INL/EXT-18-44979-Revision-0

# RELAP-7 Input Manual

David Andrs, Joshua Hansel, Russell Johns, Richard Martineau, Gregg Swindlehurst

March 2018

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

INL/EXT-18-44979-Revision-0

# RELAP-7 Input Manual

David Andrs, Joshua Hansel, Russell Johns, Richard Martineau, Gregg Swindlehurst

March 2018

**Idaho National Laboratory**
**Idaho Falls, Idaho 83415**

**http://www.inl.gov**

**Prepared for the**
**U.S. Department of Energy**

**Under DOE Idaho Operations Office**
**Contract DE-AC07-05ID14517**

INL/EXT-18-44979

# RELAP-7 Input Manual

Gregg Swindlehurst

David Andrs

Joshua Hansel

Russell Johns

Richard Martineau

March 2018

U.S. Department of Energy Office of Nuclear Energy

**DISCLAIMER**

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/EXT-18-XXXXX

# RELAP-7 Input Manual

Gregg Swindlehurst, David Andrs, Joshua Hansel,
Russell Johns, Richard Martineau

March 2018

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

## Table of Contents

# SUMMARY

The RELAP-7 code is the next generation nuclear reactor system safety analysis code being developed at the Idaho National Laboratory (INL). The code is based on the INL's modern scientific software development framework, MOOSE (Multi-Physics Object Oriented Simulation Environment). The overall design goal of RELAP-7 is to take advantage of the previous thirty years of advancements in computer architecture, software design, numerical integration methods, and physical models. The end result will be a reactor systems analysis capability that retains and improves upon RELAP5's capability and extends the analysis capability for all reactor system simulation scenarios.

RELAP-7 will be the next generation tool in the RELAP reactor safety/systems analysis application series. The key to the success of RELAP-7 is the simultaneous advancement of physical models, numerical methods, and software design while maintaining a solid user perspective. Physical models include both PDEs (Partial Differential Equations) and ODEs (Ordinary Differential Equations) and experimental based closure models. RELAP-7 utilizes well-posed governing equations for compressible two-phase flow, which can be strictly verified in a modern verification and validation effort. Closure models used in the TRACE code have been selected to reflect the progress made during the past three decades. RELAP-7 uses modern numerical methods, which allow implicit time integration, second-order schemes in both time and space, and strongly coupled multi-physics.

RELAP-7 is written with object oriented programming language C++. By using the MOOSE development environment, the RELAP-7 code is developed by following the same modern software design paradigms used for other MOOSE development efforts. The code is easy to read, develop, maintain, and couple with other codes. Most importantly, the modern software design allows the RELAP-7 code to evolve efficiently with time. MOOSE is an HPC development and runtime framework for solving computational engineering problems in a well planned, managed, and coordinated way. By leveraging millions of lines of open source software packages, such as PETSc (a nonlinear solver developed at Argonne National Laboratory) and LibMesh (a Finite Element Analysis package developed at University of Texas), MOOSE reduces the expense and time required to develop new applications. MOOSE provides numerical integration methods and mesh management for parallel computation. Therefore RELAP-7 code developers have been able to focus more upon the physics and user interface capability. There are currently over 20 different MOOSE based applications ranging from 3-D transient neutron transport, detailed 3-D transient fuel performance analysis, to long-term material aging. Multi-physics and multi-dimensional analysis capabilities, such as radiation transport and fuel performance, can be obtained by coupling RELAP-7 and other MOOSE-based applications through MOOSE and by leveraging with capabilities developed by other DOE programs. This allows restricting the focus of RELAP-7 to systems analysis type simulations and gives priority to retain and significantly extend RELAP5's capabilities.

This RELAP-7 Input Manual provides the users with a description of the input parameters for which character strings or numerical values are provided to build a RELAP-7 input file. Some of the input parameters are required for any RELAP-7 input file, and many others represent optional input file parameters that can be provided at the discretion of the user. Some of the input parameters have default values that can be superseded by user specified values.

The RELAP-7 Theory Manual describes the theoretical basis, the development of the equations and models used in the simulation, and the numerical solution methods. The RELAP-7 User's Guide focuses on assisting users in developing input models by use of a progression of sample problems of increasing complexity. These two manuals, in concert with this RELAP-7 Input Manual, provide the necessary information to apply the RELAP-7 code to system thermal-hydraulic analysis.

# 1.0 INPUT FILE DESCRIPTION

RELAP-7 uses a block-structured input file syntax, with each block having a standard name to indicate the required and optional input data. Each block name is identified with brackets (“[standard name]“) at the beginning of the block, and empty brackets (“[ ]”) at the end.

Some input file blocks may contain sub-blocks identified similarly with (“[./name]”at the beginning of the sub-block and “[../]”) at the end. Each sub-block must have an unique name (i.e. cannot use reserved names such as ‘pipe’ – e.g. need to use ‘pipeXX”) when compared with all other subblocks in each specified block. Each sub-block also has a standard type identified with (“type = standard name”) to indicate the sub-block specific input file parameters.

Input files are generally specified using syntax that consists of parameter and value pairs with an “equal sign” between them. The parameter is a block-specific and sub-block-specific standard string, and the value may be a string, an integer, a real number, or a list of strings, integers, or real numbers. Lists are given in single quotes and are separated by whitespace.

If a “parameter = value” pair is repeated in the input file, then the last appearance will be used.

Comment text is preceded by a hash symbol (“#”) in the input file.

RELAP-7 uses SI units for all input file parameters.

The open source “Atom” editor (https://atom.io) is recommended for editing RELAP-7 input files as it has plugins that assist with input file creation and editing. Drop-downs of parameter lists show the available parameters, the required parameters, default values, and parameter descriptions.

The following report sections have detailed descriptions of input file parameters for each input file block, and for commonly used sub-blocks. Additional parameters that are not commonly used are not included in the table. A file with a complete list of RELAP-7 input file parameters can be obtained via the command (“`./relap-7-opt –dump >filename`”).

# 2.0 GLOBAL PARAMETERS

The global parameters [GlobalParams] block includes those input parameters used by RELAP-7 that can specified generically, such as the choice of stabilization scheme to be used in the numerical solution, and the initial values of pressure , velocity, and temperature of the system being modeled. The generic parameter values specified in the global parameters block are used throughout the model unless they are superseded by specifying those same parameters in any block or subblock. None of the parameters are required. Typically, any input that can be specified generically or mostly generically are included in the [GlobalParams] block to reduce the volume of the input file.

In the following table the input file parameters for the [GlobalParams] block are listed.

| **Input File Parameters**<br>(Note: None are required but those that are typically specified are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| alpha_vapor_bounds | **'0.0001 0.9999'** | Lower and upper bounds imposed on the vapor volume fraction. Non-zero values are required. |
| chf_table | **aecl-ippe-1995** | The lookup table used for critical heat flux. The AECL/IPPE 1995 table is the default. |
| **closures_type** | **simple** | The 'trace' closures are recommended for water. Another option is 'helium'. |
| **gravity** | **'0 0 -9.8'**<br>'0 0 -9.81' | The standard (x, y, z) gravity vector '0 0 -9.81' is recommended |
| heat_exchange_coef_liquid | | A constant heat transfer coefficient for liquid if closures not used |
| heat_exchange_coef_vapor | | A constant heat transfer coefficient for vapor if closures not used |
| initial_T | | Initial temperature for fluid if single-phase fluid properties used |
| **initial_T_liquid** | | Initial temperature for liquid |
| **initial_T_vapor** | | Initial temperature for vapor |
| **initial_alpha_vapor** | | Initial vapor volume fraction for fluid |
| initial_p | | Initial pressure for fluid if single- |

| | | |
|---|---|---|
| | | phase fluid properties used |
| **initial_p_liquid** | | Initial pressure for liquid |
| **initial_p_vapor** | | Initial pressure for vapor |
| initial_vel | | Initial velocity for fluid if single-phase fluid properties used |
| **initial_vel_liquid** | | Initial velocity for liquid |
| **initial_vel_vapor** | | Initial velocity for vapor |
| interface_transfer | **true** | Interface heat and mass transfer on |
| **stabilization** | | User-selected name of the stabilization option specified in the Stabilizations block. |
| wall_mass_transfer | **true** | Turns on the wall boiling model and is recommended |

# 3.0 STABILIZATIONS

The stabilizations [Stabilizations] block creates the stabilization options that can be selected by the GlobalParams block or by Pipe components to stabilize the numerical solution method used in RELAP-7. The stabilization options can differ with regard to the applicability to single-phase or two-phase conditions. More than one option can be created for subsequent selection in the same input file.

Other stabilization options are included in RELAP-7 but are not specified here.

In the following sections the input file parameters that typically require values to be entered (e.g. required and no default values) for each of the [Stabilizations] sub-blocks are listed.

## 3.1 Entropy Viscosity

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block name |
| **type** | **EntropyViscosity** | The entropy viscosity option is applicable to single or two-phase conditions |
| use_first_order | **false** | ‘true” means a first-order scheme is used. ‘false’ means the second-order scheme is used and is the default and recommended. |

## 3.2 Lapidus

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block name |
| **type** | **Lapidus** | The Lapidus option is applicable to single or two-phase conditions |
| cl | **1**<br>(zero to 2) | Coefficient for single phase. Recommended values range from zero to 2 with a default value of 1. |
| cl_liquid | **1**<br>(zero to 2) | Coefficient for liquid phase. Recommended values range from |

|  |  |  |
|---|---|---|
|  |  | zero to 2 with a default value of 1. |
| cl_vapor | **1**<br>(zero to 2) | Coefficient for vapor phase. Recommended values range from zero to 2 with a default value of 1. |
| use_first_order | **false** | ‘true” means a first-order scheme is used. ‘false’ means the second-order scheme is used and is the default and recommended. |

## 3.3 SUPG

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block name |
| **type** | **SUPG** | The Streamline Upwind Petrov Galerkin (SUPG) option is applicable to single-phase conditions |

## 3.4 Pressure

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block name |
| **type** | **StabilizationPressure** | The stabilization pressure option is applicable to single-and two-phase conditions |
| **order** |  | ‘FIRST’ means dissipation is added according to the first derivative (gradient) of pressure. ‘SECOND’ means dissipation is added according to the second derivative (curvature) of pressure. |
| ce | **1.5**<br>(zero to 2) | Coefficient for single phase. Recommended values range from zero to 2 with a default value of |

| | | |
|---|---|---|
| | | 1.5. |
| ce_liquid | **0.5**<br>(zero to 2) | Coefficient for liquid phase. Recommended values range from zero to 2 with a default value of 0.5. |
| ce_vapor | **0.5**<br>(zero to 2) | Coefficient for vapor phase. Recommended values range from zero to 2 with a default value of 0.5. |

# 4.0 FLUID PROPERTIES

The fluid properties [FluidProperties] block creates sub-blocks of various fluid properties coded in RELAP-7 (e.g. water, air/water, helium, or nitrogen) that can then be referencedin other blocks by components including fluid. For water the IAPWS 95 water properties are used. In addition, if only the liquid or steam phase is present, the IAPWS 95 water properties for either single phase can be specified for computational efficiency, but are not necessary.

