LA-13265

A Primer for Criticality Calculations with DANTSYS

Los Alamos
NATIONAL LABORATORY

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

Edited by Patricia W. Mendius, Group CIC-1
Prepared by M. Ann Nagy, Group XTM

An Affirmative Action/Equal Opportunity Employer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither The Regents of the University of California, the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by The Regents of the University of California, the United States Government, or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of The Regents of the University of California, the United States Government, or any agency thereof. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

LA-13265

UC-714
Issued: August 1997

A Primer for Criticality Calculations with DANTSYS

*Robert D. Busch**

**Contractor with Los Alamos National Laboratory, from the Nuclear Criticality Safety Group at The University of New Mexico, Albuquerque, NM.*

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

TABLE OF CONTENTS

ABSTRACT xi

INTRODUCTION xiii

Chapter 1 — DANTSYS Quickstart 1-1

1.1 WHAT YOU WILL BE ABLE TO DO: 1-1
1.2 DANTSYS INPUT FILE FORMAT 1-1
1.2.A Title Cards 1-2
1.2.B General Card Format 1-3
1.3 EXAMPLE PROBLEM 1-3
1.3.A Problem Description 1-4
1.3.B Title Cards 1-4
1.3.C Block 1 Input 1-5
1.3.D Block 2 Input 1-6
1.3.E Block 3 Input 1-6
1.3.F Block 4 Input 1-7
1.3.G Block 5 Input 1-8
1.3.H Block 6 Input 1-9
1.4 RUNNING DANTSYS 1-10
1.4.A Output 1-11
1.5 SUMMARY 1-12

Chapter 2 — Input Control (Block 1) 2-1

2.1 WHAT YOU WILL BE ABLE TO DO: 2-1
2.2 PROBLEM DESCRIPTION 2-1
2.2.A Reflected Sphere Geometry 2-1
2.2.B Title 2-2
2.2.C Block 1 — Controls Input 2-3
2.2.C.1 Geometry 2-3
2.2.C.2 Cross Sections 2-4
2.2.C.3 Quadrature 2-4
2.2.C.4 Coarse Meshes 2-6
2.2.C.5 Fine Meshes 2-7
2.2.C.6 Materials and Zones 2-7
2.2.C.7 Other Useful Block 1 Keywords 2-8
2.2.C.8 Complete Title and Block 1 Input for Example Problem 2-9
2.2.D Other Input for Example Problem 2-9
2.2.D.1 Block 2 Input 2-10
2.2.D.2 Block 3 Input 2-10
2.2.D.3 Block 4 Input 2-10
2.2.D.4 Block 5 Input 2-11
2.2.D.5 Block 6 Input 2-11

2.2.E Complete Input for Bare Configuration 2-12
2.2.F Running Problem 2a 2-13
2.2.G Output 2-13
2.3 CONFIGURATION 2 — BARE SYSTEM 2-14
2.3.A Block 2 Changes 2-14
2.3.B Block 1 Changes 2-15
2.3.C Output for the Bare Configuration 2-15
2.4 SUMMARY 2-16

Chapter 3 — Geometry Description (Block 2) 3-1

3.1 WHAT YOU WILL BE ABLE TO DO: 3-1
3.2 PROBLEM DESCRIPTION 3-1
3.2.A Reflected Sphere Geometry 3-1
3.2.B Title and Block 1 Input 3-2
3.2.C Block 2 — Geometry Description 3-2
3.2.C.1 Coarse Meshes 3-3
3.2.C.2 Fine Meshes 3-5
3.2.C.3 Materials and Zones 3-8
3.2.D Other Input for Example Problem 3-9
3.2.D.1 Block 3 Input 3-10
3.2.D.2 Block 4 Input 3-10
3.2.D.3 Block 5 Input 3-10
3.2.D.4 Block 6 Input 3-11
3.2.E Complete Input for Reflected Sphere 3-11
3.2.F Running Problem 3a 3-12
3.2.G Output 3-12
3.3 EFFECTS OF MESHING AND QUADRATURE ON K-EFFECTIVE 3-13
3.4 CONFIGURATION 3b — REFLECTED CYLINDER 3-15
3.4.A Example Problem 3b Data 3-16
3.4.B Block 2 Changes 3-17
3.4.B.1 Coarse Meshes 3-17
3.4.B.2 Fine Meshes 3-18
3.4.B.3 Material Zone Assignment 3-19
3.4.B.4 Block 1 Changes 3-21
3.4.C Running Problem 3b 3-22
3.4.D Output for the Cylinder Configuration 3-22
3.5 SUMMARY 3-23

Chapter 4 — Material Description (Block 4) 4-1

4.1 WHAT YOU WILL BE ABLE TO DO: 4-1
4.2 PROBLEM DESCRIPTION 4-1
4.2.A Reflected Sphere Geometry 4-1
4.2.B Title, Block 1, Block 2, and Block 3 Input 4-2
4.2.C Block 4 — Materials and Mixtures 4-3
4.2.C.1 Material Types and Zones 4-3

4.2.C.2 Material Types 4-4
4.2.C.3 Material Zones 4-5
4.2.D Other Input for Example Problem 4-7
4.2.D.1 Block 5 Input 4-7
4.2.D.2 Block 6 Input 4-7
4.2.E Complete Input for Reflected Sphere 4-8
4.2.F Running Problem 4a 4-9
4.2.G Output 4-9
4.3 BLOCK 4 INPUT FOR TWO-DIMENSIONAL PROBLEMS 4-10
4.4 EXAMPLE PROBLEM 4b USING WEIGHT AND ATOM FRACTIONS 4-10
4.4.A Specifying Weight or Atom Fractions 4-10
4.4.B Weight Fractions 4-11
4.4.C Atom Fractions 4-12
4.4.D Block 4 Input 4-13
4.4.E Block 1 Changes 4-13
4.4.F Running Problem 4b 4-14
4.4.G Output for the Weight and Atom Fraction Specification Approach 4-14
4.5 SUMMARY 4-15

Chapter 5 — Solver Input (Block 5) 5-1

5.1 WHAT YOU WILL BE ABLE TO DO: 5-1
5.2 PROBLEM DESCRIPTION 5-1
5.2.A Reflected Sphere Geometry 5-1
5.2.B Title, Block 1, Block 2, Block 3, and Block 4 Input 5-2
5.2.C Block 5 — Solver Input 5-3
5.2.C.1 IEVT Keyword 5-3
5.2.C.2 ISCT Keyword 5-4
5.2.C.3 ITH Keyword 5-4
5.2.C.4 IBL and IBR Keywords 5-4
5.2.C.5 Applying Boundary Conditions to Physical Problems 5-7
5.2.C.6 Optional Block 5 Keywords 5-9
5.2.D Other Input for Example Problem 5-11
5.2.E Complete Input for Reflected Sphere 5-12
5.2.F Running Problem 5a 5-13
5.2.G Output 5-13
5.3 BLOCK 5 INPUT FOR TWO-DIMENSIONAL PROBLEMS 5-13
5.4 EXAMPLE PROBLEM 5b FINITE CYLINDER 5-13
5.4.A Specifying Additional Boundary Conditions 5-14
5.4.B Block 5 Input 5-14
5.4.C Complete Input File 5-14
5.4.D Running Problem 5b 5-15
5.4.E Output 5-16
5.5 EXAMPLE PROBLEM 5c FINITE CYLINDER — SYMMETRY PLANE 5-16
5.5.A Specifying Boundary Conditions with Symmetry Planes 5-16
5.5.B New Input for Problem 5c 5-16

7.2.E Output 7-4
7.2.F Dimensional Search with a Given k_{eff} 7-4
7.2.G Output for Problem 7aa 7-5
7.3 DIMENSIONAL SEARCH FOR TWO-REGION PROBLEMS 7-5
7.3.A Basic Input for Two-Region Problems 7-5
7.3.B Radius Modifiers for Two-Region Problems 7-6
7.3.C Problem 7b Results 7-7
7.3.D Two-Region Problem With Fixed Outside Radius 7-7
7.3.E Basic Input for Problem 7c 7-8
7.3.F Block 5 Input With Fixed Outside Dimension 7-9
7.3.G Output for Problem 7c 7-9
7.4 THREE REGION PROBLEMS 7-10
7.4.A Fixed Outside Radius And Constant Thickness Middle Zone 7-10
7.4.B Basic Input for Problem 7d 7-10
7.4.C Block 5 Input 7-11
7.4.D Problem 7d Results 7-12
7.5 DIMENSIONAL SEARCH FOR A TWO-DIMENSIONAL PROBLEM 7-12
7.5.A Two-dimensional Example Problem 7-12
7.5.B Basic Input for Dimensional Searches in r-z Geometry 7-13
7.5.C Block 5 Input For Two-Dimensional Search 7-14
7.5.D Problem 7e 7-15
7.6 CONCENTRATION SEARCHES 7-15
7.6.A Block 4 Input for Concentration Searches 7-15
7.6.B Example Problem for Concentration Search 7-17
7.6.C Input for Concentration Search Example 7-17
7.6.D Problem 7f 7-19
7.7 SUMMARY 7-19

Chapter 8 — REFERENCES R-1

Appendix A — Commonly Encountered Error Messages in DANTSYS A-1

Appendix B — Calculating Atom Densities B-1

I. Single Constituent — Given: mass density B-1
II.a. Multiple Materials — Given: weight fractions and mixture density B-2
II.b. Multiple Materials — Given: weight fractions and individual material densities B-3
III. Multiple Materials — Given: atom fractions and atom mixture density B-4
IV. Calculating weight fractions, atom fractions, and average atomic weight when one set of fractions is known B-5
V.a. Molecules — Given: chemical structure and mass density B-6
V.b. Molecules with Mixtures of Isotopes — Given: chemical structure and mass density B-7

VI. Solution Systems B-10
VI.a. Solutions — Given: H/X ratio, fissile component density, and chemical formula B-10

Appendix C — Specifications & Atom Densities Of Selected Materials C-1

Appendix D — The 167 Isotope Hansen-Roach Cross-Section Library D-1

Appendix E — Resonance Absorption and the Hansen-Roach Cross Sections E-1

A PRIMER FOR CRITICALITY CALCULATIONS WITH DANTSYS

by

Robert D. Busch

ABSTRACT

With the closure of many experimental facilities, the nuclear safety analyst has to rely on computer calculations to identify safe limits for the handling and storage of fissile materials. Although deterministic methods often do not provide exact models of a system, a substantial amount of reliable information on nuclear systems can be obtained using these methods if the user understands their limitations. To guide criticality specialists in this area, the Nuclear Criticality Safety Group at the University of New Mexico (UNM) in cooperation with the Radiation Transport Group at Los Alamos National Laboratory (LANL) has designed a primer to help the analyst understand and use the DANTSYS deterministic transport code for nuclear criticality safety analyses. (DANTSYS is the new name of the group of codes formerly known as: ONEDANT, TWODANT, TWOHEX, TWOGQ, and THREEDANT.) The primer assumes a college education in a technical field, but there is no assumption of familiarity with neutronics codes in general or with DANTSYS in particular. The primer is designed to teach by example, with each example illustrating two or three DANTSYS features useful in criticality analyses.

Starting with a Quickstart chapter, the primer gives an overview of the basic requirements for DANTSYS input and allows the user to quickly run a simple criticality problem with DANTSYS. This chapter is not designed to explain either the input or the DANTSYS options in detail; it introduces basic concepts that are further explained in following chapters. Each chapter has a list of basic objectives at the beginning identifying the goal of the chapter and the individual DANTSYS features covered in detail in the chapter example problems. On completion of the primer, it is expected that the user will be comfortable doing criticality calculations with DANTSYS and can handle 60-80% of the situations that normally arise in a facility. The primer provides a set of basic input files that can be selectively modified by the user to fit each particular problem.

This primer provides a starting point for the criticality analyst using DANTSYS. Complete descriptions of input, output, and algorithms are provided in the DANTSYS manual. Although self contained, the primer is intended as a companion volume to the DANTSYS manual. Specific examples of using DANTSYS for criticality analyses are provided in the primer while the manual provides information on the use of DANTSYS in all aspects of particle transport calculations. A little extra time spent going through the primer and doing the examples will save many hours of confusion and embarrassment later. After studying the primer, the user should find it a valuable tool to help make good, solid criticality analyses with DANTSYS.

INTRODUCTION

With the closure of many experimental facilities, the nuclear criticality safety analyst is increasingly required to rely on computer calculations to identify safe limits for the handling and storage of fissile materials. However, in many cases the analyst has little experience with the specific codes available at his or her facility. Typically, two types of codes are available: deterministic codes such as ANISN or DANTSYS that solve an approximate model exactly, and Monte Carlo Codes such as KENO or MCNP that solve an exact model approximately. Often the analyst feels that the deterministic codes are too simple and will not provide the necessary information, so most modeling uses Monte Carlo methods. This sometimes means that hours of effort are expended to produce results available in minutes from deterministic codes. A substantial amount of reliable information on nuclear systems can be obtained using deterministic methods if the user understands their limitations. To guide criticality specialists in this area, the Nuclear Criticality Safety Group at the University of New Mexico (UNM) in cooperation with the Radiation Transport Group at Los Alamos National Laboratory (LANL) has designed a primer to help the analyst understand and use the DANTSYS deterministic transport code for nuclear criticality safety analyses. (DANTSYS is the new name of the group of codes formerly known as: ONEDANT, TWODANT, TWOHEX, TWOGQ, and THREEDANT.) The primer assumes a college education in a technical field, but there is no assumption of familiarity with neutronics codes in general or with DANTSYS in particular. The primer is designed to teach by example, with each example illustrating two or three DANTSYS features useful in criticality analyses.

Starting with a Quickstart chapter, the primer gives an overview of the basic requirements for DANTSYS input and allows the user to quickly run a simple criticality problem with DANTSYS. This chapter is not designed to explain either the input or the DANTSYS options in detail; it introduces basic concepts that are further explained in following chapters. Each chapter has a list of basic objectives at the beginning identifying the goal of the chapter and the individual DANTSYS features covered in detail in the chapter example problems. On completion of the primer, it is expected that the user will be comfortable doing criticality calculations with DANTSYS, and can handle 60 to 80% of the situations that normally arise in a facility. The primer provides a set of basic input files that can be selectively modified by the user to fit each particular problem.

After describing the input requirements for criticality calculations in the previous six chapters, the seventh chapter of the primer provides instruction on using DANTSYS to perform searches on dimension and concentration. There is a wealth of understanding to be gained through simple sensitivity calculations. Deterministic codes in general, and DANTSYS in particular, are ideal for these types of analyses. In less than an hour, with a few input changes, several analyses can be done to determine the effects of parameters on system multiplication. Examples of these analyses are: effect of uranium enrichment, impact of

replacing aluminum with steel, or effect of solution height in a vessel on the multiplication factor of the system. This last chapter and the ones on input requirements provide a useful reference for the specialist. Besides the primer chapters, there are five appendices detailing: typical error messages and remedies, formulas for calculating atom densities, common materials with compositions and densities, isotopes in the 167 isotope Hansen-Roach library, and methods for determining sigma-p for Hansen-Roach cross sections. This information is provided in appendices so as not to obscure the basic information illustrated in each example.

Although much of the information to do an analysis is provided for the user in the primer, there is no substitute for understanding the problem and the theory of neutron interactions. The DANTSYS code can only analyze the problem as it is specified; it will not necessarily identify inaccurate modeling of the geometry nor will it know when the wrong material has been specified. The user is reminded that a single calculation of k-effective with DANTSYS or any other code is meaningless without an understanding of the context of the problem, the quality of the solution, and a reasonable idea of what the result should be.

The primer provides a starting point for the criticality analyst using DANTSYS. Complete descriptions are provided in the DANTSYS manual. Although self contained, the primer is intended as a companion volume to the DANTSYS manual. Specific examples of using DANTSYS for criticality analyses are provided in the primer while the manual provides information on the use of DANTSYS in all aspects of particle transport calculations. A little extra time spent going through the primer and doing the examples will save many hours of confusion and embarrassment later. After studying the primer, the user should find it a valuable tool to help make good, solid criticality analyses with DANTSYS.

Chapter 1
DANTSYS *Quickstart*

1.1 WHAT YOU WILL BE ABLE TO DO:

- Describe the structure of DANTSYS input files.
- Explain the format requirements for title cards and keywords in DANTSYS input files.
- Set up and run a simple criticality problem on DANTSYS.
- Interpret k_{eff} information from DANTSYS mini-print output.

1.2 DANTSYS INPUT FILE FORMAT

The DANTSYS[1] input file is set up in a block structure with each block relating to a different part of the problem description.

Block 1 (*Controls*) describes basic problem information such as the geometry, numbers of coarse and fine meshes, quadrature, and number of energy groups.

Block 2 (*Geometry*) describes the information related to the geometry such as the positions of coarse mesh boundaries, number of fine meshes in each coarse mesh region, and the zone identifier for each region.

Block 3 (*Cross Sections*) describes neutron cross-section information such as the location and type of cross-section library, number of isotopes, types of cross sections and location of types in library.

Block 4 (*Mixing*) describes information about the various mixtures and isotopes in the system such as the isotope, atom density/fraction, which material zone a mixture is assigned to, and the volume fraction of each mixture.

Block 5 (*Solver*) describes solver information such as the type of evaluation, the boundary conditions, and print options for cross sections, fluxes, etc.

Block 6 (*Edit*) describes information used to set up edit options and obtain printout of data such as collapsed group fluxes at a point or in a zone, or reaction rates for a particular isotope.

A DANTSYS input file consists of some or all of the above blocks depending on the type of problem being analyzed and the amount and type of output desired. In addition to the data blocks, an input file starts with title information. Figure 1.1 shows the input file structure for a problem with all six data blocks.

Title Cards
.....
Block 1 Cards - (Controls)
.....
..... **t**
Block 2 Cards - (Geometry)
.....
..... **t**
Block 3 Cards - (Cross Sections)
.....
..... **t**
Block 4 Cards - (Mixing)
.....
..... **t**
Block 5 Cards - (Solver)
.....
..... **t**
Block 6 Cards - (Edit)
.....
..... **t**

Figure 1.1 — DANTSYS input file structure.

1.2.A Title Cards

The first card in the DANTSYS file is the title line control card. This card is the only one in a DANTSYS file which has a fixed format requirement. The entry in the 6th column of the card indicates the number of title lines which follow the title line control card. Each title line card is used to describe the problem and may contain up to 72 characters. There is no limit on the number of title lines but most users find that 4 lines are sufficient. The first title line is echoed in various places throughout the DANTSYS output.

1.2.B General Card Format

For all cards other than the title line control card, there are no fixed format requirements. Each block has a series of keywords associated with it. These keywords can be located anywhere on the lines within a block, but <u>each keyword must be immediately followed by an equals sign</u>. So the general format is: `keyword=  values`

The `values` may be placed anywhere on the card or on following cards if required.

```
keyword=  value1   value2    value3                 value4
 value5                    value6                      value7
                 value8
```

If you wish, values may be separated by commas rather than by spaces.

The other format requirement is that each block is terminated by a `t`. Because each block has numerous keywords associated with it and you seldom need all of the keywords for any given problem, the `t` is used to terminate the block and tell the input processor that data entry for that block is complete.

Comments may be added to any card by starting with a / and then providing the comment after that. The / may appear anywhere on a card; if it is the first character on a card, then the entire card is a comment card. No data may be entered after a / ; all characters and spaces on a card after a / are treated as comments. Comments end when you go to the next card. Thus, if you need to continue a comment, then each continuation card must have a / at the beginning. Other than / and a `t` by itself, there are no limits on characters which may be used in DANTSYS input files.

1.3 EXAMPLE PROBLEM

This introduction should provide enough information to run a simple example problem. It is our intent that you gain confidence in using DANTSYS right away, so we walk through this sample problem step by step, explaining each line of input. For the present, it is important that you enter this problem exactly as we describe it. As you gain more experience with DANTSYS, you may find other ways to set up input files that are more logical to you. For example, you may find it easier to identify values for the Geometry information in Block 2 before providing geometry-specific Controls information in Block 1.

1.3.A Problem Description

This problem is a bare sphere of delta-phase plutonium metal (density of 15.61 g/cc) with a coating of nickel (also known as the Jezebel reactor). Experimental parameters are:

delta-phase Pu metal sphere:

radius	= 6.38493	cm
N_{239} = Atom density of Pu-239	= 3.7047e-2	atoms/b-cm
N_{240} = Atom density of Pu-240	= 1.751e-3	atoms/b-cm
N_{241} = Atom density of Pu-241	= 1.17e-4	atoms/b-cm
N_{Ga} = Atom density of Ga	= 1.375e-3	atoms/b-cm

Spherical nickel coating:

thickness	= 0.0127	cm
N_{Ni} = Atom density of Ni	= 9.1322e-2	atoms/b-cm

Now you are ready to begin entering the example problem. First open a new file named `example1`. All text shown in the `courier` font is what you need to type in. In this *Quickstart* chapter, as each new card is discussed, it is indicated by an arrow in the left margin. The first line in the file must be the title lines control card. This is then followed by the title lines and then the six input blocks.

1.3.B Title Cards

The first card is the title lines control card which indicates the number of title lines used in this problem. For the example problem, there will be two title lines so a 2 is placed in column 6. The overstrike character, **ƀ** , is used throughout this primer to identify a required blank space on a line.

→ **ƀƀƀƀƀ**2

The first title line will read as follows:

→ `Jezebel problem.  Bare plutonium sphere w/ nickel shell`

The second title line will read:

→ `Using 167 isotope Hansen-Roach cross sections, 16 groups`

1.3.C Block 1 Input

The Block 1 input provides the control information to DANTSYS and identifies numbers of parameters which will be provided to the code. We will provide a comment line which identifies this data as belonging to Block 1.

→ `/ *** block 1 ***`

Now there are some required keywords for Block 1. These include:

igeom	Geometry type (slab, cylinder, or sphere)
ngroup	Number of energy groups
niso	Number of isotopes in the cross-section library
isn	S_n order to be used (must be an even integer)
im	Number of coarse mesh intervals in problem
it	Total number of fine mesh intervals in problem
mt	Number of mixtures (material types) defined in Block 4
nzone	Number of material zones defined in Block 4

There is one other Block 1 keyword that is optional, but we will use it to get a condensed output.

miniprt	Mini-print, **0** = No / **1** = Yes (or use the words, **no** or **yes**)

We will enter these keywords on two lines as follows (remember it is free format; the keywords are in no particular order but each keyword must be immediately followed by an equals sign):

→ `igeom= sph  ngroup=16  niso=167  isn=16`

→ `im=2  it=27  mt=2  nzone=2  miniprt=yes  t`

Note that the terminator, `t`, for Block 1 is included on the second card. This concludes the Block 1 input section.

The entire input file at this point should look like:

```
ƀƀƀƀƀ2
Jezebel problem.  Bare plutonium sphere w/ nickel shell
Using 167 isotope Hansen-Roach cross sections, 16 groups
/ *** block 1 ***
  igeom= sph  ngroup=16  isn=16  niso=167
  im=2  it=27  mt=2  nzone=2  miniprt=yes  t
```

1.3.D Block 2 Input

The Block 2 input provides the geometry information to DANTSYS. We will provide a comment line which identifies this data as belonging to Block 2.

```
→    / *** block 2 ***
```

These are the required keywords for Block 2.

xmesh	Coordinates of the edges of the coarse meshes
xints	`Number` of fine meshes in each coarse mesh
zones	Zone number for each coarse mesh (any entry must not be greater than the value given to `nzone` in Block 1)

We will enter these on three lines as follows:

```
→    xmesh= 0.0    6.38493   6.39763
→    xints=      25          2
→    zones=       1          2            t
```

Note that the terminator, `t`, for Block 2 is included on the third card. This concludes the Block 2 input section.

1.3.E Block 3 Input

The Block 3 input provides the information concerning the cross-section library. Because this is a *Quickstart* chapter, only a brief description of the cross-section information will be provided. As cross sections are the heart of any neutronics calculation, extensive discussion of cross sections and their use in DANTSYS will be provided in Chapter 6. Note that throughout this primer, the Hansen-Roach cross-section library is used. However, this library is not a part of DANTSYS; in fact, DANTSYS has no library directly associated with it. The 167 isotope Hansen-Roach library is available from the Nuclear Criticality Safety Group, ESH-6, at LANL. It is used throughout the primer because it has shown to be applicable to most criticality problems, and it does not require reprocessing for each new problem.

We will provide a comment line which identifies this data as belonging to Block 3.

```
→    /*** block 3 ***
```

If the cross-section library is in binary format, there is only one required keyword for Block 3.

`lib`	Name and form of cross-section data file

For our example problem, we have a binary format library so only the one keyword is required in Block 3. We will enter this as follows:

```
→    lib= bxslib          t
```

Again, the terminator, `t`, for this block is included on the card. This concludes the Block 3 input section.

1.3.F Block 4 Input

The Block 4 input provides the material and mixing information to DANTSYS. First, we provide a comment line which identifies this data as belonging to Block 4.

```
→    / *** block 4 ***
```

These are the required keywords for Block 4.

`matls` Instructions for mixing the isotopes into mixtures (material types)

`assign` Assignment of mixtures to material zones

We will use eight lines to provide the Block 4 information in this example problem. First, we will identify two mixtures: the fuel and the nickel plate. The fuel will contain the plutonium and gallium isotopes while the plate contains the nickel. Determination of what isotopes go where is somewhat up to the user but all of the isotopes in a given mixture must be homogeneously mixed. After the mixtures are specified, then we need to identify how these mixtures are combined in material zones and what the volume fraction or density of each mixture is in a zone.

First we will enter the isotopes for the mixture called `fuel`. The input is as shown:

```
→    matls= fuel    pu492e1    3.7047e-2 /pu239 atom density
→                   pu405e1    1.7510e-3 /pu240 atom density
→                   pu41       1.1700e-4 /pu241 atom density
→                   ga         1.3750e-3;/gallium atom density
```

After entering the last isotope for this mixture, there is a semicolon, `;`, at the end of the atom density for gallium. This tells the program that all of the isotopes for this mixture have been entered. The semicolon is only necessary when two or more mixtures are to be entered. Note that since we have not finished the data for Block 4, no terminal `t` has been entered.

Now we will enter the information for the mixture called `plate`.

```
→         plate     ni         9.1322e-2 /nickel atom density
```

This is the end of the mixture information but we still need to assign the mixtures to the material zones. In Block 2, we associated two zones, 1 and 2, with the coarse meshes. The first material zone is the inner sphere which contains the fuel and gallium; the second material zone is the nickel plating. For each mixture zone, there is only 1 mixture and that mixture occupies 100% of the volume of the material zone. We will use a separate line for each zone although there is no requirement to do so.

```
→    assign=   core      fuel      1.0;
→              shell     plate     1.0
→    t
```

(For use in Block 2, the zone <u>number</u> is simply the entry number in the `assign=` array, e.g., `core` is the first entry — zone 1, `shell` is the second entry — zone 2, etc.) Note that the terminator, `t`, for this block has been placed on a separate line. This is just another way of doing the input for any of the six blocks. Also note that since there was more than one zone, the information for the first zone, `core`, is terminated by a semicolon before the information for the second zone is entered. This concludes the Block 4 input section.

1.3.G Block 5 Input

The Block 5 input provides the solver information to DANTSYS. This includes problem type (eigenvalue, search, etc.), boundary conditions, and types of information to print. We will provide a comment line which identifies this data as belonging to Block 5.

```
→    / *** block 5 ***
```

These are the required keywords for Block 5.

ievt	Type of calculation desired (=**1** for a k_{eff} calculation)
isct	Highest order of Legendre scattering to be used in problem
ib**r**	Right Boundary Condition (=**0** for vacuum, =**1** for reflective)

Fluxes and Neutron Densities calculated in DANTSYS are not normalized to any particular value. It is usually easier to analyze a problem when the fission density is normalized to one, so we use an optional keyword, **norm**, to do this. In most criticality calculations, complete convergence of the flux is not required. To reduce run time, there is an optional keyword, **kcalc**, which converges the k_{eff} to 0.001 and does not fully converge the angular fluxes. This option is used in most of the primer example problems. We enter these five keywords on one line as follows:

→ `ievt=1 isct=1 ibr=0 norm=1.0 kcalc=1`

Because the 167 isotope H-R library does not contain values of χ , we must enter the fission fractions appropriate to plutonium. This is done using the keyword, **chi**.

→ `chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0 t`

The `10r0.0` is shorthand for repeat the value `0.0` ten times. Note that the terminator, t, for Block 5 is included on the card. This concludes the Block 5 input section.

1.3.H Block 6 Input

Block 6 provides the edit information which allows you to get reaction rates, fluxes, or sources at individual points or in individual zones. This block also allows you to do a mass edit on your materials to indicate total mass and density by zone. Mass edits require that atomic weights be given in the cross-section library. The 167 isotope Hansen-Roach library contains atomic weights so mass edits can be performed with this library. It is a good idea to use the mass edit feature on each problem so that you can be assured that you have accounted for all of the materials that you believe are in the system. We will provide a comment line which identifies this data as belonging to Block 6.

→ `/ *** block 6 ***`

There are no required keywords for Block 6, but we will use the optional keyword for mass edit.

massed Print mass inventories by zone (**=1** for a print by solver zone)

We enter this on one line as follows:

→ `massed=1 t`

Again, the terminator, `t`, for Block 6 is included on the card. This concludes the Block 6 input section.

The input required to run this example is now complete and should look like the following:

```
ƀƀƀƀƀ2
Jezebel problem.  Bare plutonium sphere w/ nickel shell
Using 167 isotope Hansen-Roach cross sections, 16 groups
/ *** block 1 ***
  igeom= sph  ngroup=16  isn=16  niso=167
  im=2  it=27  mt=2  nzone=2  miniprt=yes          t
```

```
/ *** block 2 ***
  xmesh= 0.0    6.38493   6.39763
  xints=      25         2
  zones=       1         2          t
/ *** block 3 ***
  lib= bxslib            t
/ *** block 4 ***
  matls=fuelpu492e13.7047e-2 /pu239 atom density
                 pu405e1 1.7510e-3 /pu240 atom density
                 pu41    1.1700e-4 /pu241 atom density
                 ga      1.3750e-3;/gallium atom density
         plate   ni      9.1322e-2 /nickel atom density
  assign= core      fuel      1.0;
         shell     plate      1.0
  t
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0  kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0        t
/ *** block 6 ***
  massed=1          t
```

1.4 RUNNING DANTSYS

We will assume that DANTSYS has been installed on the machine you are using and that the executable is named `dant.x`. The default names of the input and the output files are ODNINP and ODNOUT, respectively. To run DANTSYS with different files names, type `dant.x` < and then the input filename followed by > and the name of the output file. Note that your cross-section library must be present in the same directory as the DANTSYS executable and the example1 input file. For this case, we are assuming that you are using the Hansen-Roach 167 isotope library and that it is in binary format. This means that a file called *bxslib* must be in the same directory as the executable and the input file.

To run this example problem creating ex1out as the output file, type

```
dant.x < example1 > ex1out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0031 as written on the screen echo.

1.4.A Output

First, let's assume that the run was successful. With the mini-print option, the output for this problem consists of the following sections:

- echo of input
- brief review of input values
- eigenvalue convergence monitor
- group edit and balance tables
- integral summary information
- mass edit

A couple of checks should be made to verify the numerical solution. First, in the group edit and balance tables, check that the *in scatter* total is equal to the *out scatter* total (within four decimal places). Then check the integral summary information to make sure that the integral-absorption plus the integral-net leakage add to 1.0 (true when `norm` is set to 1.0). Finally, check the mass edit and make sure that the total mass of material and the total density in each zone are reasonable. In the example problem, you should have a total mass in zone 1 of 17.02 kg with an average density of 15.61 g/cm^3. These are reasonable values for a critical delta-phase plutonium sphere.

If your model ran successfully but there was no mass edit output, then probably there were no atomic weights found in the cross-section library. To alleviate this problem, either enter atomic weights for each isotope you are using or choose a library which contains atomic weights. Mass edits are not necessary for the analysis of a problem, but they are quite useful in assuring you that the problem was modeled as you intended.

If your input did not run successfully, the error messages will be listed on the screen during execution or in the output file. In most cases, the errors are related to input data problems. Check to make sure your input file has the same data as the one listed in this chapter. Although it is free format, there are a couple of fixed requirements. Check to make sure all keywords have the equals sign immediately following (no spaces); that each of the six blocks is terminated with a `t`, and that keywords with multiple values have the semicolon at the end of each set of values. Also note that on many workstations, DANTSYS will only handle lower case entries, and DANTSYS does not allow the use of tabs or other special characters for spacing. One other caveat — the error checker in DANTSYS is sequential. Once the first error is found, due to the disruption of

the normal input stream, other errors may be listed that are not truly errors. Thus, you need to correct the first identified error and then re-run the program. This is especially true if you omit a terminator. Keywords for one block will not be correctly recognized in other blocks, but that doesn't mean the keywords are in error.

1.5 SUMMARY

This chapter has helped you to:

- Describe the structure of DANTSYS input files. Know that there are six input blocks with each block requiring certain information.
- Explain the format requirements of DANTSYS input files. Know that except for the title control, all input is free format, using keywords.
- Set up and run a simple criticality problem on DANTSYS.
- Interpret k_{eff} information from DANTSYS mini-print output. Perform simple checks for reasonableness on your output.

Now that you have successfully run DANTSYS, you are ready to learn in detail the options available in each input block and how to set up more complex problems. The following chapters present these details in a similar format to that used in this *Quickstart* chapter.

Chapter 2
Input Control (Block 1)

In the *Quickstart* chapter you ran a simple problem with DANTSYS and gained some confidence in using the code. This chapter and subsequent chapters provide a more detailed explanation of the commands used in the *Quickstart* chapter. Example problems are taken from LA–10860–MS[2] and represent computational models of criticality benchmark experiments. Each chapter and associated example problems are selected to focus on a block of the DANTSYS input file.

2.1 WHAT YOU WILL BE ABLE TO DO:

- Describe Block 1 input and explain the relationships between this input and input for other Blocks.
- Discuss the differences in input requirements between one-dimensional and two-dimensional problems.
- Describe the output which is available from the mini-print option.
- Use the balance print table output to check your results and to determine fractions of neutrons which leak and which are absorbed.

2.2 PROBLEM DESCRIPTION

This chapter examines a plutonium metal sphere in two different configurations (LA–10860–MS, p.101); a bare (unreflected) system and a graphite reflected system. In each configuration, the plutonium sphere has a radius of 5.465 cm. The graphite in the reflector has a density of 1.63 g/cc and is 3.83 cm thick.

2.2.A Reflected Sphere Geometry

The reflected plutonium sphere is modeled first. This example is similar to that used for the *Quickstart*, but will focus on the input required for Block 1. As noted in LA–10860–MS, the Plutonium is delta-phase containing 1 weight percent gallium and is coated with 0.013 cm of nickel. (The table in LA–10860–MS gives the Pu density as 15.8 g/cc which is assumed to be the density of the plutonium-gallium mixture.) The data for this example follows.

Core Material (ρ_{mix} = *15.8 g/cc-mixture)*

Delta phase Pu metal (100 percent Pu-239, 99 wt %)

N_{239} = 3.9404e-2 atoms/b-cm

Gallium (1 wt %)

N_{Ga} = 1.3647e-3 atoms/b-cm

Shell Material

Nickel

ρ = 8.9 g/cc-mixture (single material)

N_{Ni} = 9.1305e-2 atoms/b-cm

Reflector Material

Graphite

ρ = 1.63 g/cc-mixture (single material)

N_C = 8.1724e-2 atoms/b-cm

The atom densities are calculated using the methods described in Appendix B.

Configuration I. *Reflected Plutonium Sphere*

Plutonium and Gallium sphere:	Radius	= 5.465 cm
Nickel Coating:	Thickness	= 0.013 cm
Graphite Reflector:	Thickness	= 3.83 cm

2.2.B Title

To start, we will provide a title for this problem. Remember the first line of a DANTSYS input file is the only one with a fixed format. This line tells the program how many title lines follow for the particular problem. For this case, three title lines are used, so we need to put a 3 in column 6 (5 blank spaces followed by a 3). The three title lines can have any information on them that you think will help describe the problem. In this case, we will label the problem with the geometry and the materials and indicate the source of our cross-section library. The first four lines of your input file should look like:

```
ƀƀƀƀƀ3
Problem 2a – Plutonium and Gallium sphere w/ Nickel Coating
             Reflected with 3.83 cm graphite
Radius = 5.465 cm , using Hansen-Roach cross sections, 16 groups
```

Remember the ƀ is used to indicate a blank space on an input line.

2.2.C Block 1 — Controls Input

As mentioned in the *Quickstart* Chapter, Block 1 input describes the basic problem information such as geometry (shape), numbers of coarse and fine meshes, quadrature, and number of energy groups. All of the input in Block 1 (as in all the blocks) is in free format with a keyword followed by its appropriate values. Because free format is used, somewhere the user needs to tell DANTSYS how many values to expect for each keyword. This is done in Block 1.

First we will provide a comment line which identifies this data as belonging to Block 1.

