td>MeV</td>
</tr>
<tr>
<td>F2:N or F2:P or F2:E</td>
<td>Flux averaged over a surface</td>
<td>particles/cm²</td>
<td>MeV/cm²</td>
</tr>
<tr>
<td>F4:N or F4:P or F4:E</td>
<td>Flux averaged over a cell</td>
<td>particles/cm²</td>
<td>MeV/cm²</td>
</tr>
<tr>
<td>F5a:N or F5a:P</td>
<td>Flux at a point or ring detector</td>
<td>particles/cm²</td>
<td>MeV/cm²</td>
</tr>
<tr>
<td>F6:N or F6:N,P or F6:P</td>
<td>Energy deposition averaged over a cell</td>
<td>MeV/g</td>
<td>jerks/g</td>
</tr>
<tr>
<td>F7:N</td>
<td>Fission energy deposition averaged over a cell</td>
<td>MeV/g</td>
<td>jerks/g</td>
</tr>
<tr>
<td>F8:P or F8:E or F8:P,E or +F8:E</td>
<td>Energy distribution of pulses created in a detector</td>
<td>pulses</td>
<td>MeV</td>
</tr>
<tr>
<td></td>
<td>Charge deposition</td>
<td>charge</td>
<td></td>
</tr>
</tbody>
</table>
2.8.7 Material and cross section specification

The cards in this section specify both the isotopic composition of the materials and the cross-section evaluations to be used in the cells.

Material (Mm) Card:

The general form of the material card is as follows.

\[ \text{Mm ZAID1 fraction1 ZAID2 fraction2} \]

The m on a material card corresponds to the material number on the cell card.

Nuclide Identification Number (ZAID) is used to identify the element or nuclide desired. The form of the number is ZZZAAA.nnX, where ZZZ is the atomic number of the element or nuclide, AAA is the mass number of the nuclide, ignored for photons and electrons, nn is the cross-section evaluation identifier; if blank or zero, a default cross-section evaluation will be used, and X is the class of data; C is continuous energy; D is discrete reaction; T is thermal; Y is dosimetry; P is photon; E is electron; and M is multigroup.

2.8.8 VOID card

The VOID card removes all materials and cross sections in a problem and sets all nonzero importances to unity. It is very effective for finding errors in the geometry description because many particles can be run in a short time. Flooding the geometry with many particles increases the
chance of particles going to most parts of the geometry in particular, to an incorrectly specified part of the geometry and getting lost. The history of a lost particle often helps locate the geometry error.

2.8.9 Problem cut off cards

Problem cutoff cards are used to specify parameters for some of the ways to terminate execution of MCNP.

<table>
<thead>
<tr>
<th>Mnemonics</th>
<th>Card type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUT</td>
<td>Cut-offs</td>
</tr>
<tr>
<td>ELPT</td>
<td>Cell-by-cell energy cut-offs</td>
</tr>
<tr>
<td>NPS</td>
<td>History cut-offs</td>
</tr>
<tr>
<td>CTME</td>
<td>Computer time cut-offs</td>
</tr>
</tbody>
</table>

2.9 PROGRAM FLOW FOR RUNNING MCNP

The following has been input into the computer.

- Source dimension, position, distribution, photons generated, their energy, probability, and the direction of propagation
- Atomic composition of sources and materials used
- Types of tallies used
- Calculation geometry and
- Number of histories etc.,

The MCNP processes the input file step by step in the following manner. Initiation (MCNP). It includes the following steps.
Read input file (INP) to get dimensions (PASS1);

- Set up variable dimensions or dynamically allocated storage (SETDAS);
- Re-read input _le (INP) to load input (RDPROB);
- Process source (ISOURC);
- Process tallies (ITALLY);
- Process materials specifications (STUFF) including masses but without loading in the data files;
- Calculate cell volumes and surface areas (VOLUME).

Interactive Geometry Plot (PLOT): It is used to check the geometry error of the problem.