Properties for other fluids, such as sodium or an ideal gas, are included in RELAP-7 but are not specified here.

In the following table the input file parameters for the [FluidProperties] sub-blocks are listed.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **IAPWS95 7 Equation Fluid Properties** | | |
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **IAPWS957EqnFluidProperties** | This option is typically used for water properties as it is applicable to two-phase conditions |
| **IAPWS95 Liquid Properties** | | |
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **IAPWS95LiquidProperties** | This option can be used if only liquid phase water is modeled |
| **IAPWS95 Vapor Properties** | | |
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **IAPWS957VaporProperties** | This option can be used if only vapor phase water is modeled |
| **Air/Water 7 Equation Fluid Properties** | | |
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **Air/Water7EqnFluidProperties** | This option can be used if an air/water problem is modeled |

| | | |
|---|---|---|
| **Helium Fluid Properties** | | |
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **HeliumFluidProperties** | Specifies helium gas properties for an ideal gas model |
| **R** | 2076.9 | Gas constant |
| beta | | Coefficient of thermal expansion |
| **gamma** | .667 | gamma value (cp/cv) |
| k | | Thermal conductivity |
| mu | | Dynamic viscosity |
| **Nitrogen Fluid Properties** | | |
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **N2FluidProperties** | Specifies nitrogen gas properties for an ideal gas model |
| chi_cp | **1** | Uncertainty coefficient on specific heat at constant pressure |
| chi_k | **1** | Uncertainty coefficient on thermal conductivity |
| chi_mu | **1** | Uncertainty coefficient on viscosity |
| chi_p | **1** | Uncertainty coefficient on pressure |

# 5.0 PRECONDITIONING

The preconditioning block [Preconditioning] specifies the preconditioner to be used by the RELAP-7 code. The single matrix preconditioner (SMP) is the recommended option.

Other preconditioners and options are included in RELAP-7 but are not specified here.

In the following table the Input File Parameters for the SMP preconditioner sub-blocks is listed.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **SMP** | Designates the recommended single matrix preconditioner (SMP) |
| full | true<br>**false** | True is recommended to obtain the full set of couplings |
| solve_type | PJFNK | Preconditioned Jacobian-Free Newton Krylov (PJFNK) is the recommended option |

# 6.0 OUTPUTS

The outputs [Outputs] block specifies the RELAP-7 output files that are generated for run-time or post-processing applications.

Other output options are included in RELAP-7 but are not specified here.

In the following sections the output file format options and parameters for the[Outputs] sub-blocks are listed.

## 6.1 Comma Separated Values

The primary use of the comma separated values (CSV) format output file is for use by the graphical user interface (GUI).

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block name |
| **type** | **csv** | Designate that the output file is in the comma separated values (CSV) format |
| execute_on | **'initial timestep_end'** | Control the execution of the output file. Options are to set to (none\|initial\|linear\|nonlinear\|timestep_end\|timestep_begin\|final\|failed\|custom) to execute only at that moment |
| show | | A list of the variables and postprocessors that should be output to the csv file (may include variables, scalarVariables, and postprocessor names). |
| sync_only | | If true only export results at sync times that execution and outputs are forced to occur |
| sync_times | | Times at which the output and solution are forced to occur |

## 6.2 Exodus

The primary use of the exodus output file is for use by the Paraview post-processor.

| **Input File Parameters** | **Parameter Values** | |
|---|---|---|

| (Note: Required are bolded) | (Note: Defaults are bolded) | |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block name |
| **type** | **Exodus** | Designates that the output file is in the Exodus format |
| execute_on | **'initial timestep_end'** | Control the execution of the output file. Options are to set to (none\|initial\|linear\|nonlinear\|timestep_end\|timestep_begin\|final\|failed\|custom) to execute only at that moment |
| show | | A list of the variables and postprocessors that should be output to the Exodus file (may include Variables, ScalarVariables, and Postprocessor names). |
| sync_only | | Only export results at sync times |
| sync_times | | Times at which the output and solution is forced to occur |
| use_displaced_mesh | **false**<br>true | Specify ‘true’ to always use displaced mesh |

## 6.3 Console

The console output format is viewable on the computer screen.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block name |
| **type** | **console** | Designates that the output file is in the format to be viewed on the console |
| execute_on | **'initial timestep_end initial timestep_begin linear nonlinear failed'** | Control the execution of the output file. Options are to set to (none\|initial\|linear\|nonlinear\|timestep_end\|timestep_begin\|final\|failed\|custom) to execute only at that moment |
| outlier_variable_norms | | If true outlier variable norms will be printed after each solve |
| perf_log | | If true only export results at sync times that execution and outputs |

| | | |
|---|---|---|
| | | are forced to occur |
| show | | Times at which the output and solution are forced to occur |
| sync_only | | Only export results at sync times |
| sync_times | | Times at which the output and solution is forced to occur |

# 7.0 POSTPROCESSORS

The post-processor [Postprocessors] block specifies sub-blocks that calculate values that are used for run-time or post-processing applications.

Other post-processor options are included in RELAP-7 but are not specified here.

In the following sections the input file parameters for the [Postprocessors] sub-blocks are listed.

Post-processors that include the “boundary” parameter refers to [Inlet] and [Outlet] components and to two-dimensional [HeatStructure] components.

Vector post-processors exist in RELAP-7 for applications such as fluid temperatures along the length of a pipe. These are not specified here.

Commonly used post-processors are the [PointValue] that outputs the value of a parameter at a specific location (x, y, z), the [ElementExtremeValue] that outputs a maximum or minimum value of a component parameter, and the [TimeStepSize] that outputs the time step size.

## 7.1 Average Element Size

The average element size postprocessor [AverageElementSize] sub-block computes an average element size (length) for the whole domain. If the component names (blocks) are specified, then the average length is for those components.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | AverageElementSize | Designates an average element size post-processor |
| **variable** | (any value) | This variable is not used here but is currently required so use any value |
| block | | The list of component names to be included in the calculation. If none are specified then all will be included. |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in |

|  |  |  |
|---|---|---|
|  |  | [Outputs] will be used. |

## 7.2 Average Nodal Variable Value

The average nodal variable value [AverageNodalVariavbleValue] sub-block computes an average nodal value of a field variable (e.g. temperature in a pipe) for a component or components.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | AverageNodalVariableValue | Designates an average nodal variable value post-processor |
| **variable** |  | The component parameter whose value is to be averaged |
| boundary |  | The unique name(s) of a boundary condition component(s) (e.g. InletXX). |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs |  | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.3 Change Over Time Step

The change over time step post-processor [ChangeOverTimeStepPostprocessor] sub-block computes the change in a post-processor value (new value minus old value), or the magnitude of its relative change (new value minus old value, divided by old value), over a time step or over the entire transient.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | ChangeOverTimeStepPostprocessor | Designates a change over time step post-processor |
| **variable** | (any value) | This variable is not used but is currently required so use any value |
| **postprocessor** |  | The name of the post-processor output value input to this post-processor |

| | | |
|---|---|---|
| Change_with_respect_to_ initial | **true** | The default ‘true’ indicates that the change is with respect to the time zero value.  ‘False’ indicates the change if based on the value from the previous time step. |
| compute_relative_change | **false** | The default ‘false’ indicates that the change in the value will be output.  ‘True’ indicates that the relative change in the value will be output. |
| execute_on | **'timestep_end'** | Control the execution of the output file.  Options are to set to (none\|initial\|linear\|nonlinear\|timestep_end\|timestep_begin\|final\|failed\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor.  If none provided then all specified in [Outputs] will be used. |

## 7.4 Cumulative Value

The cumulative value post-processor [CumulativeValuePostprocessor] sub-block computes a cumulative sum of a post-processor value over a transient.

| **Input File Parameters**<br>(Note:  Required are bolded) | **Parameter Values**<br>(Note:  Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | CumulativeValuePostprocessor | Designates a cumulative value post-processor |
| **postprocessor** | | The name of the post-processor value input to this post-processor |
| execute_on | **' timestep_end'** | Control the execution of the output file.  Options are to set to (none\|initial\|linear\|nonlinear\|timestep_end\|timestep_begin\|final\|failed\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor.  If none provided then all specified in [Outputs] will be used. |

## 7.5 Difference

The difference post-processor [DifferencePostprocessor] sub-block computes the difference between two post-processors.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **Type** | DifferencePostprocessor | Designates a difference post-processor |
| **value1** | | Name of the post-processor whose output is the first value in the difference calculation |
| **value2** | | Name of the post-processor whose output is the second value in the difference calculation |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.6 Element Average Value

The element average value post-processor [ElementAverageValue] sub-block computes the average value of a specified parameter for a component or components.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | ElementAverageValue | Designates an element average value -processor |
| **variable** | | The component parameter whose value is to be averaged |
| block | | The list of component names to be included in the calculation. If none are specified then all will be included. |
| execute_on | **timestep_end** | Set to |

| | | |
|---|---|---|
| | | (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.7 Element Extreme Value

The element extreme value post-processor [ElementExtremeValue] sub-block determines the extreme value (maximum or minimum) for a component parameter.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | ElementAverageValue | Designates an element extreme value post-processor |
| **variable** | | The component parameter whose extreme value is to be determined |
| block | | The list of component names to be included in the calculation. If none are specified then all will be included. |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |
| value_type | **max** | Type of extreme value to return. 'max' returns the maximum value. 'min' returns the minimum value. |

## 7.8 Element Heat Flux

The element heat flux [ElementHeatFluxPostprocessor] post-processor sub-block computes the integrated heat flux (positive value indicates into the heat structure) across a heat structure [HeatStructure] component.

| **Input File Parameters** | **Parameter Values** | **Description** |
|---|---|---|

| (Note: Required are bolded) | (Note: Defaults are bolded) | |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | ElementHeatFluxPostprocess or | Designates an element heat flux post-processor |
| **Hw** | | Heat transfer coefficient of the wall |
| **P_hf** | | Heat flux perimeter |
| **T_wall** | | Temperature of the wall |
| **T_fluid** | | Temperature of the fluid |
| block | | The list of heat structure names to be included in the calculation. If none are specified then all will be included. |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.9 Extreme Value Over Time

The extreme value over time post-processor [TimeExtremeValue] reports the maximum or minimum value, or the absolute value of the maximum or minimum value, of the output value of another post-processor.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | TimeExtremeValue | Designates a extreme value post-processor |
| **postprocessor** | | The name of the postprocessor whose output value is evaluated for extreme values |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|ti |

| | | |
|---|---|---|
| | | mestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |
| value_type | **max** | Type of extreme value to return: 'max returns the maximum value; 'min' returns the minimum value; 'abs_max' returns the maximum absolute value; 'abs_min' returns the minimum absolute value. |

## 7.10 Function Value

The function value post-processor [FunctionValuePostprocessor] sub-block specifies the function that provides the value to be used in another post-processor. A typical use would be a function that samples a parameter value in time at a specified point in space. The point in space is specified by this post-processor.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | FunctionValuePostprocessor | Designates a function value post-processor |
| **function** | | The name of the function that supplies the input value to this postprocessor |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |
| point | **'0 0 0'** | A point in space (x,y,z) to be given to the function |
| use_displaced_mesh | **false**<br>true | Specify ‘true’ to always use displaced mesh |

## 7.11 Linear Combination

The linear combination post-processor [LinearCombinationPostprocessor] sub-block calculates the sum of pairs of post-processor output values (input to this post-processor) multiplied by coefficients (parameter values specified in this postprocessor), plus a constant value (specified in this post-processor).