```
/ *** block 1 ***
```

Now there are some required keywords for Block 1. These include:

igeom	Geometry type (slab, cylinder, or sphere)
ngroup	Number of energy groups
niso	Number of isotopes in the cross-section library
isn	Sn order to be used (must be an even integer)
im	Number of coarse mesh intervals in problem
it	Total number of fine mesh intervals in problem
mt	Number of mixtures (material types) defined in Block 4
nzone	Number of material zones defined in Block 4

In the *Quickstart* chapter, we entered values for each of these keywords without much discussion of why or what options were available. In this chapter, we will give details on each of the Block 1 keywords typically used in criticality analyses.

2.2.C.1 Geometry

The first keyword, **igeom**, is used to identify the geometry of the problem. There are three different one-dimensional geometries available in DANTSYS:

slab	(infinite slab of a given thickness)
cylinder	(infinite cylinder of a given radius)
sphere	(sphere of a given radius)

The value for the **igeom** keyword can be entered as the full word describing the geometry or as an abbreviation (e.g., `slab`, `plane`, `cylinder`, `cyl`, `sphere`, `sph`).

In addition to the one-dimensional geometries, DANTSYS also has three two-dimensional geometries available:

x-y slab	(infinite in z direction)
r-z cylinder	(finite cylinder with no angular dependence)
r-theta	(infinite cylinder with angular dependence in the radial direction)

The values for these are: `x-y`, `r-z`, and `r-theta`. For this problem, we are analyzing a sphere so we will use the three character abbreviation, `igeom=sph`.

2.2.C.2 Cross Sections

There are two Block 1 keywords that deal with the cross-section library: **ngroup** and **niso**. The **ngroup** keyword describes the number of energy groups in the cross-section library while **niso** defines the number of isotopes in the cross-section library for which data is available. Each cross-section library has a different group structure with different numbers of energy groups and isotopes. There is no default available for either of these two keywords; you must enter the values appropriate to your cross-section library. Even with the so-called "standard" Hansen-Roach library,[3] there are variations in the number of isotopes depending on the source of the library. (The 167 isotope Hansen-Roach library is available from ESH–6 at LANL.) However, this library is not a part of DANTSYS; in fact, DANTSYS has no library directly associated with it. The Hansen-Roach cross-section library is used with DANTSYS throughout the primer because it has been shown to be applicable to most criticality problems, and because it does not require reprocessing for each new problem. For this problem and throughout the primer, we characterize the cross-section library with `ngroup=16` and `niso= 167`.

2.2.C.3 Quadrature

Because DANTSYS is a discrete ordinates code, both the Spatial and the Angular variables must be 'converted' from continuous values to discrete values. For the spatial variables, this is done through meshing while the angular variables use quadrature. These quadratures are

discrete directions in space. Each discrete direction can be visualized as a point on the surface of a unit sphere with an associated surface area, w_m. The w_m denotes the weight of the discrete direction. The combination of discrete direction cosines and their associated weights is called a quadrature set.

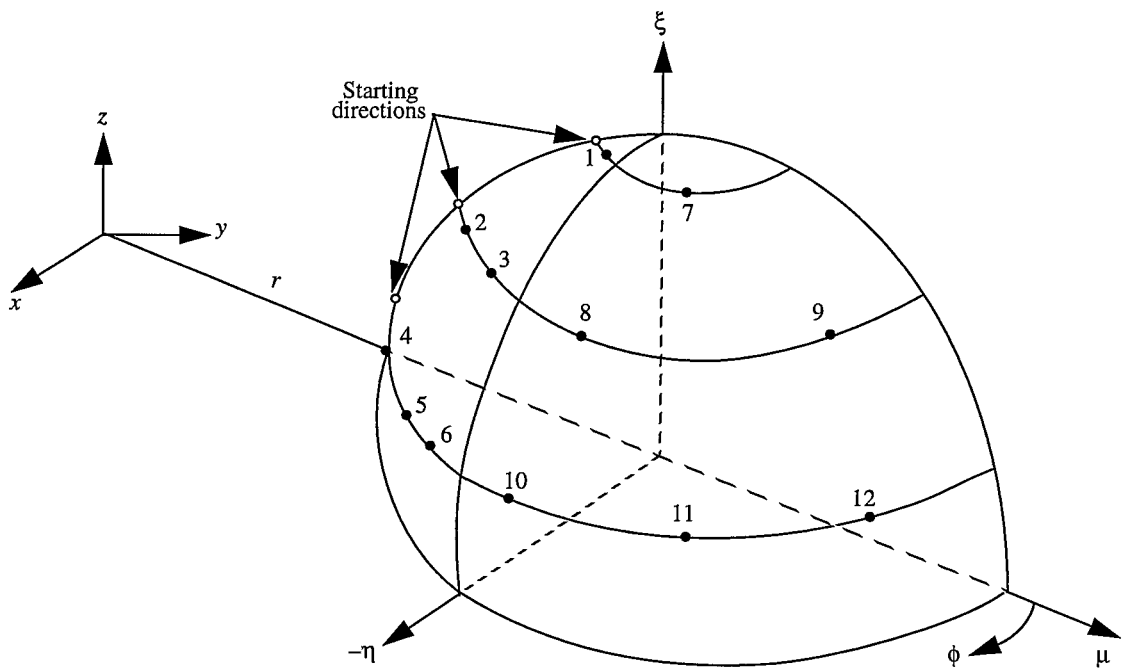


Figure 2.1 — Fully Symmetric S_6 Quadrature.

According to O'Dell and Alcouffe,[4] "Considerable work has been devoted to developing suitable quadrature sets for discrete ordinates codes. Although characterized by the name 'discrete ordinates method' and customarily referred to as simply the S_n method, the selection of a quadrature set to be used within the method is somewhat arbitrary. Accordingly, two S_n calculations, identical in all respects except differing quadrature sets, may yield differing results. For most problems, the differences are small, but the user should be aware of the potential for non-negligible differences."

For most criticality problems, the general default quadrature sets available in DANTSYS are sufficient. No user input is required to select the default set. However, the quadrature order is

a required user input. The quadrature order can be thought of as the number of direction cosines which are to be used to characterize the angular dependence in a problem. Similar to meshes, the higher the order of quadrature, the finer the angular distinction; the higher the order of quadrature, the more time will be required to solve the problem. Thus, the user must make a trade-off between precision of angular specification and problem solution time. For most one-dimensional cases, a quadrature of 16 should be sufficient, although a reasonably high quadrature, 1-D runs are usually quite short. The effect of quadrature order on k_{eff} will be discussed in Chapter 3.

The value assigned to the **isn** keyword determines the S_n subscript as used in a problem (must be an even integer). This S_n order should be entered as a positive number which will select the P_n based default quadrature set. Since this set is not the best for one-dimensional cylinders, a Block V keyword, iquad, needs to be set to obtain the Galerkin quadrature set which is appropriate for 1D cylinders. The value for iquad is 4. In this problem, we will use a quadrature order of `16` which will be entered as `isn=+16`. (Note: the positive sign is discretionary and is only entered here for illustrative purposes.)

2.2.C.4 Coarse Meshes

Just as the angles must be divided into discrete increments, so must the spatial dimension(s). This is done through a combination of *coarse meshes* to define homogeneous volumes and *fine meshes* to divide those volumes into discrete intervals. For a one-dimensional problem, the user must decide on how many coarse meshes are required in the x- or in the r-direction. A detailed discussion of coarse meshes is provided in Chapter 3. The philosophy of coarse mesh boundaries is the same for two-dimensions as that described for one-dimension. For each dimension (direction), the user determines the number of regions required to characterize the problem. Then the number of regional boundaries in each direction is the number of coarse meshes in that direction. For a one-dimensional problem, **im** is the number of coarse mesh intervals in either the x- or the r-direction depending on the geometry selected (slab = x, cylinder and sphere = r). For a two-dimensional problem, **im** is the number of coarse mesh intervals in either the x- or the r-direction while **jm** is the number of coarse mesh intervals in the y-, z-, or theta-direction depending on the geometry selected (x-y, r-z, or r-theta).

For this problem, we will have three regions so there will be three coarse meshes. This means that we will have `im=3`.

2.2.C.5 Fine Meshes

The fine meshes represent the subintervals in each region or coarse mesh. Within each region, the fine mesh spacing is constant. If you need different fine mesh spacings in a region, then you need to add a coarse mesh and divide the region into two parts. Chapter 3 provides a detailed discussion of the philosophy of determining fine mesh requirements in each region. Allocation of fine meshes to each region is done in Block 2; Block 1 only needs to know the total number of fine meshes in the problem. For a one-dimensional problem, **`it`** is the total number of fine mesh intervals in either the x- or the r-direction depending on the geometry selected (slab = x, cylinder and sphere = r). For a two-dimensional problem, **`it`** is the total number of fine mesh intervals in either the x- or the r-direction while **`jt`** is the total number of fine mesh intervals in the y-, z- or theta-direction depending on the geometry selected (x-y, r-z, or r-theta). For this problem, we will have a total of sixteen fine meshes. This means that `it=16`.

2.2.C.6 Materials and Zones

The remaining Block 1 information relates to the number of mixtures (material types) that we will define in Block 4 and the number of material zones that we will create to contain the mixtures. There is no practical limit to the number of mixtures we can define for a DANTSYS run (actual limit = 500). In most cases, mixtures are defined for the fuel, the reflector, a shell, an annulus, etc., rather than for an individual isotope. For example, the fuel may be uranium oxide, so a mixture could be defined in Block 4 which specifies the atom densities of U^{235}, U^{238}, and O present in the fuel. The combination of these isotopes is treated as a single material type. In this problem, we have three mixtures: a core containing plutonium and gallium, a shell containing nickel, and a graphite reflector. Thus, we enter `mt=3` in Block 1 so DANTSYS knows to look for the definitions of three mixtures (material types) in Block 4.

Coarse Meshes and Zones

For most one-dimensional problems, the number of material zones will equal the number of coarse meshes which equals the number of homogeneous regions. However, if you have divided a homogeneous region into two or more subregions (more than one coarse mesh in a region), then you will have more coarse meshes than zones. On the other hand if you wish to make a number of parametric runs substituting one material zone for another, then the number of zones will be greater than the number of coarse meshes. The end result of all of this is that there is no magic formula which tells you or DANTSYS how many material zones you have based on the number of coarse meshes. There is even less correlation between zones and coarse meshes in two-dimensional problems. Examples of material zones and their assignments to coarse meshes are given in Chapter 4.

Again, the purpose of Block 1 is to provide control information to DANTSYS so it knows how many materials and zones to look for when it starts reading the Block 4 input. Details on the Block 4 input can be found in Chapter 4. In this example problem, we have three regions so we will create three material zones with different materials; `nzone=3`.

2.2.C.7 Other Useful Block 1 Keywords

Because DANTSYS is designed to run on many different platforms, memory requirements may vary among the various machines. DANTSYS uses two keywords, **`maxlcm`** and **`maxscm`** to indicate memory requests. The **`maxlcm`** keyword indicates the amount of large core memory requested while **`maxscm`** indicates the amount of small core memory requested. On most machines, we have found that setting `maxlcm=800000` and `maxscm=100000` is usually sufficient for all but the larger two-dimensional runs. When DANTSYS is executed, it will indicate how much lcm and scm are required. If you have not allotted enough, the run will terminate with an error. You need to check the output to determine how much is needed for your run. We have found that a ratio of 8:1 of lcm to scm works well. Thus, if the run requires 500,000 lcm, then setting maxlcm=560000 and maxscm=70000 should provide more than enough memory to complete the analysis. (Note that the commas are used for illustrative purposes; commas must not be used to separate thousands or millions in DANTSYS input numbers. Commas are only used to separate entries and can be replaced with spaces if desired.) DANTSYS does have defaults of 140,000 and 40,000 for these keywords, but we have found that many runs

require more than the default allocations. Thus, we suggest always including `maxlcm=800000` and `maxscm=100000` which usually enables you to run without concern for memory requirements.

There is one other Block 1 keyword that we find useful: **`miniprt`**. This substantially reduces the input echo on the output and saves paper. For this problem and all subsequent problems in the primer, we will use `miniprt=yes`.

2.2.C.8 Complete Title and Block 1 Input for Example Problem

This concludes the Block 1 input section. The entire input file at this point should look like:

```
ƀƀƀƀƀ3
Problem 2a – Plutonium and Gallium sphere w/ Nickel Coating
             Reflected with 3.83 cm graphite
Radius = 5.465 cm , using Hansen-Roach cross sections, 16 groups
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=3  it=16  mt=3
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes       t
```

Note that the terminator, `t` , for Block 1 is included at the end of the last card. Remember after the first card, it is free format. The only requirement is that each keyword be immediately followed by an equals sign.

2.2.D Other Input for Example Problem

To complete the input for Problem 2a, we need to provide the information for Blocks 2 through 6. These will be very similar to those done for the *Quickstart* problem. The input for each block will be given with a brief description of what the values indicate. Details of the input, available keywords, and options will be given in Chapters 3 through 7 as each input block is discussed.

2.2.D.1 Block 2 Input

The Block 2 input provides the geometry information to DANTSYS. For Problem 2a, the Block 2 input looks like:

```
/ *** block 2 ***
  xmesh= 0.0    5.465    5.478    9.308
  xints=      11       1        4
  zones=       1       2        3            t
```

The first coarse mesh is located at 5.465 cm which is the radius of the core, the second is at 5.478 cm which is the core radius plus shell thickness, and the third is at 9.308 cm which includes the reflector thickness. We have 11 fine meshes in the first region, 1 in the second, and 4 in the third. There are three material zones: zone 1 (core), zone 2 (shell), and zone 3 (reflector).

2.2.D.2 Block 3 Input

Information concerning the cross-section library is given in Block 3. Assuming that the cross-section library is in the DANTSYS binary format, there is only one required keyword for Block 3.

```
/ *** block 3 ***
  lib= bxslib          t
```

2.2.D.3 Block 4 Input

The Block 4 input provides the material and mixing information to DANTSYS.

```
/ *** block 4 ***
  matls=  fuel    pu492e1   3.9404e-2    /pu239 atom density
                  ga        1.3647e-3;   /gallium atom density
          plate   ni        9.1305e-2;   /nickel atom density
          graph   c         8.1724e-2    /graphite atom density
  assign= core    fuel         1.0;      /material zone 1
          shell   plate        1.0;      /material zone 2
          refl    graph        1.0       /material zone 3
      t
```

(The material zone <u>number</u> referenced in Block 2 is simply the entry number in the `assign=` array, e.g., `core` is the first entry — zone 1, `shell` is the second entry — zone 2, etc.) Note that

after entering the gallium atom density for the first mixture, there is a semicolon, `;`, at the end of the entry. This tells the program that all of the isotopes for this mixture, fuel, have been entered. The semicolon is only necessary when two or more mixtures are to be defined using the **`matls`** keyword. Also because there was more than one zone, the information for the first zone, `core`, is terminated by a semicolon before the information for the second zone is entered; the information for the second zone is terminated by a semicolon before the information for the third zone is entered. This concludes the Block 4 input section.

2.2.D.4 Block 5 Input

The Block 5 input provides the solver information to DANTSYS.

```
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0        t
```

Because the 167 isotope H-R library does not contain values of χ, we enter the fission fractions appropriate to plutonium. The `10r0.0` is shorthand for repeat the value 0.0 ten times. This concludes the Block 5 input section.

2.2.D.5 Block 6 Input

Block 6 provides the edit information which allows you to get reaction rates, fluxes, or sources at individual points or in individual zones. We will use the mass edit keyword to indicate total mass and density by zone.

```
/ *** block 6 ***
  massed=1            t
```

This concludes the Block 6 input section.

2.2.E Complete Input for Bare Configuration

```
ƀƀƀƀƀ3
Problem 2a – Plutonium and Gallium sphere w/ Nickel Coating
             Reflected with 3.83 cm graphite
Radius = 5.465 cm, using Hansen-Roach cross sections, 16 groups
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=3  it=16  mt=3
  nzone=3  maxlcm=800000  maxscm=100000 miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   5.465   5.478   9.308
  xints=     11      1       4
  zones=      1      2       3           t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matls=  fuel    pu492e1   3.9404e-2    /pu239 atom density
                  ga        1.3647e-3;   /gallium atom density
          plate   ni        9.1305e-2;   /nickel atom density
          graph   c         8.1724e-2    /graphite atom density
  assign= core    fuel        1.0;       /material zone 1
          shell   plate       1.0;       /material zone 2
          refl    graph       1.0        /material zone 3
      t
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0       t
/ *** block 6 ***
  massed=1          t
```

2.2.F Running Problem 2a

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable, `dant.x` and the `prob2a` input file. For this case, we assume that you are using the Hansen-Roach 167 isotope library and that it is in the DANTSYS binary format. This means that *bxslib* must be in your directory. To execute DANTSYS, type

```
dant.x < prob2a > prob2a.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated for each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0119 as written on the screen echo.

2.2.G Output

First, let's look at the group edit and balance tables. Recall from the *Quickstart* chapter that these tables are the fourth section of mini-print output. The first column indicates the neutron energy group number; the second column lists the fraction of source (external source) neutrons within each energy group. (All listed values are based on a normalization to 1.0 because `norm=1.0` was entered in Block 5.) For this problem and for most criticality problems, there will be no external source so all entries will be zero. The next column lists the fraction of fission neutrons born into each group; this should match the chi values that you entered in Block 5. The next three columns are: in scatter, self scatter, and out scatter fractions by group. Of primary concern here is that the in scatter total (3.3399063e-1) match the out scatter total (3.3399063e-1) to within three decimal places. *Note that the values obtained for in scatter and out scatter may not match those given above, but should be close.*

The next set of columns starts with the absorption by energy group. In this particular problem, 34.73% of the neutrons were absorbed in the system. The table does not indicate which region the neutrons were absorbed in, only the fraction absorbed by energy group. If a problem requires knowledge of where the neutrons were absorbed, then you would need to run an edit with appropriate Block 6 input. Also found in the balance table is the net leakage (66.01%). The leakage plus absorption should add to 1.0 for normalized runs. In this case, they add to 1.0074 which is due to only converging k to three decimal places. If more accuracy is required, then the convergence criteria can be tightened as described in Chapter 5 on solver controls. Finally, the

last column shows the fraction by energy group of neutrons causing fission. This provides an indication of the neutron spectrum for the system. In our case, over 96% of the fissions occurred in the first five energy groups (greater than 100 keV). This is expected because we have modeled a fast system.

The integral summary information table gives the system k_{eff} and the totals of each of the columns from the group balance table. For this run, k_{eff} = 1.0118921, but remember it is only converged to the third decimal place (i.e., 0.001).

The last section of output is the mass edit table. For this run it shows a total mass in zone 1 (the core) of 10.8021 kg with an average density of 15.80 g/cc and a volume of 683.69 cc. For zone 2 (the nickel shell), there is a total mass of 0.0435192 kg with an average density of 8.9 g/cc. Note that this is the same density given in the problem statement which indicates that the atom density entered matched the problem statement. For the reflector, we have 4.3837 kg of graphite with an average density of 1.63 g/cc. The last line of the edit gives the total mass, total volume, and average density. These system values are probably less helpful than the individual zone values, but can be useful as another reality check.

2.3 CONFIGURATION 2 — BARE SYSTEM

Now that we have analyzed the reflected system, we are going to return to the bare system to indicate how few changes need to be made to run with fewer zones. We will use the Problem 2a input file, but remove the graphite reflector information. This requires changing some input in Blocks 1 and 2. Because the changes to Block 1 depend on what values are entered in Block 2, we will do the Block 2 changes first.

2.3.A Block 2 Changes

By removing the reflector from the system, we will have fewer coarse mesh intervals, fewer fine meshes, and fewer zones. The new Block 2 input should look like:

```
/ *** block 2 ***
  xmesh= 0.0    5.465    5.478
  xints=      11       1
  zones=       1       2            t
```

We now have two coarse meshes. The first is still located at 5.465 cm, the radius of the core, and the second is still at 5.478 cm, the core radius plus shell thickness. We have 11 fine meshes in the first region, and 1 in the second. There are two zones: zone 1 is the core, and zone 2 is the shell.

2.3.B Block 1 Changes

With the changes in Block 2, we need to change the number of coarse mesh intervals and fine meshes in Block 1. Changes are indicated in **`bold Courier`** type. The new title cards and Block 1 input should look like:

```
ƀƀƀƀƀ3
Problem 2b – Bare Plutonium and Gallium sphere w/ Nickel Coating
Radius = 5.465 cm , using Hansen-Roach cross sections, 16 groups
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=2  it=12  mt=3
  nzone=3  maxlcm=800000  maxscm=100000 miniprt=yes       t
```

Note that although we are not using the mixture named `graph` or the zone labeled `refl`, we can still leave their definitions in Block 4 without any effect on problem validity. This is the reason that the values for `mt` and `zone` do not change. These tell DANTSYS how many mixtures and how many material zones are defined in Block 4; we have not changed Block 4 so these remain unchanged in Block 1.

2.3.C Output for the Bare Configuration

First, let's look at the group edit and balance tables. The in scatter and out scatter fractions match. The last column shows the fraction by energy group of neutrons causing fission. For this problem, over 97% of the fissions occurred in the first five energy groups (greater than 100 keV). This is slightly higher than for the reflected system where the graphite slightly softens the spectrum.

The integral summary information table gives the system k_{eff} and the totals of each of the columns from the group balance table. For this run, k_{eff} = 0.89537278, but remember it is only converged to the third decimal place (i.e., 0.001).

The last section of output is the mass edit table. For this run, it shows the same values in zones 1 and 2 as for the reflected system. Note that although we defined a material zone 3 in Block 4, there is no mass edit on that zone as it was not used in the problem.

2.4 SUMMARY

This chapter discussed the Block 1 input and its relationship to entries in Blocks 2, 3, and 4. You should be able to identify the keywords associated with one- and two-dimensional geometries as well as describe the output from the mini-print option. The neutron balance tables were described, and we indicated how you can use the balance tables and the mass edit table to check the quality of your input. The following chapters provide detailed discussions of the input requirements for Blocks 2 through 6.

Chapter 3
Geometry Description (Block 2)

After describing the input control information in Block 1, a detailed description of the problem geometry is done with Block 2 input. This chapter begins with the one-dimensional example problem from Chapter 2 and then changes the geometry to a two-dimensional cylinder.

3.1 WHAT YOU WILL BE ABLE TO DO:

- Describe the differences between coarse meshes and fine meshes and their associated input.
- Discuss the differences in input requirements between one-dimensional and two-dimensional problems.
- Explain the relationship between the zone map in Block 2 and the material zones defined in Block 4.
- Describe the effects of quadrature and meshing on k_{eff}.

3.2 PROBLEM DESCRIPTION

This first example uses the graphite reflected plutonium metal sphere from Chapter 2 (LA–10860–MS, p.101). The plutonium sphere has a radius of 5.465 cm. The graphite in the reflector has a density of 1.63 g/cc and is 3.83 cm thick.

3.2.A Reflected Sphere Geometry

The reflected plutonium sphere is identical to that used in Chapter 2. As noted in LA–10860–MS, the plutonium is delta-phase containing 1 weight percent gallium and coated with 0.013 cm of nickel. The table gives the Pu density as 15.8 g/cc which is assumed to be the density of the core (plutonium –gallium mixture).

The data for this example follows.

Core Material (ρ_{mix} = 15.8 g/cc-mixture)

Delta phase Pu metal (100 percent Pu-239, 99 wt %)

N_{239} = 3.9404e-2 atoms/b-cm

Gallium (1 wt %)

N_{Ga} = 1.3647e-3 atoms/b-cm

Shell Material

Nickel

ρ	= 8.9 g/cc-mixture (single material)
N_{Ni}	= 9.1305e-2 atoms/b-cm

Reflector Material

Graphite

ρ	= 1.63 g/cc-mixture (single material)
N_C	= 8.1724e-2 atoms/b-cm

The atom densities are calculated using the formulas found in Appendix B.

Configuration I. *Reflected Plutonium Sphere*

Plutonium and Gallium sphere:	Radius	= 5.465 cm
Nickel Coating:	Thickness	= 0.013 cm
Graphite Reflector:	Thickness	= 3.83 cm

3.2.B Title and Block 1 Input

This problem is identical to Problem 2a in the last chapter. Changes in the title lines and in some of the Block 1 parameters relating to the Block 2 geometry input are indicated in **`Courier bold`** type. Type your input file as shown; the changes will be discussed as we get to each topic in Block 2. The first lines of your input file should look like:

```
ƀƀƀƀƀ3
Problem 3a – Plutonium and Gallium sphere w/ Nickel Coating
             Reflected with 3.83 cm graphite, Radius = 5.465 cm,
using Hansen-Roach cross sections, 16 groups, sn=16, mesh=1cm
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=3  it=11  mt=4
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes        t
```

Remember the terminator, `t`, for Block 1 is included at the end of the last card, and the **ƀ** is used to indicate a blank space on an input line.

3.2.C Block 2 — Geometry Description

As mentioned in the *Quickstart* Chapter, Block 2 input describes the basic problem geometry including location and assignment of coarse and fine meshes, and the assignment of zones to coarse mesh regions. As with Block 1, all of the input in Block 2 is in free format with a

keyword followed by its appropriate values. We have already provided information in Block 1 above telling DANTSYS how many values to expect for each keyword in Block 2.

First we will provide a comment line which identifies this data as belonging to Block 2.

```
/ *** block 2 ***
```

These are the required keywords for Block 2.

xmesh Coordinates of the edges of the coarse meshes

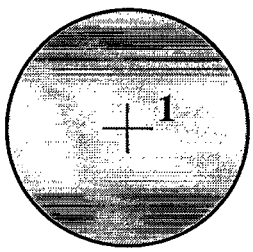
xints Number of fine meshes in each coarse mesh

zones Zone number for each coarse mesh (any entry must not be greater than the value given to nzone in Block 1)

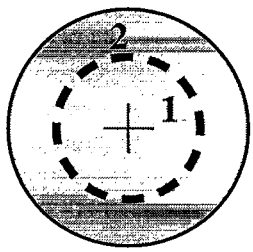
There are no optional keywords available in Block 2. In the *Quickstart* chapter, we entered values for each of the required keywords without much discussion of why or what options were available. In this chapter, we will give details on each of the Block 2 keywords typically used in criticality analyses.

3.2.C.1 Coarse Meshes

All of the geometry descriptors in Block 2 are specified relative to the coarse meshes. Each coarse mesh defines a homogeneous volume with constant size fine meshes within that coarse mesh to divide it into discrete intervals. For a one-dimensional problem, the user must decide on how many coarse meshes are required in the x- or r-direction. There must be at least one coarse mesh per region where *a region is defined as a thickness which is homogeneous in material properties.* A region may be a void, a reflector, the core, a thin shell of plating, or a fissile solution. A fuel pin surrounded by cladding is a single region only if you mix the fuel and clad materials and then characterize the mixture with a single set of macroscopic cross sections. Although each region has to have at least one coarse mesh, there is no limit on the number of subregions that can be placed in a region. For example, if you are particularly interested in what is happening in a 1 cm shell on the outside layer of the Jezebel core, then you can define two coarse meshes: the center region and the 1 cm outside shell (see Figure 3-1). The boundaries of each region are the locations of the coarse meshes, so the number of coarse meshes in a one-dimensional problem is equal to the number of regional outside boundaries.



One Coarse Mesh



Two Coarse Meshes

Figure 3-1

The philosophy of coarse mesh boundaries is the same for two-dimensions as that described for one-dimension. For each dimension (direction), the user determines the number of regions required to characterize the problem. Then the number of regional boundaries in each direction is the number of coarse meshes in that direction. For a one-dimensional problem, **im** is the number of coarse mesh intervals in either the x- or the r-direction depending on the geometry selected (slab = x, cylinder and sphere = r). For a two-dimensional problem, **im** is the number of coarse mesh intervals in either the x- or the r-direction while **jm** is the number of coarse mesh intervals in the y-, z-, or theta-direction depending on the geometry selected (x-y, r-z, or r-theta).

For this one-dimensional problem, we will have three regions. Therefore, we will define three coarse meshes; so `im=3` as already indicated in Block 1. The positions of the coarse mesh boundaries are defined in Block 2 using the **xmesh** keyword. The first coarse mesh begins at the center of the sphere at location `0.0` and ends at the outside of the plutonium sphere at a radius of `5.465` cm. The second coarse mesh begins at the end of the first coarse mesh (`5.465` cm) and ends at the outside of the nickel shell at a radius of `5.478` cm. The third coarse mesh begins at the end of the second coarse mesh (`5.478` cm) and ends at the outside of the graphite reflector at a radius of `9.308` cm. With the exception of the first coarse mesh, the beginning of each coarse mesh coincides with the end of the previous coarse mesh; thus, there is no reason to specify both boundaries (see Figure 3-2).

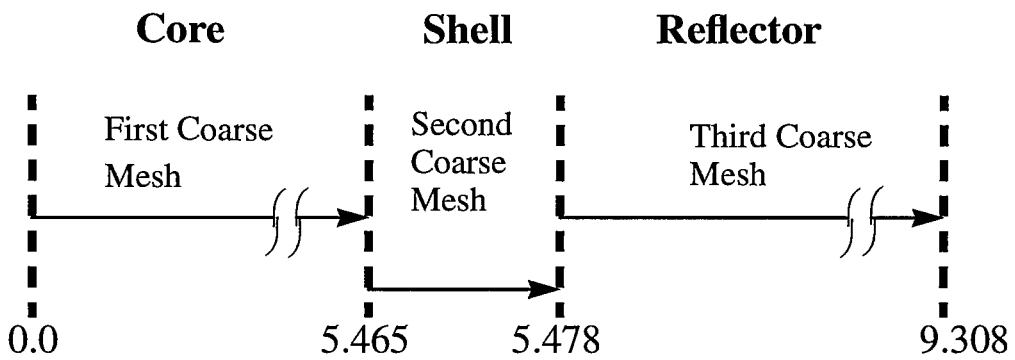


Figure 3-2 — Coarse Mesh for Reflected Pu Sphere Example Problem.

For the first coarse mesh, we must indicate its starting location; then for each coarse mesh (including the first), we indicate its end position. Thus, the keyword, **xmesh**, will have *1* (for the beginning of the first mesh) + *im* (for each coarse mesh) entries. This problem's four entries are:

```
xmesh= 0.0    5.465    5.478    9.308
```

Units Used in DANTSYS

As indicated in the discussion of coarse mesh boundaries, DANTSYS requires that all physical dimensions of a system (other than angles) be given in units that correlate with the cross sections. In most cases, these are **centimeters;** cross sections are given in barns (cm^{-2}) and atom densities are given in atoms per barn-cm. Thus the common unit of dimensions is the centimeter. If you have a geometry whose units are not centimeters and your cross sections are in barns or cm^{-2}, the positional values for meshes must be converted to centimeters before they are entered in Block 2.

3.2.C.2 Fine Meshes

The fine meshes represent the subintervals in each region or within each coarse mesh. Within each region, the fine mesh spacing is constant. If you need different fine mesh spacings in a region, then you need to add a coarse mesh and divide the region into two parts. To accurately model the effects on k_{eff} of neutron interactions, the width of a fine mesh interval in each region should be no greater than the mean free path in that region.

Mean Free Path and Fine Mesh Size

In a criticality analysis, we are only concerned with those neutrons that are scattered or absorbed. The probability of neutron interaction in a given region is directly proportional to the macroscopic cross section for that region. As the mean free path is inversely proportional to the macroscopic cross section, then an increase in the probability of interaction (and hence in the absorption or scattering macroscopic cross section) leads to a decrease in the mean free path. If the k_{eff} of a system is to be adequately characterized, then the DANTSYS model must have fine mesh sizes small enough to ensure that all interactions are appropriately accounted for. This usually translates to saying that the fine mesh size must not be greater than the mean free path in a region.

Mean Free Path and Neutron Cross Sections

For thermal systems, there is a significant amount of moderator with a microscopic scattering cross section on the order of 2 to 10 barns. The atom density of the moderator may be as large as 0.1 atom per barn-cm so the macroscopic scattering cross section in these systems may be as large as 1.0 cm^{-1}. This means the mean free path ($\lambda_s = 1 / \Sigma_s$) will usually not be less than 1.0 cm. Based on this bounding case, a fine mesh width not greater than 1.0 cm should be adequate for most regions with moderating materials. Now, the microscopic absorption cross section of fissile material in thermal systems may be on the order of 500 barns. However, the fissile atom density in these systems is usually on the order of 2×10^{-4} atoms per barn-cm. Thus, the macroscopic absorption cross section will be about 0.1 cm^{-1} with a mean free path of 10 cm. So for thermal systems, the scattering mean free path dominates and is useful for determining maximum the fine mesh size.

For systems with minimal moderation (typically fast metal systems), the microscopic scattering cross section is about 10 barns. These systems have maximum fissile atom densities of 0.05 atoms per barn-cm. Then the macroscopic scattering cross section in such a region is not greater than 0.5 cm^{-1} with a minimum associated mean free path of 2.0 cm. Although the fissile atom density in such systems is relatively high, the microscopic absorption cross section is only a few barns. This indicates a macroscopic absorption cross section of 0.1 cm^{-1} and a mean free path of 10 cm. Thus, even for predominantly fast systems, the scattering mean free path should be used as a guideline in determining the fine mesh size.

The use of mean free paths to determine fine mesh size is only a suggestion, not a requirement within DANTSYS. For most one-dimensional problems, significantly decreasing the fine mesh size (by increasing the number of equally spaced fine mesh intervals) will have little effect on memory requirements and problem run time. However, it will increase the accuracy of the estimated k_{eff} value. Later in this chapter we will see how changes in fine mesh size affect the estimate of k_{eff}.

