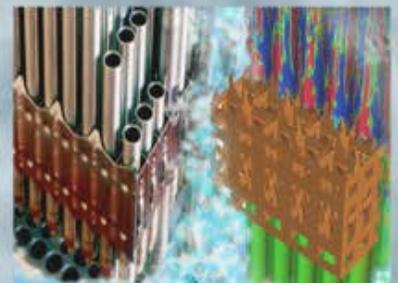
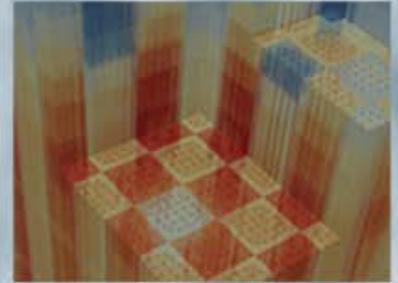


CTF – COBRA-TF Subchannel Thermal- Hydraulics Code (CTF) Pre-processor Users' Manual

Revision 0

Robert K. Salko
Maria N. Avramova
Pennsylvania State University

>abi Ufm1- , 2015



THE PENNSYLVANIA STATE UNIVERSITY
DEPARTMENT OF MECHANICAL AND NUCLEAR ENGINEERING
REACTOR DYNAMICS AND FUEL MANAGEMENT GROUP



CTF Pre-processor User's Manual

Robert K. Salko
Postdoctoral Researcher
(814) 865-2289
rks171@psu.edu

Maria N. Avramova
*Assistant Professor of
Nuclear Engineering*
(814) 865-0043
mna109@psu.edu

January 19, 2015

LEGAL NOTICE

This document was prepared by Reactor Dynamics and Fuel Management Group (RDFMG), Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, University Park, USA. Neither RDFMG, nor any person acting on behalf of RDFMG:

- A Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this document, or that the use of any information, apparatus, method, or process disclosed in this document may not infringe on privately owned rights
- B Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method or process disclosed in this document.

RDFMG Group
206 Reber Building
University Park, PA 16802
Phone: +1-814 865-0040
Fax: +1-814 865-8499 www.mne.psu.edu/rdfmg

Abstract

This document describes how a user should go about using the CTF pre-processor tool to create an input deck for modeling rod-bundle geometry in CTF. The tool was designed to generate input decks in a quick and less error-prone manner for CTF. The pre-processor is a completely independent utility, written in Fortran, that takes a reduced amount of input from the user. The information that the user must supply is basic information on bundle geometry, such as rod pitch, clad thickness, and axial location of spacer grids—the pre-processor takes this basic information and determines channel placement and connection information to be written to the input deck, which is the most time-consuming and error-prone segment of creating a deck. Creation of the model is also more intuitive, as the user can specify assembly and water-tube placement using visual maps instead of having to place them by determining channel/channel and rod/channel connections. As an example of the benefit of the pre-processor, a quarter-core model that contains 500,000 scalar-mesh cells was read into CTF from an input deck containing 200,000 lines of data. This 200,000 line input deck was produced automatically from a set of pre-processor decks that contained only 300 lines of data.

ACRONYMS

PWR Pressurized-Water Reactor

BWR Boiling-Water Reactor

CONTENTS

- 1 Introduction** **2**

- 2 Control.inp File** **4**
 - 2.1 Numerical Solution Options 8
 - 2.2 Model Options 10
 - 2.3 Initial Condition Data 14
 - 2.4 Boundary Condition Data 16
 - 2.5 Transient Data 20
 - 2.6 Convergence Criteria 23
 - 2.7 Output Options 25

- 3 Geo.inp File** **29**

- 4 Assembly.inp File** **37**
 - 4.1 Heater Types {heater types} 42
 - 4.1.1 Nuclear Fuel Rods 42

4.1.2	Electric Heater Tubes	45
4.2	Guide Tubes {guide tube types}	46
4.3	Guide Tube Positions	48
4.4	Nozzles	49
4.5	Spacer Grids	49
5	Power.inp File	51
5.1	Assembly-level Power Factor Maps	54
5.1.1	No Map	54
5.1.2	Single Map	55
5.1.3	Multiple Assembly Power Factor Maps {number of assembly maps}	56

LIST OF FIGURES

4.1 Example of a 5x5 fuel rod lattice 38

CHAPTER 1

INTRODUCTION

The CTF pre-processor utility produces CTF input decks from a reduced set of user-input data. The utility produces models geared towards Pressurized-Water Reactor (PWR) rod-bundle geometry. It does this by using basic characteristics of the rod bundle, including rod pitch, fuel rod dimensions, and visual maps for assembly and guide-tube placement. This user input is provided by four text-based input decks that take information broken up into four general categories. These four input decks include the *control*, *geo*, *assembly*, and *power* files. This user's manual documents the instructions for constructing these four input files for the purpose of building a CTF model.

Sets of example pre-processor-input files are typically distributed with the CTF pre-processor—in this case, it is a simple matter of modifying the example file values to suit the needs of your own model. If no example files are available, this document provides sufficient instruction for generating the files from scratch. Prior to discussing the construction of these input files, some notes are made on their general structure.

First, any line of input that starts with the asterisk character, “*”, is taken to be a comment line by the pre-processor. In other words, the pre-processor ignores these lines. This is useful for sufficiently describing the meaning of the various flags and values that appear in the input files. If you possess one of the sets of example files, you will find that commentary on the input values is utilized heavily in order to make the pre-processor input as intuitive and user friendly as possible. Secondly, the file is formatted in such a way that, if opened with the text editor, “VIM”, with syntax highlighting turned on, the headers and options comments will conveniently be color coded, making the files more

readable.

During development of the preprocessor, a simple tagging system was introduced in order to prevent breaking backwards compatibility when the input decks are modified and also to allow for accurate error messaging. Currently, this tagging system is only used for new terms that are introduced into the input decks, so not all terms will be tagged. A tag simply consists of some specific text inside curly braces. A tag and its corresponding data forms a sort of “key-value” pair. The tag is the “key” and its value is the data found on the following line or lines. This manual gives the tag name for each data set that has an optional or required tag.

The following four chapters discuss the meaning of each input value required by the pre-processor input files. After these files are constructed, you may simply run the pre-processor executable or, more preferably, a link to the pre-processor executable in the same directory where the four input files are residing. If no errors result during the input deck generation process, a completed CTF input deck, “deck.inp”, should appear. This file may then be moved to the directory where the CTF simulation will be run and executed.