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | LinearCombinationPostprocessor | Designates a linear combination post-processor |
| **pp_coefs** | | List of coefficients to be multiplied by post-processor values listed in the 'pp_names' parameter. These are paired values. |
| **pp_names** | | List of post-processor names whose values are multiplied by the list of coefficients in the 'pp_coefs' parameter. These are paired values. |
| b | **0** | A constant value to be add to the sum of pairs of multiplied values. |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.12 Mass Flux Integral

The mass flux integral post-processor [MassFluxIntegral] sub-block calculates the integrated mass flux (total mass) across a boundary condition component (e.g. [Inlet]).

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | MassFluxIntegral | Designates a mass flux integral post-processor |

| | | |
|---|---|---|
| **arhouA** | | Product of void fraction * density *velocity*area |
| **boundary** | | The unique name(s) of a boundary condition component(s) (e.g. InletXX). |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.13 Momentum Flux Integral

The momentum flux integral post-processor [MomentumFluxIntegral] sub-block calculates the integral of the momentum flux across a boundary component, such as an [Outlet].

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | MomentumFluxIntegral | Designates a momentum flux integral post-processor |
| **A** | | Area |
| **arhouA** | | Product of void fraction * density *velocity*area |
| **boundary** | | The unique name(s) of a boundary condition component(s) (e.g. OutletXX). |
| **p** | | Pressure |
| **vel** | | Velocity |
| alpha | **1.0** | Volume fraction (two-phase only) |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post- |

| | | |
|---|---|---|
| | | processor. If none provided then all specified in [Outputs] will be used. |

## 7.14 Nodal Energy Flux

The nodal energy flux post-processor [NodalEnergyFluxPostprocessor] sub-block calculates the energy flux across a boundary component, such as an [Outlet].

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | NodalEnergyFluxPostprocessor | Designates a nodal energy flux post-processor |
| **H** | | Specific total enthalpy is the sum of the specific enthalpy and the specific kinetic energy. |
| **arhouA** | | Alpha(liquid void fraction)*rho*u*A |
| boundary | | The unique name(s) of a boundary condition component(s) (e.g. OutletXX). |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.15 Nodal Extreme Value

The nodal extreme value post-processor [NodalExtremeValue] sub-block determines the extreme value (maximum or minimum) for a boundary component (e.g. [Outlet]) parameter.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | NodalExtremeValue | Designates a nodal extreme value post-processor |
| **variable** | | The component parameter whose |

| | | |
|---|---|---|
| | | extreme value is to be determined |
| boundary | | The unique name(s) of a boundary condition component(s) (e.g. OutletXX). |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |
| value_type | **max** | Type of extreme value to return: 'max' returns the maximum value. 'min' returns the minimum value. |

## 7.16 Nodal Sum

The nodal sum value post-processor [NodalSum] sub-block sums the values of a parameter for all of the nodes in a component at each time step. An example would be summing nodal mass values to obtain a total mass output for this post-processor.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **Type** | NodalSum | Designates a nodal sum post-processor |
| **variable** | | The component parameter value that will be summed. |
| block | | The component containing the nodes for which the parameter values will be summed each time step. |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.17 Number of Elements

The number of elements post-processor [NumElems] sub-block provides the total number of elements in the simulation model.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | NumElems | Designates a number of elements post-processor |
| elem_filter | **active** | Specifies that only active elements (those are used in the simulation model) are counted |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.18 Number of Linear Iterations

The number of linear iterations post-processor [NumLinearIterations] sub-block outputs the number of linear iterations at each time step.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | NumLinearIterations | Designates a number of linear iterations post-processor |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.19 Number of Nodes

The number of nodes post-processor [NumNodes] sub-block outputs the number of nodes.

| **Input**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | NumNodes | Designates a number of nodes post-processor |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.20 Number of Non-Linear Iterations

The number of non-linear iterations post-processor [NumNonlinearIterations] sub-block outputs the number of non-linear iterations at each time step.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | NumNonlinearIterations | Designates a number non-lonear iterations post-processor |
| accumulate_over_step | **false**<br>true | Specify 'true' to count the number of non-linear iterations for each time step |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.21 Number of Residual Evaluations

The number of residual evaluations post-processor [NumResidualEvaluations] sub-block outputs the number of residual evaluations at each time step.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | NumResidualEvaluations | Designates a number of residual evalations post-processor |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.22 Point Value

The point value post-processor [PointValue] sub-block outputs the value of a parameter (e.g. temperature) at a specific location (x, y, z) in the displaced mesh.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | PointValue | Designates a point value post-processor |
| **point** | **‘0 0 0’** | The physical point where the solution will be evaluated.(‘x y z’) |
| **variable** | | The parameter name (e.g. ‘T’, ‘p’ whose value is to be output. |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) [initial timestep_end] to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be |

| | | |
|---|---|---|
| | | used. |
| use_displaced_mesh | **false**<br>true | Specify 'true' to always use displaced mesh |

## 7.23 Real Component Parameter Value

The real component parameter value postprocessor [RealComponentParameterValuePostprocessor] sub-block outputs the numerical value of a component parameter.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | RealComponentParameterValuePostprocessor | Designates a real component parameter value post-processor |
| **component** | | The name of the component whose parameter value is output. |
| **parameter** | | The name of the component parameter whose numerical value is output |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.24 Real Control Data Value

The real control data value postprocessor [RealContolDataValuePostprocessor] sub-block outputs the numerical value of a control.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | Unique name | Designates sub-block component name |
| **type** | RealControlDataValuePostprocessor | Designates a real control data value post-processor |
| **control_data_name** | | The name of the control whose numerical value is output. |
| execute_on | **timestep_end** | Set to |

| | | |
|---|---|---|
| | | (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.25 Relative Difference

The relative difference post-processor [RelativeDifferencePostprocessor] sub-block computes the absolute value of the relative difference between two post-processor values.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | RelativeDifferencePostprocessor | Designates a difference post-processor |
| **value1** | | First post-processor that provides a value |
| **value2** | | Second post-processor that provides a value to be used as the base for relative difference compared to the first value |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.26 Scalar Variable

The scalar variable post-processor [ScalarVariable] sub-block outputs a scalar variable value for a specified component.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |

| | | |
|---|---|---|
| **type** | ScalarVariable | Designates a scalar variable post-processor |
| **variable** | | Name of the component scalar parameter |
| component | **0** | Component whose scalar variable is output |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.27 Scaling Factor

The scaling factor post-processor [ScalePostprocessor] multiplies a post-processor value by a specified scaling factor.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | ScalePostprocessor | Designates a scaling post-processor |
| **value** | | The post-processor whose output value is to be multiplied by the scaling factor |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |
| scaling_factor | **1** | The scaling factor that is used as a multiplier applied to the post-processor value |

## 7.28 Side Average Value

The side average value post-processor [SideAverageValue] computes the average value of a parameter for a heat structure [HeatStructure] surface.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | SideAverageValue | Designates a side average value post-processor |
| **boundary** | | The name of the heat structure component |
| **variable** | | The name of the parameter to be averaged (e.g. T_surface) |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.29 Side Flux Average

The side average flux post-processor [SideFluxAverage] computes the average of the flux for a heat structure [HeatStructure] surface.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | SideFluxAverage | Designates a side average flux post-processor |
| **boundary** | | The name of the heat structure component |
| **diffusivity** | | The name of the diffusivity material property that will be used in the flux computation. |
| **variable** | | The name of the flux parameter (e.g. heat flux) to be averaged |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, |

| | | |
|---|---|---|
| | | csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.30 Side Flux Integral

The side flux integral post-processor [SideFluxIntegral] computes the integral of the flux for a heat structure [HeatStructure] surface.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | SideFluxIntegral | Designates a side flux integral post-processor |
| **boundary** | | The name of the heat structure component |
| **diffusivity** | | The name of the diffusivity material property that will be used in the flux computation. |
| **variable** | | The name of the flux parameter (e.g. heat flux) to be integrated |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.31 Side Integral Variable

The side integral variable post-processor [SideIntegralVariablePostprocessor] computes the integral of a heat structure component [HeatStructure] surface parameter value.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | SideIntegralVariablePostprocessor | Designates a side integral variable post-processor |
| **boundary** | | The name of the heat structure |

| | | |
|---|---|---|
| | | component |
| **variable** | | The name of the component parameter to be integrated |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.32 Sum

The sum post-processor [SumPostprocessor] computes the sum of two post-processors.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **Type** | SumPostprocessor | Designates a sum post-processor |
| **a** | | Name of first postprocessor whose value is to summed |
| **b** | | Name of second postprocessor whose value is to summed |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor. If none provided then all specified in [Outputs] will be used. |

## 7.33 Time Step Size

The time step size post-processor [TimeStepSize] outputs the time step size.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |

| | | |
|---|---|---|
| **type** | TimeStepSize | Designates a time step size post-processor |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor.  If none provided then all specified in [Outputs] will be used. |

## 7.34 Total Variable Value

The total variable value size post-processor [TotalVariableValue] sub-block integrates the output value of a post-processor.

| **Input File Parameters**<br>(Note:  Required are bolded) | **Parameter Values**<br>(Note:  Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | TotalVariableValue | Designates a total variable post-processor |
| execute_on | **timestep_end** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| outputs | | The output names (i.e. console, csv, exodus) for this post-processor.  If none provided then all specified in [Outputs] will be used. |
| value | | The name of the postprocessor whose output value is to be integrated |

# 8.0 EXECUTIONER

The executioner [Executioner] block specifies either the input file as either a transient [Transient] or a control logic [ControlLogic] execution, and includes parameters and values related to time step selection and solver tolerances.

Other executioners and options are included in RELAP-7 but are not specified here.

In the following sections the input file parameters for the [Executioner] blocks and sub-blocks are listed.