Fine Mesh Size in Two-dimensional Problems

For two-dimensional problems, adding enough fine meshes to get their size down to the appropriate mean free path often requires substantial increases in memory and run time. In many cases this may not be practical. A possible solution to this problem is as follows. Remembering that the importance of a neutron is highly dependent on its access to fissile material, those regions with fissile material or those adjacent to fissile material should be modeled with as many fine meshes as possible. Regions further away from the fissile material can then be modeled with fine mesh widths of 2, 3, or even 5 cm. If this still does not reduce the problem to one of manageable size, take the adjacent regions and subdivide them with a few coarse meshes. Then take the major outlying portion of the region and give it fine meshes of a few cm in width. The new, smaller coarse mesh region adjacent to the fissile material can then be given a fine mesh structure appropriate to its mean free path.

For our problem, we want to have fine meshes whose widths don't exceed 1.0 cm. Remembering that each coarse mesh is divided into equally spaced regions according to the number of fine meshes, we assign the fine meshes as follows. For the first coarse mesh region whose width is 5.465 cm, we will need 6 fine meshes. If we only have 5 fine meshes, then the width of each fine mesh would be 1.093 cm (= 5.465 / 5) which is greater than our <u>self-imposed limit</u> of 1.0 cm. With 6 fine meshes in the first coarse mesh region, the width of each fine mesh subinterval will be 0.9108333 cm (= 5.465 / 6). For the second coarse mesh region whose width is 0.013 cm (5.478 5.465), we will have 1 fine mesh of width 0.013 cm.

Note that although 0.013 cm is a very small distance, DANTSYS requires that *every coarse mesh region have at least one fine mesh. This is true even if the coarse mesh region is a void.* Void regions must have at least one fine mesh but can have more if desired.

The width of the third coarse mesh region is 3.830 cm (9.308 - 5.478); with 4 fine meshes their width would be 0.9575 cm (= 3.830 / 4). In a one-dimensional problem, the keyword used to specify the number of fine meshes in each coarse mesh region is **xints**. If there are im coarse mesh regions, then there must be im entries for **xints**. In our particular problem, the 3 entries for **xints** would appear as:

```
xints=        6        1        4
```

Note that we have a total of 11 (= 6+1+4) fine mesh intervals which we specified in Block 1 using the **it** keyword. Allocation of fine meshes to each region is done in Block 2; Block 1 only needs to know the total number of fine meshes in the problem.

3.2.C.3 Materials and Zones

The number of mixtures (material types) and of material zones for a problem are identified in Block 1. For DANTSYS the term *zone* is always used to refer to combinations of mixtures which create a homogeneous mix and hence are represented by a single set of cross sections in the calculations. Each combination of mixtures is assigned to a material zone in Block 4. The order in which the zones are described in Block 4 gives rise to the zone number associated with each material zone. Although the term zone may be used elsewhere to refer to a physical area, in DANTSYS it will only refer to a material zone. (For reference to a physical area, DANTSYS uses the coarse mesh region as the means to define the physical boundaries of an area.) Once the region has been defined, then the material contents of the region must be defined. The last part of the Block 2 input is to assign material zones (*zones*) to physical areas (*coarse mesh regions*). This is done using the **zones** keyword. Each coarse mesh region must have a material zone assigned to it; thus if there are im regions, there must be im entries for zones. If a coarse mesh region contains a void, then a material zone number of 0 is assigned to it.

For our problem, we will define three material zones in Block 4. Each material zone will be some combination of the mixtures we define in Block 4. For illustrative purposes we are going to define four different mixtures in Block 4:

1.) fuel containing Pu-239;
2.) Ga to be mixed in with the fuel to create the core;
3.) Ni for the shell; and
4.) C (graphite) for the reflector.

Note that in the previous chapter we combined the plutonium and gallium into one mixture; we have separated them here to show how they could be combined at the material zone level. The first material zone to be defined (hence zone number 1) is a single mixture (graphite). This will be called the refl zone, but will be referenced in DANTSYS by its number (1) rather than its name. The second material zone is a combination of plutonium and gallium; it will be called `core`. As the second zone defined, it has a zone number of 2. The third material zone is the nickel shell with the name of `shell`; zone number of 3. Again, note that the numbering of zones is a direct result of the order in which they are defined in Block 4.

For Block 2, we take the material zone numbers from Block 4 and associate them with a coarse mesh region. For the example problem, the first coarse mesh region is the core and the material zone associated with the core is zone number 2. The second coarse mesh region is the shell, and it is associated with material zone number 3. The last coarse mesh region is the reflector which is associated with zone number 1. The entries on the zones card should look like:

```
zones=      2      3      1            t
```

where the first entry is the zone number to fill the first coarse mesh, the second entry is the zone number for the second coarse mesh, and so on. The first coarse mesh is located at the lower left of the system. For a sphere or infinite cylinder, this is the coarse mesh starting at the center. For a slab, it is the leftmost coarse mesh. Because there are no more keywords in Block 2, we have terminated the Block 2 input by placing a t at the end of the `zones` line.

Note that the definition of material zones would make more sense if the first material zone corresponded to the first coarse mesh. This is the way it was done in the previous chapter, but for illustration purposes, the order of material zones was changed here. Any order is acceptable; use whichever order makes the most sense for the problem at hand.

3.2.D Other Input for Example Problem

To complete the input for problem 3a, we need to provide the information for Blocks 3 through 6. These will be similar to the Chapter 2 example with the exception of the mixtures and ordering of the material zones. The input for each block will be given with a brief description of what the values indicate. Details of the input, available keywords, and options will be discussed in Chapters 4 through 7 as the other input Blocks are discussed in detail.

3.2.D.1 Block 3 Input

Information concerning the cross-section library is given in Block 3. Assuming that the cross-section library is in binary format, there is only one required keyword for Block 3.

```
/ *** block 3 ***
  lib= bxslib          t
```

3.2.D.2 Block 4 Input

The Block 4 input provides the material and mixing information to DANTSYS.

```
/ *** block 4 ***
  matls=  fuel        pu492e1 3.9404e-2; /pu239 atom density
          gallium     ga      1.3647e-3; /gallium atom density
          plate       ni      9.1305e-2; /nickel atom density
          graph       c       8.1724e-2  /graphite atom density
  assign= refl      graph     1.0;                    /zone 1
          core      fuel      1.0  gallium   1.0;     /zone 2
          shell     plate     1.0                     /zone 3
  t
```

(The zone <u>number</u> referenced in Block 2 is simply the entry number in the `assign=` array, e.g., `refl` is the first entry — zone 1, `core` is the second entry — zone 2, etc.) After entering the plutonium atom density for the first mixture, there is a semicolon, `;`, at the end of the entry. This tells the program that all of the isotopes for this mixture, `fuel`, have been entered. Also because there was more than one zone, the information for the first zone, `refl`, is terminated by a semicolon before the information for the second zone is entered; the information for the second zone is terminated by a semicolon after volume fractions for both mixtures have been entered. This concludes the Block 4 input section.

3.2.D.3 Block 5 Input

The Block 5 input provides the solver information to DANTSYS.

```
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0        t
```

Because the 167 isotope H-R library does not contain values of χ , we enter the fission fractions appropriate to plutonium. The `10r0.0` is shorthand for repeat the value 0.0 ten times. This concludes the Block 5 input section.

3.2.D.4 Block 6 Input

Block 6 provides the edit information which allows you to get reaction rates, fluxes, or sources at individual points or in individual zones. We use the mass edit keyword to indicate total mass and density by zone.

```
/ *** block 6 ***
  massed=1          t
```

This concludes the Block 6 input section.

3.2.E Complete Input for Reflected Sphere

```
ƀƀƀƀƀ3
Problem 3a – Plutonium and Gallium sphere w/ Nickel Coating
              Reflected with 3.83 cm graphite, Radius = 5.465 cm,
using Hansen-Roach cross sections, 16 groups, sn=16, mesh=1cm
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=3  it=11  mt=4
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   5.465   5.478   9.308
  xints=      6      1       4
  zones=      2      3       1           t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matls=  fuel        pu492e13.9404e-2; /pu239 atom density
          gallium     ga      1.3647e-3; /gallium atom density
          plate       ni      9.1305e-2; /nickel atom density
          graph       c       8.1724e-2  /graphite atom density
```

```
  assign= refl      graph     1.0;                    /zone 1
          core      fuel      1.0  gallium   1.0;     /zone 2
          shell     plate     1.0                     /zone 3
  t
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0        t
/ *** block 6 ***
  massed=1          t
```

3.2.F Running Problem 3a

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable, `dant.x` and the `prob3a` input file. For this case, we are assuming that you are using the Hansen-Roach 167 isotope library[3] and that it is in binary format. This means that *bxslib* must be in your directory. To run, type

```
dant.x < prob3a > prob3a.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0120 as written on the screen echo.

3.2.G Output

First, look at the group edit and balance tables. Make sure that the in scatter total (3.3408360e-1) matches the out scatter total (3.3408360e-1) to within three decimal places. Note the values obtained for in scatter and out scatter may not match those given above, but should be close. The integral summary information table gives the system k_{eff} and the totals of each of the columns from the group balance table. For this run, k_{eff} = 1.0120454, but remember it is only converged to the third decimal place (i.e., 0.001).

This system as modeled is not exactly critical probably because information from LA–10860–MS may not be exactly as the experiment was run. In many cases, there were additional isotopes present which have been neglected or folded into the values given in the Tables. It is always best to go to the original reference for the data if you are trying to benchmark or validate a code for a specific area of applicability. In this case, Pu-240 and Pu-241 were present in the experiment but not specified in the table.

The last section of output is the mass edit table. This run has a total mass in zone 1 (the reflector) of 4.3837 kg of graphite with an average density of 1.63 g/cc. Note that this is the same density given in the problem statement which indicates that the atom density entered matched the problem statement. Remember, zones refer to the material zones created in Block 4. For this problem, we entered the reflector as the first zone which altered the order of the zones from that used in Problem 2a.

For zone 2 (the core which combined plutonium and gallium), there is a total mass of 10.8022 kg with an average density of 15.80 g/cc and a volume of 683.69 cc. Note that the mass of the plutonium alone is 10.6942 kg which is slightly lower than the 10.8 kg listed in Table 32, pg. 101 of LA–10860–MS. This provides another check for the user to assure that the proper information has been entered. For the nickel shell, we have 0.0435192 kg with an average density of 8.9 g/cc. The last line of the edit gives the total mass, total volume, and average density. These system values are probably less helpful than the individual zone values, but can be useful as another reality check.

3.3 EFFECTS OF MESHING AND QUADRATURE ON K-EFFECTIVE

As mentioned in the discussion of quadrature and fine mesh size, varying either parameter will affect the estimated value of k_{eff}. In many criticality problems, the exact value of k_{eff} to many decimal places is less important than an estimate within a few percent of the actual value. For most problems, fine mesh sizes of 1 cm or less and a quadrature of 16 should provide reasonable

estimates of k_{eff}. However, for those cases where a reasonable estimate is not accurate enough, then a parametric study of the effect of mesh size and quadrature should be done.

Sensitivity to Quadrature Order

No matter what quadrature set is used, problem solutions should be tested for dependence on the order of quadrature. This is not to say that every problem should be calculated several times to test the effects of quadrature sets and/or quadrature order. It often suffices to perform a series of calculations on a problem typical of the class of problems with that you often analyze. Large systems with large homogeneous regions are often insensitive to quadrature order. On the other hand, small systems and systems with local heterogeneities are likely to be very sensitive to quadrature order. For example, running the Jezebel model with a quadrature of 4 gives a keff of 1.0152 while running it with a quadrature of 48 gives a k_{eff} of 1.0023. This problem is an extreme case in regards to quadrature sensitivity. It is a small system (radius = 6.385 cm) with high leakage (65.9%) so the angular flux is peaked in the outward-flowing direction. Many problems do not display this sensitivity as they usually are larger systems with significantly less leakage. For these problems, low order quadratures will suffice. However, for the system or systems that are routinely evaluated, it is important that the criticality analyst perform a series of calculations to determine the sensitivity to quadrature order.

In most cases, you will find that the estimate of k_{eff} is more dependent on quadrature than on mesh size. The best way to determine the most accurate estimate is to run an S-infinity calculation.

S-infinity Calculations

An S-infinity study starts with a small quadrature and an small number of fine meshes within each coarse mesh (this means a relatively large fine mesh width). In most cases, you would start with Sn = 2 and a fine mesh width of 2 cm. After this run is made, then you double the quadrature and double the number of fine meshes (halving the fine mesh width). This process is repeated through Sn = 32. Then the estimate of k_{eff} for each run is plotted against the inverse of the quadrature. As an infinite quadrature has an inverse of 0.0, the y-intercept of the curve indicates the best estimate of k_{eff} (best meaning least affected by quadrature or fine mesh size).

When we did an S-infinity calculation for Problem 3a, we obtained the following results, Sn=4, k=1.0354; Sn=8, k=1.0223; Sn=16, k=1.0191; Sn=32, k=1.0181, and Sn=48, k=1.0180. The Sn=48 run was done in lieu of a run at Sn=64 since this quadrature set is not in the library. When doing the Sn=48 run, the number of fine meshes was increased by a factor of 1.5 rather than 2 as Sn=48 equals 1.5 times Sn=32. A plot of k_{eff} versus 1/Sn indicates a y-intercept at 1.0178 which would then be the 'best' estimate of k_{eff}. Note that the run with Sn=16 (k_{eff} = 1.0191) is within 0.14% of the 'best' value. This indicates that using quadratures of 16 will usually provide reasonable results for one-dimensional problems.

After doing a few S-infinity studies on various system types, you will have a good feel for the best quadrature and fine mesh sizes to use with these systems. When analyzing a new system (different materials or moderator concentrations, etc.), it is always a good idea to perform an S-infinity study to determine the impact of quadrature and mesh size on the estimate of k_{eff}.

3.4 CONFIGURATION 3b — REFLECTED CYLINDER

Now that we have analyzed the reflected sphere, we are going to change the geometry to two dimensions. This example uses the graphite reflected plutonium metal cylinder (Table 32, LA–10860–MS, p.101).[2] The plutonium cylinder has a diameter of 9.87 cm with h/d = 0.86. The graphite in the reflector has a density of 1.63 g/cc and is 5 cm thick. As in the previous example, the plutonium has 1 wt% gallium and is coated with a nickel shell.

3.4.A Example Problem 3b Data

Core Material (ρ_{mix} = 15.8 g/cc-mixture)

Delta phase Pu metal (100 percent Pu-239, 99 wt %)

N_{239} = 3.9404e-2 atoms/b-cm

Gallium (1 wt %)

N_{Ga} = 1.3647e-3 atoms/b-cm

Shell Material

Nickel

ρ = 8.9 g/cc-mixture (single material)

N_{Ni} = 9.1305e-2 atoms/b-cm

Reflector Material

Graphite

ρ = 1.63 g/cc-mixture (single material)

N_C = 8.1724e-2 atoms/b-cm

Note the atom densities are the same as those used for Example 3a; the only change will be in the geometry from a sphere to a cylinder.

Configuration II. *Reflected Plutonium Cylinder*

Plutonium and Gallium cylinder:	Diameter	= 9.87 cm
	Radius	= 4.935 cm
	h/d	= 0.86
	Height	= 8.4882 cm

For cylinders and spheres, the diameter is often provided. DANTSYS requires that the user enter the radius for such geometries so make sure that the correct information is provided in your input file.

Nickel Coating:	Thickness	= 0.013 cm
Graphite Reflector:	Thickness	= 5 cm

We will use the Problem 3a input file with the same mixtures and material zones, but change the geometry. This requires only changes to Blocks 1 and 2. Since the changes to Block 1 depend on what values are entered in Block 2, we will do the Block 2 changes first.

3.4.B Block 2 Changes

3.4.B.1 Coarse Meshes

By changing to a cylindrical geometry, we now have to define coarse meshes in two dimensions. For r-z cylindrical geometry, the values of the **xmesh** keyword refer to coarse meshes in the *r* direction while the values of the **ymesh** keyword refer to coarse meshes in the *z* direction. The coarse meshing in the radial direction follows the same 'philosophy' as we used for the sphere. The first coarse mesh will be the plutonium/gallium core region, the second will be the nickel shell, and the third will be the graphite reflector. The **xmesh** entries should look like:

```
xmesh= 0.0    4.935    4.948    9.948
```

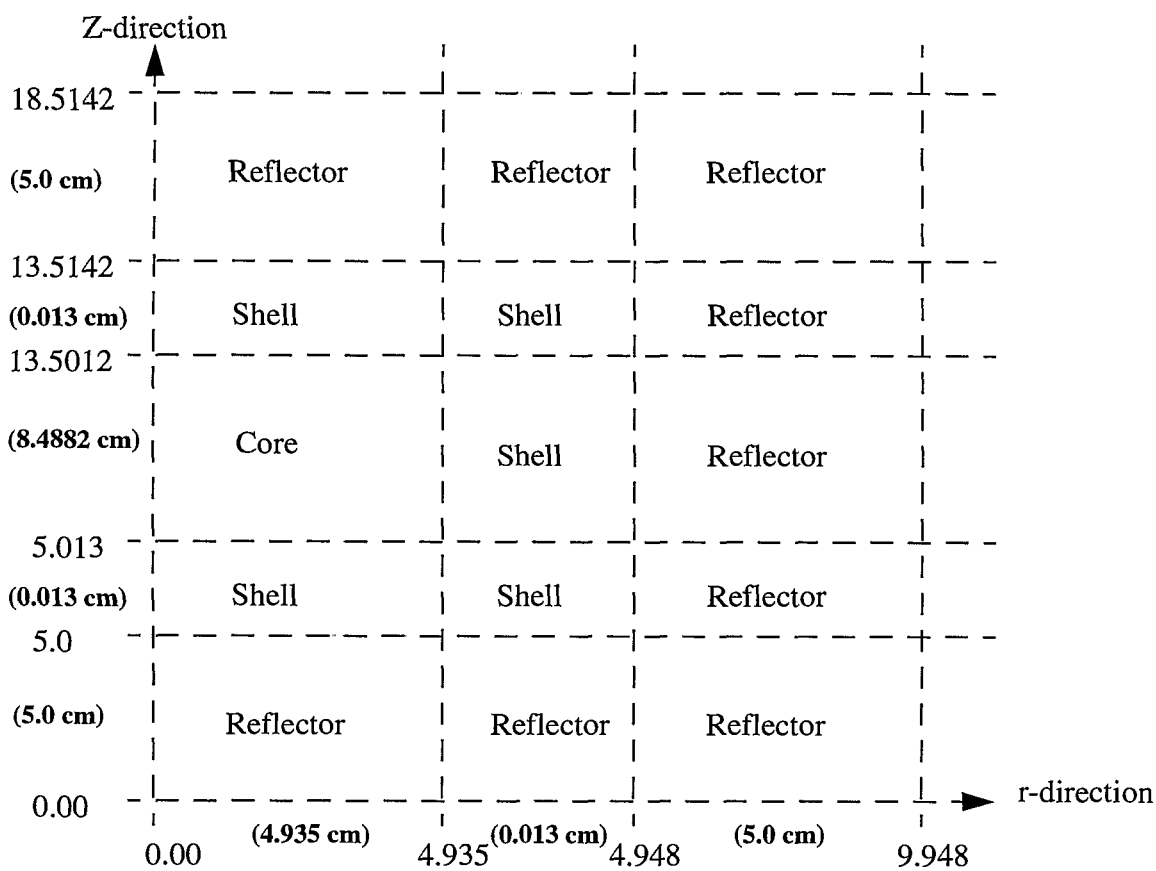


Figure 3-3 — Coarse Mesh Arrangement for the Graphite Reflected Cylinder.

As shown in Figure 3-3, we now have three coarse meshes in the r direction with the first ending at 4.935 cm which is the radius of the core, the second at 4.948 cm which is the core radius plus shell thickness, and the third at 9.948 cm which includes the 5 cm reflector thickness.

Now we need to set up the coarse meshes in the z direction (see Figure 3-3). Just as the radial meshes are defined from the center out, the axial meshes are defined from the lower left, bottom corner up. In this problem, the bottom of the system is the bottom, axial graphite reflector. This will be the first axial coarse mesh, the bottom plane of the reflector will be given the z = 0.0 coordinate. The first axial coarse mesh extends from 0.0 to 5.0 cm which is the thickness of the graphite reflector. The second axial coarse mesh will be the shell around the core with a thickness of 0.013 cm. This mesh will extend from 5.0 cm to 5.013 cm. The third axial coarse mesh is the core region with a height of 8.4882 cm. This mesh will extend from 5.013 cm to 13.5012 cm. The fourth axial coarse mesh is the top nickel shell which will extend from 13.5012 cm to 13.5142 cm. The fifth and last coarse mesh in the z direction is the top graphite reflector which extends from 13.5142 cm to 18.5142 cm. Based on this, the **ymesh** entries should look like:

```
ymesh= 0.0    5.0    5.013    13.5012    13.5142    18.5142
```

3.4.B.2 Fine Meshes

Having defined the radial and axial coarse meshes (geometry regions), we need to determine how many fine meshes will be in each coarse mesh region. This is done in the same manner as was done for one-dimension. For the radial meshing, we will have a fine mesh width of no greater than 1.0 cm so that means 5 fine meshes in the first coarse mesh, 1 in the second and 5 in the third. Thus, the entries for **xints** look like:

```
xints=    5    1    5
```

In the same manner, we determine the number of fine meshes in the axial direction based on a maximum thickness of 1.0 cm. We will have 5 fine meshes in the first axial coarse mesh, 1 in the second, 9 in the third, 1 in the fourth, and 5 in the fifth. Thus the entries for **yints** look like:

```
yints=    5    1    9    1    5
```

Notice that although the process of determining fine mesh size seems straightforward, there is now an overlap between the radial fine meshes and the axial fine meshes. Thus, if you want to have very small fine meshes in the core, this requires small fine meshes in the reflector above, below and to the outside of the core. In the example problem, if you want to have a fine mesh width of 0.5 cm in the core, then you would double the number of radial and axial meshes in

the core. But because each radial fine mesh extends from the bottom of the problem to the top of the problem, then any other regions below or above the core will now have a fine mesh width of 0.5 cm.

Fine Mesh Grids in Two-Dimensions

In two dimensions, the fine mesh is a grid laid over the problem so each grid line is continuous and essentially extends from -infinity to + infinity. Because of this, it is important to remember that having very small fine mesh widths in a particular region (a large number of fine meshes) will require that many other regions have the same fine mesh width. This creates a problem with a large number of fine mesh points. As the transport equation is solved at each fine mesh point, having small fine mesh sizes, particularly in two-dimensional problems, may dramatically increase memory requirements and problem runtime. In our one-dimensional example, there were 11 radial mesh points so there were only 11 points which required solution to the transport equation. When we went to the two-dimensional problem, there were 11 radial mesh points and 21 axial mesh points. This means that the transport equation must be solved at 231 (11 x 21) mesh points. Although only 5 radial points and 9 axial points are in the core (45 total core points), doubling the number of core fine meshes, 10 radial and 18 axial, means that the transport equation must be solved at 480 (16 x 30) points rather than just an additional 135 points in the core. Thus a desire to have a finer mesh in one region may need to be traded off against realistic problem size.

Also remember that increasing problem size often requires an increase in **maxlcm** and **maxscm** values for Block 1. The maxima of these keywords is machine dependent so a large problem that ran on one machine may be too large to run on another machine. The best way of determining fine mesh requirements is to do an S-infinity calculation as described earlier. In a two-dimensional problem, the quadrature, x-, and y-meshes are doubled for each new run. Plotting the values versus 1/Sn will give an indication of the system k_{eff}. Based on this analysis, you can choose reasonable fine mesh sizes to adequately evaluate the problem at hand.

3.4.B.3 Material Zone Assignment

For two dimensions, a material zone is assigned to a coarse mesh using the same **zones** keyword that we used for one dimension; there is no separate keyword for x-, y-, r-, or z-directions. When the zones entries are completed, it should look like a map of the system. For

two-dimensional problems, there are two ways of looking at the geometry: specifying the entire system, or using symmetry planes to specify a portion of the system. For our cylinder example, we have specified the entire problem; however, we could have used to symmetry plane through the core middle to only specify the top or bottom half of the system. (See Chapter 5 for a detailed description of symmetry planes and boundary conditions.) In either case, the entries for the **zones** keyword start with the lower left hand corner of the system. For our full system cylinder, this means the centerline of the cylinder at the bottom of the reflector. Entries are made for each z-coarse mesh and then followed by a semicolon.

In our example problem, we have `im=3` radial coarse meshes and `jm=5` axial coarse meshes. That means we enter the bottom 3 material zones; all three radial coarse meshes at the first axial coarse mesh. In our case, the first axial layer consists of reflector for the first radial coarse mesh, reflector for the second radial coarse mesh, and reflector for the third radial coarse mesh. The reflector is still material zone 1 so we would enter `1 1 1;`. The second axial layer consists of nickel shell, nickel shell, and reflector so we would enter `3 3 1;`. The third axial layer consists of the core, nickel shell, and reflector so we would enter `2 3 1;`. The fourth axial layer is identical to the second while the fifth axial layer is identical to the first.

Thus, our entries for **zones** would look like:

```
zones=      1      1      1;
            3      3      1;
            2      3      1;
            3      3      1;
            1      1      1            t
```

Notice that the last line has no semicolon because that is last of the axial coarse meshes to be associated with a zone. However, putting a semicolon at the end of the last zones entry is okay and will not affect the validity of the input. Again, since **zones** is the last keyword for Block 2, we have used the `t` to terminate the Block 2 input. The entries for **zones** look like a map of the system, but it is a mirror image map with the bottom of the system at the top. One quick way to check if you have incorrectly associated materials with coarse meshes is to look at the zones input and see if the 'map' looks like the model.

DANTSYS is still free format in two-dimensions; there is no requirement that each set of entries for an axial mesh be on a separate line. However, it is much easier to check your input and to visualize the entries if you do make all the radial entries for each axial mesh on a separate line.

Also note that with the symmetry of the system, we could have specified just the first three lines and added a reflective boundary condition. This approach will be discussed in more detail in Chapter 5 when boundary condition inputs are described. The entire Block 2 input looks like:

```
/ *** block 2 ***
  xmesh= 0.0   4.935   4.948   9.948
  xints=     5       1       5
  ymesh= 0.0   5.0   5.013   13.5012   13.5142   18.5142
  yints=     5     1       9         1         5
  zones=      1      1     1;
              3      3     1;
              2      3     1;
              3      3     1;
              1      1     1           t
```

Note that we moved the **xints** entries under the **xmesh** and the **yints** entries under the **ymesh** for clarity to quickly identify the number of fine meshes in each coarse mesh region. This completes the Block 2 input so we can go back and enter the correct values for the Block 1 keywords.

3.4.B.4 Block 1 Changes

With the changes in Block 2, we need to go from one dimension to two and change the number of coarse mesh intervals and fine meshes in Block 1. The changes are indicated in **Courier bold** type. The new title cards and Block 1 input should look like:

```
ƀƀƀƀƀ3
Problem 3b – Plutonium and Gallium cylinder w/ Nickel Coating
            Reflected with 5 cm graphite, Radius = 4.935 cm, using
Hansen-Roach cross sections, 16 groups, sn=6, mesh= 1 cm
/ *** block 1 ***
  igeom= r-z  ngroup=16  niso=167  isn=6  im=3  it=11
       jm=5   jt=21  mt=4  nzone=3
       maxlcm=800000  maxscm=100000  miniprt=yes        t
```

The first Block 1 change is in the geometry; we have changed from a sphere (sph) to a finite cylinder (r-z). *Note that cyl denotes a one-dimensional infinite cylinder; r-z or r-theta must be used for two-dimensional cylinders.* For cylinders (either infinite or finite), setting **isn=6** usually produces the best results. Because we have the same numbers of coarse and fine meshes in the x-direction (radial direction), **im** and **it** remain unchanged. With two dimensions, r- and z-, we have added the **jm** and **jt** keywords to specify the number of coarse and fine meshes in the second (axial) dimension. These Block 1 and Block 2 changes are all that are required for Problem 3b. Taking the Block 3, 4, 5, and 6 input from Problem 3a, we can now run the two dimensional cylinder.

3.4.C Running Problem 3b

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable, `dant.x` and the `prob3b` input file. To run, type

```
dant.x < prob3b > prob3b.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0074 as written on the screen echo.

3.4.D Output for the Cylinder Configuration

First, let's look at the group edit and balance tables. The in scatter and out scatter fractions match within four decimal places. Because this is a two-dimensional run, there are additional columns which show right leakage (leakage in the radial direction), horizontal leakage, and left leakage. (For cylinders the horizontal leakage will equal the right leakage. For slabs the horizontal leakage is the sum of the right and left leakage.) There are also columns for top leakage, vertical leakage, and bottom leakage. For this problem, the total top leakage equals the total bottom leakage due to symmetry. The vertical leakage is the total axial leakage. In the example problem, 44.1% of the neutrons leak radially, 21.2% leak axially, and 34.7% are absorbed. This is slightly different from that with the same materials in the spherical configuration (problem 3a) where 35.3% of the neutrons were absorbed while 64.7% leaked. The last column (nprod spectrum) shows the fraction by energy group of neutrons causing fission. For this problem, about 96% of the fissions occurred in the first five energy groups (greater than 100

keV). This is slightly lower than for the spherical system where there was less graphite to soften the spectrum.

The integral summary information table gives the system k_{eff} and the totals of each of the columns from the group balance table. For this run, k_{eff} = 1.0074240, but remember it is only converged to the third decimal place (i.e., 0.001).

The last section of output is the mass edit table. For this run it shows a total mass in zone 1 (the reflector) of 8.31491 kg of graphite with an average density of 1.63 g/cc. Note that this is the same density given in the problem statement which indicates that the atom density entered matched the problem statement. Remember zones refer to the material zones created in Block 4. For zone 2 (the core which combined plutonium and gallium), there is a total mass of 10.2611 kg with an average density of 15.80 g/cc and a volume of 649.441 cc. Note that the mass of the plutonium alone is 10.1585 kg which is within the bounds of the 10.1 kg 1% listed in Table 32, pg. 101 of LA–10860–MS. This provides another check for the user to assure that the proper information has been entered. For the nickel shell, we have 0.0482818 kg with an average density of 8.9 g/cc.

3.5 SUMMARY

This chapter presented you with two examples designed to teach the basic geometry input and the concepts of coarse meshes, fine meshes, and material zones. The two examples demonstrated the input differences between one-dimensional and two-dimensional models. You also learned how to assign material mixtures (zones) to coarse meshes with the zone map. Finally, the effects on k_{eff} of fine mesh size and quadrature were demonstrated. This completes the discussion on Block 2 input which describes the problem geometry.

Chapter 4
Material Description (Block 4)

Blocks 1 and 2 provide the geometry description for the problem. This chapter describes the input in Block 4 which provides the description and location of the various materials within the system. This chapter begins with the one-dimensional example problem from Chapter 3, examines Block 4 input requirements for two-dimensional problems, and then changes material specifications from atom densities to weight and atom fractions.

4.1 WHAT YOU WILL BE ABLE TO DO:

- Combine different isotopes into a mixture.
- Assign materials (mixtures) to material zones.
- Describe the differences in weight fraction, atom fraction, and atom density input.
- Use the mass edit output for evaluation of materials input.

4.2 PROBLEM DESCRIPTION

This example uses the graphite reflected plutonium metal sphere from Chapter 3 (LA–10860–MS, p.101). The plutonium sphere has a radius of 5.465 cm. The graphite in the reflector has a density of 1.63 g/cc and is 3.83 cm thick.

4.2.A Reflected Sphere Geometry

The reflected plutonium sphere specifications are identical to those used in Chapter 3. As noted in LA–10860–MS, the plutonium is delta-phase containing 1 weight percent gallium and coated with 0.013 cm of nickel. The table gives the Pu density as 15.8 g/cc which is assumed to be the density of the core (plutonium–gallium mixture). The data follows with atom densities calculated based on given densities and atomic weights from the 14th Chart of the Nuclides.[5]

Core Material (ρ_{mix} = *15.8 g/cc-mixture)*

Delta phase Pu metal (100 percent Pu-239, 99 wt %)

N_{239} = 3.9404e-2 atoms/b-cm

Gallium (1 wt %)

N_{Ga} = 1.3647e-3 atoms/b-cm

Shell Material

Nickel

ρ = 8.9 g/cc-mixture (single material)

N_{Ni} = 9.1305e-2 atoms/b-cm

Reflector Material

Graphite

ρ = 1.63 g/cc-mixture (single material)

N_C = 8.1724e-2 atoms/b-cm

The atom densities are calculated using the formulas found in Appendix B.

Configuration I. *Reflected Plutonium Sphere*

Plutonium and Gallium sphere:	Radius	= 5.465 cm
Nickel Coating:	Thickness	= 0.013 cm
Graphite Reflector:	Thickness	= 3.83 cm

4.2.B Title, Block 1, Block 2, and Block 3 Input

Because the problem is identical to that from the last chapter, we will repeat the title, block 1, block 2, and block 3 information in our input file. The changes in the title lines are indicated in **Courier bold** type. Type your input file as shown; the only change is in the problem number in the title. The first lines of your input file should look like:

```
bbbbb3
Problem 4a – Plutonium and Gallium sphere w/ Nickel Coating
              Reflected with 3.83 cm graphite, Radius = 5.465 cm,
using Hansen-Roach cross sections, 16 groups, sn=16, mesh=1 cm
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=3  it=11  mt=4
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes        t
```

```
/ *** block 2 ***
  xmesh= 0.0    5.465    5.478    9.308
  xints=       6       1       4
  zones=       2       3       1            t
/ *** block 3 ***
  lib= bxslib          t
```

Remember the terminator, t, for each Block is included at the end of the last card in the block, and the **b** is used to indicate a blank space on an input line.

4.2.C Block 4 — Materials and Mixtures

4.2.C.1 Material Types and Zones

The number of material types and of material zones for a problem are identified in Block 1. For DANTSYS, the term *zone* is always used to refer to a combination of isotopes which creates a homogeneous mixture. Each mixture (a combination of isotopes) is assigned to a material zone in Block 4. The order in which the zones are described in Block 4 gives rise to the zone number associated with each mixture. Although the term zone may be used elsewhere to refer to a physical area, in DANTSYS it will only refer to a material zone or mixture. (For reference to a physical area, DANTSYS uses the coarse mesh region as the means to define the physical boundaries of an area.) The assignment of material zones to coarse mesh regions is done in Block 2 and is described in Chapter 3 of this Primer.

There are two primary keywords which are required in Block 4. (Actually there are some instances where these keywords are not required, but for the majority of criticality calculations, this is not an issue.) The two are:

matls - Provides the recipes or instructions for creating mixtures from individual isotopes.

assign - Assigns the mixtures created with the MATLS keyword to material zones.

The number of mixtures or material types to be defined in Block 4 is given in Block 1 by the **mt** keyword. Similarly, the number of material zones to be defined in Block 4 is given by the Block 1 **nzone** keyword.

4.2.C.2 Material Types

The keyword **`matls`** is used to define the mixtures in the problem. It is followed by entries which describe each mixture or material type. The entries for each material type are: **the name of the mixture or material**, followed by **the isotope label** for each isotope in the mixture, and **the atom density** of that isotope. (Atom fractions or weight fractions can be used in place of atom density as described later in this chapter.) Each set of entries for a mixture must be separated from the next by a semicolon. In general it is suggested that you start each new mixture specification on a new line.

According to our Block 1 input, we need to define four different material types (or mixtures) in Block 4 (`mt=4`). For illustration purposes, we will make each isotope a separate mixture (even though each only consists of one isotope). Then the four mixtures to be defined in Block 4 are:

1.) Pu-239 for the fuel;

2.) Ga to be mixed in with the fuel to create the core;

3.) Ni for the shell; and

4.) C (graphite) for the reflector.

