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

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

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 each keyword must be immediately followed by an equals sign. 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, b̸ , is used throughout this primer to identify a required blank space on a line.

→ b̸b̸b̸b̸b̸2

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:

```
bbbbb2
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 number 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:

```
bbbbb2
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:

```
bbbbb3
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 **b** 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.

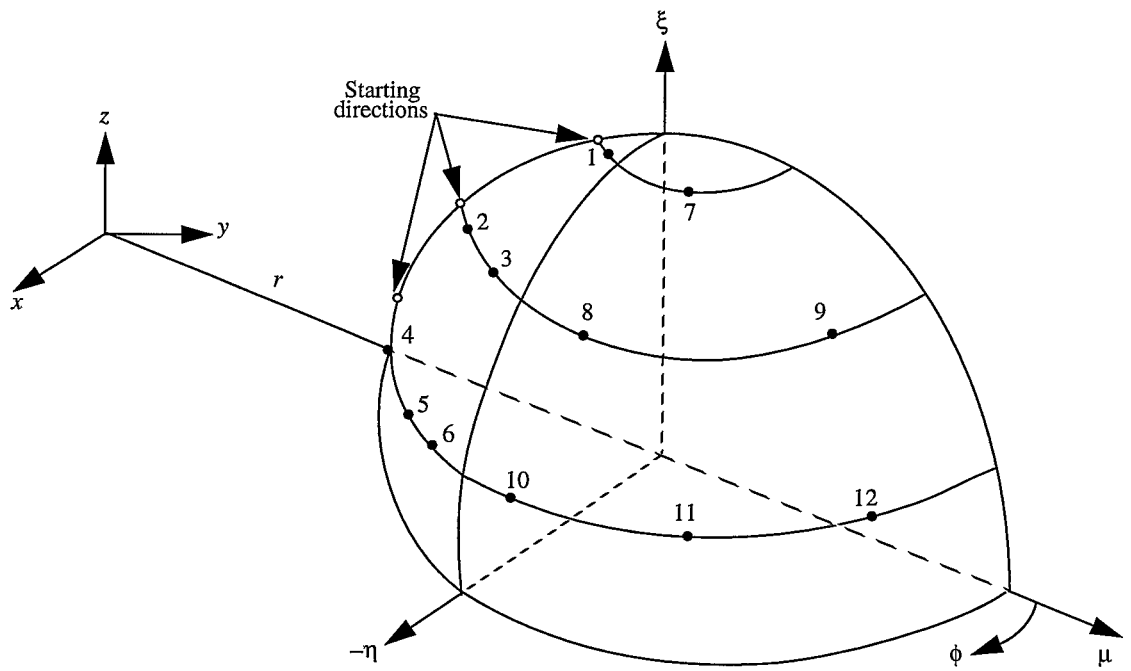

Figure 2.1 — Fully Symmetric S_6 Quadrature.

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:

```
bbbbb3
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 number 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

```
bbbbb3
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:

```
bbbbb3
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.