## 8.1 Transient

The transient [Transient] sub-block specifies most of the parameters that control a transient execution.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **type** | **Transient** | Designates execution of a transient problem |
| dt | **1** | The time step size between solves if the adaptive time step option is not used |
| dtmax | **1e+30** | The maximum time step size in an adaptive run. As an option this can be specified to control output file size. |
| dtmin | 1e-04<br>**2e-14** | The minimum time step size in an adaptive run. The recommended value is 1e-04. |
| **end_time** | **1e+30** | The end time of the simulation is to be determined based on problem-specific considerations. |
| l_max_its | 100<br>**10000** | Maximum number of linear iterations. The recommended value is 100. |
| l_tol | 1e-02<br>**1e-05** | Linear iteration tolerance. The recommended value is 1e-02. |
| nl_abs_tol | 1e-04<br>**1e-50** | Non-linear iteration absolute tolerance. The recommended value is 1e-04. |
| nl_max_its | 10<br>**50** | Maximum number of non-linear iterations. The recommended value is 10. |
| nl_rel_tol | 1e-06<br>**1e-08** | Non-linear iteration relative tolerance. The recommended |

| | | |
|---|---|---|
| | | value is 1e-06. |
| scheme | 1) bdf2<br>**2) implicit-euler** | The backward difference time integration scheme (bdf2) is recommended |
| start_time | **0** | The start time of the simulation is time zero. |

### 8.1.1 Time Stepper

The time stepper [TimeStepper} sub-block specifies the time advancement for the numerical method.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **TimeStepper** | Specifies the time stepper sub-block |
| **type** | 1) SolutionTimeAdaptiveDT<br>**2) FunctionDT**<br>3) ConstantDT | Specifies the time step option. The recommended option is the adaptive time step. The functionDT is a table of time (time_t)/time step (time_dt) pairs. |
| **dt** | | Value of first time step for adaptive time step option, and for the function option. Constant time step value for constant option. |
| percent_change | **0.1** | This is the percent that the time step is allowed to change using the adaptive time step. |
| **time_t** | | GUI pairs these with "time_dt" |

### 8.1.2 Quadrature

The quadrature [Quadrature] sub-block specifies the type of numerical integration (quadrature) with the two options being trapezoidal and Gaussian (recommended), and the order of the numerical integration with the options being first and second (recommended). For some problems that have stability issues with the recommended quadrature the other option (TRAP/FIRST) may have some benefit.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Quadrature** | Specifies the Quadrature sub-block |
| **type** | **TRAP**<br>GAUSS | Specifies the type of quadrature. The recommended type is GAUSS (Gaussian). |
| **order** | **FIRST** | Specifies the order of the |

| | | |
|---|---|---|
| | SECOND | quadrature.  The recommended order is SECOND. |

# 9.0 FUNCTIONS

The functions block [Functions] provides any functions used by RELAP-7 code such as reactor power as a function of time, pipe flow area as a function of length, or a pressure boundary condition as a function of time. Specific functions are available for selection as sub-blocks.

Other functions are included in RELAP-7 but are not specified here.

In the following sections the input file parameters for the [Functions] sub-blocks are listed.

## 9.1 Constant

The constant function [ConstantFunction] returns a constant value.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block function name |
| **type** | **ConstantFunction** | Designates function is a constant function |
| **value** | **0** | The constant value |

## 9.2 Linear

The linear function [LinearFunction] calculates the value from a linear equation (a+b*x) given constant values of 'a' and 'b', and a function 'x".

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block function name |
| **type** | **LinearFunction** | Designates function is a linear function |
| **a** | | The constant in a + b * x |
| **b** | | The gradient value in a + b *x |
| **x_func** | | The x function |

## 9.3 Parsed

The parsed function calculates a value from a user-defined function consisting of variable/value pairs, numerical values, generic parameters (e.g. "t" (time), "x, y, z" (coordinates), "pi" (π), and mathematical functions (e.g. "sin' for sine). Refer to http://warp.povusers.org/FunctionParser/fparser.html#functionsyntax for a complete list of syntax.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block function name |
| **type** | **ParsedFunction** | Designates function is a parsed function |
| **value** | | User defined function (equation) consisting of variables ('vars' with paired numerical values), numerical constants, generic parameters (t, x, y, z, pi), and mathematical functions (e.g. 'sin' for sine). Also, variables can be paired with post-processor names (as vals), and then the post-processor output values are used in the function. |
| vals | | List of numerical values corresponding to 'vars/vals' pairs. Also, postprocessor names can be used and paired to 'vars' variables. |
| vars | | List of variable names (excluding t, x, y, z) corresponding to 'vars/vals' pairs used in the function. |

## 9.4 Piece-Wise Bi-Linear

The piece-wise bi-linear [PiecewiseBilinear] function interpolates between pairs of any of the spatial coordinates or time values (x, y, z, or t) from a user-specified comma-separated (".csv") file. The input parameters designate which of the four types of values are stored in the rows and columns of the ".csv" file. For example, the stored values may be values of "x' vs. 't', or values of 'y' vs. 'z'. The input files specifies values of pairs of the two selected variables and performs the interpolation.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block function name |
| **type** | **PiecewiseBilinear** | Designates function is a piece-wise |

| | | |
|---|---|---|
| | | bi-linear function |
| axis | **1** | The first row in the ".csv" file are spatial data with ('0', '1', or '2') designating the ( x, y,or z) coordinate. The first column are values of time. Do not provides values for 'xaxis' or 'yaxis' parameters. |
| data_file | | Name of file holding .csv data |
| x | | The x abscissa values to be interpolated |
| xaxis | **1** | The first row in the ".csv" file are spatial data with ('0', '1', or '2') designating the ( x, y,or z) coordinate. Do not provide a value for the 'axis' parameter. A value for 'yaxis' must be provided. |
| y | | The y abscissa values to be interpolated |
| yaxis | **1** | The first column in the ".csv" file are spatial data with ('0', '1', or '2') designating the ( x, y,or z) coordinate. A value for the 'xaxis' must be provided. |
| z | | The ordinate values to be interpolated |

## 9.5 Piece-Wise Constant

The piece-wise constant function [PiecewiseConstant] defines a set of x/y data pairs with no interpolation.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block function name |
| **type** | **PiecewiseConstant** | Designates function is a piecewise constant function |
| axis | | The axis used (x,y, or z) if this is to be a function of position |
| x | | The abscissa values |

## 9.6 Piece-Wise Linear

The piece-wise linear function [PiecewiseLinear] linearly interpolates between values of (x, y) data pairs.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block function name |
| **type** | **PiecewiseLinear** | Designates function is a piecewise linear function |
| axis | | The axis used (x y or z) if this is to be a function of position |
| x | | The abscissa values |
| y | | The ordinate values |

## 9.7 Power Profile

The power profile function [PowerProfileFunction] specifies a standard chopped cosine axial shape and can be used by the [HeatGeneration] and [Subchannel] components.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block function name |
| **type** | **PowerProfileFunction** | Designates function is a piecewise linear function |
| **component** | 2 | The component (x or y or z) with the recommended value of ‘2’ corresponding to a vertical fuel assembly |
| **length** | | The heated length of the fuel assembly |
| coef | **1** | The axial peaking factor |

# 10.0 HEAT STRUCTURE MATERIALS

The heat structure materials [HeatStructureMaterials] block specifies the RELAP-7 parameters necessary to describe the physical properties of solid materials such as ceramics and metals. The input is specified in solid materials properties [SolidMaterialProperties} sub-blocks for each material type.

In the following table the input file parameters for the [SolidMaterialPropertiess] sub-blocks are listed.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block solid material, typically a specific metal or ceramic |
| **type** | **SolidMaterialProperties** | Designates sub-block is solid material properties |
| Cp | | Specific heat |
| k | | Thermal conductivity |
| rho | | Density |

# 11.0 CONTROLS

The controls [Controls] block specifies the modeling of control systems in RELAP-7 input files. A set of sub-blocks describe specific control system parameters.

All of the controls that are included in RELAP-7 are specified here.

Some of the parameters that are not typically used are not included.

In the following sections the input file parameters for the [Controls] sub-blocks are listed.

## 11.1 Get Function Value

This control [GetFunctionValueControl] is called by an upstream control, and specifies the function that will return the parameter value to be used by the upstream control.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block control name |
| **type** | **GetFunctionValueControl** | Designates control is a get function value control |
| **function** | | The name of the function prescribing a value to be used on a control |

## 11.2 Post-Processor Value

This control [GetPostprocessorValueControl] specifies the output of a post-processor to be used in a downstream control.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block control name |
| **type** | **GetPostprocessorValueControl** | Designates control is a get post-processor value control |
| **postprocessor** | | The name of the post-processor prescribing a value to be used in a control |

## 11.3 Proportional Integral Derivative

This control [PIDControl] is used in RELAP-7 to model a proportional integral derivative (PID) control.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block control name |
| **type** | **PIDControl** | Designates control is a PID control |
| **initial_value** | | The initial value for the integral part |
| **input** | | The name of the control data that is the input |
| **K_d** | | The coefficient for the derivative term |
| **K_i** | | The coefficient for the integral term |
| **K_p** | | The coefficient for the proportional term |
| **set_point** | | The name of the control data with the set point |

## 11.4 Real Function

This control [RealFunctionControl] sets the value of a input parameter to the value of a specified function.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block control name |
| **type** | **RealFunctionControl** | Designates control is real function control |
| **function** | | The function to use to determine the value of the specified parameter. |
| **parameter** | | The input parameter that will be assigned the value from the function |
| depends_on | | The upstream control that this control relies on (i.e. upstream control must execute first) |
| execute_on | **'initial timestep_end'** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |

## 11.5 Set Real Value

This control [SetRealValueControl] specifies the component and parameter to be controlled, along with the control that returns the value of the parameter.

| Input File Parameters<br>(Note: Required are bolded) | Parameter Values<br>(Note: Defaults are bolded) | Description |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block control name |
| **type** | **SetRealValueControl** | Designates control is a set real value control |
| **component** | | The name of the component to be controlled. |
| **parameter** | | The name of the parameter in the component to be controlled. |
| **value** | | The name of the control that determines the value to be set in the component. |

## 11.6 Terminate

This control [TerminateControl] terminates the simulation when the ‘output’ value of the parameter ‘threshold’ in a UnitTripContol reaches the specified threshold value.

| Input File Parameters<br>(Note: Required are bolded) | Parameter Values<br>(Note: Defaults are bolded) | Description |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block control name |
| **type** | **TerminateControl** | Designates control is a terminate control |
| **input** | | The name of the UnitTripControl with the parameter ‘threshold’ and its value used for simulation termination. |

## 11.7 Time Function

This control [TimeFunctionControl] specifies for a component and parameter the control of the value of that parameter using a function that includes time as a variable.

| Input File Parameters<br>(Note: Required are bolded) | Parameter Values<br>(Note: Defaults are bolded) | Description |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block control name |
| **type** | **TimeFunctionControl** | Designates control is a time function control |
| **component** | | Name of the component that will be controlled |
| **function** | | Name of the function (including time as a variable) prescribing the |

| | | |
|---|---|---|
| | | value |
| **parameter** | | Name of the parameter in the component that will be controlled |

## 11.8 Time Period

This control [TimePeriod] controls the enabled/disabled state of objects with time

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block control name |
| **type** | **TimePeriod** | Designates control is a time period control |
| depends_on | | The upstream control that this control relies on (i.e. upstream control must execute first) |
| disable_objects | | A list of objects that are enabled/disabled when the control time period starts and ends |
| end_time | | The time at which the objects are disabled |
| start_time | | The time at which the objects are enabled |

## 11.9 Unit Trip

This control [UnitTripControl] specifies the control output value(e.g. post-processor value) that is used as the input value for comparison to the specified value of the threshold (setpoint) parameter.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block control name |
| **type** | **UnitTripControl** | Designates control is a unit trip control |
| **input** | | The name of the control output value that is compared to the threshold (setpoint) value. |
| **threshold** | | The threshold (setpoint) value that will actuate a downstream control |

# 12.0 AUXILIARY KERNELS

The auxiliary kernels [AuxKernels] block includes a variety of sub-blocks that are used to calculate specific quantities of interest using component parameter output values as inputs, along with constants specified as sub-block inputs. The auxiliary kernel output values can then be used for a variety of purposes.