We will enter each of these isotopes using the isotope name found on the selected cross-section library (in this case, the 167 isotope Hansen-Roach library).[3] A listing of the isotopes found in the 167 H-R library is given in Appendix D.

Isotope Labels on cross-section Libraries

Because the name of a material must match with an entry in the cross-section library you are using, it is important to know how the isotopes are labeled in that particular library. In many cases, there will be a list available for the library, and the labels will be easily identified. However, sometimes such a listing is not available so how do you determine the isotope labels? All you need to do is set up your run with `miniprt=no` in Block 1, the appropriate inputs in Blocks 1, 2, and 3, and a `matls= unk` in Block 4. This run will terminate abnormally with the error message of unknown isotope, but the output file will provide a listing of the isotope labels in the Block 3 echo. Once you have the isotope labels, you can complete your input for Blocks 4, 5, and 6 and then run your problem.

Again we start the Block 4 input with a comment card indicating the beginning of Block 4 data.

```
/ *** block 4 ***
```

For the mixture or material type name, you can use anything except an isotope label. For our first mixture, we will call the plutonium mixture *fuel.*

The first isotope is plutonium-239 and the appropriate label is *pu492e1*. (Looking at the label listings for the H-R 167 library, you will notice that there are a number of entries for Pu-239; each entry begins with *pu49*. The various entries are for different resonance treatments of the Pu-239 cross sections. A complete discussion of the resonance treatments and how to select the appropriate treatment for your problem is found in Chapter 6 and in Appendix E. For the present, we will use the resonance treatment labeled *pu492e1*.)

The atom density from the problem specification is *3.9404e-2*. Thus, the entry for pu-239 is:

```
matls=  fuel     pu492e1 3.9404e-2;/pu239 atom density
```

We have called the mixture `fuel`, used the isotope label `pu492e1`, and entered the appropriate atom density, `3.9404e-2`. These values are followed by a semicolon indicating additional entries to be made for this keyword. Also, note the use of the comment string to the right to further describe the entry. The remaining material types for this problem will be gallium, plate, and graph. The Block 4 entries for these mixtures are as follows:

```
gallium ga 1.3647e-3;/gallium atom density
plate    ni 9.1305e-2;/nickel atom density
graph     c 8.1724e-2/graphite atom density
```

Because the graphite is the last material type to be defined, there is no semicolon required after its atom density. This completes the definition of material types for this problem.

4.2.C.3 Material Zones

Now that we have four mixtures, we need to combine them as required and assign these combinations to appropriate material zones. A material zone is characterized by a single set of cross sections, that is, a homogeneous mixture with a macroscopic cross section for absorption, one for fission, one for self-scattering, etc. Note that in chapter 2 we combined the plutonium and gallium into one material; we have separated them here to show how they could be combined at the material zone level.

The keyword **assign** is used to match each mixture with a material zone. One material zone may have many mixtures or as few as one mixture assigned to it. The **assign** keyword is followed by entries which describe each material zone assignment. The entries for each material zone assignment are: **the name of the material zone**, followed by **the name of each mixture** in the zone, and **the volume fraction for that mixture** in the zone. Each zone's specification must be separated from the next by a semicolon.

For our problem, we will define three material zones in Block 4. Each material zone will be some combination of the four material types we defined above. The first material zone to be defined (hence zone number 1) is a single material (graphite). This will be called the *refl* zone, but will be referenced in Block 2 by its number (1) rather than its name. The second material zone is a combination of plutonium and gallium; it will be called *core* and be zone number 2. The third material zone consists of only the nickel shell with the name of *shell* and zone number of three. Again, note that the number of a zone is a direct result of the order in which it is defined. DANTSYS does not care what order the zones are defined in; for this example we chose an arbitrary ordering. The definition of material zones would make more sense if the first material zone corresponded to the first coarse mesh. This is the way it was done in Chapter 2, but for illustration purposes, the order of material zones was changed here. Any order is acceptable, so use whichever order makes the most sense. In most cases, you will find it easier to define the zones from the inside out, but the choice is up to you.

The material zone input for Block 4 should look something like:

```
assign= refl      graph     1.0;
        core      fuel      1.0  gallium   1.0;
        shell     plate     1.0
t
```

Note that because there was more than one zone, the information for the first zone, `refl`, is terminated by a semicolon before the information for the second zone is entered; the information for the second zone is terminated by a semicolon after volume fractions for both materials have been entered. This concludes the Block 4 input section.

4.2.D Other Input for Example Problem

To complete the input for problem 4a, we need to provide the information for Blocks 5 and 6. These will be identical to those done for the Chapter 3. The input for each block will be given with a brief description of what the values indicate. Details of the input, available keywords, and options is given in Chapters 2 through 7 where the other input blocks are discussed in detail.

4.2.D.1 Block 5 Input

The Block 5 input provides the solver information to DANTSYS.

```
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0        t
```

Because the 167 isotope H-R library does not contain values of χ , we enter the fission fractions appropriate to plutonium. The `10r0.0` is shorthand for repeat the value 0.0 ten times. This concludes the Block 5 input section.

4.2.D.2 Block 6 Input

Block 6 provides the edit information which allows you to get reaction rates, fluxes, or sources at individual points or in individual zones. We will use the mass edit to indicate total mass and density by zone.

```
/ *** block 6 ***
  massed=1            t
```

This concludes the Block 6 input section.

4.2.E Complete Input for Reflected Sphere

```
bbbbb3
Problem 4a – Plutonium and Gallium sphere w/ Nickel Coating
              Reflected with 3.83 cm graphite, Radius = 5.465 cm,
using Hansen-Roach cross sections, 16 groups, sn=16, mesh=1 cm
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=3  it=11  mt=4
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   5.465   5.478   9.308
  xints=      6      1       4
  zones=      2      3       1           t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matls=  fuel     pu492e1 3.9404e-2;/pu239 atom density
          gallium ga 1.3647e-3;/gallium atom density
          plate    ni 9.1305e-2;/nickel atom density
          graph    c 8.1724e-2/graphite atom density
  assign= refl      graph     1.0;
          core      fuel      1.0  gallium   1.0;
          shell     plate     1.0
  t
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0       t
/ *** block 6 ***
  massed=1          t
```

4.2.F Running Problem 4a

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable and the `prob4a` input file. As before, with the Hansen-Roach 167 isotope library in binary format, the *bxslib* must be in your directory. To run, type

```
twodant < prob4a > prob4a.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0120 as written on the screen echo.

4.2.G Output

First, check the group edit and balance tables. Make sure that the in scatter total (3.3408360e-1) matches the out scatter total (3.3408360e-1) to within three decimal places. For this run, k_{eff} = 1.0120454, but remember it is only converged to the third decimal place (i.e., 0.001).

The last section of output is the mass edit table. For this run it shows a total mass in zone 1 (the reflector) of 4.3837 kg of graphite with an average density of 1.63 g/cc. This matches the problem statement. Remember zone numbers refer to the order that the material zones were created in Block 4, so zone 1 is *refl.* For zone 2 (the core which combined plutonium and gallium), there is a total mass of 10.8022 kg with an average density of 15.80 g/cc and a volume of 683.69 cc. Note that the mass of the plutonium alone is 10.6942 kg which is slightly lower than the 10.8 kg listed in Table 32, pg. 101 of LA–10860–MS. This provides another check for the user to assure that the proper information has been entered. For the nickel shell, we have 0.0435192 kg with an average density of 8.9 g/cc. The last line of the edit gives the total mass, total volume, and average density of all the material in the problem. These values are probably less helpful than the individual zones values, but can be useful as another reality check.

4.3 BLOCK 4 INPUT FOR TWO-DIMENSIONAL PROBLEMS

For Blocks 1 and 2 there were a number of changes required when the problem geometry was changed from one-dimension to two-dimensions. However, with Block 4, there are no changes required when the geometry changes. As Block 4 only describes the materials in the system, the same Block 4 input can be used for both one- and two-dimensional problems. In fact, if you go back to Chapter 3 and examine the Block 4 inputs for Problems 3a and 3b, you will note that the inputs are identical. This is somewhat analogous to the independence of material and geometric bucklings; changes in geometry of a system do not affect the material buckling. The only time changes are necessary in Block 4 is when materials are added or removed or when mixture compositions change.

4.4 EXAMPLE PROBLEM 4b USING WEIGHT AND ATOM FRACTIONS

The specifications for example Problem 4b are identical to those for Problem 4a. The difference is that we will use *weight fractions* to specify the material in the core and *atom fractions* to specify the nickel shell and graphite reflectors.

4.4.A Specifying Weight or Atom Fractions

To use either weight fractions or atom fractions in Block 4, a new keyword, **`matspec`**, is required. **`matspec`** tells the code whether the mixing in the **`matls`** array is in terms of atom densities (`atdens`), atom fractions (`atfrac`), or weight fractions (`wtfrac`). If you make the entry `matspec=atdens` or omit the `matspec=` entry, the mixing is done with atom densities as described earlier in this chapter. For each material type or mixture defined in Block 4, you need to tell DANTSYS how you are specifying the amount of material. In this example problem, we will have three mixtures (fuel, plate, and graph) so we need three entries after `matspec=`. In Problem 4a, all of the mixtures were defined using atom densities so there was no need for the **`matspec`** keyword as the default is `matspec=atdens`.

In this example, Problem 4b, we will define the first mixture using weight fractions and the last two using atom fractions so we need to add the following entry to Block 4:

```
MATSPEC= WTFRAC ATFRAC ATFRAC
```

This indicates that the first mixture, *fuel*, will be defined using weight fractions while the last two mixtures, *plate* and *graph*, will be defined using atom fractions.

4.4.B Weight Fractions

Data for the core materials as given in LA–10860–MS is:

Core Material (ρ_{mix} = *15.8 g/cc-mixture)*
Delta phase Pu metal (100 percent Pu-239, 99 wt %)
Gallium (1 wt %)

Previously, we used atom densities to describe the amounts of Plutonium-239 and Gallium in the core. DANTSYS also allows the use of weight fractions; which for this problem would be easier as the data is given as weight fractions.

Identifying Number of Mixtures with Weight or Atom Fractions

When using either weight or atom fractions, the total fraction for each *mixture* must add to 1.0. If it doesn't, the weight fractions or atom fractions for the mixture will be normalized to 1.0. In this case, we must specify only three mixtures since the weight fractions for pu-239 and gallium add to 1.0. If we specified the same four mixtures as before, then we would end up with a core with 1.0 weight fraction of gallium and 1.0 weight fraction of pu-239 with an equivalent density of 2*15.8 or 31.6 g/cc. So when using weight or atom fractions, determine the number of mixtures based on the number of material types. Then the sum of fractions in each mixture should add to 1.0.

Using weight fractions instead of atom densities requires two changes in the Block 4 input. The first is entering the weight fraction of the material where the atom density was entered. When weight fractions are used, the entries for each material type after the `MATLS=` keyword become: **the name of the mixture or material**, followed by **the isotope label** for each isotope in the mixture, and **the weight fraction** of that isotope. So for the fuel (99 wt% pu-239) and the gallium (1 wt% ga), the new Block 4 entries are:

```
matls=  fuel pu492e1 0.99;  /pu239 core weight fraction
             ga 0.01;      /gallium core weight fraction
```

The second change is to the assign keyword entries where instead of entering a volume fraction, the mixture density is entered. When weight fractions are used, the entries for each

material zone assignment after the `ASSIGN=` keyword become: ..... **the name of the material zone**, followed by **the name of each mixture** in the zone, and **the mass density for that mixture** in the zone. So for the fuel mixture with a density of 15.8 g/cc, the `assign` entry becomes:

```
core        fuel        15.8;
```

4.4.C Atom Fractions

Data for the shell and reflector materials from LA–10860 is:

Shell Material, ρ = 8.9 g/cc-mixture (single material)
Nickel

Reflector Material, ρ = 1.63 g/cc-mixture (single material)
Graphite

Note that the reference, LA–10860–MS, only provides mass densities for these materials. Previously, we calculated atom densities to describe these but as DANTSYS allows us to use atom fractions, specifying these materials by atom fractions is much easier.

Using atom fractions instead of atom densities also requires two changes in the Block 4 input. The first is entering the atom fraction of the material where the atom density was entered. When atom fractions are used, the entries for each material type after the `MATLS=` keyword become: **the name of the mixture or material**, followed by **the isotope label** for each isotope in the mixture, and **the atom fraction** of that isotope. So for the shell (atom fraction 1) and for the reflector (atom fraction 1), the new Block 4 entries are:

```
plate     ni    1.00;/nickel atom fraction in plate
graph     c     1.00;/carbon atom fraction in refl
```

The second change is to the **`assign`** keyword entries where instead of entering a volume fraction, the mixture density is entered. (Note that for both weight and atom fractions, the mixture mass density is entered in the assign keyword entries.) When atom fractions are used, the entries for each material zone assignment after the `ASSIGN=` keyword become: **the name of the material zone**, followed by **the name of each mixture** in the zone, and **the mass density for that mixture** in the zone. So for the plate mixture with a density of 8.9 g/cc and the refl mixture with a density of 1.63 g/cc, the assign entries become:

```
assign=     refl        graph       1.63;
            shell       plate       8.9
```

4.4.D Block 4 Input

The complete input for Block 4 using the weight and atom fractions looks like:

```
/ *** block 4 ***
  matspec= wtfrac atfrac atfrac
  matls=fuel pu492e1 0.99;/pu239 core weight fraction
           ga         0.01;/gallium core weight fraction
         plate ni     1.00;/nickel atom fraction in plate
         graph c      1.00;/carbon atom fraction in refl
  assign=refl graph 1.63;
         core fuel 15.8;
         shell plate 8.9
  t
```

4.4.E Block 1 Changes

With the changes in block 4, we need to change the number of material types in block 1. The changes are indicated in **Courier bold** type. The new title cards and block 1 input should look like:

```
bbbbb3
Problem 4b – Plutonium and Gallium sphere w/ Nickel Coating
wt & atom fraction, Refl with 3.83 cm graphite, Radius = 5.465 cm,
using Hansen-Roach cross sections, 16 groups, sn=+16, mesh= 1 cm
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=3  it=11  mt=3
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes       t
```

The only Block 1 change is in the number of material types (from 4 back to 3 as we are no longer specifying each component of the core as an individual mixture. These Block 1 and Block 4 changes are all that are required for Problem 4b. Taking the Block 2, 3, 5, and 6 input from

Problem 4a we can now run the reflected sphere using weight and atom fractions to specify the materials.

4.4.F Running Problem 4b

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable and the `prob4b` input file. To run, type

```
twodant < prob4b > prob4b.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0121 as written on the screen echo.

4.4.G Output for the Weight and Atom Fraction Specification Approach

First, the k-effective for this problem should be identical to that for problem 4a (k=1.0120). In this case, the difference is round-off beyond the third decimal place. Then look at the mass edit table. For this run it shows a total mass in zone 1 (the reflector) of 4.38373 kg of graphite with an atom density of 8.17245e-2 atoms/barn-cm. Note that this is the same atom density (to four decimal places) as used in problem 4a. Remember the zone numbers refer to the order that material zones were created in Block 4. For zone 2 (the core containing plutonium and gallium), there is a total mass of 10.8023 kg. The atom density for gallium is 1.36472e-03 and for pu-239 is 3.94043e-02 (both values match those used in problem 4a). For the nickel shell, we have 0.0435267 kg with an atom density of 9.13208e-2. This is slightly different than the 9.1305e-2 used in problem 4a. The difference is attributable to using an atomic weight of 58.70 (obtained from the 14th edition of the Chart of the Nuclides) in the atom density calculation instead of 58.69 which is the value listed on the cross-section file. If you expect your results to be identical, then you need to be consistent in the values used. In this case, to maintain consistency requires the use of the atomic weights given in the cross-section data file.

The last line of the edit gives the total mass, total volume, and average density. These values match with those from Problem 4a which indicates that using either atom densities or weight/atom fractions provides the same material descriptions.

4.5 SUMMARY

This chapter presented you with two examples designed to teach the input requirements to describe the materials in a problem. You learned how to mix isotopes to create a material type and then how to assign this material type to a material zone. The two examples demonstrated the use of atom fractions, weight fractions, and atom densities to describe amounts of various isotopes in mixtures. And finally, you have had additional experience using the output from massed to verify input parameters for the materials in your problem. This completes the discussion on Block 4 input which describes the materials involved in a problem.

Chapter 5
Solver Input (Block 5)

The previous chapters described the input requirements for geometry and materials. This chapter discusses the input requirements necessary to solve the neutron transport equation in finite difference form. Beginning again with the one-dimensional example problem from Chapter 3, this chapter examines Block 5 input requirements for boundary conditions, convergence criteria, and limits on iterations. It then continues to a two-dimensional problem to see what new input is required when the dimensionality of the problem changes.

5.1 WHAT YOU WILL BE ABLE TO DO:

- Use the `kcalc` keyword to reduce runtime for criticality calculations.
- Describe the differences among the various boundary conditions.
- Use the `norm` and `chi` keywords to specify a fission source level and a fission source spectrum.

5.2 PROBLEM DESCRIPTION

This example uses the graphite reflected plutonium metal sphere from Chapter 3 (LA–10860–MS, p.101). The plutonium sphere has a radius of 5.465 cm. The graphite in the reflector has a density of 1.63 g/cc and is 3.83 cm thick.

5.2.A Reflected Sphere Geometry

As noted in Chapter 3 and LA–10860–MS, the plutonium is delta-phase containing 1 weight percent gallium and coated with 0.013 cm of nickel. The table gives the Pu density as 15.8 g/cc which is assumed to be the density of the core (plutonium–gallium mixture). The data follows with atom densities calculated based on given densities and atomic weights from the 14th Chart of the Nuclides.[5]

Core Material (ρ_{mix} = *15.8 g/cc-mixture)*

Delta phase Pu metal (100 percent Pu-239, 99 wt %)

N_{239} = 3.9404e-2 atoms/b-cm

Gallium (1 wt %)

N_{Ga} = 1.3647e-3 atoms/b-cm

Shell Material

Nickel

ρ = 8.9 g/cc-mixture (single material)

N_{Ni} = 9.1305e-2 atoms/b-cm

Reflector Material

Graphite

ρ = 1.63 g/cc-mixture (single material)

N_C = 8.1724e-2 atoms/b-cm

The atom densities are calculated using the formulas found in Appendix B.

Configuration I. *Reflected Plutonium Sphere*

Plutonium and Gallium sphere:	Radius	= 5.465 cm
Nickel Coating:	Thickness	= 0.013 cm
Graphite Reflector:	Thickness	= 3.83 cm

5.2.B Title, Block 1, Block 2, Block 3, and Block 4 Input

Because the problem is identical to that from the last chapter, our input file will repeat the title, block 1, block 2, block 3, and block 4 information from example problem 4a. The changes in the title lines are indicated in **`Courier bold`** type. Type your input file as shown; the only change is in the problem number in the title. The first lines of your input file should look like:

```
bbbbb3
Problem 5a – Plutonium and Gallium sphere w/ Nickel Coating
        Reflected with 3.83 cm graphite, Radius = 5.465 cm, using
Hansen-Roach cross sections, 16 groups, sn=16, mesh=1 cm
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=3  it=11  mt=4
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   5.465   5.478   9.308
```

```
  xints=      6      1      4
  zones=      2      3      1          t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matls=  fuel     pu492e1  3.9404e-2;   /pu239 atom density
          gallium  ga       1.3647e-3;   /gallium atom density
          plate    ni       9.1305e-2;   /nickel atom density
          graph    c        8.1724e-2    /graphite atom density
  assign= refl      graph     1.0;
          core      fuel      1.0  gallium   1.0;
          shell     plate     1.0
  t
```

Remember the terminator, t, for each Block is included at the end of the last card in the block, and the **b** is used to indicate a blank space on an input line.

5.2.C Block 5 — Solver Input

The Block 5 input describes the type of calculation required, information on boundary conditions, printing options, normalization of fission source rate and group fission fractions, and cross-section scattering order. For a one-dimensional problem, there are five keywords required in Block 5. These are:

- **IEVT** - Type of calculation desired (source, k_{eff}, or search)
- **ISCT** - Legendre order of scattering desired in the problem evaluation
- **ITH** - Direct or adjoint calculation
- **IBL** - Left boundary condition
- **IBR** - Right boundary condition

Default values are provided for all of these, but the **IEVT** default is not suitable for criticality (k_{eff}) calculations so there must be at least one entry in Block 5 using the **IEVT** keyword.

5.2.C.1 IEVT Keyword

As mentioned above, the **ievt** keyword describes the type of calculation to be made with DANTSYS. For most criticality calculations, a k_{eff} calculation is desired so the entry in Block 5

would be `ievt=1` indicating an eigenvalue calculation. Other possible values for **`ievt`** indicate a type of search desired; these will be discussed in detail in Chapter 7 of this primer.

5.2.C.2 ISCT Keyword

The **`isct`** keyword is used to define the order of Legendre scattering to be used in the analysis of the problem. A value of `0` indicates that only isotropic scattering will be considered in the problem. The maximum value for **`isct`** is determined by whichever cross-section library is being used for the analysis. If hydrogen is involved in the problem, particularly with water or plastics, then the scattering will not be isotropic so **`isct`** needs to be set to a value of `1` or greater if these data are found in the cross-section library. With the 167 isotope Hansen-Roach library, the maximum Legendre order available is 1 so for most criticality analyses using this library, `isct=1` is the appropriate entry.

5.2.C.3 ITH Keyword

Although adjoint calculations are particularly useful in shielding or dose calculations, there are very few criticality calculations where they apply. As such a direct calculation is required so `ith=0` is the appropriate entry. The default for the **`ith`** keyword is `0` so if the keyword is not present in Block 5 then DANTSYS will do a direct calculation. For criticality calculations, it is suggested that defaults be used where the value of a keyword does not change; so we will not provide an entry for **`ith`** in Block 5.

5.2.C.4 IBL and IBR Keywords

The last two required Block 5 keywords describe the boundary conditions for one-dimensional problems. These are the values for the left, **`ibl`**, and right, **`ibr`**, boundaries. However, the left boundary condition is only applicable to slab geometries; for spheres or cylinders, the centerline boundary condition of symmetry is implicit in the equations so any entry for **`ibl`** in these geometries will be ignored. The possible values for the boundary conditions are: `0` indicating a vacuum, `1` indicating reflective, `2` indicating periodic, and `3` indicating white. The default for any boundary condition is vacuum, but because boundary conditions change with problem geometry and symmetry considerations, it is strongly suggested that no defaults be used.

5.2.C.4.1 Vacuum Boundary Condition

A vacuum boundary condition means that no neutrons will renter the boundary. Thus, any neutron exiting the system through a vacuum boundary is permanently lost to the system. This condition is shown in Figure 5.1. *In the following figures, a dashed arrow indicates neutrons leaving the system while a solid arrow represents those returning to the system. The length of the arrow is proportional to the number of neutrons so longer arrows represent more neutrons than shorter arrows. Note that angles are shown as discrete based on the quadrature of the system.* The vacuum boundary is selected by setting the keyword, either `ibl= 0` or `ibr= 0`.

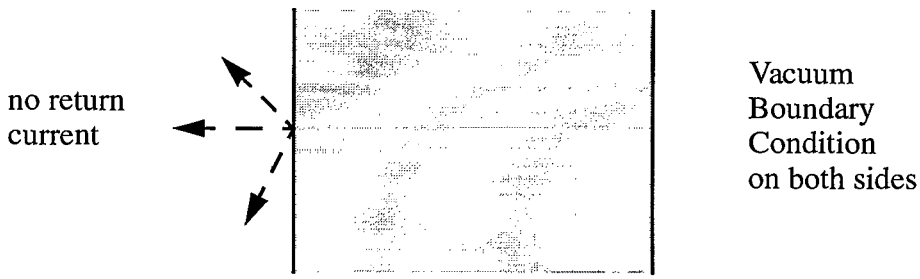


Figure 5.1 — Vacuum Boundary Conditions.

5.2.C.4.2 Reflective Boundary Conditions

For the reflective boundary condition, the incoming angular flux is set equal to the outgoing angular flux in the direction corresponding to specular reflection. The reflective boundary is selected by setting the keyword, either `ibl= 1` or `ibr= 1`. As shown in Figure 5.2, a given quantity of neutrons leaving a boundary in a particular direction will be returned to the system with the same quantity but a specular angle to the initial leakage direction.

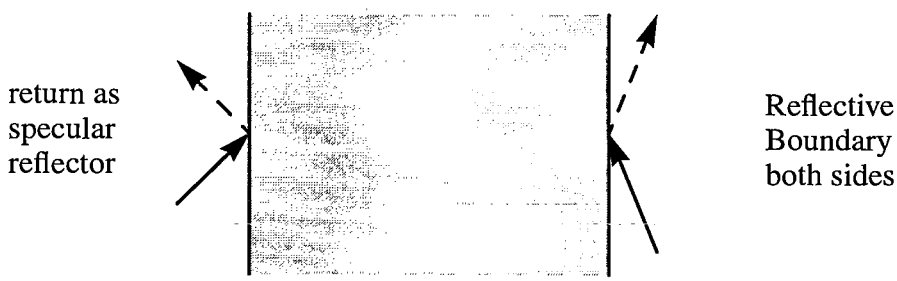


Figure 5.2 — Reflective Boundary Conditions.

5.2.C.4.3 Periodic Boundary Condition

For the periodic boundary condition, the incoming angular flux on a boundary is set equal to the outgoing angular flux in the same direction on the opposite boundary. Figure 5.3 shows the leakage being returned at the same quantity and angle but entering the system on the opposite boundary. The periodic boundary is selected by setting the keyword, either `ibl= 2` or `ibr= 2`.

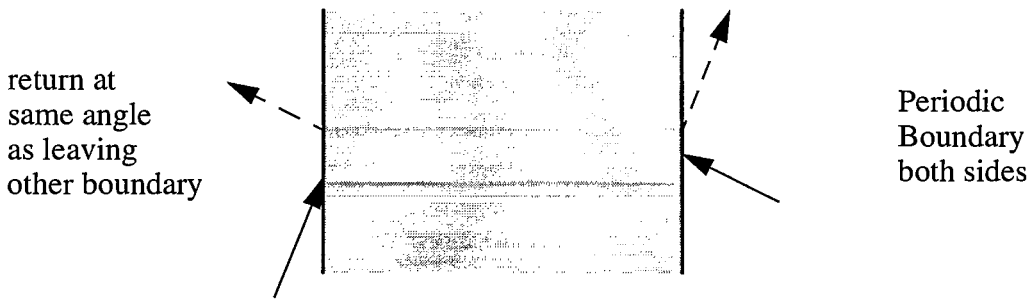


Figure 5.3 — Periodic Boundary Condition.

5.2.C.4.4 White Boundary Condition

For the white boundary condition, the incoming angular fluxes on the value are each set equal to a single value chosen such that the net flow across the boundary is zero. The number of angular fluxes depends on the quadrature selected for the problem. The white boundary is selected by setting the keyword, either ibl or ibr = 3. Figure 5.4 shows that any neutrons leaving the system are returned to the same boundary as three (or whatever quadrature is selected) equal angular fluxes.

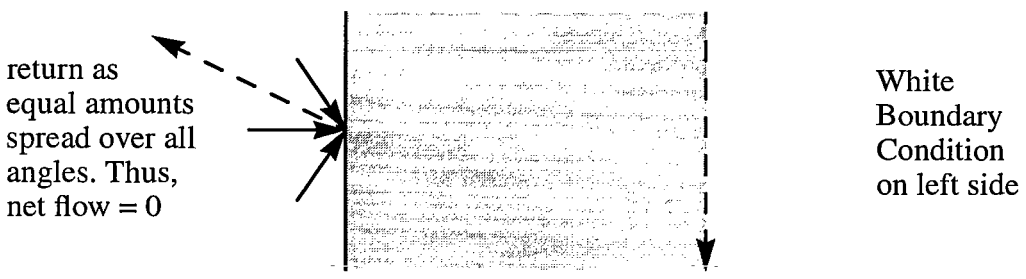


Figure 5.4 — White Boundary Condition.

The non-vacuum boundary conditions have some constraints on their use. First, the reflective and periodic boundary conditions should only be used for planar surfaces. This means for slabs in one-dimensional analyses and for slabs and tops/bottoms of cylinders in two-dimensional analyses. Second, as periodic represents a repetition of a shape, it must be applied to both `ibl` and `ibr` in one-dimension or to `ibb` and `ibt` in two dimensions. In other words, if you have a periodic boundary on the right side of a slab, there must be a periodic boundary on the left side of the slab. Or if you have a periodic boundary at the bottom of a cylinder or slab, you must have a periodic boundary at the top of the cylinder or slab. Finally, although the white boundary condition is in a sense non-physical, it should be applied to all curved surfaces where you want a 'reflective-type' boundary.

All of the above description is accurate but doesn't provide much guidance to a criticality specialist in determining which boundary condition is appropriate for the problem at hand.

5.2.C.5 Applying Boundary Conditions to Physical Problems

Based on geometry considerations rather than flux considerations, you would use the boundary conditions in the following way. For a single unit, the outside boundary or boundaries would be set to the vacuum condition. This isolates the system. If a system has symmetry in one or more directions, then you would use a reflective boundary along a symmetry plane. The example below shows the use of a reflective boundary condition.

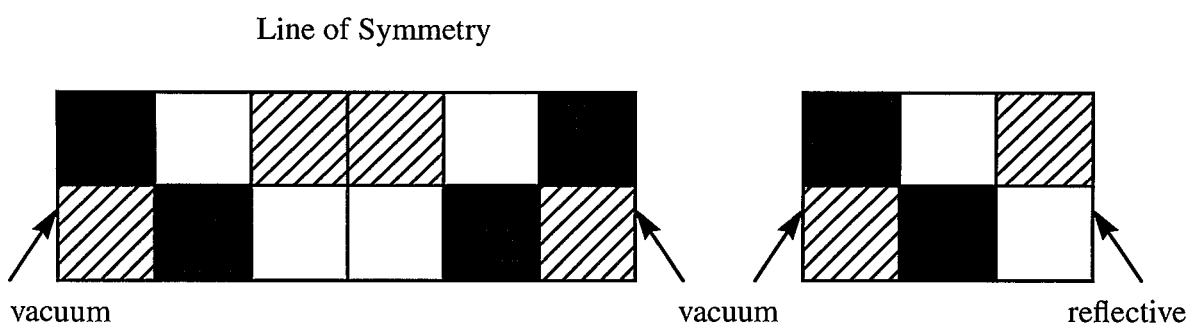


Figure 5.5 — Use of Reflective Boundary Condition Along a Line of Symmetry.

Example 3b was a complete cylinder; with a reflective boundary it can be modeled as just the upper or lower half (see section 5.5). Using a symmetry line through the middle, we only need to model the upper half of the cylinder. (Note that for items depending on volume, the use of a symmetry line reduces the volume by one-half. Parameters such as mass from a mass edit will be 0.5 of the actual value when a reflective boundary is used. See Section 5.5.D for an example.)

If you have a system that repeats infinitely in a direction, then you can use either a reflective or a periodic boundary condition. For example, consider a stack of cans of alternating material. If the height to diameter ratio is very large, then you can consider the stack to be infinite in the z-direction. You could apply the boundary conditions as shown in the following example.

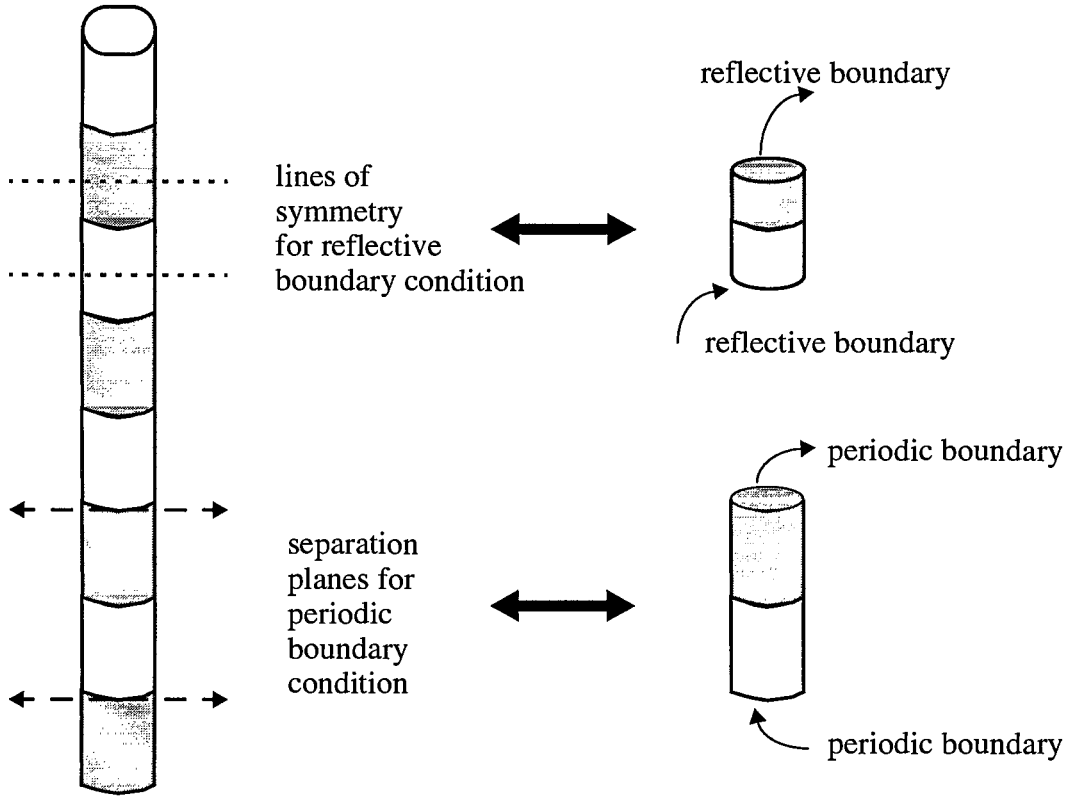


Figure 5.6 — Reflective and Periodic Boundaries.

Reflective boundary conditions represent lines of symmetry while periodic boundary conditions represent basic 'building blocks' of a model. In Figure 5.6, there are two different materials

repeated in the z-direction: the white can and the grey can. The basic building block is one unit on top of the other. It is this building block that the periodic boundary condition repeats in the upward and in the downward directions. When using periodic boundaries remember that they must be applied in pairs (that is, if either left or right *or* top or bottom is periodic, then both left and right *or* both top and bottom must be periodic).

Note that using either a reflective or a periodic boundary condition effectively eliminates leakage in that direction. Evaluation of a system without leakage provides the infinite multiplication factor for the system.

Analyses of Infinite Multiplication Factors

In many cases a criticality specialist is concerned with the possibility of a quantity of material or a mixture to achieving criticality. The exact geometry of such a system is of less concern than the infinite multiplication factor. The k_∞ of the system indicates the properties of the material mixture and provides guidance on whether the system can achieve criticality. If k_∞ is less than 1.0, the system will be subcritical in every configuration so extensive analyses of specific geometries are unnecessary. To perform a k_∞ calculation on planar geometries, you need to set all boundary conditions to a value of **1** indicating reflective conditions. This effectively eliminates all system leakage and provides a multiplication factor which is independent of geometry. To perform a k_∞ calculation on curved geometries (either spherical or cylindrical), you need to set **ibr** to a value of **3** indicating white boundary conditions. For an r-z cylinder, the planar surface boundaries, **ibb** and **ibt**, would be set to **1** for reflective while the radial boundary would have **ibr** = **3**.