Cross Section Processing (XACT):

- Load libraries (GETXST);
- Eliminate excess neutron data outside problem energy range (EXPUNG);
- Doppler broaden elastic and total cross sections to the proper temperature if the problem temperature is higher than the library temperature (BROADN);
- Process multigroup libraries (MGXSPT);
- Process electron libraries (XSGEN) including calculation of range tables, straggling tables, scattering angle distributions, and bremsstrahlung.
MCRUN: It sets up multitasking and multiprocessing, runs histories (by calling TRNSPT, which calls HISTORY), and returns to OUTPUT to print, write RUNTPE dumps, or process another criticality (KCODE) cycle.

Under MCRUN, MCNP runs neutron, photon, or electron histories (HISTORY), calling ELECTR for electron tracks:

- Start a source particle (STARTP);
- Find the distance to the next boundary (TRACK), cross the surface (SURFAC) and enter the next cell (NEWCEL);
- Find the total neutron cross section (ACETOT) and process neutron collisions (COLIDN) producing photons as appropriate (ACEGAM);
- Find the total photon cross section (PHOTOT) and process photon collisions (COLIDP) producing electrons as appropriate (EMAKER);
- Use the optional thick-target bremsstrahlung approximation if no electron transport (TTBR);
- Follow electron tracks (ELECTR);
- Process optional multigroup collisions (MGCOLN, MGCOLP, MGACOL);
- Process detector tallies (TALLYD) or DXTRAN;
- Process surface, cell, and pulse height tallies (TALLY).

Periodically write output_le, restart dumps, update to next criticality (KCODE) cycle, rendezvous for multitasking and updating detector and DXTRAN Russian roulette criteria, etc.
Output

- Go to the next criticality cycle (KCALC);
- Print output le summary tables (SUMARY, ACTION);
- Print tallies (TALLY P);
- Generate weight windows (OUTWWG).

Plot tallies, cross sections, and other data (MC PLOT).

GKS graphics simulation routines.

PVM distributed processor multiprocessing routines.

Random number generator and control (RANDOM).

Mathematics, character manipulation and other slave routines.

2.10 APPLICATIONS OF MONTE CARLO TECHNIQUE IN MEDICINE

Numerical analysis techniques are an indispensable tool in many branches of Physics, including Radiation Physics. The Monte Carlo method use random variables, which is well suited for radiation transport and to measure the interested quantities. In radiation transport, the physical process, leading to the deposition of energy in a medium irradiated with photon beam are random in nature. When a photon beam irradiates a medium (patient), it sets in motion the secondary electrons and scattered photons, which go on to produce their own interactions. The generation of these particles is random in nature; the resulting path lengths are randomly distributed; the type of radiation produced, the amount of energy transfer and the scatter direction are also random.
A Monte Carlo simulation of the physical processes may be considered if we know the probability distribution function that governs each step. Using such functions, one can simulate each step with numerical variables, construct the correct sequence of such events, and measure the appropriate statistics for results. The results can be compared with the experimentally measured data and the initial parameters of the models may be revised appropriately i.e. the Monte Carlo techniques can be used to benchmark the photon and electron beam data. Radhe Mohan (1997) has suggested that “there is a potential for an improvement in clinical outcome if the accuracy in dose is improved with the aid of Monte Carlo simulation of radiation transport”. This is supported by Lymperopoulou et al (2004). The Monte Carlo technique serves the above statement and are extensively useful in various branches of Medicine such as Radiation Oncology (Teletherapy and Brachytherapy), Diagnostic Radiology, Nuclear Medicine, shielding calculations in radiation protection and in radiobiology studies, are discussed below.