This manual discusses each term that may be needed as input to the preprocessor via the four input files. Each term will have several components that are discussed. The title of each term is given, followed by a box containing the term definition, its datatype (integer or float), its options (if any), its units (if any), its tag (if it has one), and whether the input is required, optional, or conditional. Note that some terms will not have any options (e.g., a value will be expected for boundary or initial conditions as opposed to the modeling choice for friction factor). Additionally, some terms will not have units associated with them (e.g., modeling choices) and will instead present an em dash (“—”). Some terms are required for every case you model—make sure you always have these terms in your input deck. Other terms will be optional—if the preprocessor doesn’t find them, it will default to some value and probably also give you a warning message. Other terms, still, will be conditional, meaning that if you choose a particular option, additional input will be required. This status characteristic will be identified in the term description to help prevent you from making input errors.

CHAPTER 2

CONTROL.INP FILE

This chapter details the construction of the *control.inp* file. The purpose of this file is to:

- select some general models that CTF should use in the simulation,
- select the initial conditions for the problem, and
- select the boundary conditions for the problem.

Note that CTF does allow fine control of what the boundary conditions can be set to. For example, the user can set different inlet mass flow rates for different rod-bundle channels in the model; however, the pre-processor does not allow this flexibility because it would require a much greater set of user input, which would defeat the purpose of this utility. If you wish to refine the boundary conditions, it is best to use the pre-processor to generate the input deck, and then modify Card 13 in the “deck.inp” file as per the instructions of the CTF User’s Manual.

Term 1: Deck Title

Description	Datatype
Title of the model that will be printed in the input deck and output files produced by CTF. A maximum of 90 characters may be included in the title. Anything past 90 characters will be truncated.	Character
Options	Units
—	—
Tag	Input Type
—	Required

Term 2: Parallel Option

Description	Datatype
Switch the parallel option on or off. If you turn the parallel option on, the preprocessor will create a set of parallel input decks instead of the normal, single, serial “deck.inp” file. A separate file will be created for each domain in the problem. Also, an additional “pmaster.inp” deck will be created. When you execute COBRA-TF on these input files, you must execute it using <code>mpirun</code> and you must specify the same number of processors as there are domains (“pdeck.inp” files) in the problem.	Integer
Options	Units
0—Produce serial input deck 1—Produce parallel input decks	—
Tag	Input Type
{parallel}	Optional

Term 3: Write Maps Option

Description	Datatype
This option specifies whether the rod and channel maps should be written to the CTF input deck on Card Group 17. If you select to write the maps, you also must provide filenames for the HDF5 and VTK files to be written by CTF on the following two lines of input.	Integer
Options	Units
1—Write maps to the deck 0—Do not write the maps	—
Tag	Input Type
—	Optional

Term 4: HDF5 Filename

Description	Datatype
The name of the HDF5 file that will be generated by CTF when HDF5 edits are to be written.	Character
Options	Units
—	—
Tag	Input Type
—	Conditional

Term 5: VTK Filename

Description	Datatype
The name of the VTK file that will be generated by CTF when VTK edits are to be written.	Character
Options	Units
—	—
Tag	Input Type
—	Conditional

Term 6: Number of Transient Power Files

Description	Datatype
<p>The number of power.inp files to be read in, with each power.inp file specifying power data for a different point in the simulation time. If you specify multiple power.inp files, then the files should be named using the convention <code>power.inp.<num></code> where <code><num></code> is a number that varies from 1 to the total number of power.inp files being specified. If you leave this option out of the <i>control</i> deck, the preprocessor will default to reading a single <i>power</i> file.</p>	Integer
Options	Units
—	—
Tag	Input Type
{num transient power files}	Optional

Term 7: Units Option

Description	Datatype
<p>The units system that CTF should use when outputting results. Note that the pre-processor currently only takes SI units for input values.</p>	Integer
Options	Units
<p>1–SI Output 3–US Output</p>	—
Tag	Input Type
—	Required

2.1 Numerical Solution Options

Term 8: Outer Iteration Convergence Criterion

Description	Datatype
The outer iteration solves the pressure corrections to be made in the mesh. This term is the non-dimensionalized error in those corrections that should be reached before judging the outer iteration to be converged. A suggested value is 0.001.	Float
Options	Units
—	—
Tag	Input Type
—	Required

Term 9: Maximum Number of Outer Iterations

Description	Datatype
The maximum number of outer iterations to be taken before the simulation is ended due to convergence problems. A suggested value is 5.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 10: Maximum Number of Inner Iterations

Description	Datatype
The maximum number of inner iterations to be taken before the simulation is ended due to convergence problems. A suggested value is 40.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 11: Courant Number

Description	Datatype
The Courant number to be used for the simulation. Traditionally, this value has been set to 0.8 in CTF. For single-phase cases, it has been found that the simulation speed could be greatly increased by raising this up to 3.0. However, convergence studies have revealed that a value of 0.4 is the safest one to use for this parameter.	Float
Options	Units
—	—
Tag	Input Type
—	Required

2.2 Model Options

Term 12: Surface Friction Factor Correlation

Description	Datatype
The correlation used for calculating the surface-friction factor.	Integer
Options	Units
$1-f = \max\left(\frac{1.691}{Re^{0.43}}, \frac{0.117}{Re^{0.14}}\right)$ $2-f = 0.204Re^{-0.2}$ $3-f = 0.390Re^{-0.249}$	—
Tag	Input Type
—	Required

Term 13: Entrainment and Deposition

Description	Datatype
Determines whether or not entrainment/deposition will be modeled.	Integer
Options	Units
0—No entrainment/deposition 1—Model entrainment/deposition	—
Tag	Input Type
—	Required

Term 14: Mixing and Void Drift

Description	Datatype
Determines if turbulent mixing and void drift will be modeled and, if so, how it will be modeled. This is referred to as the IMIX parameter.	Integer
Options	Units
0–No turbulent mixing or void drift	—
1–User-specified constant for two-phase turbulent-mixing coefficient	
2–Single-phase mixing coefficient calculated using Rogers and Rosehart	
3–User-specified constant single-phase turbulent-mixing coefficient	
Tag	Input Type
—	Required

Term 15: k_m

Description	Datatype
Distribution weighting factor for the void-drift model. Entered if the IMIX parameter is 1, 2, or 3. A value of 0.0 makes void drift inactive. A suggested value is 1.4.	Float
Options	Units
—	—
Tag	Input Type
—	Conditional

Term 16: β_m

Description	Datatype
Constant turbulent-mixing coefficient. If IMIX is 1, this is the the two-phase mixing coefficient. If IMIX is 3, this is the single- phase mixing coefficient. This is not entered if IMIX is 2.	Float
Options	Units
—	—
Tag	Input Type
—	Conditional