5.2.C.6 Optional Block 5 Keywords

There are three other Block 5 keywords which although optional are often used in criticality calculations. We have used these, **kcalc**, **norm**, and **chi**, in previous example problems. Now we will provide the details on each of these three keywords.

kcalc - invoke a special convergence scheme for criticality calculations

norm - normalize the fission source rate to a given value

chi - provide group fission fractions by zone

In most criticality analyses we are not concerned with exact values of the angular fluxes but are concerned with a good value of the eigenvalue. When **kcalc=1** , the eigenvalue is

converged but a final convergence on the angular fluxes is not performed. This reduces the problem run time without losing information required in criticality analyses. If `kcalc=0` , then both the eigenvalue and the pointwise fluxes are converged. For normal cases when `kcalc=0` , the eigenvalue convergence default is 0.0001 while when `kcalc=1` , the default is less restrictive, 0.001.

Using the **norm** keyword allows you to set the total fission source rate in a problem to any value desired. For most criticality problems, you will find it convenient to set the total fission source rate to 1.0 neutron per second. This normalizes all of the processes occurring in the system such that their sum will be 1.0 (i.e., the total absorptions in a system plus the total leakage from all outside surfaces should equal 1.0). This allows the user to check the validity of the run and also provides relative fractions for absorption and leakage.

For problems containing combinations of fissile material, you need to specify groupwise fission fractions. As discussed, some libraries have these chi values built in while others do not. For the Hansen-Roach 167 isotope library, there are no built in chi values so the user must specify them. This can be done either in Block 3 with the **chivec** keyword or in Block 5 with the **chi** keyword. Entering the chi values in Block 5 provides more flexibility because any variation of the values with material zones can only be handled with Block 5 input. *Note: if you are using a bxslib that does not have chi values, entering them in Block 3 will not work. DANTSYS overwrites your chi values with those from the bxslib which are zero so you will get an error indicating no fissions. Thus, it is strongly suggested that chi values always be entered in Block 5.* The chi values for each zone should be representative of the dominant fissionable isotope in the zone. The availability of zone-dependent chi values is particularly useful for problems in which one zone may have a different dominant fissile isotope than another zone. Data for the **chi** keyword is entered by zone and by group. Thus, for the Hansen-Roach library with 16 groups, there would be 16 entries which would apply to the first zone, and then the next 16 entries apply to the second zone. If there are more zones than zone-group entries, the last 16 entries will be applied to all remaining zones.

The block 5 input for our problem looks like:

```
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0        t
```

Because the 167 isotope H-R library does not contain values of χ, we enter the fission fractions appropriate to plutonium. The `10r0.0` is shorthand for repeat the value 0.0 ten times. Note that although we have three material zones in the problem there is only one set of chi values. These apply to the first zone which is the reflector. As there are less sets of chi values than material zones, DANTSYS will apply the 'last set', in this case the first set, to all remaining material zones. Which means that the given set of chi values will also be applied to zone 2, the core, and zone 3, the shell. This concludes the Block 5 input section.

5.2.D Other Input for Example Problem

To complete the input for problem 5a, we only need to provide the information for Block 6 — the mass edit input. This is identical to that for the previous example problems; details on the **massed** keyword are given in Chapter 2.

Block 6 provides the edit information which allows you to get reaction rates, fluxes, or sources at individual points or in individual zones. We use the mass edit to indicate total mass and density by zone.

```
/ *** block 6 ***
  massed=1          t
```

This concludes the Block 6 input section.

5.2.E Complete Input for Reflected Sphere

```
bbbbb3
Problem 5a – Plutonium and Gallium sphere w/ Nickel Coating
        Reflected with 3.83 cm graphite, Radius = 5.465 cm, using
Hansen-Roach cross sections, 16 groups, sn=16, mesh=1 cm
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=3  it=11  mt=4
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   5.465   5.478   9.308
  xints=      6      1       4
  zones=      2      3       1           t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matls=  fuel     pu492e1  3.9404e-2;   /pu239 atom density
          gallium  ga       1.3647e-3;   /gallium atom density
          plate    ni       9.1305e-2;   /nickel atom density
          graph    c        8.1724e-2    /graphite atom density
  assign= refl      graph     1.0;
          core      fuel      1.0  gallium   1.0;
          shell     plate     1.0
  t
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0       t
/ *** block 6 ***
  massed=1          t
```

5.2.F Running Problem 5a

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable and the `prob5a` input file. For this case, we are assuming that you are using the Hansen-Roach 167 isotope library and that it is in binary format. This means that *bxslib* must be in your directory. To run, type

```
twodant < prob5a > prob5a.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0120 as written on the screen echo.

5.2.G Output

First, check the group edit and balance tables. Make sure that the in scatter total (3.3408360e-1) matches the out scatter total (3.3408360e-1) to within three decimal places. For this run, k_{eff} = 1.0120454, but remember it is only converged to the third decimal place (i.e., 0.001).

The last section of output is the mass edit table. As this is the same problem as we ran in Chapter 3 all of the mass edit information should be identical. To make a spot check, look at the zone 2 information. For zone 2 (the core which combined plutonium and gallium), there should be a total mass of 10.8022 kg with an average density of 15.80 g/cc and a volume of 683.69 cc.

5.3 BLOCK 5 INPUT FOR TWO-DIMENSIONAL PROBLEMS

The only change required in Block 5 input when performing a two-dimensional analysis is the addition of two more boundary condition keywords, **`ibt`**, for the top boundary condition and **`ibb`**, for the bottom boundary condition. The next sections provide a number of two-dimensional example problems with various values for the top, bottom, and right boundary conditions.

5.4 EXAMPLE PROBLEM 5b FINITE CYLINDER

The specifications for example problem 5b are identical to those for problem 3b, a two-dimensional cylinder. The difference is that we will discuss the additional boundary conditions applicable to two-dimensional problems.

5.4.A Specifying Additional Boundary Conditions

In the two-dimensional geometries useful for criticality calculations, r-z cylinders and x-y slabs, there are two more boundary condition keywords which should be specified, **ibt** and **ibb**. These are the top and bottom of the cylinder or the upper y-surface and the lower y-surface of the slab. Values for these keywords are the same as discussed in Section 5.2.C.4. As all of these surfaces are plane surfaces, you can use either reflective or periodic boundary conditions with them.

5.4.B Block 5 Input

For this example problem, 5b, we will use vacuum boundary conditions for both the top and bottom of the cylinder. The complete input for Block 5 with all the boundary conditions looks like:

```
/ *** block 5 ***
  ievt=1  isct=1  ibr=0 ibb=0 ibt=0  norm=1.0 kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0        t
```

The input for the other blocks (1, 2, 3, 4, and 6) remain the same as those given in example problem 3b.

5.4.C Complete Input File

As this is a two-dimensional example problem, we will use the input given for example problem 3b. The only changes we make are in the title, and in the Block 5 input as discussed above. The changes are indicated in **Courier bold** type. The complete input for example 5b should look like:

```
bbbbb3
Problem 5b – Plutonium and Gallium cylinder w/ Nickel Coating
            Reflected with 5 cm graphite, Radius = 4.935 cm, using
Hansen-Roach cross sections, 16 groups, sn=6, mesh= 1 cm
/ *** block 1 ***
  igeom= r-z  ngroup=16  niso=167  isn=6  im=3  it=11
       jm=5   jt=21  mt=4  nzone=3
       maxlcm=800000  maxscm=100000  miniprt=yes        t
```

```
/ *** block 2 ***
  xmesh= 0.0   4.935   4.948   9.948
  xints=     5       1       5
  ymesh= 0.0   5.0   5.013   13.5012   13.5142   18.5142
  yints=     5     1       9         1         5
  zones=       1      1     1;
               3      3     1;
               2      3     1;
               3      3     1;
               1      1     1           t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matls=  fuel     pu492e1  3.9404e-2;   /pu239 atom density
          gallium  ga       1.3647e-3;   /gallium atom density
          plate    ni       9.1305e-2;   /nickel atom density
          graph    c        8.1724e-2    /graphite atom density
  assign= refl      graph     1.0;
          core      fuel      1.0  gallium   1.0;
          shell     plate     1.0
  t
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  ibb=0  ibt=0  norm=1.0  kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0       t
/ *** block 6 ***
  massed=1          t
```

5.4.D Running Problem 5b

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable and the `prob5b` input file. To run, type

```
twodant < prob5b > prob5b.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0074 as written on the screen echo.

5.4.E Output

First, the k_{eff} for this problem should be identical to that for problem 3b (k=1.0074). Then look at the mass edit table. For this run it shows a total mass in zone 1 (the reflector) of 8.31491 kg of graphite with an atom density of 8.17240e-2 atoms/barn-cm. For zone 2 (the core containing plutonium and gallium), there is a total mass of 10.2611 kg. For the nickel shell, we have 0.0482818 kg.

5.5 EXAMPLE PROBLEM 5c FINITE CYLINDER — SYMMETRY PLANE

The specifications for example problem 5c are identical to those for problem 5b except that we will use a reflective boundary condition through the center plane of the cylinder. This cuts the cylinder in half so we only have to model the top half with reflection to get the same value for problem 5b.

5.5.A Specifying Boundary Conditions with Symmetry Planes

In terms of boundary conditions, the only change that needs to be made is to change **`ibb`** from `0`, a vacuum to `1`, reflective. However, if we don't change anything else we end up with a cylinder that is twice as tall as we originally had. Thus, we also need to redo the Block 1 and Block 2 input describing the meshing and cylinder height.

5.5.B New Input for Problem 5c

We will start at the midplane of the cylinder and call that z=0. The height of the core region is 4.2441 or one-half the given height of 8.4882 cm. The shell goes from 4.2441 cm to 4.2581 cm to account for its thickness of 0.013 cm. Then we include 5 cm of graphite at the top going from 4.2581 cm to 9.2581 cm. We will have 5 fine meshes in the core (we had 9 and halved it to 4.5 but need an integer number so we used 5), 1 in the shell and 5 in the graphite. Thus there are now three coarse mesh regions in the z-direction with a total of 11 fine meshes (`jm=3`

jt=11). The new input for Blocks 1, 2, and 5 looks like (again changes are noted in **bold Courier** type):

```
/ *** block 1 ***
  igeom= r-z  ngroup=16  niso=167  isn=6  im=3  it=11
  jm=3   jt=11  mt=4  nzone=3
  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   4.935   4.948   9.948
  xints=     5       1       5
  ymesh= 0.0   4.2441 4.2571 9.2571
  yints=     5       1       5
  zones=     2       3       1;
             3       3       1;
             1       1       1           t
/ *** block 5 ***
  ievt=1  isct=1  ibr=0 ibb=1 ibt=0  norm=1.0 kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0       t
```

Note that since the z meshing changed, changes were required to the **zones** keyword entries to delete the bottom two coarse mesh regions.

5.5.C Running Problem 5c

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable and the prob5c input file. To run, type

```
twodant < prob5c > prob5c.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0076 as written on the screen echo.

5.5.D Output

The k_{eff} for this problem should have been identical to that for problem 5b (k=1.0074) but it was slightly higher (k=1.0076). The difference could be a result of only converging to 3

decimal places or due to the change in fine mesh size in the core because we couldn't have 4.5 fine meshes. Remember that changes in mesh size and quadrature will affect your results. In this case, there is only a minor difference, but with other problems it may be significant. Now look at the mass edit table. For this run it shows a total mass in zone 1 (the reflector) of 4.15746 kg of graphite which is 0.5 times the 8.31491 kg of graphite that we had in problem 5b. Using the midplane as a reflective surface gives the correct k_{eff} but reduces the material inventories by one-half. For zone 2 (the core containing plutonium and gallium), there is a total mass of 5.13055 kg which again is one-half the 10.2611 kg listed in problem 5b. For the nickel shell, we have 0.0241409 kg which is one-half of 0.0482818 kg.

5.6 EXAMPLE PROBLEM 5d CYLINDER WITH INFINITE REFLECTOR

If we want to specify an infinite axial reflector for this cylinder problem, then we change the top boundary condition. We will change **`ibt`** from `0`, a vacuum to `1`, reflective. This makes the 5 cm graphite reflector effectively infinite in the axial direction but still only 5 cm thick in the radial direction.

5.6.A New Input for Problem 5d

The only change required is in Block 5 where we change the top boundary condition. The new input for Block 5 looks like (again changes are noted in **`bold Courier`** type):

```
/*** block 5 ***
  ievt=1  isct=1  ibr=0  ibb=1  ibt=1  norm=1.0  kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0        t
```

5.6.B Running Problem 5d

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable and the `prob5d` input file. To run, type

```
twodant < prob5d > prob5d.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0674 as written on the screen echo.

5.6.C Output

The k_{eff} for this problem (k=1.0674) is greater than that for problem 5c (k=1.0076) which it should have been due to the addition of the infinite axial reflector. This should reduce the axial leakage to zero so we will check the integral summary information table. Note that the entries for both top leakage and vertical leakage are 0.00 because of the effectively infinite reflector. The horizontal leakage is 0.6295 which is significantly greater than the horizontal leakage of 0.4408 for a cylinder with 5 cm of graphite in all directions.

5.7 EXAMPLE PROBLEM 5e CYLINDER, k-infinity CALCULATION

To specify an infinite multiplication factor problem, we set all boundary conditions to reflective for planar surfaces and white for curved surfaces. Since we already have both axial boundary conditions set to reflective for problem 5d, we only need to change the radial condition to white, **`ibr`** from `0`, a vacuum to `3`, white. This makes the system effectively infinite in both the axial and the radial directions.

5.7.A New Input for Problem 5e

The only change required is in Block 5 where we change the right boundary condition. The new input for Block 5 looks like (again changes are noted in **`bold Courier`** type):

```
/*** block 5 ***
  ievt=1  isct=1  ibr=3  ibb=1  ibt=1  norm=1.0  kcalc=1
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0        t
```

5.7.B Running Problem 5e

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable and the `prob5e` input file. To run, type

```
twodant < prob5e > prob5e.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 2.4703 as written on the screen echo.

5.7.C Output

The k_{eff} for this problem (k=2.4703) is substantially greater than that for problem 5d (k=1.0674) which it should have been due to the elimination of all system leakage. Checking the integral summary information table, we find that there is no leakage and 1.000955 relative absorption. The absorption is larger than 1.000 because we are doing a kcalc which is only converged to the third decimal place (i.e., 0.001).

5.8 EXAMPLE PROBLEM 5f INFINITE 1-D CYLINDER

Although we specified an infinite axial reflector and effectively infinite reflector in problems 5d and 5e, these are not the same as running an infinite, one-dimensional cylinder with the same radial configuration. For an infinite 1-D cylinder calculation, we need to make changes to Blocks 1, 2, and 5 much like we did in problem 5c. We change the right boundary condition back to a vacuum. As we are only in 1-D, there are no other boundary conditions to deal with. However, we also need to remove the non-pertinent 2-D information from Blocks 1 and 2. Note in Block 5 we added the iquad=4 for the one-dimensional cylinder case.

5.8.A New Input for Problem 5f

We will remove the axial region and mesh information so the new input for Blocks 1, 2, and 5 looks like (again changes are noted in **bold Courier** type):

```
/ *** block 1 ***
  igeom= cyl  ngroup=16  niso=167  isn=6  im=3  it=11
  mt=4  nzone=3
  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   4.935   4.948   9.948
  xints=     5       1       5
  zones=      2       3       1      t
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0  kcalc=1   iquad=4
  chi= 0.225 0.347 0.161 0.170 0.084 0.013 10r0.0       t
```

5.8.B Running Problem 5f

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable and the `prob5f` input file. To run, type

```
twodant < prob5f > prob5f.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.364 as written on the screen echo.

5.8.C Output

The k_{eff} for this problem (k=1.364) is substantially greater than that for problem 5d (k=1.0674) which it should have been due to making the core, shell, and reflector all infinite in the radial direction. Note that this geometry as modeled results in an infinite core and shell as well as an infinite radial reflector; this is different than problem 5d where the core and shell were finite in height while the reflector was made infinite. The caveat to the user is to make sure that boundary conditions are properly applied so that the model accurately reflects what is desired.

5.9 SUMMARY

This chapter presented you with six examples designed to teach the input requirements for the solver; particularly the boundary condition keywords. You learned what the kcalc keyword does to speed up runtime, and you learned how to provide chi values which are representative of the major fissile constituent in each zone. Finally, through a number of examples, you learned how to use different boundary conditions to take advantage of symmetry, to reduce leakage, or to evaluate the infinite multiplication factor for a system. This completes the discussion on Block 5 input which describes the solution process and boundary conditions involved in a problem.

Chapter 6
Cross-Section Description (Block 3)

The first five chapters provided you with all the input required to perform DANTSYS calculations on individual units. However, as noted earlier, results of these calculations are highly dependent on which cross-section library was used. One major misconception regarding most neutronic codes is that they have cross-section libraries built in. This is not true. It is up to the user to select or generate the cross-section library which best fits the problem. DANTSYS requires a multigroup cross-section library, but **it can be any multigroup library**. Throughout the years, the criticality community has used the Hansen-Roach cross sections[3] and found them applicable within a few percent to most systems. Yet even stating that you are using the Hansen-Roach library is not a complete enough description. Hansen-Roach primarily refers to a sixteen group energy structure and associated cross sections. However, there are a number of cross-section libraries which use the sixteen group structure but have different values of cross sections. For this primer we have used the 167 isotope Hansen-Roach library which is available from ESH-6 at LANL. As this library is in the DANTSYS binary form, our input in Block 3 has been quite simple using only a single keyword to indicate the form of the cross-section library. Even with the binary form cross-section library, there are still some issues which must be addressed for proper use of the Hansen-Roach library: how to select the correct "isotopes" and which order of scattering to be used. If you decide to use a different cross-section library, then there are a number of additional Block 3 keywords that will be needed to describe the structure of the library. This chapter will focus on specific information needed for the Hansen-Roach library and then on general information regarding the use of other libraries. As with the other input Blocks, there is input control information in Block 1 which relates to Block 3 so this will be reviewed also.

6.1 WHAT YOU WILL BE ABLE TO DO:

- Calculate the potential scattering cross section and use it to select the appropriate "isotope" from the 167 isotope Hansen-Roach library.
- Describe the differences in input requirements among various cross-section libraries.
- Explain when anisotropic scattering cross sections are needed in a problem.

6.2 PROBLEM DESCRIPTION

For the identification of the proper fissile isotope, we will use a reflected solution cylinder containing uranyl fluoride with the data obtained from (LA–10860–MS, p.33).[2] The table provides information for various H/U_{235} atomic ratios. For this example we will use an H/X (H/U_{235}) ratio of 524 which gives an infinite cylinder critical diameter of 31.8 cm. The solution is 4.89% enriched in U-235 and has a U-235 density of 0.0425 g/cc.

6.2.A Reflected Infinite Cylinder Geometry

The reflected infinite solution cylinder $(U4.89)O_2F_2$ will be analyzed with the following dimensions and atom densities.

Solution Material (ρ_{U235} = 0.0425 g/cc-mixture)

Uranium (4.89 percent U-235)

N_{235}	= 1.0889e-4 atoms/b-cm
N_{238}	= 2.09135e-3 atoms/b-cm

Solution Materials

N_F	= 4.40048e-3 atoms/b-cm
N_H	= 5.705836e-2 atoms/b-cm
N_O	= 3.292966e-2 atoms/b-cm

Cylinder Material

Aluminum

ρ	= 2.70 g/cc-mixture (single material)

Reflector Material

Water

ρ	= 1.00 g/cc-mixture (single material)

The atom densities are calculated using the formulas found in Appendix B.

Configuration I. *Reflected Infinite Cylinder*

Solution:	Diameter	= 31.8 cm
Aluminum Cylinder:	Thickness	= 0.1587 cm
Water Reflector:	Thickness	= 30.00 cm

6.2.B Title, Block 1, and Block 2 Input

Now that you have been through chapters on each of the blocks except Block 3, you should be able to set up the input for the title, Block 1, and Block 2. For this problem we decided to have 3 material zones, 3 mixtures, 3 coarse meshes, and a fine mesh size of approximately one cm. This means that the first lines of our input file look like:

```
ƀƀƀƀƀ3
Problem 6a – Infinite cylinder 31.8 cm in diameter
Reflected with 30 cm water, U(4.89)O2F2 w/ H/X=524
using Hansen-Roach cross sections, 16 groups, isn=-6, mesh=1cm
/ *** block 1 ***
  igeom= cyl  ngroup=16  niso=167  isn=6  im=3  it=47  mt=3
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   15.9   16.0587   46.0587
  xints=     16      1       30
  zones=      1      2       3           t
```

Remember the terminator, `t`, for Block 1 is included at the end of the last card, and the ƀ is used to indicate a blank space on an input line. Your title lines may be completely different from those shown; that is okay as long as the number of title lines matches that given on the first card. If your input in Blocks 1 or 2 is different, that may be okay but check to make sure that your coarse meshes have the same positional values (remember the cylinder diameter was given in the problem specifications but DANTSYS requires that you enter the cylinder radius). We have indicated 3 material zones with the first being the solution, the second being the shell, and the third being the reflector. As we have done before we could turn these around and define them in any order, but the most logical is from the inside out as shown above. Also note that because we are working with a cylinder, we need to use iquad=4 in Block 5.

6.2.C Sources of Cross Sections in the Hansen-Roach Library

Appendix D shows the isotopes available in the 167 isotope Hansen-Roach library. Reviewing that list indicates that there are multiple isotopes available for hydrogen, deuterium, magnesium, silicon, calcium, copper, cadmium, tungsten, thorium-232, uranium-233, uranium-234, uranium-235, uranium-236, uranium-238, plutonium-238, plutonium-239, plutonium-240, and plutonium-241. There are basically three reasons for the multiple isotopes: application to a different spectrum for scattering, application to a different spectrum for absorption, and different source for the cross section.

For most of the isotopes listed above, the reason for multiple listings is that there are two sources of cross sections for those materials: either Oak Ridge (OR) or Los Alamos (indicated by absence of OR). The listing in Appendix D gives the source for each cross-section set for each isotope. The ones listed as LAMS 2941[6] came from the original Hansen-Roach report. The others have been derived along the way. At the present, there appears to be no adequate basis for choosing one source over another. If a particular material is of significance in your problem, then it is suggested that you analyze the problem using each of the available cross-section sets to determine the effect of that material on the results. No guidelines can be given because each set is particularly suited to one spectrum while applicable to many. The advantage of the Hansen-Roach data is that it has been proven over the years to be reasonably applicable to a multitude of spectra.

6.2.D Selecting Cross Sections Based on Scattering Spectrum

Within the 167 isotope library, the Oak Ridge generated cross sections and hydrogen and deuterium have anisotropic scattering cross sections as well as isotropic scattering cross sections. The effects of the two moderating materials: hydrogen and deuterium, will be highly dependent on the characteristics of the neutron spectrum. For these materials there are two sets of cross sections: one weighted with a fission spectrum (CHI) and one weighted with a 1/E slowing down spectrum (DE). There is also one other cross-section set for deuterium (dLA) whose origin is unknown. For the majority of criticality calculations, it is suggested that the DE set be used whenever hydrogen or deuterium is present as a moderator. This is primarily because the presence of one of these materials tends to create a softer spectrum where 1/E slowing down will have a major effect.

6.2.E Selecting Cross Sections Based on Absorption Spectrum

As detailed in Appendix E, the resonances for the fissile nuclides have different effects depending on the neutron spectrum. For cross-section libraries other than Hansen-Roach, the cross sections are usually collapsed based on the spectrum appropriate to the problem. In the Hansen-Roach library, this spectral collapsing is replaced by having a number of different cross-section sets for each fissionable nuclide. The user then selects the set which is most appropriate for the spectrum. This selection process is based on the calculation of a potential scattering cross section which is then used as an indicator of the neutron spectrum in the problem. Because there

are a number of cross-section sets for these nuclides, the Hansen-Roach library is labeled as having 167 isotopes, but in fact, many of these 'isotopes' are just variations in the absorption cross sections for each fissionable nuclide.

6.2.E.1 Determining Potential Scattering Cross Sections

The process for determining the potential scattering cross section of a system uses the scattering cross sections for the light isotopes and calculates a total macroscopic light isotope scattering cross section. Essentially what is required is the summation of the macroscopic scattering cross sections for all isotopes with $A \leq 15$, and then for oxygen-16, one-half the macroscopic scattering cross section is added to the sum. A complete description of this process and suggested values of microscopic scattering cross sections for the lighter isotopes are given in Appendix E.

6.2.E.2 Potential Scattering Cross Section for Example Problem

For the example problem, there are only two isotopes with $A \leq 16$: hydrogen in H_2O, and oxygen in H_2O and in UO_2F_2. To calculate the potential scattering cross section of the system, we multiply the atom density of hydrogen, 5.705836e-2 atom/b-cm by its scattering cross section, 20.4 barns, and add that to one-half the product of the atom density of oxygen, 3.292966e-2 atom/b-cm, and its scattering cross section, 3.8 barns. (Remember, the one-half is only applied to the macroscopic scattering cross section of oxygen.) This gives a value of 1.2266 cm^{-1} for the system potential scattering cross section.

6.2.E.3 Determining Which Fissionable Isotopes to Select from the Library

Now that you have the system potential cross section, you can determine which fissionable 'isotopes' to use in the problem. In the example problem, there are only two fissionable isotopes: Uranium-235 and Uranium-238. However, in the Hansen-Roach library, there are 13 different cross-section sets for U-235 and 18 different sets for U-238. The correct set for each nuclide is based on the system potential scattering cross section. To determine which set to use for U-235, divide the system potential scattering cross section of 1.2266 cm^{-1} by the atom density of U-235, 1.0889e-4 atom/b-cm, which gives a value of 11,264 b. This value is used to select the correct cross-section set. Note, in the library, each of the U-235 cross sections is labeled in the following form: *u25xey* where the *u25* indicates uranium-235 and the *xey* is the value of the system potential

cross section that the set is based on. For our example problem, the closest listed potential cross section is 1e4 or 10,000 b. This means that we will use the *u251e4* cross-section set in the problem. Although we are discussing this in the Block 3 input, the `u251e4` will be entered in the materials listing of Block 4. For uranium-238, the cross-section sets are identified in the following form: *u28xey* where the *u28* indicates uranium-238 and the *xey* is the potential scattering cross section for the set. In a similar fashion to the U-235, we determine the value for uranium-238 as follows: divide the system potential scattering cross section of 1.2266 cm^{-1} by the atom density of U-238, 2.09135e-3 atom/b-cm, which gives a value of 586 b. This value is between the set based on 4e2 (400 b) and 6e2 (600 b) but is closer to 600. In this case we will use the cross-section set labeled *u286e2* for the uranium 238 material.

Impact of sigma-p on K-effective

As discussed in Appendix E, it is not particularly important to match the value for the nuclide, in this case 586, by mixing some 400 material with some 600 material, but it is extremely important that a set close to the calculated value be used. This example problem represents a critical system. When analyzed with the 600 cross-section set for U-238, $k_{eff} = 1.0090$; when analyzed with the 400 set, $k_{eff} = 1.0237$. However, if we choose a value of *u28inf* Infinitely dilute (low concentration of uranium) for the material then we get a k_{eff} of 0.7615 while choosing a value of *u280e0* (metallic uranium with no moderation) for the material gives a k_{eff} of 1.0892. Because the Hansen-Roach cross sections are not calculated from the problem spectrum, it is important that the appropriate cross-section set be selected if the resonance absorption impacts on system multiplication are to be correctly characterized. As seen from this example, the calculated k_{eff} of a system may vary from 0.76 to 1.09 depending on the cross-section set selected. It is vital that the potential scattering cross section be calculated and used to identify the appropriate cross-section set. Also note that the sigma-p effect is stronger for Th-232, U-238, and Pu-240 than for other fissile isotopes. This is further discussed in Appendix E.

6.2.F Other Input for Example Problem

To complete the input for problem 6a, we need to provide the information for Blocks 3 through 6. These will be similar to the previous examples with the exception of the mixtures and ordering of the material zones.

6.2.F.1 Block 3 Input

Information concerning the cross-section library is given in Block 3. Using a Hansen-Roach cross-section library in binary format, there is only one required keyword for Block 3.

```
/ *** block 3 ***
  lib= bxslib          t
```

6.2.F.2 Block 4 Input

The Block 4 input provides the material and mixing information to DANTSYS. We have previously identified the correct cross-section sets for u-235 and u-238 as *u251e4* and *u286e2*. For this problem we will enter the data on the water reflector and the aluminum container using atom fractions while the UO_2F_2 solution data is entered based on atom densities. Using these approaches, the entire Block 4 input should look like:

```
/ *** block 4 ***
  matspec=  atfrac  atfrac  atdens
  matls=  water     hDE       0.6667       /hydrogen atom fraction
                    o         0.3333;      /oxygen atom fraction
          alum      al        1.0000;      /aluminum atom fraction
          uo2f2     u251e4    1.0889e-4    /u-235 atom density
                    u286e2    2.09135e-3   /u-238 atom density
                    f         4.40048e-3   /fluorine atom density
                    hDE       5.705836e-2  /hydrogen atom density
                    o         3.292966e-2  /oxygen atom density
  assign= 1         uo2f2     1.0;         /zone 1
          2         alum      2.70;        /zone 2 al density=2.7
          3         water     1.0          /zone 3 h2o density=1.0
  t
```

(Note that in place of material zone names such as `refl` or `core`, we have labeled the material zones with numbers. The zone <u>number</u> referenced in Block 2 is simply the entry number in the `assign=` array, e.g., If we decided to label the first entry in the **`assign`** array as 13 or any other number, this would have no effect on its referenced zone number for Block 2 input. It is the first entry in the **`assign`** array and is therefore zone number 1 regardless of how we label it in

the **assign** array. If you have a number of material zones, it is often useful to label them with their correct zone numbers so there is a quick cross reference to the zone assignments in Block 2. This concludes the Block 4 input section.

6.2.F.3 Block 5 Input

The Block 5 input provides the solver information to DANTSYS. There are two Block 3 related keywords that we need to provide values for in Block 5: **isct** and **chi**. The **isct** keyword indicates the highest order of scattering which is to be used in problem calculations. Most nuclides scatter isotropically so `isct=0` is the default for this keyword. However, two isotopes, hydrogen and deuterium, have significant anisotropic scattering which must be accounted for. If the **isct** keyword is not present in Block 5, it defaults to `0`, and any anisotropic scattering is ignored. The Hansen-Roach cross-section library has anisotropic scattering sets for both hydrogen and deuterium; **isct** should be set to `1` so that anisotropic scattering from these materials is accounted for. If the example problem is run with `isct=0` then the k_{eff} increases from 1.0090 to 1.0929. Just as with the potential scattering cross sections, it is important to make sure that anisotropic scattering due to hydrogen or deuterium is accounted for by using the correct value for **isct**. Note other cross-section libraries may have higher order Legendre scattering cross sections available for many isotopes so the value of **isct** may be 3, 4, or higher depending on the highest order of scattering cross section available in the library. However, for the Hansen-Roach library, the highest order of scattering is 1 so **isct** should not be greater than `1`. For most criticality problems, you will want to use all of the scattering orders available so set **isct** to the highest order available from the library you are using. Do not let **isct** default to `0` for criticality calculations.

The other Block 5 keyword associated with the cross sections is **chi** which provides the fraction of fission neutrons born into each energy group. Some libraries have values for chi provided; in fact the Hansen-Roach 118 isotope library does. However, in many cases, these are fixed chi values which may be appropriate for one element, say uranium, but not necessarily for another such as plutonium. It is always best to supply values for **chi** which are appropriate to the materials in the problem. For the previous examples we have dealt with plutonium systems and have provided the chi values appropriate to plutonium. For this problem we have a uranium

system so we need to use the appropriate uranium chi values. These are shown below with all of the Block 5 input for example problem 6a.

```
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0 kcalc=1  iquad=4
  chi= 0.204 0.344 0.168 0.180 0.090 0.014 10r0.0        t
```

Note the differences in chi values between those given above for uranium and those used in previous examples for plutonium. This concludes the Block 5 input section.

6.2.G Complete Input for Solution Cylinder

The Block 6 input remains unchanged from previous example problems so the complete input for the reflected solution cylinder should look like:

```
bbbbb3
Problem 6a - Infinite cylinder 31.8 cm in diameter
Reflected with 30 cm water, U(4.89)O2F2 w/ H/X=524
using Hansen-Roach cross sections, 16 groups, isn=6, mesh=1cm
/ *** block 1 ***
  igeom= cyl  ngroup=16  niso=167  isn=6  im=3  it=47  mt=3
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   15.9   16.0587   46.0587
  xints=     16      1       30
  zones=      1      2       3           t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matspec=  atfrac  atfrac  atdens
  matls=  water     hDE       0.6667      /hydrogen atom fraction
                    o         0.3333;     /oxygen atom fraction
          alum      al        1.0000;     /aluminum atom fraction
          uo2f2     u251e4    1.0889e-4   /u-235 atom density
```

```
                    u286e2     2.09135e-3  /u-238 atom density
                    f          4.40048e-3  /fluorine atom density
                    hDE        5.705836e-2 /hydrogen atom density
                    o          3.292966e-2 /oxygen atom density
  assign= 1         uo2f2      1.0;        /zone 1
          2         alum       2.70;       /zone 2 al density=2.7
          3         water      1.0         /zone 3 h2o density=1.0
  t
/ *** block 5 ***
  ievt=1  isct=1  ibr=0  norm=1.0 kcalc=1  iquad=4
  chi= 0.204 0.344 0.168 0.180 0.090 0.014 10r0.0       t
/ *** block 6 ***
  massed=1          t
```

6.2.H Running Problem 6a

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable, `dant.x` and the `prob6a` input file. For this case, we are assuming that you are using the Hansen-Roach 167 isotope library and that it is in binary format. This means that *bxslib* must be in your directory. To run, type

```
dant.x < prob6a > prob6a.out
```

DANTSYS will write information to the screen showing the title of the run and the value of k_{eff} calculated on each iteration. The calculation for this problem should take much less than a minute. The k_{eff} value determined with this input on a SUN SPARC 10 was 1.0090 as written on the screen echo.

6.2.I Output

First, look at the group edit and balance tables. Make sure that the in scatter total (9.2680984e0) matches the out scatter total (9.2680982e0) to within three decimal places. Note the values obtained for in scatter and out scatter may not match those given above, but should be close. The integral summary information table gives the system k_{eff} and the totals of each of the columns from the group balance table. For this run, k_{eff} = 1.0090433, but remember it is only converged to the third decimal place (i.e., 0.001)

The last section of output is the mass edit table. For zone 1 (the core containing UO_2F_2 and water), there is a total mass of 1.5713 kg with an average density of 1.98 g/cc and a volume of 794.23 cc. Note that the mass of the uranium-235 alone is 0.033755 kg. For the aluminum cylinder, we have 0.0430209 kg with an average density of 2.7 g/cc and an atom density of 6.02617e-2 which are reasonable values for aluminum.

This run shows a total mass in zone 3 (the water reflector) of 6.0 kg of water with an average density of 1.00 g/cc. Note that this is the same density given in the problem statement which indicates that the atom density entered matched the problem statement. Remember, zones refer to the material zones created in Block 4.