Other auxiliary kernels are included in RELAP-7 but are not specified here.

In the following sections the input file parameters for the [AuxKernels] sub-blocks are listed.

## 12.1 Constant

The constant auxiliary kernel [ConstantAux] sub-block specifies an input numerical value that is assigned as a component parameter value or a component boundary parameter value in the simulation model.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates auxiliary kernel sub-block name |
| **type** | **ConstantAux** | Designates a constant auxiliary kernel |
| **variable** | | The name of the component parameter that the constant value is assigned to. Only one is allowed. |
| block | | The list of component names for which the parameter is assigned the constant value |
| boundary | | The list of [Inlet] component names for which the parameter is assigned the constant value |
| execute_on | **linear** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |
| value | **0** | The constant value to be assigned to the specified components. |

## 12.2 Function

The function auxiliary kernel [FunctionAux] sub-block samples a function output value in space and time, as applicable, and assigns it as a component parameter value or a component boundary parameter value in the simulation model.

| **Input File Parameters** | **Parameter Values** | **Description** |
|---|---|---|

| (Note: Required are bolded) | (Note: Defaults are bolded) | |
|---|---|---|
| **[./name]** | **Unique name** | Designates auxiliary kernel sub-block name |
| **type** | FunctionAux | Designates a function auxiliary kernel |
| **function** | | The function that provides the input value in space and time |
| **variable** | | The name of the component parameter that the function output value is assigned to. Only one is allowed. |
| block | | The list of component names for which the parameter is assigned the function output value |
| boundary | | The list of component names ([InletXX]) for which the parameter is assigned the function output value |
| execute_on | **linear** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |

## 12.3 Mach Number

The Mach number auxiliary kernel [MachNumberAux] calculates the Mach number for a specified component.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates auxiliary kernel sub-block name |
| **type** | **MachNumberAux** | Designates a Mach number auxiliary kernel |
| **e** | | Specific internal energy from the specified component |
| **fp** | | The name of the fluid properties model |
| **v** | | Specific volume from the specified component |
| **variable** | | The parameter name assigned to the Mach number output value (e.g. 'Mach_number_PipeXX') |
| **vel** | | Fluid velocity from the specified component |

| | | |
|---|---|---|
| block | | The list of component names to be included in the calculation |
| execute_on | **linear** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |

## 12.4 Material Real

The material average property auxiliary kernel [MaterialRealAux] outputs component volume-averaged material properties.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates auxiliary kernel sub-block name |
| **type** | MaterialRealAux | Designates a material real auxiliary kernel |
| **property** | | The material property parameter name to be averaged |
| **variable** | | The parameter name assigned to the component average parameter value (e.g. 'PipeXX_average_temperature') |
| block | | The list of component names to be included in the calculation |
| execute_on | **linear** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |

## 12.5 Mixture Quantity

The mixture quantity auxiliary kernel [MixtureQuantityAux] calculates a component mixture quantity from vectors of independent values (that sum to 1.0) and dependent variable values. An example is a vector of independent values (liquid void fraction, vapor void fraction), and dependent variables (liquid density, vapor density).

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates auxiliary kernel sub-block name |
| **type** | **MixtureQuantityAux** | Designates a mixture quantity auxiliary kernel |
| **a** | | Vector of independent parameter |

| | | |
|---|---|---|
| | | names whose values (decimal values) sum to 1.0 (e.g. liquid void fraction, vapor void fraction) |
| **b** | | Vector of dependent parameter values that correspond to the vector of independent variable values (e.g. density of liquid, density of vapor) |
| **variable** | | The parameter name assigned to the component mixture parameter value (e.g. 'PipeXX_mixed_density') |
| block | | The list of component names to be included in the calculation |
| execute_on | **linear** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |

## 12.6 Prandtl Number

This auxiliary kernel [PrandtlNumberAux] calculates the Prandtl number for a component.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates auxiliary kernel sub-block name |
| **type** | **PrandtlNumberAux** | Designates a Prandtl number auxiliary kernel |
| **e** | | Specific internal energy from the specified component |
| **fp** | | The name of the fluid properties model |
| **v** | | Specific volume from the specified component |
| **variable** | | The parameter name assigned to the Prandtl number output value (e.g. 'Prandtl_number_PipeXX') |
| block | | The list of component names to be included in the calculation. If none are specified then all will be included. |
| execute_on | **linear** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |

## 12.7 Reynolds Number

This auxiliary kernel [ReynoldsNumberAux] calculates the Reynolds number for a component.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates auxiliary kernel sub-block name |
| **type** | **ReynoldsNumberAux** | Designates a Reynolds number auxiliary kernel |
| **D_h** | | Hydraulic diameter from the specified component |
| **e** | | Specific internal energy from the specified component |
| **fp** | | The name of the fluid properties model |
| **rho** | | Density from the specified component |
| **v** | | Specific volume from the specified component |
| **variable** | | The parameter name assigned to the Reynolds number output value (e.g. 'Reynolds_number_PipeXX') |
| **vel** | | Velocity from the specified component |
| alpha | **1.0** | Liquid void fraction from the specified component |
| block | | The list of component names to be included in the calculation |
| execute_on | **linear** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |

## 12.8 Sound Speed

The sound speed auxiliary kernel [SoundSpeedAux] calculates the speed of sound as a function of specific internal energy and specific volume for a component.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates auxiliary kernel sub-block name |
| **type** | **SoundSpeedAux** | Designates a sound speed auxiliary |

| | | |
|---|---|---|
| | | kernel |
| **e** | | Specific internal energy from the specified component |
| **fp** | | The name of the fluid properties model |
| **v** | | Specific volume from the specified component |
| **variable** | | The parameter name assigned to the speed of sound output value (e.g. ‘Sound_speed_PipeXX’) |
| block | | The list of component names to be included in the calculation |
| execute_on | **linear** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |

## 12.9 Saturation Temperature

The saturation temperature auxiliary kernel [TemperatureSaturationAux] calculates saturation temperature from pressure for a component.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates auxiliary kernel sub-block name |
| **type** | **TemperatureSaturationAux** | Designates a saturation temperature auxiliary kernel |
| **fp** | | The name of the fluid properties model |
| **p** | | Pressure from the specified component |
| **variable** | | The parameter name assigned to the speed of sound output value (e.g. ‘Tsat_PipeXX’) |
| block | | The list of component names to be included in the calculation |
| execute_on | **linear** | Set to (nonlinear\|linear\|timestep_end\|timestep_begin\|custom) to execute only at that moment |

# 13.0 COMPONENTS

The components [Components] block specifies within sub-blocks the parameters used for each of the component types used in RELAP-7 simulation models. In RELAP-7 the terminology ‘component’ refers to a zero, one, or two-dimensional element of a fluid system simulation model. A component can represent a physical structure such as a pipe, valve, pump, or heat exchanger. A component can also be a boundary condition such as a pressure sink or a mass flowrate. A network of components is specified to represent the system being simulated using a three-dimensional mesh.

For each component type most of the parameters are required to fully describe the component, although the list of parameters can vary significantly depending on the component type. Parameter values specified generically in the [GlobalParams] block are superseded by specifying component-specific values.

The input file parameters for each component type are listed in the following sections. Some component parameters that are not typically used are not included.

## 13.1 Check Valve

The check valve [CheckValve] component is similar to a junction component in that it connects two 1D components, such as pipes. The check valve opens when the dynamic pressure difference exceeds an input value, and then reseats by one of three input values indicating reverse flow, or the static or dynamic differential pressure reversal. Forward and reverse loss coefficients and a reference area are inputs. The valve open/close response time is also an input.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **CheckValve** | Designates component is a check valve |
| **A_ref** | | Area |
| **back_pressure** | | Dynamic Δp for the valve to open |
| **center** | | Geometric center position (x, y, z) |
| **initial_status** | **OPEN**<br>CLOSE | Initial status – either ‘OPEN’ or ‘CLOSE’ (i.e. closed) |
| **inputs** | | Name of the 1D component at the inlet (e.g. ‘PipeXX(out)) |
| **K** | | Forward flow form loss coefficients at the inlet and at the outlet |
| **outputs** | | Name of the 1D component at the |

| | | |
|---|---|---|
| | | outlet (e.g. ‘PipeXX(in)) |
| **response_time** | | Response time to open or close |
| **volume** | | Volume |
| check_valve_type | **FLOW**<br>STATIC<br>DYNAMIC | Check valve type: ‘FLOW’ (close by flow reversal); ‘STATIC’ (close by static Δp); ‘DYNAMIC’ (close by dynamic Δp) |
| initial_T | **300** | Initial temperature |
| initial_alpha_vapor | **1** | Initial vapor volume fraction |
| initial_p | **100000** | Initial pressure |
| initial_vel | **0** | Initial velocity |
| K_reverse | | Reverse flow form loss coefficients at the inlet and at the outlet |

## 13.2 Compressible Valve

The compressible valve [CompressibleValve] component is used to model relief and safety valves. It is similar to a junction component in that it connects two 1D components, such as pipes. Typically the compressible valve opens when the dynamic Δp exceeds an input value, and then reseats when the Δp decreases to less than an input value. As an alternative the valve can open based on inlet pressure. Forward and reverse loss coefficients and a reference area are inputs. The valve open/close response time is also an input. Currently the compressible valve component is restricted to vapor flow.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **CompressibleValve** | Designates component is a compressible valve |
| **A_ref** | | Area |
| **center** | | Geometric center position (x, y, z) |
| **initial_status** | **CLOSE** | Initial status of the valve (‘OPEN’ or ‘CLOSE’ (i.e. closed)) |
| **inputs** | | Name of the 1D component at the inlet (e.g. ‘PipeXX(out)) |
| **is_actuated_by_physics** | | This parameter is under development so specify ‘false’ |

| | | |
|---|---|---|
| **is_actuated_by_pressure_difference** | | ‘true’ if the valve is actuated by Δp; ‘false’ if actuation is by inlet pressure |
| **K** | | Forward flow form loss coefficients at the inlet and at the outlet |
| **outputs** | | Name of the 1D component at the outlet (e.g. ‘PipeXX(in)) |
| **response_time_close** | | Response time to close |
| **response_time_open** | | Response time to open |
| **volume** | | Volume |
| delta_p_close | **1e+100** | Minimum Δp to close |
| delta_p_open | **1e+100** | Minimum Δp to open |
| initial_T | **300** | Initial temperature |
| initial_alpha_vapor | **1** | Initial vapor volume fraction |
| initial_p | **100000** | Initial pressure |
| initial_vel | **0** | Initial velocity |
| K_reverse | | Reverse flow form loss coefficients at the inlet and at the outlet |
| valve_action | NO_ACTION | Valve control action, either ‘OPEN’ or ‘CLOSE’ (i.e. closed), or ‘NO_ACTION’ |