Term 17: DFROD

Description	Datatype
Outside diameter of the fuel rod. Enter only for IMIX=2.	Float
Options	Units
—	m
Tag	Input Type
—	Conditional

Term 18: Θ_m

Description	Datatype
Ratio between maximum two-phase turbulent-mixing coefficient and single-phase turbulent-mixing coefficient. Entered only if IMIX=2 or if IMIX=3.	Float
Options	Units
—	—
Tag	Input Type
—	Conditional

Term 19: Property Evaluations Option

Description	Datatype
Specifies which set of property evaluations should be used for determining fluid state properties (e.g. density, conductivity, viscosity, etc.). The default is currently "asme".	String
Options	Units
iapws - IAPWS-IF97 property evaluations asme - ASME 1967 property evaluations	—
Tag	Input Type
property evaluations	Optional

Term 20: Solver

Description	Datatype
The method for solving the pressure-correction-equation matrix	Integer
Options	Units
0-Direct Gaussian elimination 1-Bi-conjugate gradient stabilized method with ILUT (Incomplete LU factorization with dual truncation strategy) preconditioner 2-Generalized minimum residual method with no preconditioner 3-Bi-conjugate gradient stabilized method with with no preconditioner 4-Generalized minimum residual method with (Incomplete LU factorization with dual truncation strategy) preconditioner	—
Tag	Input Type
—	Required

Term 21: Boiling Heat Transfer Model

Description	Datatype
Selects model used to calculate subcooled and nucleate boiling heat transfer. The default, if this is left out, is Thom.	string
Options	Units
thom chen	—
Tag	Input Type
boiling ht correlation	Optional

2.3 Initial Condition Data

Term 22: Mass Flow Rate Initialization

Description	Datatype
The mass flow rate applied throughout the entire momentum-cell mesh.	Float
Options	Units
—	kg/s
Tag	Input Type
—	Required

Term 23: Rod Temperature Initialization

Description	Datatype
The initial temperature applied throughout the conductors in the model.	Float
Options	Units
—	°C
Tag	Input Type
—	Required

Term 24: Pressure Initialization

Description	Datatype
The initial pressure throughout the scalar-cell mesh. Note that CTF will consider the effect of elevation and gravity on pressure prior to starting the flow-field solution and re-calculate the pressure field. This starts the simulation with a more correct pressure field than if a uniform one were assumed.	Float
Options	Units
—	bar
Tag	Input Type
—	Required

Term 25: Water Enthalpy Initialization

Description	Datatype
The initial enthalpy of the liquid/steam mixture in the model. You can also specify as a temperature instead by entering the negative of the temperature (e.g. specify 200 °C as -200)	Float
Options	Units
—	kJ/kg or °C
Tag	Input Type
—	Required

Term 26: Non-condensable Gas Enthalpy Initialization

Description	Datatype
The initial enthalpy of the non-condensable gases.	Float
Options	Units
—	kJ/kg
Tag	Input Type
—	Required

2.4 Boundary Condition Data

Term 27: DHFRAC

Description	Datatype
The fraction of heat produced by the rods that is released directly to the coolant.	Float
Options	Units
—	—
Tag	Input Type
—	Required

Term 28: Inlet Boundary Condition Type

Description	Datatype
The type of boundary condition applied at the inlet.	Integer
Options	Units
1-Pressure and enthalpy	—
2-Mass flow rate and enthalpy	
3-Mass flow rate only	
4-Mass source	
5-Pressure sink	
Tag	Input Type
—	Required

Term 29: Outlet Boundary Condition Type

Description	Datatype
The type of boundary condition applied at the outlet.	Integer
Options	Units
1-Pressure and enthalpy	—
2-Mass flow rate and enthalpy	
3-Mass flow rate only	
4-Mass source	
5-Pressure sink	
Tag	Input Type
—	Required

Term 30: Inlet Mass Flow Rate

Description	Datatype
Mass flow rate boundary condition applied to the inlet. Only entered if inlet boundary condition type is 2 or 3. Enter 0.0 otherwise.	Float
Options	Units
—	kg/s
Tag	Input Type
—	Required

Term 31: Inlet Enthalpy

Description	Datatype
Enthalpy of liquid entering the model. Only entered if the inlet boundary condition type is 1 or 2. Enter 0.0 otherwise. You may also specify as a temperature by entering the negative of the fluid temperature.	Float
Options	Units
—	kJ/kg or °C
Tag	Input Type
—	Required

Term 32: Outlet Enthalpy

Description	Datatype
Enthalpy of liquid leaving the model. Only entered if the outlet boundary condition type is 1 or 2. Enter 0.0 otherwise. Note that this value is only used if reverse flow occurs. Otherwise, if flow is out of the model at the outlet, the value of enthalpy in the exit auxiliary cell will be set to the value of enthalpy in the scalar-cell just prior to the exit. You may also enter this as a temperature by specifying the negative of the fluid temperature.	Float
Options	Units
—	kJ/kg or °C
Tag	Input Type
—	Required

Term 33: Inlet Pressure

Description	Datatype
Pressure at the inlet. Only entered if the boundary condition type at the inlet is 1 or 5. Enter 0.0 otherwise.	Float
Options	Units
—	bar
Tag	Input Type
—	Required

Term 34: Outlet Pressure

Description	Datatype
Pressure at the outlet. Only entered if the boundary condition type at the outlet is 1 or 5. Enter 0.0 otherwise.	Float
Options	Units
—	bar
Tag	Input Type
—	Required

2.5 Transient Data

Term 35: Transient Option

Description	Datatype
Declares whether the simulation should be run as transient or steady-state. If run as steady-state, then convergence criteria will be checked after each timestep iteration and the simulation will be terminated when the criteria are satisfied. If run as transient, convergence criteria are not checked. The simulation is run until TEND (see below) is reached. If you omit this entry, the preprocessor will default to running the simulation to steady state.	string
Options	Units
yes—Run as transient no—Run as steady-state	—
Tag	Input Type
{run as transient}	Optional

Term 36: DTMIN

Description	Datatype
The minimum allowable timestep size for the simulation.	Float
Options	Units
—	s
Tag	Input Type
—	Required

Term 37: DTMAX

Description	Datatype
The maximum allowable timestep size for the simulation.	Float
Options	Units
—	s
Tag	Input Type
—	Required

Term 38: TEND

Description	Datatype
The time to end the transient simulation. Note, though, that the preprocessor, by default, will setup the simulation to be a pseudo-transient, which will cause COBRA-TF to ignore the TEND value. Instead, COBRA-TF will continue the transient until convergence is reached.	Float
Options	Units
—	s
Tag	Input Type
—	Required

