An Introduction to MCNP

Edward T. Norris

December 14, 2015

Contents

Chapter 1

Introduction

Why did I write this book? The manual sucks. LOL, not really. But it does. A lot.

1.1 Forewarning

First, reader be forewarned, the contents herein pertain to MCNP6. Since MCNP6 was based directly on MCNP5 (and hence earlier versions), many of the same concepts can be drag-and-dropped verbatim. However, there are some differences between the two. MCNP6 was a merger between the standard at the time, MCNP5 and MCNPX.

If discrepancies are found, always err on the side of the manual.

This book is not comprehensive, it should be regarded as an aide in learning the foundations of MCNP.

¡Purpose of 5/X¿

Every version of MCNP comes with a comprehensive manual with the code itself.

1.2 Obtaining MCNP6

MCNP must be obtained through the Radiation Safety Information Computational Center at Oak Ridge National Laboratory (https://rsicc.ornl.gov). MCNP is written and maintained by scientists at Los Alamos National Laboratory, funded by the U. S. Department of Energy. As government code developed for analysis of nuclear systems, it is heavily regulated. MCNP itself is subject to 10 CFR §810.

Particularly for non-U.S. citizens.

The Fortran source code is under special restriction

Go to the RSICC.

1.3 Outline

The structure of this book is built to suit an undergraduate understanding of MCNP6. The emphasis herein is not on the theory and mathematics, though they will be mentioned when they are applicable, but rather on the mechanical operation of MCNP6 as a computational tool to perform analysis. The following chapter will provide a quick summary of the entire book, mapping out the structure of an MCNP6 file as well as the core components needed. Subsequent chapters will focus on a single element of the input file at a time.

1.4 Formatting

This is my book. It will occasionally have comments about giraffes. Because they're fucking awesome.

1.5 Logistics

Uh, yeah, there might be some.

Chapter 2

A Crash Course

I suspect that some readers require more of an immediate start as opposed to an in depth documentary on the intricacies of each and every option MCNP6 has to offer. They are more interesting in running a simple problem or modifying an existing input file to perform some additional calculation.

This chapter will do two things; first it will go over the overall structure of a MCNP6 input, second, it will provide a single-chapter crash course on running an input file for those that simple need to run a file.

2.1 Anatomy of an Input File

An input file is split into three sections called decks. The first deck, contains the problem title card and the cells in the problem. The second deck contains surfaces used to compose the cells in the aforementioned deck. Finally, the last deck contains everything else in the file such as material compositions, source specifications, and tally information, needed to run MCNP. The three decks are traditionally called the cell deck, surface deck, and data deck respectively to reflect their purpose.

Chapter 3

Surfaces

There are a couple different kinds of surfaces.

3.1 Surfaces Defined by Equations

Surfaces are the most primitive geometric objects in MCNP. Surfaces are defined by a mathematic equation in 3D space. Surfaces can be planes, cylinders, toroidal, or one of many other types. Each type of surface in MCNP is defined by a mnemonic that identifies it and informs MCNP which type of surface it is.

Mnemonic	Description	Equation	Variables
P	Arbitrary Plane	x	x
PX	Plane perpendicular to X axis	x	x
PY	Plane perpendicular to Y axis	x	x
PZ	Plane perpendicular to Z axis	x	x
S	Arbitrary sphere	x	x
SX	Sphere centered on X axis	x	x
SY	Sphere centered on Y axis	x	x
SZ	Sphere centered on Z axis	x	x
SO	Sphere centered at origin	x	x
C/X	Cylinder parallel to X axis	x	x
C/Y	Cylinder parallel to Y axis	x	x
C/Z	Cylinder parallel to Z axis	x	x
CX	Cylinder centered along X axis	x	x
CY	Cylinder centered along Y axis	x	x
CZ	Cylinder centered along Z axis	x	x
K/X	Cone parallel to X axis	x	x
K/Y	Cone parallel to Y axis	x	x
K/Z	Cone parallel to Z axis	x	x
KX	Cone centered on X axis	x	x
KY	Cone centered on Y axis	x	x
KZ	Cone centered on Z axis	x	x
SQ	2^{nd} order equation parallel to axis	x	x
GQ	Arbitrary 2^{nd} order equation	x	x
TX	Torus parallel to X axis	x	x
TY	Torus parallel to Y axis	x	x
TZ	Torus parallel to Z axis	x	x

3.2 Macrobodies

There are a number of geometric volumes used repeatedly that can be constructed out of the simple

3.3 Practice

The first and foremost piece of advice I have for anyone writing MCNP files, particularly if the geometry is hard complex or otherwise difficult to visualize, don't start writing your input file until every surface and cell is drawn and labeled with pen and paper!

My second piece of advice is to adopt a numbering convention throughout your input file and remain consistent.

3.4 Examples

Use pen and paper!

Problems

1. squid.
2. dog.
3. Create a surface.

Chapter 4

Simple Cells

Math for the win.

Chapter 5

Repeated Structures

Yeah, they repeat.

5.1 Like X But Y

This is like that, but... well, not.

5.2 Lattices

They suck a lot. Yeah, a whole lot.

5.3 Universes

They go inside themselves.

Chapter 6

Materials

Like copper, iron, and cotton candy.

6.1 Cross Section Libraries

They have data. Making your own custom libraries.

Chapter 7

Source Definitions

This is going to be a really long chapter

7.1 Point Sources

They are simple

7.2 Energy Distributions

There are quite a few types of them

7.3 Beam and Cone Sources

Pencil beams, and cones for ice cream.

7.4 Volume Sources

Lots of them!

Chapter 8

Tallies

These are useful.

8.1 Surface Tallies

Across the surface, ho!

8.2 Volume Tallies

Through the volume, ho!

8.3 Point Detectors

They are magical and perfect.

8.4 Mesh Tallies

They're all meshy.

8.5 Modeling real detectors

It's hard. I quit.

Chapter 9

Debugging Problems

This chapter is devoted to finding and fixing problems. All problems can generally be split into one of two categories. The first category contains verbose, or fatal, errors. These errors cause the program to cease execution and usually print an error message (or just a stack trace of garbage). The second, while it is hard to believe, are much more severe: silent errors. Silent errors do not stop execution, nor do they produce warnings or messages to the user otherwise. They simply produce incorrect results.

9.1 Geometry Errors

This section will cover some simple, yet useful techniques to analyze geometry and identify errors.

Chapter 10

Visualizing

If you're blind, just skip this chapter.

10.1 MCNP Plotter

It sucks.

10.2 VisEd

It's not quite as bad as the plotter, but it still sucks. A lot.

Chapter 11

Postprocessign Data

Like Matlab?

Chapter 12

Theory

You’re reading the wrong book. Sorry buddy.