6.3 Other Cross-Section Libraries

Although this primer has discussed the use of the Hansen-Roach library, **it is certainly not the only library which can be used with DANTSYS**. Almost any other cross-section library can be used with DANTSYS as long as it is formatted in a manner compatible with the DANTSYS options. For a library that is in the DANTSYS binary format, you only need to use the **`lib`** keyword with the value `bxslib`. In some cases, the binary forms are not easily transportable so the cross-section library may be sent in the ASCII form of the binary library. (LANL uses this method for the 167 isotope Hansen-Roach library.) If your library is in this format, then you need to use the following input along with the Block 1, 2, 4, 5, and 6 input for your run:

```
/ *** block 3 ***
  lib= xslibb      savbxs=1          t
```

This will read the ASCII form of the binary and then write a binary form (a result of using `savbxs=1`). You then need to take the binary form which will be called bxslib and then save it in the appropriate directory under some other name such as bxshr167 (for the 167 isotope Hansen-Roach library) or bxsmendf (for a mendf library). Once you have saved the binary, then future runs will use the `lib= bxslib libname= bxshr167` input for Block 3. Libname is a keyword of Block 3 that allows the user to specify the name of the bxslib if it is not bsxlib.

Many other libraries are ASCII files written in a DTF/ANISN format. These libraries can be used by entering `lib= xslib`. However, there are a number of optional Block 3 keywords which will be required to read these libraries and create a useful binary (`savbxs=1`). These

keywords are described in the DANTSYS manual [see 2-35 to 2-40 or 3-35 to 3-40]. One important note is that you must use the **names** keyword in Block 3 and the **atwt** keyword in Block 4 to create a binary which has names and atomic weights that allow the use of the **massed** keyword in Block 6. Only if this information is present will you be able to generate a mass edit output for checking parameters such as atom density, mass density, or volume. [A more complete discussion on the use of non-DANTSYS binary cross-section libraries is provided in Chapter 10 of the DANTSYS manual.]

6.4 SUMMARY

This chapter presented you with information on calculating the potential scattering cross sections for the Hansen-Roach library and demonstrated the effects of fissionable cross-section set on k_{eff}. You should be able to describe the differences in input requirements among various cross-section libraries, and explain the difference between isotropic and anisotropic scattering cross sections. Finally, the use of various cross-section libraries was discussed with reference to the DANTSYS manual for specifics on integrating those libraries with DANTSYS. Remember, DANTSYS requires a cross-section library to calculate a value for k_{eff}. Although this primer uses the 167 isotope Hansen-Roach library, it is not the only library which can be used in criticality calculations. Check with people at your facility to determine which libraries are available and which may have be used previously with DANTSYS for criticality calculations.

Chapter 7
Dimensional and Concentration Searches

Now that you have been through detailed explanations of the six input blocks, this chapter will focus on the search capabilities of DANTSYS. Many times in criticality problems, you will be looking for the size of a system which produces a given k_{eff} or for a concentration of a material that makes a system critical or subcritical. DANTSYS has a search capability which allows you to do these calculations and more; however, in this primer we will only focus on dimensional and concentration searches.

7.1 WHAT YOU WILL BE ABLE TO DO:

- Search for a critical radius or thickness in a bare system.
- Search for a radius or thickness that will give a desired k_{eff}.
- Search for thicknesses of multiple regions while keeping other regions constant in size.
- Find the atom density of a material or materials that will give a desired k_{eff}.

7.2 PROBLEM DESCRIPTION

The first problem will be to find the critical radius of a bare sphere of U(93.71) with a density of 18.74 g/cc. The value calculated by DANTSYS should be close to that given in Table 29, page 97 of LA–10860–MS.

7.2.A Sphere Geometry

As this is a simple geometry with only two materials, we will use the weight fraction input in Block 4.

Core Material (ρ_{mix} = *18.74 g/cc-mixture)*
Uranium 235, 93.71 wt %
Uranium 238, 6.29 wt %

For dimensional search problems, we need to provide an estimate of the size which will be the initial value used in the search. In this example, we will use 10 cm as the starting radius.

7.2.B Title, Block 1, Block 2, Block 3, and Block 4 Input

A dimensional search problem differs from normal k-calculation problems only in the Block 5 input. As noted above, you have to supply an initial guess for the radius in the Block 2 input. This is a simple sphere problem with one coarse mesh, one material mixture, and one material zone. With small, bare systems, the result is highly dependent on quadrature. We will use isn=16 for these search problems, but if you require very precise dimensions, then use isn=48 for spheres. Your input file should look similar to:

```
bbbbb2
Problem 7A – U(93.71) sphere, no reflector, radius search
using Hansen-Roach cross sections, 16 groups, sn=16, mesh=1 cm
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=1  it=10  mt=1
  nzone=1  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   10.0
  xints=     10
  zones=      1           t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matspec= wtfrac
  matls=  fuel   u252e1    0.9371  /u235 wt. fraction
                 u280e0    0.0629  /u238 wt. fraction
  assign=  core   fuel   18.74           t
```

When entering a value for the initial size, it is best to be small rather than large. In this example, if we used a value of 100 for the initial radius, we would likely find that DANTSYS would give us the following error message, `NEG C.M. DELTA.` To do the search, DANTSYS calculates the k_{eff} with the initial estimate and then a k_{eff} with a smaller estimate. These two values are then used to extrapolate to the radius where k_{eff} equals the desired value. If the initial estimate is a long ways off of the actual value, that extrapolation may give a negative size for the system; hence the error message, `NEG C.M. DELTA.` If you get this message, simply reduce your initial estimate to a value that should be smaller than the final one and rerun DANTSYS.

Another item to note is that although the size of the system will change during the search, the number of fine meshes in each coarse mesh region remains constant. Thus, it is wise to over-mesh the regions whose dimensions are changing. This means that as the dimension is increasing, the fine mesh width will also increase, but should stay around 1 to 1.5 cm per fine mesh. In this example, we have started with 1cm fine meshes. If the final dimension is less than 15 cm then we will still be in the normal range. If the final dimension is greater than 15 cm, then it might be useful to rerun the problem with the final dimension as the initial guess and a number of fine meshes that will give a mesh size around 1 cm.

7.2.C Block 5 Input for Dimensional Searches

For a dimensional search, the Block 5 input requires some more keywords that we haven't used before. To select a dimensional search, the value for the **ievt** keyword is changed from 1 (for an eigenvalue problem) to 4 (dimensional search). Values for **isct**, **ibr**, **norm**, and **kcalc** remain as before. The values for the **chi** keyword are changed to reflect a uranium system rather than a plutonium system. The major changes are in the introduction of these keywords:

rm Radius modifiers for each coarse mesh interval, (for use with dimension searches only).

evm Initial search parameter increment (required for dimensional searches, no default available).

pv Value of k_{eff} which you are searching for (default = 1.0).

We will spend several sections of this chapter on the appropriate values for **rm** depending on system configuration, but for a bare system, you need to set `rm=1.0` in a dimensional search. In most searches, using `evm=-0.1` will work will so this is a good value for this keyword. For this example, we are doing a critical search so we will let **pv** default to 1.0.

With these entries the rest of the input for example problem 7a should look like:

```
/ *** block 5 ***
  ievt=4  rm=1.0  evm=-0.1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.204 0.344 0.168 0.180 0.090 0.014 10r0.0       t
/ *** block 6 ***
  massed=1          t
```

This concludes the input for this example problem.

7.2.D Running Problem 7a

Remember, whichever cross-section library you are using must be present in the same directory as the DANTSYS executable and the `prob7a` input file. For this case, we are assuming that you are using the Hansen-Roach 167 isotope library and that it is in binary format. This means that *bxslib* must be in your directory. To run, type

```
dant.x < prob7a > prob7a.out
```

DANTSYS will write information to the screen showing the title of the run, but instead of writing the value of k_{eff}, the value of the search eigenvalue will be written. The search eigenvalue is used in determining the new dimensions but you don't need to worry about what value DANTSYS calculates; the results of the search are given in the output file.

7.2.E Output

First, check the group edit and balance tables. Make sure that the in scatter total (2.7938379e-1) matches the out scatter total (2.7938379e-1) to within three decimal places. For this run, the search eigenvalue was -1.2157843e-1, but remember it depends on the initial radius guess and is not the final answer to the search.

The next section after the integral summary table provides the results of the dimensional search. It lists the final coarse mesh boundaries. Coarse mesh boundary 1 is still at 0.0, the center of the sphere. Coarse mesh boundary 2 (the outside boundary of coarse mesh region 1) has moved from the initial estimate of 10.0 to 8.784 (which is its value for a k_{eff} of 1.0).

The last section of output is the mass edit table. For search runs, the values given in the mass edit are based on the final results of the search. In this case, this means based on a uranium sphere with a radius of 8.784 cm which equates to a volume of 2.839e3 cc as given in the mass edit. The critical mass of U-235 in a spherical configuration of U(93.71) is calculated as 49.86 kg. This compares with 49.12 kg of U-235 and 8.55 cm radius as given in LA–10860–MS, pg. 97.

7.2.F Dimensional Search with a Given k_{eff}

In this example, we will search for the radius of a U(93.71) sphere which has a k_{eff} of 0.9. The only change required to the input file used for problem 7A is the addition of `pv=0.9` in Block 5. This indicates that instead of searching based on a k_{eff} of 1.0, DANTSYS will search based on a k_{eff} of 0.9.

The Block 5 input for example problem 7aa should look like:

```
/ *** block 5 ***
  ievt=4  pv=0.9  rm=1.0  evm=-0.1  isct=1  ibr=0
  norm=1.0 kcalc=1
  chi= 0.204 0.344 0.168 0.180 0.090 0.014 10r0.0        t
```

It would also be helpful to change your title card to indicate this is problem 7aa and that k=0.9.

7.2.G Output for Problem 7aa

Running DANTSYS using the same commands as before with the `prob7aa` input file should give you a search eigenvalue of -2.25419e-1. Note this is different from that of problem 7aa as the k_{eff} for this case is different.

Checking the results from the dimensional search shows coarse mesh boundary 2 (the outside boundary of coarse mesh region 1) has moved from the initial estimate of 10.0 to 7.746 (which is its value for a k_{eff} of 0.9). For search runs, the values given in the mass edit are based on the final results of the search. In this case, this means based on a uranium sphere with a radius of 7.746 cm which equates to a volume of 1.9467e3 cc as given in the mass edit. The spherical mass of U-235 for a k_{eff} of 0.9 is 34.19 kg versus the calculated critical mass of 49.86 kg.

7.3 DIMENSIONAL SEARCH FOR TWO-REGION PROBLEMS

For multiple regions, you must provide a value of **rm** in each region. Those regions where the size or thickness is not changing will have an `rm=0.0`. Those problems where the dimension of only one region is changing will have an `rm=1.0` applied to that region and `rm=0.0` for all other regions. In this example, we will evaluate the critical radius of U(93.71) (density = 18.5 g/cc) with a graphite reflector, density of 1.66 g/cc and a thickness of 20 cm.

7.3.A Basic Input for Two-Region Problems

The input for this example problem is very similar to that for problem 7a. The only differences are the addition of a second coarse mesh and a second material. Input for the first four Blocks should look like:

```
ƀƀƀƀƀ2
Problem 7b – U(93.71) sphere, 20cm graph reflector, radius search
using Hansen-Roach cross sections, 16 groups, sn=16, mesh=1 cm
```

```
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=2  it=20  mt=2
  nzone=2  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   10.0   30.0
  xints=     10     10
  zones=      1      2     t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matspec= wtfrac
  matls=  fuel   u252e1    0.9371  /u235 wt. fraction
                 u280e0    0.0629; /u238 wt. fraction
          graph  c         1.00    /graphite wt. fraction
  assign=  core   fuel   18.5;
           refl   graph   1.66           t
```

We again use 10 cm as the initial estimate of the core radius; then with a 20 cm reflector, the outside of the system will be at 30 cm. As graphite has a larger mean free path than uranium, we increase our fine mesh size to 2 cm in the reflector. Note the semicolons in the Block 4 input to separate the two mixture specifications and the two material zone descriptions.

7.3.B Radius Modifiers for Two-Region Problems

As mentioned earlier, there must be **im** entries for the **rm** keyword to correspond to the **im** entries in the coarse mesh definitions. In this example, we have 2 coarse mesh regions so there will be 2 values for the dimension modifiers; the first value will be the value for the first coarse mesh region (core) and the second is for the second coarse mesh region (reflector). We are keeping the thickness of the reflector constant and searching for the core dimension which provides a critical system. For a region whose dimension doesn't change, we use `0.0` for its **rm** entry. With only two regions in the problem and only one changing, we will use `1.0` for the **rm** of the changing region.

With these entries the rest of the input for example problem 7b should look like:

```
/ *** block 5 ***
  ievt=4  rm=1.0,0.0  evm=-0.1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.204 0.344 0.168 0.180 0.090 0.014 10r0.0       t
```

```
/ *** block 6 ***
  massed=1            t
```

As there is no entry for **pv**, the default of `1.0` will be used so the final configuration will represent a critical system. This concludes the input for this example problem.

Comments on the signs for pv and evm

Up to this point we have provided a negative sign for **evm** and a positive sign for **rm** where the dimension is changing. When **ev** is defaulted to 0, then the product of **rm** and **evm** represents a direction of change for the dimensions. If a calculated k_{eff} is greater than its desired value (i.e., k_{eff} (calc) - k_{eff} (desired) > 0), the dimension of a fuel region or a region of positive reactivity must be reduced to get to the desired value. Thus, the product of **rm** and **evm** should have a negative sign so the dimension is changed in the correct direction. If however, the region is an absorber or has negative reactivity, then the product of **rm** and **evm** should be positive indicating that the dimension of such a region should be increased if the k_{eff} is too large. The negative **evm** provides some stability in the search calculation so the sign of **rm** is chosen to match the required conditions. As the example problems have dealt with changes in the dimension of a fuel region, we have used a positive **rm** to go along with the negative **evm**.

7.3.C Problem 7b Results

After analyzing problem 7b with DANTSYS, you should have the following results. The dimensional search shows coarse mesh boundary 2 (the outside boundary of coarse mesh region 1, the core) at 6.444 cm while the position of coarse mesh boundary 3 (the outside of coarse mesh region 2, the reflector) is now at 26.444 cm. Note that the thickness of the graphite reflector remained at 20 cm even though the core size was reduced from the initial estimate. Also, note that the addition of a reflector reduced the critical core radius from 8.784 cm to 6.444 cm which seems logical. The total critical mass of uranium, U(93.71), with the reflector is 20.74 kg as noted in the mass edit output while the critical mass of U-235 is 19.43 kg.

7.3.D Two-Region Problem With Fixed Outside Radius

Let's now look at the problem where we are replacing the graphite reflector with an aluminum reflector (density of 2.7 g/cc) but we want the outside dimension of the system to remain the same (i.e., outside radius = 26.444 cm). In this case we will need to compensate for

any reduction in the core region size by increasing the reflector region by the same amount. It is relatively easy to state this in words, but to get DANTSYS to do it requires more effort. In the DANTSYS manual[1] [Input Details, pg. 7-34], there is an equation which indicates how coarse mesh boundaries are altered. This equation indicates that the new coarse mesh boundary ($\tilde{R}_{k+1}$) for region k depends on the new boundary ($\tilde{R}_k$) for region k-1 and on the old values of those boundaries. When this equation is evaluated for our example, $RM_2 = -R_1 / (R_2 - R_1)$; assuming that `rm=1.0` for the first region. This means that the rm value for the reflector is equal to the negative of the initial estimate of the core outside boundary (in this case, 10) divided by the initial thickness of the reflector region (in this case, 26.444-10=16.444). So the second value for **rm** should be `-0.6081` if we want to keep the outside dimension constant.

7.3.E Basic Input for Problem 7c

The input for this example problem is almost identical to that for problem 7b. The only differences are on the title card, the location of the outside boundary, and changing from graphite to aluminum in the reflector. Input for the first four Blocks should look like:

```
ƀƀƀƀƀ3
Problem 7c – U(93.71) sphere, aluminum reflector, radius search
outside dimension fixed at 26.444 cm
using Hansen-Roach cross sections, 16 groups, sn=16, mesh=1 cm
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=2  it=20  mt=2
  nzone=2  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   10.0   26.444
  xints=     10     10
  zones=      1      2     t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matspec= wtfrac
  matls=  fuel   u252e1    0.9371  /u235 wt. fraction
                 u280e0    0.0629; /u238 wt. fraction
          alum   al        1.00    /graphite wt. fraction
```

```
assign=  core   fuel   18.5;
         refl   alum    2.7          t
```

We again use 10 cm as the initial estimate of the core radius while the outside of the system is fixed at 26.444 cm. We have changed the reflector material from graphite to aluminum.

7.3.F Block 5 Input With Fixed Outside Dimension

As mentioned earlier, there must be **im** entries for the **rm** keyword to correspond to the **im** entries in the coarse mesh definitions. In this example, we have two coarse mesh regions so there will be 2 values for the dimension modifiers; the first value will be the value for the first coarse mesh region (core) and the second is for the second coarse mesh region (reflector). In section 7.3.D, we calculated the values of rm required to keep the outside dimension fixed. We enter `rm=1.0, -0.6081` for the two values. This will change the reflector size in the opposite direction and by the opposite amount of any change to the core size. With these entries the rest of the input for example problem 7c should look like:

```
/ *** block 5 ***
  ievt=4  rm=1.0,-0.6081 evm=-0.1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.204 0.344 0.168 0.180 0.090 0.014 10r0.0        t
/ *** block 6 ***
  massed=1          t
```

Again we use the default for **pv**, so the final configuration will represent a critical system. This concludes the input for this example problem.

7.3.G Output for Problem 7c

After analyzing problem 7c with DANTSYS, you should have the following results. The dimensional search shows coarse mesh boundary 2 (the outside boundary of coarse mesh region 1, the core) at 7.259 cm while the position of coarse mesh boundary 3 (the outside of coarse mesh region 2, the reflector) remained constant at 26.444 cm. Note that the thickness of the aluminum reflector is 19.185 while the graphite reflector had a thickness of 20 cm. This implies that graphite is a better reflector in this system because less fuel (radius = 6.444 cm) was required than with the aluminum reflector (7.259 cm). The total critical mass of U(93.71) with the aluminum reflector is 29.64 kg while the mass of the U-235 is 27.77kg.

7.4 THREE REGION PROBLEMS

7.4.A Fixed Outside Radius And Constant Thickness Middle Zone

Let's now look at the problem where we have a constant thickness aluminum shell around the core and a graphite reflector on the outside. In addition to keeping the shell thickness constant, we want the outside dimension of the system to remain the same (i.e., outside radius = 26.444 cm). In this case we will need to compensate for any reduction in the core region size by increasing the reflector region by the same amount. We apply the same equation from the DANTSYS manual[1] [Input Details, pg. 7-34] but with three regions. If we assume that **rm** for the core region is set to `1.0`, then **rm** for the shell region will be `0.0` (thickness unchanged during the search). This leaves us with the **rm** for the reflector. When the equation is evaluated for this problem, $RM_3 = -R_1 / (R_3 - R_2)$. There is a pattern here: set `rm=1.0` for the first region, then the value of rm required for the compensating region (i.e., the region whose size is changing to compensate for a change in the first region) is equal to minus the estimated thickness of the first region divided by the initial estimated thickness of the compensating region. In this example, we have an initial estimate of 10 cm for the core region which leaves an initial thickness of 14.444 cm (26.444 - 2 - 10) for the graphite reflector region. So the third value for **rm** should be `-0.6923` (= -10 / 14.444) if we want to keep the outside dimension constant.

7.4.B Basic Input for Problem 7d

The input for this example problem is very similar to that for problem 7c. The only differences are the addition of a third coarse mesh and third material and changing the reflector material back to graphite.

Input for the first four Blocks should look like:

```
bbbbb3
Problem 7d – U(93.71) sphere, al shell, c reflector, radius search
keeping al thick at 2cm, & outside dimension fixed at 26.444 cm
using Hansen-Roach cross sections, 16 groups, sn=16, mesh=1 cm
/ *** block 1 ***
  igeom= sph  ngroup=16  niso=167  isn=+16  im=3  it=22  mt=3
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   10.0   12.0  26.444
  xints=     10      2    10
  zones=      1      2     3        t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matspec= wtfrac
  matls=  fuel   u252e1    0.9371  /u235 wt. fraction
                 u280e0    0.0629; /u238 wt. fraction
           alum  al        1.00;   /aluminum wt. fraction
          graph  c         1.00;   /graphite wt. fraction
  assign=  core   fuel   18.5;
           tank   alum    2.7;
           refl   graph   1.66         t
```

We again use 10 cm as the initial estimate of the core radius, give the aluminum shell a thickness of 2 cm and, fix the outside of the system at 26.444 cm.

7.4.C Block 5 Input

The only change to the Block 5 input is the requirement for three values for **rm** as we now have three coarse regions. In section 7.4.A, we calculated the values of rm required to keep the outside dimension fixed. We enter rm=1.0, 0.0, -0.6923 for the three values. This will keep the shell thickness constant and change the reflector size in the opposite direction and by the opposite amount of any change to the core size.

With these entries the rest of the input for example problem 7d should look like:

```
/ *** block 5 ***
  ievt=4  rm=1.0,0.0,-0.6923 evm=-0.1  isct=1  ibr=0
  norm=1.0 kcalc=1
  chi= 0.204 0.344 0.168 0.180 0.090 0.014 10r0.0        t
/ *** block 6 ***
  massed=1          t
```

Again we use the default for **pv**, so the final configuration will represent a critical system. This concludes the input for this example problem.

7.4.D Problem 7d Results

After analyzing problem 7d with DANTSYS, you should have the following results. The dimensional search shows coarse mesh boundary 2 (the outside boundary of coarse mesh region 1, the core) at 6.572 cm, the position of coarse mesh boundary 3 (the outside of coarse mesh region 2, the shell) moved to 8.572 maintaining a constant thickness of 2 cm. The outside system boundary remained constant at 26.444 cm. The critical mass of U(93.71) with the aluminum shell and graphite reflector is 22.00 kg with a mass of 20.62 kg for U-235.

7.5 DIMENSIONAL SEARCH FOR A TWO-DIMENSIONAL PROBLEM

Just as you must provide a value of the dimensional modifier for all regions, you must provide values in both dimensions when doing two-dimensional searches. The major change going from a one-dimensional to a two-dimensional search is the change in the keyword from **rm** to **xm** (for r in r-z geometries or x in x-y geometries) and **ym** (for z in r-z geometries or y in x-y geometries). Values are provided in the same way as for a one dimensional problem.

7.5.A Two-dimensional Example Problem

As an example of a search in a two-dimensional problem, we will seek the critical height of U(93.71)NO_3 in an aluminum tank with no external reflector. The following dimensions and atom densities are from the Criticality Benchmark Project, Volume II.b.[7]

Solution Material (ρ_U = 0.05489 g/cc-mixture, ρ_{sol} = 1.0758g/cc)

Uranium (93.71 percent U-235)

N_{235} = 1.3103e-4 atoms/b-cm

N_{238} = 9.5108e-6 atoms/b-cm

Solution Materials

N_N = 3.4432e-4 atoms/b-cm

N_H = 6.5441e-2 atoms/b-cm
N_O = 3.4003e-2 atoms/b-cm

Cylinder Material
Aluminum
N_{Al} = 5.9469e-2 atoms/b-cm

Configuration *Bare R-Z Cylinder*

Solution:	Diameter	= 33.01 cm
Aluminum Cylinder:	Thickness	= 0.32 cm
	Height	= 50.00 cm

To select the correct cross section, a potential scattering cross-section analysis must be done for the two uranium isotopes. For U-235, the potential scattering cross section is calculated as 10,800 which indicates that `u251e4` is the appropriate cross section. For U-238, the potential scattering cross section is 150,000 indicating that `u28inf` is the appropriate cross section.

7.5.B Basic Input for Dimensional Searches in r-z Geometry

The input for this example problem should refresh your memory on two-dimensional input requirements. We will have two x coarse meshes: one for the solution and one for the tank; and three y coarse meshes: one for the tank bottom, one for the solution, and one for the void above the solution. With this information, input for the first four Blocks should look like:

```
ƀƀƀƀƀ3
Problem 7e – U(93.71)no3 cylinder, no refl. al tank, height search
radius=16.505 cm w/ rho=1.0758g/cc and 54.89gU/liter
using Hansen-Roach cross sections, 16 groups, sn=-6, mesh=1 cm
/ *** block 1 ***
  igeom= r-z ngroup=16 niso=167 isn=6 im=2 it=19 jm=3 jt=43
  mt=2 nzone=2  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   16.505   16.825
  xints=     17        2
  ymesh=  -0.64  0.0  35.5  50.0
  yints=        2   36     5
```

```
  zones=       2   2;
               1   2;
               0   2      t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matspec= wtfrac
  matls=  fuel   u251e4    1.3103e-4  /u235 atom density
                 u28inf    9.5108e-6  /u238 atom density
                 o         3.4003e-2  /oxygen atom density
                 n         3.4432e-4  /nitrogen atom density
                 hDE       6.5441e-2; /hydrogen atom density
          alum   al        5.9469e-2  /aluminum atom density
  assign=  core   fuel    1.0;
           tank   alum    1.0           t
```

We use 35.5 cm as the initial estimate of the solution height but the outside boundary is fixed by the tank height of 50 cm.

7.5.C Block 5 Input For Two-Dimensional Search

Analogous to the one-dimensional case, there must be **im** entries for the **xm** keyword and **jm** entries for the **ym** keyword corresponding to the **im** and **jm** entries in the coarse mesh definitions. In this example, we have 2 radial coarse mesh regions so there will be 2 values for **xm** and 3 axial coarse mesh regions so there will be 3 values for **ym**. We are not changing any of the radial dimensions, so the two values of xm will be zero. In the axial direction, we are searching for the height of the solution which will make the system critical, while holding the tank height constant. The three values for **ym** correspond to the aluminum tank bottom plate, the solution height, and the top of the tank. The bottom thickness will not change so its **ym** value will be zero. The solution height does change so its **ym** value will be 1.0.

Then to keep the tank height constant at 50 cm, we use the formula from 7.3.D reconfigured for an axial modifier; $\{YM_3 = -Z_2 / (Z_3 - Z_2)\}$. Substituting Z_3=50 and Z_2=35.5, we get -2.4483 for the third value of ym. With these entries the rest of the input for example problem 7e should look like:

```
/ *** block 5 ***
  ievt=4  xm=0.0,0.0  ym=0,1.0,-2.4483 evm=-0.1
  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.204 0.344 0.168 0.180 0.090 0.014 10r0.0        t
/ *** block 6 ***
  massed=1          t
```

As there is no entry for **pv**, the default of `1.0` will be used so the final configuration will represent a critical system. This concludes the input for this example problem.

7.5.D Problem 7e

After analyzing problem 7e with DANTSYS, you should have the following results. The dimensional search shows results for the i mesh boundaries (the radial boundaries) and the j mesh boundaries (the axial boundaries). The modified r and original r values are the same as we did not search in the radial direction. In the axial direction, the first two boundaries remain at -0.64 and 0.0 indicating the thickness of the aluminum tank remains unchanged. The third boundary representing the solution height changed from our initial guess of 35.5 cm to a final, critical height of 41.18 cm. The last boundary remains unchanged at 50 cm. The critical mass of U-235 is 18.02 kg as noted in the mass edit output.

7.6 CONCENTRATION SEARCHES

Although a search on dimension is the most common search used in criticality problems, there are times when we would like to know what concentration of material or what uranium enrichment is required for criticality. These concentration searches use the same Block 5 keywords, **ievt**, **evm**, and **pv** as do the dimensional searches. However, the modifiers for concentration are entered in Block 4 with the other materials information instead of in Block 5 with the solver data. For a concentration search, **ievt** is set to 3 (`ievt=1` for eigenvalue problem, `ievt=3` for concentration search, and `ievt=4` for dimensional search). The values for the other keywords, evm and pv remain as discussed in 7.2.C.

7.6.A Block 4 Input for Concentration Searches

For a concentration search, one additional Block 4 keyword is required:

asgmod C1 parameters used in concentration searches.

The C1 parameters act just like the radius modifiers in that they are multiplied by the search eigenvalue and added to a previous value to get a new concentration. The general form of the **asgmod** instruction is: ASGMOD= zone mat_m $C1_m$ mat_n $C1_n$ ... where zone is the name of the material zone, mat_m is the character name of the mth material whose concentration will be modified and $c1_m$ is its modifier.

Because the c1 modifier is applied to a mixture, you will need to separate the material of interest from the other materials. As an example, if you are searching on boron concentration in a water mixture, the original eigenvalue model may have combined boron, oxygen, and hydrogen into a single mixture. However, when doing the concentration search, you will need to have two mixtures: one for oxygen and hydrogen, and one for the boron. With the two separate mixtures, then you can apply a concentration modifier only to the boron while leaving the concentrations of oxygen and hydrogen unchanged.

Determining the appropriate value of c1 for a material is done in the same way we determined the radius modifier for a dimension search. If you are only varying the concentration of a single material, then the appropriate c1 value is 1.0. If you are varying two materials whose concentrations are dependent on each other, then choose a c1 value of 1.0 for the first and a value of either (initial concentration of second / initial concentration of first) if the concentrations increase together, or a c1 value of - (initial concentration of second / initial concentration of first) if their concentrations are inversely related. (See grey box on uranium enrichment for an example of inversely related concentrations.)

Searches on Uranium Enrichment

In doing searches on enrichment for U-235, you need to separate U-235 and U-238 from the rest of the fuel or core. Then you give the name of the zone that the uranium is in, e.g., core followed by the name of the uranium-235, e.g., u235, followed by a 1.0 for its modifying parameter. Then you follow this by the name of the uranium-238, e.g., u238, followed by a modifying parameter of — (initial u235 atom density) / (initial u238 atom density). Because the output from a concentration search is a concentration multiplier (i.e., a value that multiplies the original concentration), a search on uranium enrichment is best done starting with an initial u-235 enrichment of 1% or 10%. This makes it easy to use the final multiplier to determine the necessary enrichment.

7.6.B Example Problem for Concentration Search

As an example of a concentration search, let's take example problem 6a and do a search on uranium enrichment. Problem 6a was a reflected solution cylinder containing uranyl fluoride. In that example, we calculated a k_{eff} of 1.009 for a u-235 enrichment of 4.89%. Let's use the same geometry and start with a u-235 enrichment of 1%. We will search on both u-235 and u-238 concentrations, as these are dependent on each other, to find the concentrations that will give a k_{eff} of 1.009. We are not changing the uranium concentration in the solution nor are we changing the concentration of solution. Therefore, the atom densities used for fluorine, hydrogen, and oxygen in problem 6a will stay the same. We will separate u-235 and u-238 from the solution and identify each of them as a separate mixture. To make it easier to determine the final enrichment, we will start with an initial u-235 enrichment of 1%; obtaining the atom density by dividing by 4.89. As the total uranium atom density will be unchanged, the initial u-238 atom density is obtained by subtracting the initial u-235 atom density (2.227e-5) from the total uranium atom density (2.20024e-3).

Although concentrations are changed during a search, the cross-section set remains fixed. This could cause problems if the final concentrations are significantly different from those initially entered. You need to select initial concentrations and cross-section sets which are as close to the final answer as possible. When using the Hansen-Roach library, the dependence on potential scattering cross section may make it difficult to know which cross-section set to select for a concentration search. In this particular problem, the initial potential scattering cross section for u-235 is about 55,000 which would indicate either `u25inf` or `u251e4` as the cross-section set. As the u-235 concentration increases, the potential scattering cross section decreases. This would indicate that using the u251e4 set should be more applicable across a wider range of u-235 concentrations. For u238, the decision is easier as the initial potential scattering cross section is about 550, and it will increase with decreasing u-238 concentration. Based on this, the `u286e2` cross-section set appears to be applicable for all concentrations that might be encountered in this problem.

7.6.C Input for Concentration Search Example

With the appropriate Block 5 keywords for a concentration search to a k_{eff} of 1.009, the input for problem 7f should look like:

```
bbbbb3
Problem 7f – inf. cylinder 31.8 cm dia, water reflected, k=1.009
U(?)O2F2 with H/U = 30.4, 30 cm water reflector, enrichment search
Hansen-Roach cross sections, 16 groups, sn=6, same as prob6a
/ *** block 1 ***
  igeom= cyl  ngroup=16  niso=167  isn=6  im=3  it=47  mt=5
  nzone=3  maxlcm=800000  maxscm=100000  miniprt=yes       t
/ *** block 2 ***
  xmesh= 0.0   15.9  16.0587  46.0587
  xints=     16     1       30
  zones=      1      2     3        t
/ *** block 3 ***
  lib= bxslib          t
/ *** block 4 ***
  matspec= atfrac atfrac atdens
  matls=  water  hDE      0.66667       /hydrogen wt. fraction
                 o        0.33333;      /oxygen wt. fraction
          alum   al       1.00;         /aluminum wt. fraction
          u25    u251e4   2.227e-5;     /u235 atom density @1%
          u28    u286e2   2.17797e-3;   /u238 wt. fraction
          sol    f        4.40048e-3    /fluorine atom density
                 hDE      5.705836e-2   /hydrogen atom density
                 o        3.292966e-2   /oxygen atom density
  assign=  fuel    sol  1.0  u25 1.0  u28  1.0;
           tank    alum   2.70;     /aluminum density
           refl    water   1.0      /water density
  asgmod=  fuel u25 1.0  u28 -0.0102
/ *** block 5 ***
  ievt=3  pv=1.009 evm=-0.1  isct=1  ibr=0  norm=1.0 kcalc=1
  chi= 0.204 0.344 0.168 0.180 0.090 0.014 10r0.0
  oitm = 30  iquad=4   t
/ *** block 6 ***
  massed=1          t
```

We select a c1 modifier of 1.0 for u-235; then the c1 modifier for u-238 is calculated as (-initial 235 / initial 238) = (-2.227e-5 / 2.17797e-3) = 0.0102. Note that we have added a new keyword in Block 5, **oitm**. This sets the limit on the number of outer iterations which

DANTSYS will perform. It defaults to 20 so is not required in most runs. However, in some searches, more than 20 outer iterations will be required. To change the limit, set **oitm** to 30 or 40 as necessary. This concludes the input for this example problem.

7.6.D Problem 7f

After analyzing problem 7f with DANTSYS, you should have the following results. The results of the concentration search are found near the end of the output. For each material number, the multiplier on its final concentration is given. For material 3 (the third mixture defined in the matls input), the multiplier is given as 4.889. This would be multiplied by the initial atom density to give the final atom density (= 4.889*2.227e-5 or 1.0888e-4) which is the value given in the mass edit table for u-235. Because we used an initial atom density for u-235 which corresponded to an enrichment of 1%, the multiplier of 4.889 indicates that the final u-235 enrichment would be 4.889% which is very close to the value of 4.89% found in problem 6a.

Note that for mixtures where the input is in atom fractions, specifically the water and the aluminum in this problem, the multiplier is the physical density of the material. Thus, if you are doing concentration searches on materials entered in atom fractions, the final result will be the density of the material. Again note that as with dimension searches, the mass edit values are for the final configuration. In this problem, there is 0.0338 kg of u-235 and 0.6567 kg of u-238 to create a system with $k_{eff} = 1.009$.

Variation of Enrichment with Cross-Section Set

As noted above, the cross-section set chosen for u-235 was u251e4 although u25inf was closer to the initial atom density. Re-running problem 7f with the u25inf cross section gave an enrichment of 4.892% rather than the 4.889% obtained using the u251e4 set. Thus, there is very little difference in the results of this problem using either set. However, if greater accuracy is desired, then you need to make two runs: the first to determine the approximate value, and the second using the approximate value and its associated cross-section set to get a final, more accurate value.