2.10.1 External radiotherapy

Monte Carlo applications in external beam radiotherapy have the calculation of standard beam parameters such as central axis depth dose and off-axis profiles. In addition, Monte Carlo is frequently used to obtain data where measurement is not possible or there is any significant uncertainty. Many investigators demonstrated with data based evidence, that Monte Carlo methods are valuable tool in predicting the dose, irradiated patient volumes with better accuracy.
2.10.2 Brachytherapy dosimetry

Monte Carlo methods have found extensive use in Brachytherapy radiation dosimetry, due to difficulties involved in direct measurement near the sources. To study various parameters of presently available radioactive sources such as exposure rate constant, radial dose functions, anisotropy functions, photon attenuation, self-absorption, scattering and shielding requirements, Monte Carlo codes are used. The application of Monte Carlo methods in brachytherapy started in early 1980s. Williamson and Li (1995) have developed a very sophisticated photon Monte Carlo code, which is optimized for dose calculations around a brachytherapy source. They have used this code to calculate extensive tables of dose distributions around the brachytherapy sources. He observed a lot of difference between his simulated data with that of the time existing data. Most of his work has found its way into commercial TPSs. These calculations and others like them have become the basis of a detailed dosimetry protocol for brachytherapy (Nath et al 1995).

Wuu et al (1996) used MCNP and DELTA code, to estimate the RBE of four brachytherapy sources. The accuracy of commercial BPS Brachytherapy Planning System (Nucletron, Veemendal, Netherland) with regard to near field dosimetry, in particular, the superposition effect of multiple source locations, was investigated by Wong et al (1999). Several investigating groups (Fessenden et al 1996 and Karaiskos et al 2003) used Monte Carlo codes to investigate the dosimetric properties of HDR Ir-192 brachytherapy source.

2.10.3 Diagnostic radiology

The last few decades have seen increasing evidence of harmful effects of radiation on human tissues (ICRU Report-48 1992). This leads to
greater concern for the quantification of radiation doses to patients undergoing diagnostic radiology examination. The appropriate parameter for this purpose is the effective dose, which implies determination of the doses to the critical organs. Since these cannot be obtained in real patients experimentally, other methods use patient entrance surface dose measured with TLDs and the output of the unit or dose-area product. But these were intended to quality control methods rather than providing information about the determinant suffered by the patient. The Monte Carlo method is one of the most under utilized area covering a wide range of diagnostic applications, from the patient exposure and dose to image contrast and resolution. By using Monte Carlo techniques Jones and Wall (1985) NRPB in UK, Drexler et al (1990) and Alonso et al (1999) separately estimated different organ dose under certain assumptions.

The Monte Carlo simulations were used to reconstruct tomographic image. Brockoff et al (1996) were the team to attempt this type of problem by using Monte Carlo simulations employing a mono-energetic source with array of detectors. A Medical Internal Radiation Dose (MIRD) committee anatomical phantom was placed between the source and detector array. The phantom was rotated and profiles obtained an interval of 2° upto 180° total. Reconstruction was performed offline. Monte Carlo simulation was used to calculate organ and effective doses from CT procedures (Jansen et al 1996).

2.10.4 Nuclear Medicine

In Nuclear Medicine imaging techniques, the quantity of radionuclides in the human body is quantified by measuring radiation by using radiation detector (Scintillation detector, NaI). During the data acquisition, a considerable amount of radiation are scattered and/or attenuated within the patient body and also the activity outside the filed of view may
contribute to image. All these factors lead an error in measurements that requires corrections. Through Monte Carlo simulations, it is possible to distinguish between scattered and unscattered radiation. The Monte Carlo methods are used to study the response of scintillation counters (Shimoyama et al 2000) to ionizing radiation used in Gamma camera and designing various nuclear detectors. The Monte Carlo techniques are found more useful and accurate method in finding the absorbed dose in different organs by radioactive medicine administered into the patient for imaging or therapy purpose where it is very difficult to find the absorbed dose directly.

2.11 Advantages and limitations of Monte Carlo method

Advantages

- Arbitrary complex geometries can be specified to the desired accuracy
- Calculations can be carried out with the desired precision
- The dose distribution within a mm range from the sources are computed precisely

Limitations

- As millions of histories are needed to generate a precise result, it is a time consuming technique
- There is a need for detailed knowledge of the incident radiation beam, dimensional data of calculation geometry and atomic composition of materials