Term 39: RTWFP

Description	Datatype
The ratio of the conduction equation timestep size to the fluid solution timestep size. This can generally be large for steady-state solutions (e.g. 1000).	Float
Options	Units
—	—
Tag	Input Type
—	Required

Term 40: MAXITS

Description	Datatype
The maximum iterations to take before ending the pseudo-transient. If COBRA-TF doesn't reach convergence by this number of iterations, the simulation is ended and simulation results are printed. A notice of the convergence problems will be passed to the user via the standard output.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

2.6 Convergence Criteria

Term 41: Global Energy Balance

Description	Datatype
The pseudo-transient option convergence criteria for the global energy balance. See the Theory Manual for further information on the definition of this term. Note that the convergence criteria terms are optional, but you must enter them all or none of them. If you do not enter them, they will be set to default values. The default for global energy balance is 0.01%. If you are providing convergence criteria, you must precede this term with the tag, {convergence criteria}.	Float
Options	Units
—	%
Tag	Input Type
{convergence criteria}	Optional

Term 42: Global Mass Balance

Description	Datatype
The pseudo-transient option convergence criteria for the global mass balance. See the Theory manual for further information on the definition of this term. Note that you must enter all 5 convergence criteria terms, or none at all. If none are entered, they will be set to default values. The global mass balance default is 0.01%	Float
Options	Units
—	%
Tag	Input Type
—	Optional

Term 43: Fluid Energy Storage

Description	Datatype
The pseudo-transient option convergence criteria for the fluid energy storage. See the Theory manual for further information on the definition of this term. Note that you must enter all 5 convergence criteria terms, or none at all. If none are entered, they will be set to default values. The default for fluid energy storage is 0.5%	Float
Options	Units
—	%
Tag	Input Type
—	Optional

Term 44: Solid Energy Storage

Description	Datatype
The pseudo-transient option convergence criteria for the solid energy storage. See the Theory manual for further information on the definition of this term. Note that you must enter all 5 convergence criteria terms, or none at all. If none are entered, they will be set to default values. The default for solid energy storage is 0.5%	Float
Options	Units
—	%
Tag	Input Type
—	Optional

Term 45: Mass Storage

Description	Datatype
The pseudo-transient option convergence criteria for the mass storage. See the Theory manual for further information on the definition of this term. Note that you must enter all 5 convergence criteria terms, or none at all. If none are entered, they will be set to default values. The default for mass storage is 0.5%	Float
Options	Units
—	%
Tag	Input Type
—	Optional

2.7 Output Options

Term 46: Channel Edits

Description	Datatype
Defines if channel data should printed. If so, it will be printed to deck.out and channels.out. If not, deck.inp will still be created, but the channels.out files will not be opened. Note that the VTK file will still be created.	Integer
Options	Units
0—Do not print channel data 1—Print the channel data	—
Tag	Input Type
{edit channels}	Optional

Term 47: Gap Edits

Description	Datatype
Defines if gap data should printed. If so, it will be printed to deck.out and gaps.out. If not, deck.inp will still be created, but the gaps.out files will not be opened. Note that you cannot display only gaps data. If you choose only gaps data, the preprocessor will default to printing both channels and gaps data. Also, the VTK file will still be created if you turn off gaps output.	Integer
Options	Units
0—Do not print gap data 1—Print the gap data	—
Tag	Input Type
{edit gaps}	Optional

Term 48: Rod Edits

Description	Datatype
Defines if rod data should printed. If so, it will be printed to deck.out. If not, deck.inp will still be created, though no rod data will be printed in the file. The VTK file will still be created if you turn off rods output.	Integer
Options	Units
0—Do not print rod data 1—Print the rod data	—
Tag	Input Type
{edit rods}	Optional

Term 49: DNB Edits

Description	Datatype
Defines if DNB data should printed. If so, it will be printed to dnb.out. If not, the dnb.out file will not be created. The VTK file will still be created if you turn off DNB output.	Integer
Options	Units
0—Do not print DNB data 1—Print the DNB data	—
Tag	Input Type
{edit dnb}	Optional

Term 50: Rod VTK Option

Description	Datatype
Defines if the rod VTK file should be generated. If so, CTF will produce a VTK file for the rods in addition to the fluids. This allows the user to visualize selected rod data. Defaults to not generating the file if this card is omitted.	Integer
Options	Units
0—Do not print rod VTK file 1—Print the rod VTK file	—
Tag	Input Type
{rods vtk}	Optional

Term 51: HDF5 Edits

Description	Datatype
Defines if the HDF5 file should be produced. Default is to produce the file.	Integer
Options	Units
0—Do not print HDF5 file 1—Print the HDF5 file	—
Tag	Input Type
{edit hdf5}	Optional

Term 52: Convergence Parameter Edits

Description	Datatype
Defines if the convergence file should be produced. The convergence file prints the 5 parameters used to determine if steady-state has been reached at each iteration of the simulation. Default is to produce the file.	Integer
Options	Units
0—Do not print file 1—Print the file	—
Tag	Input Type
{edit convergence}	Optional

CHAPTER 3

GEO.INP FILE

The *geo* file gives information on the global, core-wide geometry of the model. The user gives information about the number of assemblies in the core and how they are placed. Additionally, the user also specifies the meshing of different axial zones.

The user gives information on assembly placement primarily via two means. First, the user specifies the dimensions of the assembly map. The assembly map is essentially a top-view of the core being modeled. It is comprised of integer values. An integer greater than 0 represents an assembly and 0's representing an empty region. The map must be square or rectangular in shape, which is why it takes 0's for denoting the absence of an assembly. An example of a map for four assemblies placed in a square configuration is as follows:

```
**1 2
 1 1 1
 2 1 1
```

Note that the first line is a header line that labels the columns of the map. Likewise, the first column is a header column that label the rows of the map. The pre-processor ignores the first row and first column. The 1's in the map denote four assemblies, viewed from above. It is necessary that, before such a map is input, the user provide the dimensions of the map. The dimensions are given as the number of columns (X) by the number of rows (Y). For the above example, the dimensions would be "2 2".