7.7 SUMMARY

This chapter presented you with numerous examples designed to illustrate the search techniques within DANTSYS. You learned how to search for a critical radius or thickness in a

bare system, search for a radius or thickness that will give a desired k_{eff}, and search for thicknesses of multiple regions while keeping other regions constant in size. You also learned how to find the atom density of a material or materials that will give a desired k_{eff}. Specifically, the concentration search example demonstrated how to determine a u-235 enrichment to produce a given k_{eff}.

This completes the primer discussion on DANTSYS. Although the primer has been limited to standard criticality applications, DANTSYS is capable of handling more complex problems. These build on the basic information about the 6 INPUT BLOCKS presented here so further analyses should begin with a thorough understanding of the material in this primer. You are reminded that a single analysis of any problem with DANTSYS is only a beginning; a complete analysis of a system requires a number of runs and a good understanding of neutron interaction fundamentals. DANTSYS is only one of many tools available to the criticality analyst and should be used within its proper context. However, many, quick one-dimensional analyses of typical criticality configurations can provide you with a substantial understanding of the system to its various parameters.

REFERENCES

1. R. E. Alcouffe, R. S. Baker, F. W. Brinkley, D. R. Marr, R. D. O'Dell, and W. F. Walters, "DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System," LA–12969–M, Manual, Los Alamos National Laboratory, June 1995.

2. H. C. Paxton and N. L. Pruvost, "Critical Dimensions of Systems Containing U-235, Pu-239, and U-233," LA–10860–MS, Los Alamos National Laboratory, July 1987.

3. G. E. Hansen and W. H. Roach, "Six and Sixteen Group Cross Sections for Fast and Intermediate Critical Assemblies," LAMS–2543, Los Alamos Scientific Laboratory, December 1961.

4. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analyses: Theory and Guidelines for Effective Use of Transport Codes," LA–10983–MS, Los Alamos National Laboratory, September 1987.

5. F. W. Walker, J. R. Parrington, and F. Feiner, **Nuclides and Isotopes, Fourteenth Edition**, GE Nuclear Energy, San Jose, CA, 1989.

6. G. I. Bell, J. J. Devaney, G. E. Hansen, C. B. Mills, W. H. Roach, and L. D. Connolly, "Los Alamos Group-Averaged Cross Sections," LAMS–2941, Los Alamos Scientific Laboratory, September 1963.

7. Nuclear Energy Agency Nuclear Science Committee, **International Handbook of Evaluated Criticality Safety Benchmark Experiments**, NEA/NSC/DOC(95)03, multiple volumes, 1995

APPENDIX A

Commonly Encountered Error Messages in DANTSYS

DANTSYS has a number of error messages to guide the user when the input or the execution is not proceeding normally. As discussed earlier, DANTSYS uses a sequential process for its input evaluation so the first error encountered will be valid. Those beyond the first error may or may not be valid messages. For example, if a terminator is left off of Block 1, there will be an error message indicating current block contains arrays belonging to other blocks. There will be other error messages indicating Hollerith data in numeric fields, or invalid array count, etc. These are only there due to the lack of a terminator in a previous block. Inserting the terminator may clear a number of error messages.

The following are a list of typical error messages that a user might encounter along with some possible solutions that should correct the problem.

Block Error

*** error *** current block contains arrays belonging to other blocks

this message will be followed by a list of arrays and number of blocks. for example:

list of arrays	from blocks
11	i
3	ii
0	iii

Interpretation: this error usually results from the omission of a terminator at the end of one of the six input blocks. The position can be determined from the table. In the above example, there are 11 arrays in block i (1) and 3 arrays in block ii (2). This would indicate that the terminator was missing from block 1.

Keyword Error

* error * card #

listing of columns in card

listing of input from card

error columns xx-yy

* error * blank preceding an array identifier (=, $, or *)

Interpretation: this error usually results from an equals sign that does not immediately follow a keyword. Remove all spaces between a keyword and the equals sign and rerun the problem.

Block 1 Keyword Count Error

* error * no. of xxxxx in yyyyy .gt. zzzzz

the xxxxx are values in the yyyyy keyword which is greater than or less than the count given in block 1 under the zzzzz keyword. for example:

no. of zones in assign .gt. nzone

Interpretation: this error usually results from changes to keywords in Blocks 2, 3, or 4 without changing the associated count in Block 1. In the above example, either change the value for **nzone** in Block 1 or make sure that ids of material zones entered in Block 2 are less than or equal to the value for nzone.

Array Error

*** error ***

array name xxxxxx

array underloaded

count loaded = yy/count desired = zz.

Interpretation: this error usually results from a mismatch between the Block 1 counts and the entries for the array, xxxxxx, in Block 2, 3, 4, or 5. The count desired should match the value for the appropriate keyword in Block 1. Either add enough entries to the array or change the Block 1 count.

Illegal Character Error

* error * card #

listing of columns in card

listing of input from card

error columns xx-yy

* error * illegal character in numeric convert

Interpretation: this error usually results from a tab in the input file. As tabs are not printable, they will not be obvious, but DANTSYS will not accept them. The easiest solution is to retype the input line.

Memory Error

* error *

scm or lcm storage exceeded

needed xxxxxx , only had yyyyy

Interpretation: this error indicates that the value for either maxlcm or maxscm was too small. Increase the value to greater than that needed and rerun the problem.

Void Mesh Error

* error *

number per coarse mesh .le. 0

Interpretation: this error results from entering 0 fine meshes in a coarse mesh that contains a void. Although a void has no material in it, there must be at least one fine mesh in every coarse mesh region. So put a fine mesh in the void region and rerun the problem.

Isotope or Mixture Error

* error *

isotope or temp mixture doesn't exist

* isotope or mixture label

Interpretation: this error usually results from specifying an isotope that doesn't exist on the library or a typographical error in identifying a mixture in the assign statement.

The errors listed above are input errors which are fatal and will preclude DANTSYS from analyzing your problem. Sometimes, you will get runtime errors from DANTSYS which indicate problems with the solver process. The most common of these is the `neg c.m. delta` which may occur during a dimensional search. See Chapter 7 for a discussion on how to remedy this problem. In addition, Chapter 14 of the DANTSYS manual has a more detailed list of input and runtime errors and possible solutions. When in doubt, consult the manual.

APPENDIX B

Calculating Atom Densities

Most neutronics codes require that the user enter values to describe the atom densities of materials involved in the analysis. The problem facing the user is that many times the data supplied are in the form of weight percent, volume percent, solution density, density of individual constituents, etc. This data is not always directly compatible with the input requirements of a particular code. This appendix is designed to cover the majority of cases for atom density calculations with a multitude of different input specifications.

I. Single Constituent — Given: mass density

In many cases, the mass density of a single isotope, element, or molecular species is given. The basic equation for atom or molecular density is then:

$$N = \frac{\rho \cdot N_a}{A} \tag{B-1}$$

where: N = atom (or molecule) density (*atoms/cm*^{3})
ρ = density of constituent (*g/cm*^{3})
N_a = Avogadro's number = 0.6022×10^{24} (*atoms/mole*)
A = atomic or molecular weight of constituent (*g/mole*).

Example 1. Calculate the atom density of uranium-238 (U-238) for a nominal mass density of 19.1 g/cc.

Given: ρ = 19.1 g/cc $N_a = 0.6022 \times 10^{24}$ atoms/mole A = 238.05 g/mole

Then: $N_{u238} = 4.832 \times 10^{22}$ atoms/cc

The atom density of *U238* in the example is 4.832×10^{22} *atoms/cm*^{3}. These units are sufficient for some calculations, but the unit for microscopic cross sections is the *barn*, which is 10^{-24} *cm*^{2}. Many computer codes require atom densities to be entered in *atoms* per *barn-cm* because the cross sections are generally given in *barns (b)*. Multiplying the result by 10^{-24} *cm*^{2}*/barn* gives an atom density of *U238* of 4.832×10^{-2} *atom/barn-cm*. Rather than going through this step for each calculation, the rest of this Appendix will express Avogadro's number as 0.6022 *atoms-cm*^{2} per

mole-barn. This representation of Avogadro's number incorporates the proper units and directly gives values of atom density in *atom/barn-cm.*

II.a. Multiple Materials — Given: weight fractions and mixture density

When there is a mixture of materials with a known density and individual weight fractions, the atom density equation becomes:

$$N_i = \frac{\rho_{mix} \cdot wf_i \cdot N_a}{A_i} \tag{B-2}$$

where:
N_i = atom (or molecule) density of the ith material (*atoms/cm^3*)
ρ_{mix} = mass density of overall mixture (*g/cm^3*)
wf_i = weight fraction of the ith material
N_a = Avogadro's number = 0.6022 (*atoms-cm^2/mole-barn*)
A_i = atomic or molecular weight of the ith material (*g/mole*).

Example 2. Calculate the atom densities of U-238 and U-235 in 3 weight percent enriched uranium[a], U(3). The nominal mass density of the uranium is 18.9 g/cc.

Given: ρ_{mix} = 18.9 g/cc A_{238} = 238.05 g/mole A_{235} = 235.04 g/mole

wf_{238} = 0.97 g-238/g-mix wf_{235} = 0.03 g-235/g-mix

Then:

$$N_{238} = \frac{18.9\ \frac{\text{g-mix}}{\text{cc}} \cdot 0.97\ \frac{\text{g-238}}{\text{g-mix}} \cdot 0.6022\ \frac{\text{atom-sq.cm}}{\text{mole-b}}}{238.05\ \frac{\text{g-238}}{\text{mole}}} = 0.046377\ \frac{\text{atoms-238}}{\text{b-cm}}$$

In a similar way, the atom density of U-235 is calculated using equation (B-2) as:

N_{235} = 0.001453 atoms/b-cm

a. Weight percent of U-235 in uranium compounds is usually put in parentheses after the symbol for uranium, e.g., U(5) would indicate 5 weight percent U-235.

Note that although the example was done with only two materials, as long as the mixture density and individual weight fractions are given, the technique applies to any number of materials in a mixture.

Based on individual atom densities, the atom fractions of each constituent can be determined.

$$af_i = \frac{N_i}{N_1 + N_2 + \ldots + N_t} \tag{B-3}$$

where af_i = atom fraction of the ith material

So the atom fractions for *Example 2* are:

$$\text{Atom-fraction of U-235} = \frac{0.001453}{0.001453 + 0.046377} = 0.0303$$

$$\text{Atom-fraction of U-238} = \frac{0.046377}{0.001453 + 0.046377} = 0.9697$$

Because the atomic weights of the two isotopes are within a few percent of each other, there is little difference between the weight fractions and the atom fractions. However, as will be demonstrated with boron, this small difference is not always true as shown in Section IV.

II.b. Multiple Materials — Given: weight fractions and individual material densities

If the individual densities and weight fractions are known, then the mixture density is determined from:

$$\frac{1}{\rho_{mix}} = \frac{wf_1}{\rho_1} + \frac{wf_2}{\rho_2} + \ldots + \frac{wf}{\rho_n} \tag{B-4}$$

Example 3. Using the data of *Example 2*, calculate the density of the mixture assuming a density for U-235 of 18.6 g/cc and for U-238 of 18.9 g/cc.

Using equation (B-4):

$$\frac{1}{\rho_{mix}} = \frac{0.03}{18.6} + \frac{0.97}{18.9} \qquad \rho_{mix} = 18.89 \ \text{g/cc}$$

Once the mixture density is known, the atom densities are then calculated using equation (B-2).

III. Multiple Materials — Given: atom fractions and atom mixture density

Although weight fractions are generally used for enrichments, atom fractions are given in publications such as the *Chart of the Nuclides* or the *CRC Handbook of Chemistry and Physics*. To use atom fractions, an average atomic weight must be determined.

$$\bar{A} = af_1 \cdot A_1 + af_2 \cdot A_2 + \ldots + af_n \cdot A_n \tag{B-5}$$

The average atomic weight as calculated in equation (B-5), is used in the calculation of the mixture atom density.

$$N_{mix} = \frac{\rho_{mix} \cdot N_a}{\bar{A}} \tag{B-6}$$

After the mixture atom density is known, then the individual constituent atom densities are given by:

$$N_i = af_i \cdot N_{mix} \tag{B-7}$$

Example 4. According to the *Chart of the Nuclides*, natural boron has a density of 2.34 g/cm^3 with atom fractions of 0.199 for ^{10}B and 0.801 for ^{11}B. What are the atom densities for the two isotopes?

Given: ρ_{Bnat} = 2.34 g/cc A_{B10} = 10.01 g/mole A_{B11} = 11.01 g/mole

af_{B10} = 0.199 af_{B11} = 0.801

Then calculate the average atomic weight of natural boron using equation (B-5):

$$\bar{A}_{Bnat} = 0.199 \cdot 10.01 + 0.801 \cdot 11.01 = 10.81 \text{ g-Bnat / mole}$$

Now use equation (B-6) to calculate the atom density of natural boron:

$$N_{mix} = N_{Bnat} = \frac{2.34 \ \frac{\text{g-Bnat}}{\text{cc}} \cdot 0.6022 \ \frac{\text{atom-sq.cm}}{\text{mole-b}}}{10.81 \ \frac{\text{g-Bnat}}{\text{mole}}} = 0.0259 \ \frac{\text{atoms-Bnat}}{\text{b-cm}}$$

Because we know what fraction of the natural boron atoms are ^{10}B atoms (19.9%), we can calculate the atom density of ^{10}B in natural boron.

$$N_{B10} = 0.199 \ \frac{\text{atom-B10}}{\text{atom-Bnat}} \cdot 0.1304 \ \frac{\text{atom-Bnat}}{\text{b-cm}} = 0.0259 \ \frac{\text{atoms-B10}}{\text{b-cm}}$$

In a similar way, the atom density of ^{11}B is calculated:

$$N_{B11} = 0.801 \ \frac{\text{atom-B11}}{\text{atom-Bnat}} \cdot 0.1304 \ \frac{\text{atom-Bnat}}{\text{b-cm}} = 0.1045 \ \frac{\text{atoms-B11}}{\text{b-cm}}$$

Note that $N_{B10} + N_{B11} = N_{Bnat}$, which it should.

IV. Calculating weight fractions, atom fractions, and average atomic weight when one set of fractions is known

In the previous section, equation (B-5) showed how to calculate the average atomic weight, when atom fractions are known. f, however, weight fractions are given, then a different equation is used to calculate average atomic weight.

$$\bar{A} = \left[\frac{wf_1}{A_1} + \frac{wf_2}{A_2} + \ldots + \frac{wf_n}{A_n}\right]^{-1} \tag{B-8}$$

Knowing the weight fractions and the average atomic weight, one can then calculate the atom fractions using the following equation:

$$af_i = wf_i \cdot \frac{\bar{A}}{A_i} \tag{B-9}$$

Or if you know the atom fractions, calculate the weight fractions as:

$$wf_i = af_i \cdot \frac{A_i}{\bar{A}} \tag{B-10}$$

Example 5. Continue with *Example 4* and calculate the weight fractions for ^{10}B and for ^{11}B in natural boron.

Given: $\rho_{Bnat} = 2.34$ g/cc $A_{B10} = 10.01$ g/mole $A_{B11} = 11.01$ g/mole

$af_{B10} = 0.199$ $af_{B11} = 0.801$

From *Example 4*, we know the average atomic weight of natural boron is 10.81 g/cc. Using equation (B-9), we find the weight fractions as:

$$wf_{B10} = 0.199 \cdot \frac{10.01}{10.81} = 0.184 \ \frac{\text{g-B10}}{\text{g-Bnat}}$$

$$wf_{B11} = 0.801 \cdot \frac{11.01}{10.81} = 0.816 \ \frac{\text{g-B11}}{\text{g-Bnat}}$$

As indicated earlier, there is a significant difference between weight and atom fractions for *B10* and *B11* in natural boron. The weight fraction of *B10* is 18.4% while the atom fraction of *B10* is 19.9%.

V.a. Molecules — Given: chemical structure and mass density

Determination of atom densities for constituents of a molecule is similar to the calculation done when the atom fractions are known. In this case, the atom "fractions" are usually greater than or equal to 1 and represent the number of atoms of a particular type per molecule.

Example 6. Determine the atom densities of hydrogen and oxygen in water with a density of 1.0 g/cc.

Given: $\rho_{water} = 1.0$ g/cc $A_{water} = 18.0$ g/mole

$atoms_H = 2$ per molecule $atoms_O = 1$ per molecule

First we calculate the molecular density of water using equation (B-1).

$$N_{water} = \frac{1.0\ \frac{g}{cc} \cdot 0.6022\ \frac{\text{atom-sq.cm}}{\text{mole-b}}}{18\ \frac{g}{mole}} = 3.34\times10^{-2}\ \frac{\text{molecules-water}}{\text{b-cm}}$$

In water, there are 2 atoms of hydrogen and 1 atom of oxygen for every molecule.

$$N_H = 2\times N_{water} = 6.68\times10^{-2}\ \frac{\text{atoms-H}}{\text{b-cm}} \qquad N_O = 1\times N_{water} = 3.34\times10^{-2}\ \frac{\text{atoms-O}}{\text{b-cm}}$$

V.b. Molecules with Mixtures of Isotopes — Given: chemical structure and mass density

In example 6, it was assumed that all hydrogen was ^{1}H and all oxygen was ^{16}O. However, for many materials encountered in criticality safety, the isotopic content is very important (e.g., boron and uranium).

Example 7. Determine the atom densities of ^{10}B, ^{11}B, and ^{12}C in boron carbide (B_4C) assuming the boron is natural boron and the mixture density is 2.54 g/cc.

Given: $\rho_{Bnat} = 2.34$ g/cc $A_{B10} = 10.01$ g/mole $A_{B11} = 11.01$ g/mole

$A_C = 12.0$ g/cmole $af_{B10} = 0.199$ $af_{B11} = 0.801$

From *Example 4*, we know the average atomic weight of natural boron is 10.81 g/cc. The molecular weight of B_4C can be calculated using equation (B-4).

$$A_{B4C} = \frac{4\ \text{moles-Bnat}}{\text{mole-B4C}} \times \frac{10.81\ g}{\text{mole-Bnat}} + \frac{1\ \text{mole-C}}{\text{mole-B4C}} \times \frac{12.00\ g}{\text{mole-Bnat}} = 55.24\ \frac{g}{mole}$$

Now use equation (B-1) to calculate the molecular density of B_4C.

$$N_{B4C} = \frac{2.54\ \frac{g}{cc} \cdot 0.6022\ \frac{\text{molecules-sq.cm}}{\text{mole-b}}}{55.24\ \frac{g}{mole}} = 2.77\times10^{-2}\ \text{molecules-B4C}$$

Because there are 4 atoms of natural boron per molecule of boron carbide, the atom density of natural boron is:

$$N_{Bnat} = 4 \times N_{B4C} = 1.108\times10^{-1} \ \frac{\text{atoms-Bnat}}{\text{b-cm}}$$

Using equation (B-7), the atom densities of ^{10}B and ^{11}B can be determined.

$$N_{B10} = af_{B10} \cdot N_{Bnat} = 0.199 \cdot 1.108\times10^{-1} = 2.205\times10^{-2} \ \frac{\text{atoms-B10}}{\text{b-cm}}$$

$$N_{B11} = af_{B11} \cdot N_{Bnat} = 0.801 \cdot 1.108\times10^{-1} = 8.875\times10^{-2} \ \frac{\text{atoms-B11}}{\text{b-cm}}$$

These are the atom densities of ^{10}B and ^{11}B in boron carbide. Now we, need to calculate the atom density of carbon in the mixture.

$$N_C = 1 \cdot N_{B4C} = 2.77\times10^{-2} \ \frac{\text{atoms-C}}{\text{b-cm}}$$

Note on atom fractions and weight fractions

In mixtures, it is important to know whether the weight or atom fractions are relative to the entire mixture or just to some constituent part of the mixture. In the example above, the atom fractions for B10 and B11 were relative to the natural boron and not to the B_4C. As another example, in $U(20)O_2$, the 0.20 weight fraction of U235 is relative to the uranium, not to the UO_2, so it can only be applied to the uranium. However, in a system of plutonium mixed with 1 weight percent gallium, the weight percent of gallium is referenced to the entire mixture so it would be applied to the mixture.

For materials where the atom fractions are known, determination of the individual atom densities is straight forward. However, for those cases where the weight fractions are known, the atom fractions are first calculated from Equation (B-9) and then used to determine atom densities, as above.

Example 8. Determine the atom densities of the individual constituents in $U(20)O_2$ whose density is 10.5 g/cc.

First, determine the average atomic weight of uranium dioxide with 20 weight percent U-235. Start by calculating the average atomic weight of U using equation (B-8).

$$\overline{A_U} = \left[\frac{0.20}{235.04} + \frac{0.80}{238.05}\right]^{-1} = 237.44 \ \frac{\text{g-U(20)}}{\text{cc}}$$

Now calculate the molecular weight of uranium dioxide:

$$\bar{A} = 237.44 + 2 \cdot 16 = 269.44 \ \frac{\text{g-UO2}}{\text{cc}}$$

Now determine the atom density of UO_2.

$$N_{UO2} = \frac{10.5\frac{\text{g-UO2}}{\text{cc}} \cdot 0.6022\frac{\text{molecules-sq.cm}}{\text{mole-b}}}{269.44\frac{\text{g-UO2}}{\text{mole}}} = 2.35\times10^{-2} \ \frac{\text{molecules-UO2}}{\text{b-cm}}$$

There is 1 atom of U and 2 atoms of O for each molecule of UO_2:

$$N_U = 1 \cdot N_{UO2} = 2.35\times10^{-2} \ \frac{\text{atoms-U}}{\text{b-cm}}$$

$$N_O = 2 \cdot N_{UO2} = 4.70\times10^{-2} \ \frac{\text{atoms-O}}{\text{b-cm}}$$

But 20 percent by weight of the uranium is U-235 while the other 80 weight percent is U-238. So we will use the weight fractions to calculate uranium atom fractions from equation (B-9).

$$af_{U235} = wf_{U235} \cdot \frac{\bar{A}_{U(20)}}{A_{U235}} = 0.20 \cdot \frac{237.44}{235.04} = 0.202$$

$$af_{U238} = wf_{U238} \cdot \frac{\bar{A}_{U(20)}}{A_{U238}} = 0.80 \cdot \frac{237.44}{238.05} = 0.798$$

Then calculate the atom densities based on the weight fractions.

$$N_{U235} = af_{U235} \cdot N_{U(20)} = 0.202 \cdot 2.35\times10^{-2} = 4.75\times10^{-3} \ \frac{\text{atoms-U235}}{\text{b-cm}}$$

$$N_{U238} = af_{U238} \cdot N_{U(20)} = 0.798 \cdot 2.35\times10^{-2} = 1.875\times10^{-2} \ \frac{\text{atoms-U238}}{\text{b-cm}}$$

This concludes the formulas and examples needed for non-solution systems.

VI. Solution Systems

Because solution systems have a number of parameters (e.g., solution density, molality, normality, single constituent density, H/U ratio, H/Pu ration, H/X ratio) that can be used to characterize them, calculation of atom densities in solution is usually more complex than for solid systems. If the solution density is given, then the atom densities are calculated using the technique given in Section V.a. (where water is used as the example).

VI.a. Solutions — Given: H/X ratio, fissile component density, and chemical formula

The H/X ratio (ratio of hydrogen atoms to fissile atoms) is often used in criticality studies to indicate the amount of moderation in a system. When the fissile mass density is provided along with H/X, all of the atom densities can be determined.

Example 9. In the uranyl fluoride example from Chapter 6 of the Primer, we have $U(4.89)O_2F_2$ in solution with water; H/X=524 and the U-235 density is 0.0425 g-U235/cc.

First, calculate the fissile atom density.

$$N_{U235} = \frac{\rho_{U235} \cdot N_a}{A_{U235}} = \frac{0.0425 \ \frac{\text{g-U235}}{\text{cc-mix}} \cdot 0.6022 \ \frac{\text{atoms-sq.cm}}{\text{mole-b}}}{235.04 \ \frac{\text{g-U235}}{\text{mole}}}$$

$$N_{U235} = 1.0889\times10^{-4} \ \frac{\text{atoms-U235}}{\text{b-cm}}$$

We know the weight fractions of U-235 and U-238, but we want the atom fractions. To get these values we need to use equation (B-8) to calculate the average atomic weight of the (U-235 & U-238) mixture.

Given: $wf_{U235} = 0.0489$ g-U235 / g-U

$wf_{U238} = (1-0.0489) = 0.9511$ g-U238 / gU

$$\bar{A}_U = \left[\frac{0.0489}{235.04} + \frac{0.9511}{238.05}\right]^{-1} = 237.90 \ \frac{\text{g-U(4.89)}}{\text{cc}}$$

Now with this atomic weight, we can use equation (B-9) to calculate the atom fractions.

$$af_{235} = wf_{235} \cdot \frac{\bar{A}}{A_{235}} = 0.0489 \cdot \frac{237.90}{235.04} = 0.04949 \ \frac{\text{atoms-U235}}{\text{atoms-U}}$$

$$af_{238} = wf_{238} \cdot \frac{\bar{A}}{A_{238}} = 0.9511 \cdot \frac{237.90}{238.05} = 0.95051 \ \frac{\text{atoms-U238}}{\text{atoms-U}}$$

The uranium atom density is determined by dividing the U-235 atom density by the U-235 atom fraction.

$$N_U = \frac{N_{235}}{af_{235}} = \frac{1.0889\times10^{-4} \ \frac{\text{atoms-U235}}{\text{b-cm}}}{0.04949 \ \frac{\text{atoms-U235}}{\text{atoms-U}}} = 2.20024\times10^{-3} \ \frac{\text{atoms-U}}{\text{b-cm}}$$

Once the uranium atom density is known, the U-238 atom density can be calculated.

$$N_{238} = af_{238} \cdot N_U = 0.95051 \ \frac{\text{atoms-U238}}{\text{atoms-U}} \cdot 2.20024\times10^{-3} \ \frac{\text{atoms-U}}{\text{b-cm}}$$

$$N_{238} = 2.09135\times10^{-3} \ \frac{\text{atoms-U238}}{\text{b-cm}}$$

Knowing that there are 2 oxygen atoms and 2 fluorine atoms per uranium atom, the atom densities of these two materials in the uranyl fluoride can then be determined.

$$N_O = 2 \cdot N_U = 4.40048\times10^{-3} \ \frac{\text{atoms-O}}{\text{b-cm}}$$

$$N_F = 2 \cdot N_U = 4.40048\times10^{-3} \ \frac{\text{atoms-F}}{\text{b-cm}}$$

With uranyl fluoride solutions, there is a substantial amount of water so we need to calculate the atom densities of the hydrogen and the oxygen. This is done using the H/X ratio.

$$N_H = \frac{H}{X} \cdot N_{235} = 524 \cdot 1.0889\times10^{-4} = 5.705836\times10^{-2} \ \frac{\text{atoms-H}}{\text{b-cm}}$$

Then the atom density of the oxygen in water is determined knowing that there are 2 hydrogen atoms for each oxygen atom.

$$N_O = N_H \cdot \frac{1 \text{ atom-O}}{2 \text{ atoms-H}} = 2.852918\times10^{-2} \ \frac{\text{atoms-O}}{\text{b-cm}}$$

The total atom density for oxygen is the sum of its atom density in UO_2F_2 and its atom density in H_2O.

$$N_O = 4.40048\times10^{-3} \ \frac{\text{atoms-O}}{\text{b-cm}} + 2.852918\times10^{-2} \ \frac{\text{atoms-O}}{\text{b-cm}} = 3.292966\times10^{-2} \ \frac{\text{atoms-O}}{\text{b-cm}}$$

The final values, in atoms/b-cm, that were used in Chapter 6 are:

N_{235} = 1.0889×10^{-4}

N_{238} = 2.09135×10^{-3}

N_{H} = 5.705836×10^{-2}

N_{O} = 3.292966×10^{-2}

N_{F} = 4.40048×10^{-3}

APPENDIX C

Specifications & Atom Densities Of Selected Materials

R. D. O'Dell
Criticality Safety Group
Los Alamos National Laboratory
April 1994

The following material compositions, specifications, and atom densities have been compiled by the above author from various sources. They are provided for the convenience of users of computer codes for nuclear analysis. No warranty is made nor is any legal liability or responsibility assumed for the accuracy, completeness, or usefulness of the following information. Reference to any specific commercial product by trade name or registered trademark does not necessarily constitute or imply its endorsement.

Atomic weights used in the following were taken from "Nuclides and Isotopes, Fourteenth Edition," General Electric Company, San Jose, California.

Atom densities are given in units of atoms/barn-cm.

BISCO® modified NS-4 with 4.5% Boron

Los Alamos National Laboratory Analysis

Density = 1.119 g/cc

Nuclide	Wt. Frac.	Atom Dens.
C	0.4771	0.02677
N	0.0356	0.00171
H	0.0787	0.05262
Si	0.0018	0.00004
Na	0.0494	0.00145
B(nat)	0.0455	0.00284
O	0.3119	0.01314

BORON CARBIDE (Natural Boron): B_4C

Density = 2.51 g/cc

A= 55.2570

Nuclide	Wt. Frac.	Atom Dens.
B(nat)	0.7826	0.10941
C	0.2174	0.027359

CELOTEX® (Lignocellulosic Fiberboard)*

Normal density = 16 2 lb/ft^3

Celotex ~ $C_6H_{10}O_5$ (cellulose)

A = 162.143

Atom Densities as Function of Celotex Density

Density(g/cc)	Density (lb/ft3)	C	H	O
0.31	19.34	6.908-3	1.1514-2	5.757-3
0.30	18.71	6.685-3	1.1142-2	5.571-3
0.29	18.09	6.463-3	1.0771-2	5.385-3
0.2886	18.0	6.431-3	1.0719-2	5.359-3
0.28	17.46	6.240-3	1.0400-2	5.200-3
0.27	16.84	6.017-3	1.0028-2	5.014-3
0.26	16.22	5.794-3	9.657-3	4.828-3
0.2565	16.0	5.716-3	9.527-3	4.763-3
0.25	15.59	5.571-3	9.285-3	4.643-3
0.24	14.97	5.348-3	8.914-3	4.457-3
0.23	14.35	5.125-3	8.542-3	4.271-3
0.2245	14.0	5.003-3	8.338-3	4.169-3
0.22	13.72	4.903-3	8.171-3	4.086-3
0.21	13.10	4.680-3	7.800-3	3.900 3
0.20	12.47	4.457-3	7.428-3	3.714-3
0.19	11.85	4.234-3	7.057-3	3.528-3
0.18	11.23	4.011-3	6.685-3	3.343-3

Ref: ASTM C-208, "Standard Specification for Insulating Board (Cellulosic Fiber), Structural and Decorative"

CONCRETE [KENO Regular Concrete Standard Mix]

Density = 2.3 g/cc

Nuclide	Wt. Frac.	Atom Dens.
H	0.010	0.01374
O	0.532	0.04606
Si	0.337	0.01662
Al	0.034	0.00175
Na	0.029	0.00175
Ca	0.044	0.00152
Fe	0.014	0.00035

CONCRETE [LOS ALAMOS (MCNP) Mix]

Density = 2.25 g/cc

Nuclide	Wt. Frac.	Atom Dens.
H	0.00453	0.006094
O	0.51260	0.043421
Si	0.36036	0.017390
Al	0.03555	0.001786
Na	0.01527	0.000900
Ca	0.05791	0.001958
Fe	0.01378	0.000334

CONCRETE [NBS Ordinary]

Density = 2.35 g/cc

Nuclide	NBS Wt. Frac	Adj.* Wt. Frac.	Adj. Atom Dens.
H	0.0056	0.006	0.00842
O	0.4956	0.500	0.04423
Si	0.3135	0.315	0.01587
Al	0.0456	0.048	0.00252
Na	0.0171	0.017	0.00105
Ca	0.0826	0.083	0.00293
Fe	0.0122	0.012	0.00030
K	0.0192	0.019	0.00069
Mg	0.0024	-	
S	0.0012	-	

* adjusted to sum to unity without minor trace elements

GYPSUM (Calcium Sulfate}

$CaSO4 \bullet 2H_2O$

Density = 2.32 g/cc

A = 172.17

Nuclide	Atom Density
Ca	0.008115
S	0.008115
O	0.048689
H	0.032460

INCONEL [KENO Standard Mix]

Density = 8.3 g/cc

Nuclide	Wt. Frac.	Atom Dens.
Si	0.025	0.00445
Ti	0.025	0.00261
Cr	0.15	0.01442
Fe	0.07	0.00626
Ni	0.73	0.06217

INCONEL X [Simplified]

Density = 8.5 g/cc

Nuclide	Wt. Frac.	Atom Dens.
Ni	0.78	0.0680
Cr	0.15	0.0148
Fe	0.07	0.0064

KYNAR®: $C_2H_2F_2$,

Density = 1.76 g/cc

A= 64.0347

Nuclide	Wt. Frac.	Atom Dens.
C	0.3751	0.0331
H	0.0315	0.0331
F	0.5934	0.0331

LEXAN®: $C_{16}H_{14}O_3$

Los Alamos National Laboratory Analysis

Density = 1.20 g/cc

A = 254.2855

Nuclide	Wt. Frac.	Atom Dens.
C	0.755749	0.045471
H	0.055494	0.039787
O	0.188757	0.008527

MAGNESIUM OXIDE: MgO

Density = 3.22 g/cc

A = 40.3044

Nuclide	Wt. Frac.	Atom Dens.
Mg	0.6030	0.04811
O	0.3970	0.04811

NYLON®: $C_{12}H_{22}N_2O_2$,

Density = 1.14 g/cc

A = 226.319

Nuclide	Wt. Frac.	Atom Dens.
C	0.63685	0.036401
H	0.09798	0.066737
N	0.12378	0.006067
O	0.14139	0.006067

PARAFFIN: $C_{25}H_{52}$

Density = 0.93 g/cc

A = 352.688

Nuclide	Wt. Frac.	Atom Dens.
C	0.8514	0.03970
H	0.1486	0.08257

PLEXIGLAS® & LUCITE®: $C_5H_8O_2$,

Density = 1.18 g/cc

A = 100.117

Nuclide	Wt. Frac.	Atom Dens.
C	0.59985	0.03549
H	0.08054	0.05678
O	0.31961	0.01420

POLYETHYLENE: CH_2

Density = 0.92 g/cc

A = 14.0269

Nuclide	Wt. Frac.	Atom Dens.
C	0.85628	0.03950
H	0.14372	0.07899

POLYURETHANE FOAM

Los Alamos National Laboratory Analysis

Density = 0.021 g/cc

Nuclide	Wt. Frac.	Atom Dens.
H	0.041	5.144E-4
C	0.544	5.728E-4
N	0.121	10.925E-4
O	0.294	2.324E-4

POLYVINYL CHLORIDE (PVC): C_2H_3Cl

Density = 1.65 g/cc

Nuclide	Wt. Frac.	Atom Dens.
C	0.3844	0.03180
H	0.0484	0.04771
Cl	0.5672	0.01590

PYREX® (Borated Glass), [KENO Standard Mix]

Density = 2.23 g/cc

Nuclide	Wt. Frac.	Atom Dens.
B(nat)	0.037	0.00460
Al	0.010	0.00050
Na	0.041	0.00240
O	0.535	0.04491
Si	0.377	0.01803

SILICON RUBBER [G.E. RTV12A]

(Weight fractions below provided by G.E. Silicone Products Div.)