## 13.3 Core Channel

The core channel [CoreChannel] component is a composite component that models the coolant flow in the reactor and the heat conduction in the fuel heat structure. The heat transfer between the fuel rods and the coolant is fully coupled, and the axial mesh must be consistent. The fuel heat structure can be either cylindrical fuel rods or plates, and can be either 1D or 2D. The fuel rods include the capability to model the fuel, the fuel-to-cladding gap, and the cladding, with include a variable mesh in the solid materials. The materials are provided by the [HeatStructureMaterials] block. The power input is provided from the [PointKinetics] component, or the [PrescribedReactorPower] component.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |

| | | |
|---|---|---|
| **type** | **CoreChannel** | Designates component is a core channel |
| **A** | | Area of coolant channel |
| **elem_number_of_hs** | | Number of cells in each heat structure material. For fuel rod this is from centerline to surface of fuel pellet, ID of cladding, and OD of cladding. For a plate this is from left surface to right surface. |
| **fuel_type** | | 'cylinder' for fuel rod, or 'plate' |
| **length** | | Total length along the main axis |
| **material_hs** | | Name of each of the materials used in the heat structures (centerline to surface for cylinder; left to right for plate) |
| **n_elems** | | Number of cells along the main axis |
| **n_heatstruct** | | Number of heat structure materials |
| **name_of_hs** | | User specified names of each heat structure |
| **orientation** | | Orientation vector (x, y, z) |
| **P_hf** | | Heat structure heated perimeter |
| **position** | | Origin (x, y, z) |
| **width_of_hs** | | Width of each heat structure |
| D_h | | Hydraulic diameter of the coolant channel |
| Hw | | Optional user-specified single-phase convective heat transfer coefficient. Normally this is not specified and the closures (boiling curve) are used. |
| Hw_liquid | | Optional user-specified liquid-phase convective heat transfer coefficient. Normally this is not specified and the closures (boiling curve) are used. |
| Hw_vapor | | Optional user-specified vapor-phase convective heat transfer |

| | | |
|---|---|---|
| | | coefficient. Normally this is not specified and the closures (boiling curve) are used. |
| PoD | **1** | Pitch to diameter ratio for fuel rod lattice |
| depth | | Thickness of plate fuel |
| dim_hs | **2** | Dimension of the mesh used for the heat structure ‘2 ‘ = 2D; ‘1” = 1D |
| f | | Friction factor for the outer surface of the heat structure |
| f_interface | **0** | Interface (interphase) friction between liquid and vapor. A user-specified value will replace the value provided by the closure. |
| fp | | Name of fluid properties |
| fuel_type | **PLATE** | ‘cylinder’ or ‘plate’ |
| heat_transfer_geom | **ROD_BUNDLE** | Convective heat transfer based on a ‘pipe’ or ‘rod bundle’ |
| initial_T | | Initial fluid temperature for single-phase |
| initial_T_liquid | | Initial fluid temperature for the liquid phase |
| initial_T_vapor | | Initial fluid temperature for the vapor phase |
| initial_Ts | | Initial temperature for each material |
| initial_alpha_vapor | | Initial vapor volume fraction |
| initial_p | | Initial pressure in the pipe |
| initial_p_liquid | | Initial pressure for the liquid phase |
| initial_p_vapor | | Initial pressure for the vapor phase |
| initial_vel | | Initial velocity for single-phase |
| initial_vel_liquid | | Initial velocity for the liquid phase |
| initial_vel_vapor | | Initial velocity for the vapor phase |
| mesh_disp_gap | **0.02** | Heat structure mesh offset for post-processing visualization |

| | | |
|---|---|---|
| n_elems_hs | | Number of cells in the heat structure along the main axis.  If none provided it will default to 'n_elems' |
| n_rods | | Number of rods |
| offset | **'0 0 0'** | Offset of the origin for post-processing visualization |
| power | | Name of the component that provides the reactor power boundary condition |
| power_fraction | | Fraction of reactor power that is deposited into each material |
| power_shape_function | | A numerical value is applied as an axial power multiplier (‘1’ is an constant axial power peaking factor).  A value corresponding to a function name can return an axial power shape, and/or a fuel material region radial power shape, either as linear functions (constant in time), or as bi-linear functions of time |
| roughness | **0** | Roughness of the outer surface of the heat structure |

## 13.4 Downcomer

The BWR downcomer [Downcomer] component connects the feedwater pipe, the separator dryer discharge, the steam dome, and the downcomer outlet.  The downcomer is separated into a vapor region above the mixture level, and a liquid region below the mixture level.  During a transient the mixture level is tracked.  The vapor pressure is the steam dome pressure.  The downcomer component is a junction component with volume.

| **Input File Parameters**<br>(Note:  Required are bolded) | **Parameter Values**<br>(Note:  Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **DownComer** | Designates component is a BWR downcomer |
| **A_ref** | | Area |
| **K** | | Forward flow form loss coefficients at the inlet and at the outlet |
| **center** | | Geometric center position (x, y, z) |

| | | |
|---|---|---|
| **dome_component** | | Steam dome component name to provide the vapor pressure |
| **initial_level** | | Initial iquid level |
| **inputs** | | Feedwater pipe component outlet |
| **outputs** | | Lower plenum component inlet |
| **volume** | | Volume |
| initial_T | **300** | Initial temperature |
| initial_alpha_vapor | **1** | Initial vapor volume fraction |
| initial_p | **100000** | Initial pressure |
| initial_vel | **0** | Initial velocity |
| K_reverse | | Reverse flow form loss coefficients at the inlet and at the outlet |

### 13.5 Elbow Pipe

The elbow pipe [ElbowPipe] component is a pipe component that includes a bend with a radius and an angle.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **ElbowPipe** | Designates component is an elbow pipe |
| **A** | | Area |
| **end_angle** | | Angle at which the curvature of the pipe ends |
| **fp** | | Name of fluid properties |
| **n_elems** | | The number of cells along the main axis |
| **orientation** | | Orientation vector (x, y, z) |
| **position** | | Origin (x, y, z) |
| **radius** | | Radius of the curvature to the centerline of the pipe |

| | | |
|---|---|---|
| **start_angle** | | Angle at which the curvature of the pipe starts |
| D_h | | Hydraulic diameter |
| f | | Wall friction |
| f_2phase_mult_liquid | | Two-phase friction multiplier for liquid phase |
| f_2phase_mult_vapor | | Two-phase friction multiplier for vapor phase |
| f_interface | | Interface (interphase) friction |
| heat_transfer_geom | **PIPE** | Convective heat transfer based on a pipe |
| Hw | | Optional user-specified single-phase convective heat transfer coefficient. Normally this is not specified and the closures (boiling curve) are used. |
| Hw_liquid | | Optional user-specified liquid-phase convective heat transfer coefficient. Normally this is not specified and the closures (boiling curve) are used. |
| Hw_vapor | | Optional user-specified vapor-phase convective heat transfer coefficient. Normally this is not specified and the closures (boiling curve) are used. |
| initial_T | | Initial temperature |
| initial_T_liquid | | Initial temperature for the liquid phase |
| initial_T_vapor | | Initial temperature in for the vapor phase |
| initial_alpha_vapor | | Initial vapor volume fraction |
| initial_p | | Initial pressure for single-phase |
| initial_p_liquid | | Initial pressure for the liquid phase |
| initial_p_vapor | | Initial pressure for the vapor phase |
| initial_vel | | Initial velocity for single-phase |
| initial_vel_liquid | | Initial velocity for the liquid phase |

| | | |
|---|---|---|
| initial_vel_vapor | | Initial velocity for the vapor phase |
| K_2phase_mult_liquid | | Two-phase form loss multiplier for liquid |
| K_2phase_mult_vapor | | Two-phase form loss multiplier for vapor |
| K_prime | | Form loss coefficient per unit length that is used for flow in both directions |
| P_hf | | Heat flux perimeter |
| offset | **'0 0 0'** | Offset of the origin for post-processing visualization |
| q_wall | | User-spedified wall heat flux. Normally this is not specified and the closures (boiling curve) are used. |
| roughness | **0** | Roughness |

## 13.6 Free Boundary

The free boundary [FreeBoundary] component provides a null boundary condition used for 1D components that must be connected to components at both ends.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **FreeBoundary** | Designates a free boundary component |
| **input** | | Name of the component connected to the free boundary component |

## 13.7 Heat Exchanger Connector

The heat exchanger connector [HXConnector] component connects the heat exchanger to the heat structure component and to the pipe component. It also specifies which side of the heat structure the heat exchanger is connected to.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |

| | | |
|---|---|---|
| **type** | **HXConnector** | Designates component is a heat exchanger connector |
| **hs** | | The name of the heat structure component |
| **hs_side** | | The side of the heat structure |
| **pipe** | | The name of the pipe component |

## 13.8 Heat Generation

The heat generation [HeatGeneration] component specifies the component and the material region that heat is generated in, and how that heat is distributed, from a power source originating in a separate component.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description.** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **HeatGeneration** | Designates a heat generation component |
| **block** | | Name of the heat structure material region in the component with heat generation |
| **hs** | | Name of the heat structure component with heat generation |
| power | | Name of the component that provides power |
| power_fraction | **1** | Fraction of power that is deposited into each material region |
| power_shape_function | | A numerical value is applied as an axial power multiplier (‘1’ is an constant axial power peaking factor). A value corresponding to a function name can return an axial power shape, and/or a fuel material region radial power shape, either as linear functions (constant in time), or as bi-linear functions of time |

## 13.9 Heat Structure

The heat structure (HeatStructure] component models a solid material that conducts energy within the solid, and that convects energy into the adjacent pipe components. The heat structure can be either 1D or 2D, and either a cylinder or plate. It can be a composite of several materials, and can be divided into a mesh.

| **Input File Parameters** | **Parameter Values** | **Description** |
|---|---|---|
| (Note: Required are bolded) | (Note: Defaults are bolded) | |
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **HeatStructure** | Designates component is a heat structure |
| **hs_type** | **plate** | ‘cylinder’ or ‘plate’ |
| **length** | | Lengths along the main axis |
| **materials** | | Material names used in the heat structures |
| **n_elems** | | Number of cells along the main axis |
| **n_part_elems** | | Number of cells in each material. For a cylinder from the centerline to the surface of each region. For a plate from the left side to the right side. |
| **names** | | User-specified heat structure region names (e.g. ‘fuel’, ‘cladding’) |
| **orientation** | | Orientation vector (x, y, z) |
| **position** | | Origin (x, y, z) |
| **widths** | | Width of each heat structure. For a cylinder from the centerline to the surface of each region. For a plate from the left side to the right side. |
| axial_offset | **0** | Radial distance of the gap at the interior of an annular fuel cylinder. Not applicable for plates. |
| depth | | Thickness of plate |
| dim | **2** | Dimension of the mesh used for the heat structure ‘2 ‘ = 2D; ‘1” = 1D |
| initial_T | | Initial temperature |
| num_rods | **1** | Number of rods represented by this heat structure. Used as a multiplier on dimensional inputs. |
| offset | **'0 0 0'** | Offset of the origin for post-processing visualization |