It is important to note that the core map *should not* be padded with zeros on any of the sides. Therefore, using the following core map to represent the 2x2 geometry would cause the preprocessor to fail.

```

**1 2 3 4
1 0 0 0 0
2 0 1 1 0
3 0 1 1 0
4 0 0 0 0

```

Another important matter, for the sake of visualizing the mesh, is that of the origin of the model. The origin is considered to be at the center corner of the core. This holds true whether you are modeling a full core, a single assembly, or even a single pin. For example, the origin will be placed at the center of a model of a single pin. It will be placed at the center of the center rod in a 17x17 assembly. For an 18x18 assembly, however, it will be placed in the center of the center channel (even-rod-lattice assemblies have a channel at the center of the model). If utilizing a symmetry option to model the core, the origin is still placed in the center of the core. In the case of odd-rod-lattices, the top-left corner of the model will be a rod and the origin will be placed at the center of this rod. In the case of even-rod-lattices, the top-left corner of the model will be a channel, and the origin will be placed at the center of this channel (the upper-left corner of the channel).

This origin is used in defining the location of the scalar-mesh cells. It is of primary importance when running coupled simulations that require the location of entities be known for performing data transfers and for visualizing results when writing them to the VTK file. Other than those two cases, CTF does not have a concept of an origin, nor does it utilize the absolute location of model entities.

A single assembly model would be created most simply by specifying a map with 1x1 dimensions that contains a single 1 as its entry. On the other hand, more complicated maps can be generated, like a map for a PWR core. An example for a map that would retain the actual geometry of the core is shown as follows:

```

** 1  2  3  4  5  6  7  8  9 10 11 12 13 14
1  0  0  0  0  0  0  1  1  0  0  0  0  0  0
2  0  0  0  0  1  1  1  1  1  1  0  0  0  0
3  0  0  0  1  1  1  1  1  1  1  1  0  0  0
4  0  0  1  1  1  1  1  1  1  1  1  0  0  0
5  0  1  1  1  1  1  1  1  1  1  1  1  1  0
6  0  1  1  1  1  1  1  1  1  1  1  1  1  0
7  1  1  1  1  1  1  1  1  1  1  1  1  1  1
8  1  1  1  1  1  1  1  1  1  1  1  1  1  1
9  0  1  1  1  1  1  1  1  1  1  1  1  1  0
10 0  1  1  1  1  1  1  1  1  1  1  1  1  0
11 0  0  1  1  1  1  1  1  1  1  1  1  0  0
12 0  0  0  1  1  1  1  1  1  1  1  0  0  0
13 0  0  0  0  1  1  1  1  1  1  0  0  0  0
14 0  0  0  0  0  0  1  1  0  0  0  0  0  0

```

The dimensions of the core map, for this case, would be “14 14”.

If all assemblies have identical geometry, then the map can just consist of 0’s and 1’s; however, you also have the option to make a model that contains assemblies with different geometries. Currently, the parameters that can differ between assemblies include:

1. Clad outside diameter
2. Clad inside diameter
3. Fuel pellet diameter
4. Pellet theoretical density

Items 3 and 4 only matter for nuclear fuel rods. No other characteristics (e.g., number of radial nodes per pellet, placement of guide tubes, gap conductance, pin pitch, bundle pitch, axial noding, spacer grid geometry, spacer grid placement, etc.) may be changed from assembly-to-assembly in the model. Using different assembly types in the model requires two things: (1) you should specify the number of different assembly types that will be specified in the model at the top of *geo.inp*, and (2) you should identify the different assemblies in the core assembly map in *geo.inp*. Then, in the assembly input file, *assem.inp*, you will be required to give an individual set of powered rod and guide tube characteristics for each different assembly type that was specified. The *assem.inp* chapter gives more details on how to do this.

An option also exists on whether the core should be modeled using quarter- or eighth-core symmetry. If this option is left out, no symmetry will be used. If you wish to model the core using symmetry, you need to enter everything as if

the full core were being modeled and then select the quarter-core option (4) or the eighth-core option (8). The preprocessor will take care of re-dimensioning the problem, dividing up the boundary conditions and getting rid of the parts of the core assembly map which are not needed.

The second thing modeled in the *geo* file is the axial zoning. In CTF, all scalar-mesh cells in a specific axial zone have the same axial height. Typically, the user would use an axial zone to fit a discrete number of scalar-mesh cells in a certain region of the core. For example, a region may consist of a grid span (a region between two successive spacer grids). In CTF, loss coefficients are applied on the faces of scalar-mesh cells (the centers of momentum-mesh cells), so we can put an integer number of scalar-mesh cells between two successive spacer grids.

A zone is specified by giving two pieces of information: first, the top of the zone is given and, second, the number of scalar-mesh cells in that zone is given. For the first zone, for example, the user would specify the location of the top of the zone—the pre-processor assumes its bottom to be the beginning of the model, which is 0.0 mm. The user could specify 4 scalar-mesh cells in that first zone and the pre-processor would determine the size of the cells necessary to achieve this discretization. Then the user would specify the end of the second zone on the following input line in the *geo* file along with the number of scalar-mesh cells that should exist in that zone. The pre-processor will use the top of the first zone as the bottom of the second zone. Again, it will determine the axial height of the scalar-mesh cells in Zone 2 necessary to fit the requested number of cells into the zone.

Note that the height of the top of the last axial zone will be used as the total model length by the preprocessor.

The remainder of this chapter gives the specific information for each term that is required by the *geo* file, similar to how was done for Chapter 2.

Term 53: Number of Fuel Assemblies

Description	Datatype
This specifies the total number of fuel assemblies in the model. Do not count the zeros in the assembly map.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 54: Number of Fuel Assembly Types

Description	Datatype
This is the number of different fuel assembly geometry types that will be specified in the core map and the <i>assem.inp</i> file.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 55: Core Dimensions

Description	Datatype
The number of rows and columns in the core map	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 56: Symmetry Option

Description	Datatype
Whether to use symmetry or not. This input is optional. If you leave it out, you will get a warning message and the preprocessor will assume that you do not want to use symmetry.	Integer
Options	Units
1—Model the map as given (no symmetry) 4—Model the map using quarter-core mirror symmetry 5—Model the map using quarter-core rotational symmetry 8—Mode the map using eighth-core symmetry	—
Tag	Input Type
{symmetry option}	Optional

Term 57: Fuel Assembly Map

Description	Datatype
A map of the assemblies, looking at the core from above. A non-zero number specifies that an assembly resides in that location and a 0 specifies that there is not an assembly. Different non-zero numbers represent different assembly types that will be specified in <i>assem.inp</i> . The dimensions of the map must agree with the dimensions given in the previous term. Also, the total number of assemblies specified in the map, must agree with the number of assemblies specified at the beginning of the <i>geo</i> file.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 58: NZONES

Description	Datatype
The number of axial zones that the model is split up into.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 59: Zone Location and Discretization Data