Density = 1.0185 g/cc

Nuclide	Wt. Frac.	Atom Dens.
C	0.3211	0.01640
Si	0.3745	0.00818
O	0.2235	0.00857
H	0.0807	0.04911

STEEL, CARBON

Density = 7.82 g/cc

Nuclide	Wt. Frac.	Atom Dens.
C	0.005	0.00196
Fe	0.995	0.08390

STAINLESS STEEL 304

Density = 7.92 g/cc

Nuclide	Wt. Frac.	Atom Dens.
Fe	0.695	0.05936
Cr	0.190	0.01743
Ni	0.095	0.00772
Mn	0.020	0.00174

STAINLESS STEEL 316

Density = 7.92 g/cc

Nuclide	Wt. Frac.	Atom Dens.
Fe	0.655	0.05594
Cr	0.170	0.01559
Ni	0.120	0.00975
Mo	0.025	0.00124
Mn	0.020	0.00174
Si	0.010	0.00170

STAINLESS STEEL 347

Density = 7.92 g/cc

Nuclide	Wt. Frac.	Atom Dens.
Fe	0.685	0.05850
Cr	0.180	0.01651
Ni	0.105	0.00853
Mn	0.020	0.00174
Si	0.010	0.00170

TEFLON®: CF_2

Los Alamos National Laboratory Analysis

Density = 2.15 - 2.20 g/cc

A = 50.0078

Nuclide	Wt. Frac.	Atom Dens.
C	0.2402	0.02650
F	0.7598	0.05298

APPENDIX D

The 167 Isotope Hansen-Roach Cross-Section Library

Criticality Safety Group
Los Alamos National Laboratory
January 1994

A 167 isotope cross-section library has been developed for use with KENO and DANTSYS. The following table gives the isotopes and their respective identification for both of the codes for which their use is intended. This information is provided for the convenience of users of computer codes for nuclear analysis. No warranty is made nor is any legal liability or responsibility assumed for the accuracy, completeness, or usefulness of the following information.

Number	TWODANT ID	KENO V.a ID	Description
1	hDE	1100	HR Hydrogen – p1 – LAMS 2941
2	hCHI	1101	OR Hydrogen – p1 – Origin Unknown
3	dDE	1200	HR Deuterium – p1 – LAMS 2941
4	dCHI	1201	HR Deuterium – p1 – LAMS 2941
5	dLA	1202	LA Deuterium – p1 – Origin Unknown
6	li6	3100	HR Lithium 6 – p0 – LAMS 2941
7	li7	3200	HR Lithium 7 – p0 – LAMS 2941
8	be	4100	HR Beryllium – p0 – LAMS 2941
9	b	5100	HR Natural Boron – p0 – LAMS 2941
10	b10	5101	OR Boron 10 – p1 – ENDF/B-IV
11	b11	5102	OR Boron 11 – p1 – ENDF/B-IV
12	c	6100	HR Carbon – p0 – LAMS 2941
13	n	7100	HR Nitrogen – p0 – LAMS 2941
14	o	8100	HR Oxygen – p0 – LAMS 2941
15	f	9100	HR Fluorine – p0 – Origin Unknown
16	na	11100	HR Sodium – p0 – LAMS 2941
17	mgOR	12100	OR Magnesium – p1 – XSDRN
18	mg	12101	HR Magnesium – p0 – Origin Unknown
19	al	13100	HR Aluminum – p0 – LAMS 2941
20	siLA	14100	LA Silicon – p0 – Origin Unknown
21	siOR	14101	OR Silicon – p1 – XSDRN
22	sOR	16100	OR Sulfur – p1 – XSDRN
23	cl	17100	HR Chlorine – p0 – LAMS 2941
24	k	19100	HR Potassium – p0 – LAMS 2941
25	ca	20100	HR Calcium – p0 – Origin Unknown
26	caOR	20101	OR Calcium – p1 – GAM 2

Number	TWODANT ID	KENO V.a ID	Description
27	ti	22100	HR Titanium – p0 – Origin Unknown
28	tiOR	22101	OR Titanium – p0 – GAM 2
29	v	23100	OR Vanadium – p0 – GAM 2
30	cr	24100	HR Chromium – p0 – LAMS 2941
31	mn	25100	OR Manganese – p1 – GAM 2
32	fe	26100	HR Iron – p0 – LAMS 2941
33	co	27100	HR Cobalt – p0 – LAMS 2941
34	ni	28100	HR Nickel – p0 – LAMS 2941
35	cu	29100	HR Copper – p0 – Origin Unknown
36	cuOR	29101	OR Copper – p1 – XSDRN
37	zn	30100	OR Zinc – p0 – GAM 2
38	ga	31100	HR Gallium – p0 – LAMS 2941
39	zr	40100	HR Zirconium – p0 – LAMS 2941
40	nb	41100	HR Niobium – p0 – LAMS 2941
41	mo	42100	HR Molybdenum – p0 – LAMS 2941
42	cd	48100	HR Cadmium – p0 – LAMS 2941
43	cdOR	48101	OR Cadmium – p0 – GAM 2
44	in	49100	OR Indium – p0 – GAM 2
45	ce	58100	HR Cerium – p0 – LAMS 2941
46	sm	62100	OR Samarium – p0 – GAM 2
47	eu	63100	OR Europium – p0 – GAM 2
48	gd	64100	OR Gadolinium – p0 – GAM 2
49	ta	73100	HR Tantalum – p0 – LAMS 2941
50	w	74100	HR Tungsten – p0 – Origin Unknown
51	wOR	74101	OR Tungsten – p0 – GAM 2
52	pb	82100	HR Lead – p0 – Origin Unknown
53	pbOR	82101	OR Lead – p1 – XSDRN

Number	TWODANT ID	KENO V.a ID	Description
54	th022e1	90200	HR Thorium 232 – p0 – LAMS 2543
55	th024e1	90201	HR Thorium 232 – p0 – LAMS 2543
56	th025e1	90202	HR Thorium 232 – p0 – LAMS 2543
57	th026e1	90203	HR Thorium 232 – p0 – LAMS 2543
58	th028e1	90204	HR Thorium 232 – p0 – LAMS 2543
59	th021e2	90205	HR Thorium 232 – p0 – LAMS 2543
60	th022e2	90206	HR Thorium 232 – p0 – LAMS 2543
61	th023e2	90207	HR Thorium 232 – p0 – LAMS 2543
62	th024e2	90208	HR Thorium 232 – p0 – LAMS 2543
63	th026e2	90209	HR Thorium 232 – p0 – LAMS 2543
64	th028e2	90210	HR Thorium 232 – p0 – LAMS 2543
65	th021e3	90211	HR Thorium 232 – p0 – LAMS 2543
66	th022e3	90212	HR Thorium 232 – p0 – LAMS 2543
67	th023e3	90213	HR Thorium 232 – p0 – LAMS 2543
68	th024e3	90214	HR Thorium 232 – p0 – LAMS 2543
69	th026e3	90215	HR Thorium 232 – p0 – LAMS 2543
70	th021e4	90216	HR Thorium 232 – p0 – LAMS 2543
71	th022e4	90217	HR Thorium 232 – p0 – LAMS 2543
72	th02inf	90218	HR Thorium 232 – p0 – LAMS 2543
73	u232e1	92300	HR Uranium 233 – p0 – LAMS 2941
74	u234e1	92301	HR Uranium 233 – p0 – LAMS 2941
75	u236e1	92302	HR Uranium 233 – p0 – LAMS 2941
76	u231e2	92303	HR Uranium 233 – p0 – LAMS 2941
77	u232e2	92304	HR Uranium 233 – p0 – LAMS 2941
78	u234e2	92305	HR Uranium 233 – p0 – LAMS 2941
79	u236e2	92306	HR Uranium 233 – p0 – LAMS 2941
80	u231e3	92307	HR Uranium 233 – p0 – LAMS 2941

Number	TWODANT ID	KENO V.a ID	Description
81	u232e3	92308	HR Uranium 233 – p0 – LAMS 2941
82	u234e3	92309	HR Uranium 233 – p0 – LAMS 2941
83	u236e3	92310	HR Uranium 233 – p0 – LAMS 2941
84	u231e4	92311	HR Uranium 233 – p0 – LAMS 2941
85	u23inf	92312	HR Uranium 233 – p0 – LAMS 2941
86	u24OR1	92400	OR Uranium 234 – o1 – ENDF/B-IV
87	u24OR2	92401	OR Uranium 234 – p0 – Mihalczo Modification
88	u252e1	92500	UR Uraniuim 235 – p0 – LAMS 2941
89	u254e1	92501	HR Uraniuim 235 – p0 – LAMS 2941
90	u256e1	92502	HR Uraniuim 235 – p0 – LAMS 2941
91	u251e2	92503	HR Uraniuim 235 – p0 – LAMS 2941
92	u252e2	92506	HR Uraniuim 235 – p0 – LAMS 2941
93	u254e2	92505	HR Uraniuim 235 – p0 – LAMS 2941
94	u256e2	92506	HR Uraniuim 235 – p0 – LAMS 2941
95	u251e3	92507	HR Uraniuim 235 – p0 – LAMS 2941
96	u252e3	92508	HR Uraniuim 235 – p0 – LAMS 2941
97	u254e3	92509	HR Uraniuim 235 – p0 – LAMS 2941
98	u256e3	92510	HR Uraniuim 235 – p0 – LAMS 2941
99	u251e4	92511	HR Uraniuim 235 – p0 – LAMS 2941
100	u25inf	92512	HR Uraniuim 235 – p0 – LAMS 2941
101	u26OR1	92600	OR Uranium 236 – p1 – ENDF/B-IV
102	u26OR2	92601	OR Uranium 236 – p0 – Mihalczo Modification
103	u280e0	92800	HR Uranium 238 – p0 – LAMS 2941
104	u282e1	92801	HR Uranium 238 – p0 – LAMS 2941
105	u284e1	92802	HR Uranium 238 – p0 – LAMS 2941
106	u286e1	92803	HR Uranium 238 – p0 – LAMS 2941
107	u281e2	92804	HR Uranium 238 – p0 – LAMS 2941

Number	TWODANT ID	KENO V.a ID	Description
108	u282e2	92805	HR Uranium 238 – p0 – LAMS 2941
109	u284e2	92806	HR Uranium 238 – p0 – LAMS 2941
110	u286e2	92807	HR Uranium 238 – p0 – LAMS 2941
111	u281e3	92808	HR Uranium 238 – p0 – LAMS 2941
112	u282e3	92809	HR Uranium 238 – p0 – LAMS 2941
113	u284e3	92810	HR Uranium 238 – p0 – LAMS 2941
114	u286e3	92811	HR Uranium 238 – p0 – LAMS 2941
115	u281e4	92812	HR Uranium 238 – p0 – LAMS 2941
116	u282e4	92813	HR Uranium 238 – p0 – LAMS 2941
117	u284e4	92814	HR Uranium 238 – p0 – LAMS 2941
118	u286e4	92815	HR Uranium 238 – p0 – LAMS 2941
119	u281e5	92816	HR Uranium 238 – p0 – LAMS 2941
120	u28inf	92817	HR Uranium 238 – p0 – LAMS 2941
121	pu481e2	94800	OR Plutonium 238 – p0 – PERSIMMON
122	pu481e3	94801	OR Plutonium 238 – p0 – PERSIMMON
123	pu481e4	94802	OR Plutonium 238 – p0 – PERSIMMON
124	pu48inf	94803	OR Plutonium 238 – p0 – PERSIMMON
125	pu492e1	94900	HR Plutonium 239 – p0 – LAMS 2941
126	pu494e1	94901	HR Plutonium 239 – p0 – LAMS 2941
127	pu496e1	94902	HR Plutonium 239 – p0 – LAMS 2941
128	pu491e2	94903	HR Plutonium 239 – p0 – LAMS 2941
129	pu492e2	94904	HR Plutonium 239 – p0 – LAMS 2941
130	pu494e2	94905	HR Plutonium 239 – p0 – LAMS 2941
131	pu496e2	94906	HR Plutonium 239 – p0 – LAMS 2941
132	pu491e3	94907	HR Plutonium 239 – p0 – LAMS 2941
133	pu492e3	94908	HR Plutonium 239 – p0 – LAMS 2941
134	pu494e3	94909	HR Plutonium 239 – p0 – LAMS 2941

Number	TWODANT ID	KENO V.a ID	Description
135	pu496e3	94910	HR Plutonium 239 – p0 – LAMS 2941
136	pu491e4	94911	HR Plutonium 239 – p0 – LAMS 2941
137	pu492e4	94912	HR Plutonium 239 – p0 – LAMS 2941
138	pu494e4	94913	HR Plutonium 239 – p0 – LAMS 2941
139	pu496e4	94914	HR Plutonium 239 – p0 – LAMS 2941
140	pu491e5	94915	HR Plutonium 239 – p0 – LAMS 2941
141	pu49inf	94916	HR Plutonium 239 – p0 – LAMS 2941
142	pu405e1	94000	HR Plutonium 240 – p0 – LAMS 2941
143	pu401e2	94001	HR Plutonium 240 – p0 – LAMS 2941
144	pu402e2	94002	HR Plutonium 240 – p0 – LAMS 2941
145	pu404e2	94003	HR Plutonium 240 – p0 – LAMS 2941
146	pu406e2	94004	HR Plutonium 240 – p0 – LAMS 2941
147	pu401e3	94005	HR Plutonium 240 – p0 – LAMS 2941
148	pu402e3	94006	HR Plutonium 240 – p0 – LAMS 2941
149	pu404e3	94007	HR Plutonium 240 – p0 – LAMS 2941
150	pu406e3	94008	HR Plutonium 240 – p0 – LAMS 2941
151	pu401e4	94009	HR Plutonium 240 – p0 – LAMS 2941
152	pu402e4	94010	HR Plutonium 240 – p0 – LAMS 2941
153	pu404e4	94011	HR Plutonium 240 – p0 – LAMS 2941
154	pu406e4	94012	HR Plutonium 240 – p0 – LAMS 2941
155	pu401e5	94013	HR Plutonium 240 – p0 – LAMS 2941
156	pu402e5	94014	HR Plutonium 240 – p0 – LAMS 2941
157	pu404e5	94015	HR Plutonium 240 – p0 – LAMS 2941
158	pu406e5	94016	HR Plutonium 240 – p0 – LAMS 2941
159	pu401e6	94017	HR Plutonium 240 – p0 – LAMS 2941
160	pu40inf	94018	HR Plutonium 240 – p0 – LAMS 2941
161	pu41	94100	HR Plutonium 241 – p0 – Origin Unknown

Number	TWODANT ID	KENO V.a ID	Description
162	pu41OR	94101	OR Plutonium 241 – p0 – GAM 2
163	pu42	94200	OR Plutonium 242 – p0 – GAM 2
164	np37OR	93700	OR Neptunium 237 – p1 – ENDF/B-IV
165	am51OR	95100	OR Americium 241 – p1 – ENDF/B-IV
166	am53OR	95300	OR Americium 243 – p1 – ENDF/B-IV
167	cm64OR	96400	OR Curium 244 – p1 – ENDF/B-IV

APPENDIX E

Resonance Absorption and the Hansen-Roach Cross Sections

The following pages are extracted from a paper by Robert D. Busch and R. Douglas O'Dell, titled "Validity of Hansen-Roach Cross Sections in Low-Enriched Uranium Systems," which was published in the Proceedings of the ICNC '91 — International Conference on Nuclear Criticality Safety, pgs. IV-45 to IV-52. The Conference was held in Oxford, United Kingdom from 9-13 September 1991.

RESONANCE ABSORPTION & THE HANSEN-ROACH CROSS SECTIONS

Robert D. Busch, Ph.D., P.E.
Department of Chemical & Nuclear Engineering
University of New Mexico

R. Douglas O'Dell
HSE–6, Criticality Safety
Los Alamos National Laboratory

July 5, 1991

ABSTRACT

Within the nuclear criticality safety community, the Hansen-Roach 16 group cross-section set has been the "standard" for use in k_{eff} calculations over the past 30 years. Yet even with its widespread acceptance, there are still questions about its validity and adequacy, about the proper procedure for calculating the potential scattering cross section, σ_p, for uranium and plutonium, and about the concept of resonance self shielding and its impact on cross sections. This paper attempts to address these questions. It provides a brief background on the Hansen-Roach cross sections. Next is presented a review of resonances in cross sections, self shielding of these resonances, and the use of σ_p to characterize resonance self shielding. Three prescriptions for calculating σ_p are given.

BACKGROUND

The sets of multigroup neutron cross-section data known as the Hansen-Roach cross sections were first formally presented in late 1961 as both six- and sixteen-group cross sections for fast and intermediate critical assemblies.[1] The six-group cross sections were for the study of fast neutron critical assemblies, and the sixteen-group cross-sections were for intermediate neutron critical assemblies. In the sixteen-group data the top five energy groups were identical to the top five groups of the six-group set. In September, 1963, a second report was issued containing six-, sixteen-, eighteen-, twenty four-, and twenty five-group cross sections.[2] These cross-section sets were designed for reactor calculations with neutron energies ranging from fast

to thermal. In all these sets the top five energy groups were the same. It is in Ref. 2 that "revised" cross sections for ^{235}U and ^{238}U were presented for the sixteen-group cross sections. The eighteen-, twenty four-, and twenty five-group specifications provided more energy groups in the epithermal and thermal energy ranges relative to the sixteen-group specification and thus would have been expected to be more suitable for thermal neutron systems. However, only the sixteen-group cross sections have survived the nearly three decades of use, and these have become known as the Hansen-Roach cross sections. Taking only the "revised" cross sections for ^{235}U and ^{238}U, the data tabulated in Ref. 2 covers 108 "nuclides." These 108 "nuclides" actually represent only 31 different elements and isotopes. (A list of these 31 elements/isotopes is provided in Appendix A).

The reason that the 108 "nuclide" sets cover only 31 different elements/isotopes is that many of the nuclides represent cross-section sets for the same fissionable isotope but for differing resonance absorption treatments. The isotopes of ^{233}U, ^{235}U, ^{238}U, ^{239}Pu, and ^{240}Pu have such multiple cross-section sets.

Up to this point, things seem to be fairly clear regarding the history of the Hansen-Roach library, but such is not actually the case. In the early 1960's multigroup neutronics computer codes for performing calculations were beginning to come of age. Since the codes required cross-section data, the 16 group Hansen-Roach library was quickly acquired and accepted by those doing criticality safety analyses. Numerous versions of the library began to appear as users began adding data for additional nuclides for their own needs. Even within the 1960's the genealogy of some of these libraries was difficult to track down. Today there are a number of data sets which bear the label "Hansen-Roach" library, and their origins and quality assurance/ validation are obscure. However, there are three prevalent sets which are widely used: one used at Los Alamos which tracks the data in Ref. 2; one used with KENO[3] and which is called the AMPX working library; and the library attached to the SCALE/CSAS code system.[4] The Hansen-Roach AMPX working library and the SCALE/CSAS library seem to be derived from similar parentage through Oak Ridge National Laboratory although it should be noted that a recent notice was issued concerning problems with the AMPX working library.[5] It appears that the LANL library and the SCALE/CSAS library can be used with confidence.

Even though the sixteen-group cross sections were not originally designed to span the complete energy range of critical assembly types, they have done a remarkable job of providing good calculational predictions of k_{eff} for virtually all types of critical assemblies and systems, *when the proper fissionable "nuclide" is used.*

RESONANCE SELF-SHIELDING AND THE CALCULATION OF σ_p

One of the major problems that seems to arise from the use of the Hansen-Roach cross sections stems from the existence of multiple sets of cross sections for the fissionable isotopes ^{233}U, ^{235}U, ^{238}U, ^{239}Pu, and ^{240}Pu. For example, there are 13 sets of cross sections for the isotope ^{235}U. The presence of these multiple sets of data for a given isotope causes much confusion for many users of the library. In most other cross-section libraries there is only one set of data for each isotope. In those cases where multiple sets are provided, there is usually a temperature associated with each set to identify the set. For the Hansen-Roach library, however, the different sets are not due to temperature differences but, instead, are provided to properly account for "resonance self shielding" of the fissionable isotopes in different mixtures of fissionable absorbers and neutron moderating material. While it is not within the scope of this paper to present the theory of resonance absorption, a few words may be helpful to provide some level of understanding of the need for, and importance of, the multiple sets of cross sections. Although much of the following applies equally well to elements and isotopes other than the fissionable isotopes of uranium and plutonium, for purposes of this paper the presentation will be specifically focused on these latter two elements.

Principles of Resonance Absorption and Self Shielding

For neutron energies between, say, 0.1 eV and 3 keV, the absorption cross section of the isotopes of uranium and plutonium varies markedly in a series of sharp, narrow (in energy), very pronounced "resonances." Most of the absorption of neutrons in this energy range takes place in these resonances. These resonances are frequently only about 0.1 eV in width, may have peak amplitudes of thousands of barns, and are separated from one another by about 20 eV. This pronounced resonance structure in the neutron-energy-dependent cross section leads to a corresponding fine energy structure in the neutron flux. In general, there will be a dip in the neutron flux at the energy and the location where there is a strong absorption resonance.

Now, the vast majority of neutrons produced by fission are born with energies greater than 50 keV; that is, they are born with energies well above the resonance energy range. For neutrons to acquire energies where they can see the resonances of uranium and plutonium, they must be slowed down, or moderated, from their birth energies by means of scattering collisions. For practical purposes, such moderation requires that relatively low atomic weight material (a moderator) must be present in addition to fissionable material. For example, unreflected systems of uranium or plutonium metal have virtually no neutrons with energies below about 10 keV and ordinary criticality calculations for these systems are not sensitive to the existence of resonances. If moderating material is present, the overall neutron flux energy spectrum in the eV–keV range varies as 1/E to good approximation. Superimposed on this overall energy shape are the local flux dips resulting from the resonances in the fissionable material or "absorber." If a flux dip due to a strong resonance is "felt" by a resonance at a lower energy, then the neutron absorption in the latter resonance will be less than would have occurred if the higher energy resonance had not been present simply because the neutron flux that "sees" the second resonance is reduced by the first resonance. In other words, the second (and third, and fourth, etc.) resonance "downstream" from the first resonance may be somewhat shielded by the first resonance. This phenomenon is called *resonance self shielding*. The greatest absorption by the resonances will occur when there is no resonance self shielding. The degree to which self shielding affects the overall neutron absorption in resonances depends strongly on the amount of scattering (moderation) present in the absorber-moderator mixture, or, conversely, on the amount of absorption present in the mixture, that is, the degree of "dilution" of the resonance-absorbing isotope in the moderator. The dilution of the resonance-absorbing isotope is measured by the ratio of the moderator macroscopic scattering cross section to the atom density of the particular resonance-absorbing fuel isotope in question. This ratio is referred to as the potential scattering cross section for that resonance-absorbing isotope. It follows, then, that the greatest absorption by the resonances (per absorber atom) will occur when the absorbing isotope is "infinitely" dilute with an enormously large potential scattering cross section so that there is no resonance self shielding. The least amount of absorption by the resonances per absorber atom will occur with the least possible dilution (a potential scattering cross section of zero) where resonance self shielding is a maximum. *Note that each resonance-absorbing isotope in the mixture will have its own unique potential scattering cross section.*

In generating multigroup cross sections, it is necessary to determine the average cross section for an element or isotope over the energy range of a given energy group. In averaging the absorption cross section of a resonance absorbing isotope over an energy range encompassing many resonances, therefore, the average cross section is dependent on the potential scattering cross section for that isotope *in that mixture*. Thus, to generate a general library which can be used for all types of mixtures of resonance-absorbing material with varying degrees of moderation, it is necessary to generate multiple cross-section sets for each resonance-absorbing isotope; one set for each characteristic mixture type. This is precisely what the Hansen-Roach library contains — multiple sets of cross sections which encompass the full range of isotope-moderator types. The range of mixture-types is characterized by the potential scattering cross-section parameter for the resonance-absorbing isotopes of uranium and plutonium. The use of the proper set of cross sections for the fissionable isotopes is very important; indeed, for low-enriched moderated systems, it is *essential*!

It is noted that the multiple sets of cross sections in the Hansen-Roach library are for *room temperature* materials. Temperature effects, most notably in the Doppler broadening of the resonances, are not accounted for.

Calculating σ_p

The remaining issue in the proper use of the Hansen-Roach cross sections is the determination of the potential scattering cross section, σ_p, for each of the extant fissionable isotopes in the system to be analyzed. The basic definition of this quantity was provided without further explanation on page 6 of Ref. 2 where it is stated "The notation SIG P, or SIGMA P, specifies a scattering cross section per absorber atom for neutrons:

$$\sigma_p \equiv \frac{\Sigma_s(\text{moderator})}{N\ (\text{atomic density of absorber atoms})}\text{"} \tag{1}$$

The definition of "moderator" has caused some differences in the way σ_p has been calculated. Three basic prescriptions used to calculate σ_p use different definitions of "moderator." The first, which we will call the KENO, or SCALE/CSAS, prescription is found in Ref. 3. It considers every isotope/nuclide as a contributor to the "moderator." There is no

apparent theoretical basis for such a definition of "moderator" since the intermediate and heavy nuclides provide little scattering energy loss to the neutrons. Because of this latter fact, the inclusion of intermediate and heavy nuclides is of little numerical consequence. The second prescription is the Hopper-Renier prescriptions.[6] It also considers all isotopes/nuclides as contributors, but the average logarithmic decrement, ξ, for each nuclide is used to weight its scattering contribution†. Since ξ decreases with increasing atomic weight, the intermediate and heavy nuclides contribute little to the scattering of the "moderator." The third prescription is suggested by Gordon Hansen and Thomas McLaughlin as being appropriate for use with the Hansen-Roach library. This form uses the classical definition of "moderator" in that it considers only the light nuclides (hydrogen through oxygen) from which neutron scattering results in large energy losses. A practical advantage of this prescription is its simplicity. Suitable values of the scattering cross sections, $\sigma_{s,i}$, for common light nuclides are provided in Appendix B.

All three prescriptions for calculating σ_p for fissionable isotope "x" can be cast in a common generic form:

$$\sigma_{p,x} = \frac{1}{N_x} \sum_{i=1}^{M} (WF_i) N_i \sigma_{s,i} \tag{2}$$

where:

N_x is the atom density of fissionable isotope x,

M is the number of nuclides in the mixture,

N_i is the atom density of nuclide i in the mixture,

$\sigma_{s,i}$ is the average scattering cross section in the resonance region for nuclide i in the mixture,

WF_i is a weighting factor for nuclide i in the mixture, different for each prescription.

SCALE/CSAS:

$$WF_i = 1.0, \quad \text{for all } i \tag{3}$$

---

† $\xi = 1 + \frac{(A-1)^2}{2A} \ln\left[\frac{A-1}{A+1}\right]$

Hopper-Renier:

$$WF_i = \frac{1 + \frac{(A_i - 1)^2}{2A_i} \ln\left[\frac{A_i - 1}{A_i + 1}\right]}{(0.14474)[\ln(1000)]}, \text{ for all i} \tag{4}$$

Hansen-McLaughlin:

(5)

In Eqs. (4) and (5), A_i is the atomic weight of nuclide i.

Sensitivity of k_{eff} on σ_p Used

The value of σ_p calculated for a given system is strongly dependent on the enrichment of the fuel and on the H/X atom density ratio where X denotes the fissionable isotope in a mixture containing hydrogen. Generally, the value of σ_p is only weakly dependent on the prescription used to calculate it. Since the Hansen-Roach library only has cross sections for selected values of σ_p, it is probable that the calculated σ_p will fall between available values. The accepted procedure in such a case is to linearly apportion the cross sections between the two available library values. For example, suppose a σ_p for ^{238}U with an atom density of 0.04 is calculated to be 550. The library contains cross sections for σ_p's of 400 (u238-6r) and 600 (u238-7r). Thus, one could model the actual ^{238}U with u238-6r at an atom density of 0.01 plus u238-7r at an atom density of 0.03. For many applications it is not necessary to go to such fine detail as just described but instead to simply use the nuclide in the library whose σ_p lies closest to the calculated σ_p. The main point is to use library data for σ_p close to the calculated σ_p. This is especially true for low-enriched uranium solutions as is dramatically shown in Figure 1. The actual system used for this figure is 5 weight percent enriched uranium in a critical (k_{eff}=1) bare infinitely long cylinder of UO_2F_2 solution with a H/^{235}U ratio of 500. The calculated σ_p for the

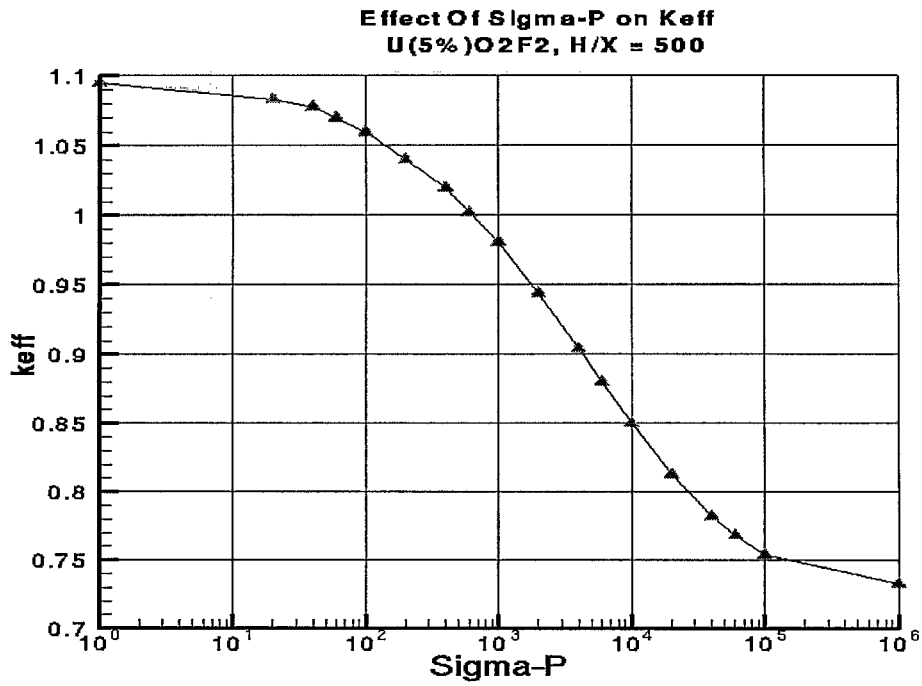


Fig. 1. Variation in k_{eff} with Different ^{238}U σ_p for a Critical System of $U(5)O_2F_2$ Solution.

^{238}U is about 550. Plotted in the figure is the calculated k_{eff} as a function of the Hansen-Roach ^{238}U nuclide used for the different σ_p's available in the library. Note that if "infinitely dilute" ^{238}U is erroneously used, a calculated k_{eff} of 0.73 results — a grossly nonconservative value. Conversely, if metallic ^{238}U cross sections are used erroneously, a value of k_{eff} of 1.09 results — a large overestimate of k_{eff}. Finally, note that using the ^{238}U with σ_p of 600 (close to the calculated value of 550) results in a k_{eff} very near to unity. Thus, the proper selection of the ^{238}U based on σ_p from the Hansen-Roach library is *essential* for low-enriched uranium solution calculations.

It is logical to ask if the proper selection of the fissile ^{235}U nuclide based on σ_p is equally important. Generally, the answer is no. The reason for this is that with the fissile isotopes resonance, self shielding affects both fission and capture resonances. Since these two processes compete with one another in their effect on k_{eff}, there tends to be a cancelling effect. The net

result is a much lesser degree of sensitivity on k_{eff} when different fissile nuclide cross-section sets are used from the Hansen-Roach library.

Also for fast neutron systems, k_{eff} is not very sensitive to the different σ_p-dependent cross section-sets. The reason for this is simply that resonance self shielding effects are minimal in fast systems since there are few neutrons with energies in the resonance energy range.

Nevertheless, it is *always* wise to use the σ_p parameter to select the correct self shielded cross-section set for both fissile and fissionable isotopes, irrespective of the type of system. When this is done, the Hansen-Roach cross sections can be used with confidence in calculating k_{eff} for all types of systems.

SUMMARY

We have presented a background history of the Hansen-Roach 16 group cross sections together with an explanation of the existence of, and need for, multiple sets of cross sections for the fissionable isotopes of uranium and plutonium. These multiple sets are necessary to provide for differences in resonance self shielding as a function of the potential scattering cross section, σ_p, for the fissile isotope in a moderating mixture. Three prescriptions for calculating σ_p were presented. Results indicate that very little difference occurs among k_{eff} calculations using any of the three prescriptions. We also showed that if no prescription is used, or if the uranium/ plutonium cross section is selected at random, very large errors (25% or larger) in k_{eff} can occur, especially in low-enriched, moderated systems. However, results indicate that with the proper use of σ_p to predict resonance self shielding effects, the Hansen-Roach cross sections are a reliable, valid set of data for virtually all types of systems.

REFERENCES

1. GORDON E. HANSEN and WILLIAM H. ROACH, "Six and Sixteen Group Cross Sections for Fast and Intermediate Critical Assemblies," Los Alamos Scientific Laboratory report LAMS–2543, (December 6, 1961).

2. G. I. BELL, J. J. DEVANEY, G. E. HANSEN, C. B. MILLS, and W. H. ROACH, "Los Alamos Group-Averaged Cross Sections," Los Alamos Scientific Laboratory report LAMS–2941, (September 24, 1963).

3. L. M. PETRIE and N. F. CROSS, "KENO IV: — An Improved Monte Carlo Criticality Program," Oak Ridge National Laboratory report ORNL–4938 (November 1975).

4. STAFF of Technical Applications, Computing and Telecommunications Division, ORNL, "SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation," Vols. 1-3, NUREG/CR-0200, U.S. Nuclear Regulatory Commission (July 1980, original issue).

5. RSIC Newsletter, No. 311, October 1990, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6362.

6. CALVIN M. HOPPER and JOHN-PAUL RENIER, "Expanded and Applied Sixteen-Energy-Group Cross-Section Library," in *Trans. Am. Nuc. Soc.*, 61, June 10-14, 1990 (pp 186-188).

APPENDIX A

Listing of the 31 Elements/Isotopes Included on the "Original" 16 Group Hansen-Roach Cross-Section Library

Note: In the table below, the number in parenthesis for certain fissionable isotopes is the number of different potential-scattering-dependent data sets that exist for that isotope.

Number	Element/Isotope	Number	Element/Isotope
1	Aluminum	17	Oxygen
2	Boron(Nat.)	18	Plutonium-239 (17)
3	Beryllium	19	Plutonium-240 (19)
4	Carbon	20	Thorium
5	Cadmium	21	Uranium-233 (13)
6	Chlorine	22	Uranium-235 (14)
7	Gallium	23	Uranium-238 (19
8	Fluorine	24	Zirconium
9	Iron	25	Nitrogen
10	Hydrogen, P_0 & P_1	26	Niobium
11	Potassium	27	Cerium
12	Lithium-6	28	Cobalt
13	Lithium-7	29	Tantalum
14	Molybdenum	30	Chromium
15	Sodium	31	Deuterium, P_0 & P_1
16	Nickel		

APPENDIX B

Suitable values for σ_s in the resonance regions for common nuclides with atomic weights ≤ 16

Nuclide	σ_s
^{1}H	20.4
^{2}D	3.4
Li	0.9
Be	6.0
B	3.5
C	4.6
N	9.0
^{16}O	3.8

This report has been reproduced directly from the
best available copy.

It is available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62,
Oak Ridge, TN 37831.
Prices are available from
(615) 576-8401.

It is available to the public from the
National Technical Information Service,
US Department of Commerce,
5285 Port Royal Rd.
Springfield, VA 22616.

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545