### 13.10 Ideal Pump

The ideal pump [IdealPump] component is a junction component that provides a mass flow rate boundary condition between two pipe components.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **IdealPump** | Designates component is an ideal pump |
| **fp** | | Name of fluid properties |
| **inputs** | | Name of the 1D component at the inlet (e.g. 'PipeXX(out)) |
| **mass_flow_rate** | | Mass flow rate boundary condition |
| **outputs** | | Name of the 1D component at the outlet (e.g. 'PipeXX(in)) |

### 13.11 Inlet

The inlet [Inlet] component provides a flow boundary condition to the inlet of a pipe component. Various combinations of parameters options are available to specify the fluid state. Selection of the parameter combinations need to avoid over-specification.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **Inlet** | Designates component is an inlet |
| **input** | | Name of the 1D component receiving the flow (e.g. 'PipeXX(in)) |
| H | | Enthalpy for single-phase |
| H_liquid | | Enthalpy for the liquid phase |
| H_vapor | | Enthalpy for the vapor phase |
| T0 | | Stagnation temperature for single-phase |
| T0_liquid | | Stagnation temperature for the liquid phase |

| | | |
|---|---|---|
| T0_vapor | | Stagnation temperature for the vapor phase |
| T_liquid | | Temperature of the liquid phase |
| T_vapor | | Temperature of the vapor phase |
| alpha_vapor | | Vapor volume fraction |
| m_dot | | Mass flow rate for single-phase |
| m_dot_liquid | | Mass flow rate of the liquid phase |
| m_dot_vapor | | Mass flow rate of the vapor phase |
| p | | Pressure |
| p0 | | Stagnation pressure for single-phase |
| p0_liquid | | Stagnation pressure for the liquid phase |
| p0_vapor | | Stagnation pressure for the vapor phase |
| p_liquid | | Pressure for the liquid phase |
| p_vapor | | Pressure for the vapor phase |
| reversible | **false** | 'false' for boundary conditions that provide a fluid mass flow type of boundary condition. 'true' for boundary conditions that are similar to an outlet component, such as a pressure sink. |
| rho | | Density for single-phase |
| rho_liquid | | Density of the liquid phase |
| rho_vapor | | Density of the vapor phase |
| rhou | | Momentum for single-phase |
| rhou_liquid | | Momentum for the liquid phase |
| rhou_vapor | | Momentum for the vapor phase |
| vel | | Velocity for single-phase |

| | | |
|---|---|---|
| vel_liquid | | Velocity of the liquid phase |
| vel_vapor | | Velocity of the vapor phase |

## 13.12 Junction

The junction [Junction] component can connect multiple pipes at the inlet and outlet but is restricted to single-phase fluids.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **Junction** | Designates component is a junction |
| **initial_T** | | Initial temperature |
| **initial_p** | | Initial pressure |
| **inputs** | | Names of the 1D components at the inlet (e.g. 'PipeXX(out)) |
| **outputs** | | Names of the 1D components at the outlet (e.g. 'PipeXX(in)) |
| K | | Form loss coefficients at the outlets. If loss coefficients are different for forward and reverse flow, then use the volume junction component. |

## 13.13 Outlet

The outlet [Outlet] component provides a backpressure boundary condition to the outlet of a pipe component. An example would be a pressure sink for a relief valve. There is an option for a reversible outlet with flow at the outlet conditions from the outlet component into the pipe component.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **Outlet** | Designates component is an outlet |
| **input** | | Name of the 1D component at the inlet that is connected to the |

| | | |
|---|---|---|
| | | backpressure boundary condition (e.g. 'PipeXX(out)) |
| p | | Pressure for single-phase |
| p_liquid | | Pressure for the liquid phase |
| p_vapor | | Pressure for the vapor phase |
| reversible | **false** | 'true' for a reversible outlet boundary condition |

## 13.14 Pipe

The pipe [Pipe] component is the fundamental 1D component for modeling a volume of fluid in space. A pipe component connects to only one other 1D component at each end of the pipe via one of several junction components.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **Pipe** | Designates component is a pipe |
| **A** | | Area can be a constant, or variable along the length of the pipe using a function sub-block |
| **fp** | | Name of fluid properties |
| **length** | | Total length along the main axis |
| **n_elems** | | Number of cells along the main axis |
| **orientation** | | Orientation vector (x, y, z) |
| **position** | | Origin (x, y, z) |
| D_h | | Hydraulic diameter |
| Hw | | Optional user-specified single-phase convective heat transfer coefficient. Normally this is not specified and the closures (boiling curve) are used. |
| Hw_liquid | | Optional user-specified liquid-phase convective heat transfer coefficient. Normally this is not specified and |

| | | |
|---|---|---|
| | | the closures (boiling curve) are used. |
| Hw_vapor | | Optional user-specified vapor-phase convective heat transfer coefficient. Normally this is not specified and the closures (boiling curve) are used. |
| K_2phase_mult_liquid | | Two-phase form loss multiplier for liquid |
| K_2phase_mult_vapor | | Two-phase form loss multiplier for vapor |
| K_prime | | Form loss coefficient per unit length applicable for both forward and reverse flow |
| P_hf | | Heat structure heated perimeter |
| PoD | **1** | Pitch to diameter ratio for fuel rod lattice |
| T_wall | | Wall temperature |
| f | | Wall friction |
| f_2phase_mult_liquid | | Two-phase friction multiplier for liquid phase |
| f_2phase_mult_vapor | | Two-phase friction multiplier for vapor phase |
| f_interface | | Interface (interphase) friction between liquid and vapor. A user-specified value will replace the value provided by the closure. |
| heat_transfer_geom | **PIPE** | Convective heat transfer based on a pipe |
| initial_T | | Initial fluid temperature for single-phase |
| initial_T_liquid | | Initial fluid temperature for the liquid phase |
| initial_T_vapor | | Initial fluid temperature for the vapor phase |
| initial_alpha_vapor | | Initial vapor volume fraction |
| initial_p | | Initial pressure for single-phase |
| initial_p_liquid | | Initial pressure for the liquid phase |
| initial_p_vapor | | Initial pressure for the vapor phase |

| | | |
|---|---|---|
| initial_vel | | Initial velocity for single-phase |
| initial_vel_liquid | | Initial velocity for the liquid phase |
| initial_vel_vapor | | Initial velocity for the vapor phase |
| offset | **'0 0 0'** | Offset of the origin for post-processing visualization |
| q_wall | | User-spedified wall heat flux. Normally this is not specified and the closures (boiling curve) are used. |
| roughness | **0** | Roughness |

## 13.15 Pipe With Heat Structure

The pipe with heat structure [PipeWithHeatStructure] component is the pipe component with a heat structure attached. The parameters are a combination of the parameters used for the pipe and heat structure components, with the same limitations.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **PipeWithHeatStructure** | Designates component is a pipe with heat structure |
| **A** | | Area can be a constant, or variable along the length of the pipe using a function sub-block |
| **elem_number_of_hs** | | Number of heat structure cells along the main axis |
| **fp** | | Name of fluid properties |
| **HS_BC_type** | **adiabatic** | Heat structure surface boundary condition type. 1) 'adiabatic'; 2) 'constant_temperature' (uses parameter 'T'), 3) convective (uses parameters 'h_amb' and 'T_amb') |
| **hs_type** | **PLATE** | 'cylinder' or 'plate' |
| **initial_Ts** | | Initial heat structure temperature |
| **length** | | Length of the pipe along the main axis |
| **material_hs** | | Material names used in the heat |

| | | |
|---|---|---|
| | | structure |
| **n_elems** | | Number of fluid cells along the main axis |
| **orientation** | | Orientation vector (x, y, z) |
| **position** | | Origin of the pipe (x ,y ,z) |
| **radius_i** | | The inner radius of the pipe wall |
| **width_of_hs** | | Thickness of the heat structure |
| D_h | | Hydraulic diameter |
| dim_hs | **2** | Dimension of the mesh used for the heat structure ‘2 ‘ = 2D; ‘1” = 1D |
| f | | Wall friction for single-phase |
| f_2phase_mult_liquid | | Two-phase friction multiplier for liquid |
| f_2phase_mult_vapor | | Two-phase friction multiplier for liquid |
| f_interface | | Interface (inter-phase) friction |
| h_amb | **0.01** | Convective heat transfer coefficient from outer surface to ambient. Used for ‘HS_BC_type’ = ‘convective’. |
| heat_source_liquid | | Volumetric heat source in liquid |
| heat_source_solid | **0** | Volumetric heat source in solid |
| heat_transfer_geom | **PIPE** | Convective heat transfer based on a pipe |
| Hw | | Optional user-specified single-phase convective heat transfer coefficient. Normally this is not specified and the closures (boiling curve) are used. |
| Hw_liquid | | Optional user-specified liquid-phase convective heat transfer coefficient. Normally this is not specified and the closures (boiling curve) are used. |
| Hw_vapor | | Optional user-specified vapor-phase convective heat transfer coefficient. Normally this is not specified and the closures (boiling curve) are |

| | | |
|---|---|---|
| | | used. |
| initial_T | | Initial temperature for single-phase |
| initial_T_liquid | | Initial temperature for the liquid phase |
| initial_T_vapor | | Initial temperature for the vapor phase |
| initial_alpha_vapor | | Initial vapor volume fraction |
| initial_p | | Initial pressure for single-phase |
| initial_p_liquid | | Initial pressure for the liquid phase |
| initial_p_vapor | | Initial pressure for the vapor phase |
| initial_vel | | Initial velocity for single-phase |
| initial_vel_liquid | | Initial velocity for the liquid phase |
| initial_vel_vapor | | Initial velocity for the vapor phase |
| K_2phase_mult_liquid | | Two-phase form loss multiplier for liquid |
| K_2phase_mult_vapor | | Two-phase form loss multiplier for vapor |
| K_prime | | Form loss coefficient per unit length applicable for both forward and reverse flow |
| offset | **'0 0 0'** | Offset of the origin for post-processing visualization |
| P_hf | | Heat flux perimeter |
| q_wall | | User-spedified wall heat flux. Normally this is not specified and the closures (boiling curve) are used. |
| roughness | **0** | Roughness |
| T | **600** | Fixed ambient temperature boundary condition. Used for 'HS_BC_type' = 'convective'. |
| T_amb | **300** | Ambient temperature |
| T_wall | | Wall temperature |