Description	Datatype
NZONES lines are entered here. In each line, two pieces of data are entered. First, a float value is entered that specifies the location of the top of the zone and the bottom of the next zone. Second, an integer value is entered that specifies how many scalar mesh cells the current zone will be broken up into. “Current zone” refers to the zone the resides below the location that was entered for the first term. Note that the last location value entered must be equal to the top of the model.	Float & Integer
Options	Units
—	mm & —
Tag	Input Type
—	Conditional

Term 60: Name of Assembly File

Description	Datatype
<p>This is the name of the <i>assembly</i> file that is next discussed in Chapter 4. The file may have any name the user wishes. This allows for the user to create many fuel-assembly files, all modeling a specific assembly type. Changing the fuel-assembly type used in your core model is as simple as changing the name of the file at the end of the <i>geo</i> file. Note that the <i>assembly</i> file should be located in the same directory as the pre-processor utility executable and the other three input files.</p>	Character
Options	Units
—	—
Tag	Input Type
—	Required

CHAPTER 4

ASSEMBLY.INP FILE

The *assembly* file specifies dimensions of the fuel rods and guide tubes that exist in an assembly. It also details information on their placement in the rod lattice. The final information entered on the assembly is the placement and loss coefficient of spacer grids. Note that, while this document refers to this file as the “assembly” file, its name may be anything the user chooses. To use the file, the user must then enter its name in the *geo* file, as was discussed in Chapter 3.

The preprocessor can model three types of rods—nuclear fuel rods, electric heater rods, and guide tubes. Guide tubes and electric heater tubes are basically the same except for the fact that electric heater tubes are powered and guide tubes are not. The guide tube and electric heater tube input only require the inside diameter and outside diameter of the tube and the tube material. Note, though, that the tube material property is currently not used at all in the preprocessor. The preprocessor will always generate material properties for Zircalloy in the input deck.

The nuclear fuel rods input requires the number of radial nodes to include in the pellet for the heat conduction solution, the pellet outer diameter, the pellet theoretical density, the constant gap conductance value, and the clad material. Again, while the clad material is required, it will always be Zircalloy no matter what is put for that parameter. Note that you are not allowed to have *both* nuclear fuel rods and electric heater rods in the same model—it’s one or the other. Guide tubes, however, can be mixed with either of the active heater types.

Note from *geo.inp*, that it is possible to have different geometry assembly types

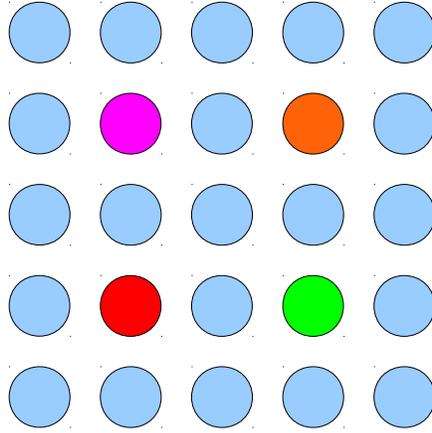


Figure 4.1: Example of a 5x5 fuel rod lattice

in the model. If you specified that there are more than one assembly type, you are required to give an equal number of heater types and guide tube types in the *assem.inp* file. For example, if you are using three different assembly types, and your active heater in the model is a nuclear fuel rod type, then you must have three sets of nuclear fuel rod data and three sets of guide tube data. Also, each set of data must be tagged, which makes it less likely that you will make an input error. Additionally, when you use multiple fuel rod types, you also need to tag where the heater types start and where the guide tube types start, as described later in the chapter.

Most of the terms in this file will be easily understood, but additional information is given here on the specification of guide tube placement. The guide tubes are placed within a “fuel-rod lattice”. For example, if a 5x5 rod bundle is being modeled, then a lattice of fuel rods exists which has 5 rods in each of 2 directions when viewed from above. The placement of a guide tube in this lattice is specified by stating the row and column location in that lattice. The origin starts at the upper-left, just as it did for specifying the assembly map in the *geo* file and the first rod location would be “1 1”.

For example, a 5x5 bundle is shown from above in Figure 4.1. In this figure, guide tubes are shown in red, green, pink, and orange. To communicate the location of the red guide tube to the pre-processor, we would specify its X and Y location as “2 2”. Likewise, the green tube is positioned at “4 2”, the pink tube is positioned at “2 4”, and the orange tube is positioned at “4 4”.

The remainder of this chapter gives the information about the specific terms required by the *assembly* file.

Term 61: Number of Fuel Rods

Description	Datatype
The number of fuel rods in the assembly (do not count the guide tubes).	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 62: Size of the Fuel Array

Description	Datatype
The number of rods in one row. In the example shown in Figure 4.1, this value would be 5.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 63: Number of Guide Tubes or Water Rods

Description	Datatype
The number of guide tubes or water rods in one assembly.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 64: Active Length of Model

Description	Datatype
The heated length of the model.	Float
Options	Units
—	mm
Tag	Input Type
—	Required

Term 65: Start of Active Length Region

Description	Datatype
The axial location where the active region begins. If you do not enter the start of active length, the preprocessor will assume that the beginning of active length is at 0.0 mm.	Float
Options	Units
—	mm
Tag	Input Type
active region start	Optional

Term 66: Bundle Pitch

Description	Datatype
The pitch between assemblies. For an inner assembly, this would also be the length of its boundaries when looking at it from above.	Float
Options	Units
—	mm
Tag	Input Type
—	Required

Term 67: Assembly Casing or Can?

Description	Datatype
This designates whether there is a solid surface surrounding the assembly that would increase the wetted perimeter of side and corner channels. This can occur for test bundles that are surrounded by a casing.	Integer
Options	Units
0–No casing 1–Casing present	—
Tag	Input Type
—	Required

Term 68: Heater Element Type

Description	Datatype
This designates whether the heater elements will be nuclear fuel rods or electric heater tubes.	Integer
Options	Units
0–Nuclear Fuel Rods 1–Electric Heater Tubes	—
Tag	Input Type
—	Required

Term 69: Conduction Model Flag

Description	Datatype
How conduction should be modeled in the rods.	Integer
Options	Units
0—No conduction	—
1—Radial conduction	
2—Radial and axial conduction	
3—Radial, axial, and azimuthal conduction	
Tag	Input Type
—	Required

4.1 Heater Types {heater types}

This section of the deck will contain information about the nuclear fuel rods or the electric heater tubes, depending on your choice of heater elements. When modeling multiple assembly types, you *must* enter as many heater data sets as there are assembly types. Also, you *must* start this section with the {heater types} tag. Then, each data set should be started with a numerical tag equal to the data set being read in. For example, if you are modeling three assembly types, and you are using nuclear fuel rods, you should have three sets of nuclear fuel rod data. The first set will be tagged with {1}, the second set will be tagged with {2}, and the third set will be tagged with {3}.

4.1.1 Nuclear Fuel Rods

Term 70: Fuel Pellet Diameter

Description	Datatype
The diameter of the fuel pellet inside the rod.	Float
Options	Units
—	mm
Tag	Input Type
—	Conditional

Term 71: Radial Nodes in the Fuel Pellet

Description	Datatype
The number of nodes that are placed inside the fuel pellet. The conduction equation is solved for each of these nodes. A temperature will be calculated for each node in the fuel pellet.	Integer
Options	Units
—	—
Tag	Input Type
—	Conditional

Term 72: Cladding Inner Diameter

Description	Datatype
The diameter of the inside surface of the fuel cladding.	Float
Options	Units
—	mm
Tag	Input Type
—	Conditional

Term 73: Cladding Outside Diameter

Description	Datatype
The diameter of the outside surface of the fuel rod.	Float
Options	Units
—	mm
Tag	Input Type
—	Conditional

Term 74: Pin Pitch

Description	Datatype
The distance between the centers of fuel rods and guide tubes in the rod lattice.	Float
Options	Units
—	mm
Tag	Input Type
—	Conditional

Term 75: Theoretical Density of Fuel Pellet

Description	Datatype
The density of the fuel pellet material. A typical value is 95 %.	Float
Options	Units
—	%
Tag	Input Type
—	Conditional

Term 76: Gap Conductivity

Description	Datatype
The pre-processor assumes constant conductivity of the fill gas that exists between the fuel pellet and cladding inside the fuel rod. This conductivity is entered here. A common assumed value is $1000 \text{ BTU/hr}\cdot\text{ft}^2\cdot\text{F}$ or $5678.3 \text{ W/m}^2\cdot\text{K}$.	Float
Options	Units
—	$\text{W/m}^2\cdot\text{K}$
Tag	Input Type
—	Conditional

Term 77: Cladding Material

Description	Datatype
The material of the cladding. Currently, only Zircaloy can be entered for this term. The standard Zircaloy material property tables are then used in CTF. You may use a different material by specifying a material property table in Group 9 of the input deck and then specifying values for that table in Group 10.	Character
Options	Units
Zircaloy	—
Tag	Input Type
—	Conditional

4.1.2 Electric Heater Tubes

Term 78: Cladding Inner Diameter

Description	Datatype
The diameter of the inside surface of the fuel cladding.	Float
Options	Units
—	mm
Tag	Input Type
—	Conditional

Term 79: Cladding Outside Diameter

Description	Datatype
The diameter of the outside surface of the fuel rod.	Float
Options	Units
—	mm
Tag	Input Type
—	Conditional

Term 80: Pin Pitch

Description	Datatype
The distance between the centers of fuel rods and guide tubes in the rod lattice.	Float
Options	Units
—	mm
Tag	Input Type
—	Conditional

4.2 Guide Tubes {guide tube types}

If modeling multiple assembly geometry types, you must start this section with the {guide tube types} tag. You must enter an equal number of sets of guide tube data as there are assembly types. Each data set should be tagged with a numerical tag. For example, if modeling three fuel assembly types, you should have three sets of guide tube data; the first set should be tagged with {1}, the second should be tagged with {2}, and the third should be tagged with {3}. You should omit this section if you specified that there were no guide tubes.

Term 81: Inner Diameter of the Guide Tube

Description	Datatype
The guide tube is modeled as a tube geometry. This is the diameter of the inside surface.	Float
Options	Units
—	mm
Tag	Input Type
—	Conditional

Term 82: Outer Diameter of the Guide Tube

Description	Datatype
Diameter of the outside surface of the guide tube.	Float
Options	Units
—	mm
Tag	Input Type
—	Conditional

Term 83: Guide Tube Material

Description	Datatype
The material from which the guide tube is made. Currently, only Zircaloy is supported, as was the case for the cladding material.	Character
Options	Units
Zircaloy	—
Tag	Input Type
—	Conditional

4.3 Guide Tube Positions

Term 84: Guide Tube Positions

Description	Datatype
Each line of input represents the X and Y position of a single guide tube. This item should include as many lines of input as there were guide tubes. Each line contains two terms—first the X position of the guide tube in the rod lattice and, second, the Y position of the guide tube in the rod lattice. Enter nothing for this term if zero guide tubes were specified.	Integer & Integer
Options	Units
—	—
Tag	Input Type
—	Conditional

4.4 Nozzles

Term 85: Data on assembly nozzles

Description	Datatype
This optional parameter is included if the model includes the assembly nozzles. If you enter this tag, you must enter two lines of data: the first gives the location, height, and loss coefficient of the bottom nozzle and the second gives the location, height, and loss coefficient of the top nozzle. Each line should contain: an integer that identifies the nozzle (not used), the axial location of the bottom of the nozzle, the axial location of the top of the nozzle, and the loss coefficient of the nozzle. WARNING: This is an experimental feature and will make a very crude model of the nozzles. The entire model will be one axial section, which means the rods and sub-channels will extend right through the two nozzles.	Integer & Float & Float & Float
Options	Units
—	— & mm & mm & mm
Tag	Input Type
nozzles	Optional

4.5 Spacer Grids

Term 86: Number of Spacer Grids

Description	Datatype
The number of spacer grids contained in the assembly.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 87: Spacer Grid Data

Description	Datatype
<p>Each line of input specifies the information for one spacer grid. There should be a line of input for each spacer grid. For example, if you specified that there were 7 spacers, then you must enter 7 lines of data here. Each line contains 4 entries. The first is the spacer grid number (enter from 1 to the total number of spacer grids here), the second entry is the location of the bottom (upstream) side of the spacer grid, the third entry is a dummy value (you may enter any value here, as it is not used for anything), and the fourth is the loss coefficient of the grid.</p>	<p>Integer & Float & Float & Float</p>
Options	Units
—	— & mm & —
Tag	Input Type
—	Conditional

CHAPTER 5

POWER.INP FILE

The final pre-processor input file is the *power* file(s). If you said there would be multiple power files in the *control* file, then you need to provide that number of *power* files. Each individual file gives power data for a different point in the transient. The files should be numbered using the following scheme:

`power.inp.<num>`

The key, `<num>` is varied from 1 to the total number of *power* files entered in the *control* file. Otherwise, if you are not providing multiple *power* files, then a single file named *power.inp* should be provided.