## 13.16 Point Kinetics

The point kinetics [PointKinetics] component solves the point kinetics equations to determine the change in reactor power. Moderator density or temperature reactivity feedback, and fuel temperature (Doppler) reactivity feedback are included via the [ReactivityFeedback] component. An additional reactivity component, such as control rod reactivity as a function of time can be included.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **PointKinetics** | Designates a point kinetics component |
| **initial_power** | | Initial reactor power |
| ANS_standard | **2005** | The year of the ANSI/ANS 5.1 decay heat standard. Options include 1979, 1994, and 2005. |
| F_U239 | **1** | U-239 atoms produced by neutron capture in U-238 per fission |
| F_alpha | **'0.97 0.03 0 0'** | Fraction of fissions from U-235, U-238, and Pu-239 for 1979 ANS standard; Fractions of fissions from U-235, U-238, Pu-239, and Pu-241 for 1994 and 2005 ANS standards |
| F_gamma | **1** | Multiplier on reactor power |
| Q | **200** | Total energy in MeV generated per fission |
| beta_over_lambda | **200** | Delayed neutron fraction divided by prompt neutron lifetime |
| eta_Np239 | **0.419** | Np-239 decay energy in MeV. The default is from the ANS standard. |
| eta_U239 | **0.474** | U-239 decay energy in MeV. The default is from the ANS standard |
| f_i | **'0.038 0.213 0.188 0.407 0.128 0.026'** | Fraction of delayed neutrons per group. Default values are for Groups 1 to 6 in six-group model. |
| feedback_components | | User-specified name of the reactivity feedback component |
| feedback_reactivity | **0** | Fuel (Doppler) and moderator density reactivity feedback |
| fission_per_fissile | **1** | Number of fissions per fissile atom |
| lam_Np239 | **3.41e-06** | Np-239 decay constant |

| | | |
|---|---|---|
| lam_U239 | **0.000491** | U-239 decay constant |
| lambda | **'0.0127 0.0317 0.115 0.311 1.4 3.87'** | Decay constants for delayed neutron groups. Default values are for Groups 1 to 6 in six-group model. |
| num_eq_FP | **23** | Number of fission product equations. The default is from the ANS standard. |
| num_eq_isotope | **4** | Number of isotopes tracked. The default value of ‘4’ is for the ANS 1994 and 2005 standards. The1979 ANS standard uses a value of ‘3’. |
| num_eq_precursor | **6** | Number of delayed neutron precursor equations. The default value ‘6’ is for the common form of the point kinetics equation. |
| operating_time | **52** | Reactor operating time before shutdown in weeks |
| reactivity_func | | Name of the function that can be used to include a reactivity vs. time component of the total reactivity. An example would be reactivity due to control rod insertion as a function of time. |
| rho | **0** | The initial reactivity. For a steady-state reactor the default value of ‘0’ is applicable. |

## 13.17 Prescribed Reactor Power

The prescribed reactor power [PrescribedReactorPower] component provides a constand value of reactor power, or a function name that returns the reactor power as a function of time.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **PrescribedReactorPower** | Designates a prescribed reactor power component |
| **function** | | A constant numerical value, or a user-specified function name that returns the reactor power as a function of time. |

## 13.18 Pump

The pump [Pump] component is a junction that increases the pressure (head) by a user-specified value, or by connection to a turbine component that provides shaft work.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **Pump** | Designates component is a pump |
| **A_ref** | | Area |
| **fp** | | Name of fluid properties |
| **initial_p** | | Initial pressure |
| **inputs** | | Names of the 1D components at the inlet (e.g. 'PipeXX(out)) |
| **K_reverse** | | Reverse flow form loss coefficients at the inlet and at the outlet |
| **outputs** | | Names of the 1D components at the outlet (e.g. 'PipeXX(in)) |
| driving_component | | Turbine component name to provide shaft work |
| head | **0** | User-specified constant value of pump head |

## 13.19 Reactivity Feedback

The reactivity feedback [ReactivityFeedback] component calculates the reactivity feedback due to a change in moderator density (using a table of pairs of values) or using a moderator temperature coefficient of reactivity, and due to a change in the fuel temperature (using a table of pairs of values) or using a fuel temperature (Doppler) coefficient of reactivity. The resulting change in reactivity is then used in the point kinetics component.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **ReactivityFeedback** | Designates a reactivity feedback component |
| **core_channel** | | User-specified name of the core channel component that provides the moderator density and temperature |

| | | |
|---|---|---|
| **fuel_block** | | In the core channel component, the user-specified name of the heat structure material region (e.g. ‘fuel’) that provides the fuel temperature |
| **fuel_temperature** | | Fuel material temperatures (axial cell average values) used as the independent parameter in a table of fuel temperature reactivity |
| **fuel_reactivity** | | Fuel temperature (Doppler) reactivity values from a table of fuel temperature reactivity |
| **fuel_temp_coefficient** | | Fuel temperature (Doppler) coefficient of reactivity (constant value) that is multiplied by a volumetric average fuel temperature to obtain the change in reactivity. |
| **fuel_weighting_factor** | | Weighting factors applied to each axial cell of the fuel material from bottom to top, and applied to the table of fuel temperature reactivity vs. fuel temperature. Normally these sum to 1.0. |
| **mod_density** | | Moderator (coolant channel) densities (axial cell average values) used as the independent parameter in a table of moderator density reactivity |
| **mod_reactivity** | | Moderator (coolant channel) density reactivity values from a table of moderator (coolant channel) density reactivity |
| **mod_temp_coefficient** | | Moderator (coolant channel) temperature coefficient of reactivity (constant value) that is multiplied by a volumetric average moderator (coolant channel) temperature to obtain the change in reactivity. |
| **mod_weighting_factor** | | Weighting factors applied to each axial cell of the moderator (coolant channel) from bottom to top, and applied to the table of moderator density reactivity vs. moderator density. Normally these sum to 1.0. |

## 13.20 Simple Junction

The simple junction [SimpleJunction] component is a junction that is only applicable when the two connecting pipes have the same area. This is the only junction that currently can model two-phase flow.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **SimpleJunction** | Designates component is a simple junction |
| **inputs** | | Name of the 1D component at the inlet (e.g. 'PipeXX(out)) |
| **outputs** | | Name of the 1D component at the outlet (e.g. 'PipeXX(in)) |

## 13.21 Solid Wall

The solid wall [SolidWall] component is a junction used as the boundary condition for a dead-ended pipe component. Therefore there is no flow circulating through the pipe, however the mass in the pipe can change due to coolant expansion or contraction.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **SolidWall** | Designates component is a solid wall |
| **input** | | Name of the 1D component at the inlet (e.g. 'PipeXX(out)) |

## 13.22 Turbine

The turbine [Turbine] component is a simplified turbine junction component applicable to a Terry turbine, and not for a large power generation turbine.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **Turbine** | Designates component is a turbine |
| **fp** | | Name of fluid properties |
| **initial_T** | | Initial temperature |

|  |  |  |
|---|---|---|
| **initial_p** |  | Initial pressure |
| **inputs** |  | Name of the 1D component at the inlet (e.g. ‘PipeXX(out)) |
| **max_mass_flow_rate** |  | Nominal maximum mass flow rate |
| **outputs** |  | Name of the 1D component at the outlet (e.g. ‘PipeXX(in)) |
| **p0_design** |  | Nominal design inlet stagnation pressure |
| **pressure_ratio_design** |  | Ratio of pressures at inlet and outlet for design conditions |
| **T0_design** |  | Nominal design inlet stagnation temperature |
| **turbine_efficiency** |  | Turbine thermal efficiency |

## 13.23 Valve

The valve [Valve] component is a junction component connecting two pipe components that opens or closes at a specified time with a specified stroke time.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **Valve** | Designates component is a valve |
| **A_ref** |  | Area |
| **center** |  | Geometric center position (x, y, z) |
| **initial_status** | **OPEN** | Initial position of the valve ‘open’ or ‘close’ (closed) |
| **inputs** |  | Name of the 1D component at the inlet (e.g. ‘PipeXX(out)) |
| **K** |  | Forward flow form loss coefficients at the inlet and at the outlet |
| **outputs** |  | Name of the 1D component at the outlet (e.g. ‘PipeXX(in)) |
| **response_time** |  | Valve stroke time |
| **trigger_time** |  | Simulation time the valve is actuated to change position |
| **volume** |  | Volume |

| | | |
|---|---|---|
| K_reverse | | Reverse flow form loss coefficients at the inlet and at the outlet |
| initial_T | **300** | Initial temperature |
| initial_alpha_vapor | **1** | Initial vapor volume fraction |
| initial_p | **100000** | Initial pressure |
| initial_vel | **0** | Initial velocity |

## 13.24 Volume Junction

The volume junction [VolumeJunction] component is a junction component with volume with a unique capability to connect multiple pipe components, which is necessary when flow streams are split or merged. The current model is limited to single-phase conditions. Typical applications would be to model a plenum or a header.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **VolumeJunction** | Designates component is a volume junction |
| **A_ref** | | Reference area |
| **center** | | Geometric center (x, y, z) |
| **inputs** | | Names of the 1D components at the inlet (e.g. ‘PipeXX(out)) |
| **K** | | Forward flow form loss coefficients at the inlets and at the outlets |
| **outputs** | | Names of the 1D components at the outlet (e.g. ‘PipeXX(in)) |
| **volume** | | Volume |
| K_reverse | | Reverse flow form loss coefficients at the inlets and at the outlets |
| initial_T | **300** | Initial temperature |
| initial_alpha_vapor | **1** | Initial vapor volume fraction |
| initial_p | **100000** | Initial pressure |

## 13.25 Wet Well

The wet well [WetWell] component is a simplified junction component with a specified gas (e.g. air or nitrogen) over a liquid water pool in a rectangular (cubic) volume representation. The major design features of a BWR wet well suppression pool are included. An example application is to condense the exhaust steam from a Terry turbine. The model is not suitable for large steam flowrates such as a LOCA blowdown.

| **Input File Parameters**<br>(Note: Required are bolded) | **Parameter Values**<br>(Note: Defaults are bolded) | **Description** |
|---|---|---|
| **[./name]** | **Unique name** | Designates sub-block component name |
| **type** | **WetWell** | Designates component is a BWR wet well |
| **Ac** | | Average cross sectional area |
| **alpha_s** | | Effective heat transfer coefficient for the gas to water pool interfacial area |
| **cooling_rate** | | Heat removal rate boundary condition from an immersed heat exchanger |
| **fp_nc_gas** | | Name of fluid properties for the non-condensible gas |
| **fp_vapor** | | Name of fluid properties for vapor |
| **fp_water** | | Name of fluid properties for water |
| **initial_Lw** | | Initial water level |
| **initial_T** | | Initial gas and water temperature |
| **initial_p_gas** | | Initial gas pressure |
| **inputs** | | Steam sources to the wet well (e.g. turbine exhaust steam) |
| **Lt** | | Total effective height of the wet well |
| **outputs** | | Dry well that is connected to gas venting line (e.g. an outlet component) |
| **z_in** | | Inlet steam pipe discharge elevation above the pool bottom |
| **z_out** | | Outlet water pipe inlet elevation above the pool bottom |
| K_i | **1** | Forward flow form loss coefficient for steam inlet pipe |
| K_ir | **0.5** | Reverse flow form loss coefficient for |

| | | |
|---|---|---|
| | | steam inlet pipe |
| K_o | **0.5** | Forward flow form loss coefficient for water outlet pipe |
| K_or | **0.5** | Reverse flow form loss coefficient for water outlet pipe |
| K_v | **0.5** | Forward flow form loss coefficient for gas venting line to dry well |
| K_vr | **1** | Reverse flow form loss coefficient for gas venting line to drywell |
| number_of_steam_lines | **1** | Number of steam injection lines |