In these files, the user specifies the total model power and its distribution in this file. There are primarily four pieces of information that are entered. First, the total core power is given, which the pre-processor uses to calculate linear heat rate that will be delivered to the CTF input deck.

The second piece of information is the axial distribution of power. This is entered via a power factor table which specifies a power factor for explicit axial locations in the core. The axial power factor table is applied core-wide. The table can contain as many elements as you desire and as few as two for a uniform power profile. Note it is possible to model unheated regions in your model. The axial power factor table should only cover the active region. Therefore, if you are modeling a core that has a 0.5 m unheated region, followed by a 3 m heated region, followed by another 0.5 m unheated region, then the active length is 3 m and the total length is 4 m. Your axial power factor table, in this case, should

go from 0.0 m to 3.0 m. It does not start at 0.5 m.

The third piece of information is the core radial power profile. This specifies the power factor on an assembly level. It is given as a map just like was done in the *geo* file. In fact, the radial power factor map must be the same dimensions as the assembly map was in the *geo* file. Note that this radial power factor map must normalize to 1.0 within a tolerance of 0.1 %.

The fourth piece of information is the assembly radial power profile(s). This set of information is optional. If you omit it, the preprocessor will assume a uniform power factor map for each assembly in the model. If you include just one assembly power factor map, that map will be applied for each assembly in the model. There is also an option to specify different assembly power factor maps for each assembly. If this last option is employed, it is necessary to specify a map of the assembly power factor maps. This map is the same dimensions as the core, containing 0's where assemblies *are not* present and non-zero integers where assemblies *are* present. The non-zero integers are indices that identify assembly power factor maps that are given after the core map of power factor maps.

If you choose to enter assembly power factor maps, you must follow two rules:

1. All guide tubes must get a power factor of 0.0, since they are not powered
2. The non-guide tube power factors must normalize to 1.0

For Rule 2, this does not mean that the rod power factors *inside* each assembly must normalize to 1.0; rather, the non-guide-tube power factors of the entire model must normalize to 1.0.

If you are providing multiple *power.inp* files for different points in the transient, you are limited to what types of data you can change throughout the transient. You are able to vary the total core power, the power factors in the axial power factor table, the power factors in the core radial power factor map, and the power factors in the local assembly power factor maps. You cannot change the number of entries in any of the maps, however. They must always remain the same shape and size. Furthermore, if providing power factor maps on a per-assembly basis, you must provide the same number of assembly maps at each point in the transient.

Term 88: Transient Time for this Power Data

Description	Datatype
This is the point in the transient that this power data is applied. Enter this piece of information only if you are specifying multiple <i>power.inp</i> files.	Float
Options	Units
—	s
Tag	Input Type
{transient time}	Conditional

Term 89: Core Power

Description	Datatype
The total power provided to the model.	Float
Options	Units
—	MW
Tag	Input Type
—	Required

Term 90: Axial Power Factor Elements

Description	Datatype
This is the number of axial locations that a power factor will be specified for in the axial power factor table that follows.	Integer
Options	Units
—	—
Tag	Input Type
—	Required

Term 91: Axial Power Factor Table

Description	Datatype
Each line of this table gives two pieces of information. First, the axial location is specified and, second, the power factor at that axial location is specified. The table should have as many lines as were specified in the previous term. Note that the power factors in this table should be normalized to average core power.	Float & Float
Options	Units
—	mm & —
Tag	Input Type
—	Required

Term 92: Radial Core Power Factor Map

Description	Datatype
This is a map that has the same dimensions as the assembly map in the <i>geo</i> input file. A power factor is specified for each assembly that was specified in the assembly map. The power factors should be normalized to average core power.	Float
Options	Units
—	—
Tag	Input Type
—	Required

5.1 Assembly-level Power Factor Maps

5.1.1 No Map

If you don't put anything for this term, a uniform power factor profile is assumed for every bundle in the model.

5.1.2 Single Map

If a single map is specified, then it will be applied to every single bundle in the model. Note that this map must normalize to 1.0. However, guide tubes do not count in the normalization. For example, if you have a 3x3 rod bundle with 1 guide tube, it will have 8 heated rods. If you wanted to manually set a power factor of 1.0 for each rod, you would set set this power map as follows:

```
1.0 1.0 1.0
1.0 0.0 1.0
1.0 1.0 1.0
```

You would not set it like this:

```
1.125 1.125 1.125
1.125 0.000 1.125
1.125 1.125 1.125
```

While COBRA-TF requires the latter type of input, the preprocessor takes the former type of input. This is because the former type of input is more intuitive. The preprocessor will take care of re-normalizing the map so that it works in COBRA-TF.

Term 93: Radial Assembly Power Factor Map

Description	Datatype
This is a map that has the same dimensions as a bundle in the core. It specifies the power factor of each rod in the assembly. Currently, only one assembly power factor map may be entered, so it must normalize to 1.0. Make sure to put a 0.0 where there are guide tubes. The preprocessor will multiply the assembly power factor (specified in the previous power factor map) by each rod power factor specified in this map. Note that this is an optional entry. If you do not enter an assembly power factor map, the pre-processor will take the assembly power factor and apply it uniformly across all rods in the assembly.	Float
Options	Units
—	—
Tag	Input Type
—	Optional

5.1.3 Multiple Assembly Power Factor Maps {number of assembly maps}

To enter multiple assembly power factor maps, tag this section with {number of assembly maps}.

Term 94: Number of Assembly Power Factor Maps

Description	Datatype
Enter the number of different assembly power factor maps that will be entered.	Integer
Options	Units
—	—
Tag	Input Type
{number of assembly maps}	Optional

Term 95: Map of Assembly Power Factor Maps

Description	Datatype
This map should match the core assembly map. Enter a 0 where this is no assembly and an integer where there is an assembly. Each integer will identify an assembly power factor map to be applied to that assembly.	Integer
Options	Units
—	—
Tag	Input Type
{assembly power factor index map}	Optional

Enter the assembly power factor maps in this section. You must enter as many maps as you said would be present in your model. Each map should have the same dimensions as the rod lattice in an assembly. Each map must be tagged with a numerical tag. For example, if you specified that there would be three assembly power factor maps, you must enter three maps here—the first map should be preceded by a {1} tag, the second map should be preceded by a {2} tag, and the third map should be preceded by a {3} tag.

Term 96: Assembly Power Factor Maps

Description	Datatype
Enter the power factors for each rod in the assembly. This map should have the same dimensions as the rod lattice in an assembly.	Float
Options	Units
—	—
Tag	Input Type
Number of the map	Optional