INL/EXT-14-33977

RELAP-7 User's Guide

Hongbin Zhang, Haihua Zhao, Ling Zou, David Andrs, Ray Berry, Richard Martineau

December 2014

U.S. Department of Energy
Office of Nuclear Energy

INL/EXT-14-33977

RELAP-7 User's Guide

Hongbin Zhang, Haihua Zhao, Ling Zou, David Andrs, Ray Berry, Richard Martineau

December 2014

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

Contents

Figures

Tables

RELAP-7 Overview

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 become the main reactor systems simulation toolkit for the LWRS (Light Water Reactor Sustainability) program's RISMC (Risk Informed Safety Margin Characterization) effort and 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 RELAP5 and newly developed models will be reviewed and selected to reflect the progress made during the past three decades and provide a basis for the closure relations that will be required in RELAP-7. 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. Multiphysics 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 document provides a user's guide to help users to learn how to run the RELAP-7 code. A number of example problems and their associated input files are presented in this document to guide users to run the RELAP-7 code starting with simple pipe problems to problems with increasing complexity. Because the code is an ongoing development effort, this RELAP-7 User's Guide will evolve with periodic updates to keep it current with the state of the development, implementation, and model additions/revisions. A complete User's Manual will be developed at a later time when the RELAP-7 code becomes more mature.

1 RELAP-7 Features

An overall description of the RELAP-7 architecture, governing theory, and computational approach is given here as an instructive, and executive overview of the RELAP-7 distinguishing features.

1.1 Software Framework

MOOSE is INL's development and runtime environment for the solution of multi-physics systems that involve multiple physical models or multiple simultaneous physical phenomena. The systems are generally represented (modeled) as a system of fully coupled nonlinear partial differential equation systems (an example of a multi-physics system is the thermal feedback effect upon neutronics cross-sections where the cross-sections are a function of the heat transfer). Inside MOOSE, the Jacobian-Free Newton Krylov (JFNK) method [1, 2] is implemented as a parallel nonlinear solver that naturally supports effective coupling between physics equation systems (or Kernels). The physics Kernels are designed to contribute to the nonlinear residual, which is then minimized inside of MOOSE. MOOSE provides a comprehensive set of finite element support capabilities (LibMesh [3], a Finite Element library developed at University of Texas) and provides for mesh adaptation and parallel execution. The framework heavily leverages software libraries from DOE SC and NNSA, such as the nonlinear solver capabilities in either the the Portable, Extensible Toolkit for Scientific Computation (PETSc [4]) project or the Trilinos project [5] (a collection of numerical methods libraries developed at Sandia National Laboratory). Argonne's PETSc group has recently joined with the MOOSE team in a strong collaboration wherein they are customizing PETSc for our needs. This collaboration is strong enough that Argonne is viewed as a joint developer of MOOSE.

A parallel and tightly coordinated development effort with the RELAP-7 development project is the Reactor Analysis Virtual control ENvironment (RAVEN). This MOOSE-based application is a complex, multi-role software tool that will have several diverse tasks including serving as the RELAP-7 graphical user interface, using RELAP-7 to perform RISMC focused analysis, and controlling the RELAP-7 calculation execution.

Together, MOOSE/RELAP-7/RAVEN comprise the systems analysis capability of LWRS RISMC ToolKit.

1.2 Governing Theory

The primary basis of the RELAP-7 governing theory includes thermal fluids flow, reactor core heat transfer, and reactor kinetics models.

With respect to the thermal fluids flow dynamics models, RELAP-7 incorporates both single- and two-phase flow simulation capabilities encompassing all-speed and all-fluids. The single phase flow models include isothermal flow and nonisotherml flow capabilities. The two-phase flow models include the homogeneous equilibrium flow model (HEM) and the well posed two fluid 7-equation model.

In addition to the fluids flow dynamics model, RELAP-7 necessarily simulates the heat transfer process with reactor kinetics as the heat source. The heat-conduction equation for cylindrical or slab geometries is solved to provide thermal history within metal structures such as fuel and clad. The volumetric power source in the heat conduction equation for the fuel comes from the point kinetics model with thermal hydraulic reactivity feedback considered [6]. The reactor structure is coupled with the thermal fluid through energy exchange (conjugate heat transfer) employing surface convective heat transfer [7] within the fluid . The fluid, heat conduction, conjugate heat transfer and point kinetics equations are solved in a fully coupled fashion in RELAP-7 in contrast to the operator-splitting or loose coupling approach used in the existing system safety analysis codes.

1.3 Computational Approach

Stated previously, the MOOSE framework provides the bulk of the ”heavy lifting” available to MOOSE-based applications with a multitude of mathematical and numerical libraries. For RELAP-7, LibMesh [3] provides the second-order accurate spatial discretization by employing linear basis, one-dimensional finite elements. The Message Passing Interface (MPI, from Argonne National Laboratory) provides for distributed parallel processing. Intel Threading Building Blocks (Intel TBB) allows parallel C++ programs to take full advantage of multicore architecture found in most large-scale machines of today. PETSc (from Argonne), Trilinos (from Sandia), and Hypre [8] (from Lawrence Livermore National Laboratory) provide the mathematical libraries and nonlinear solver capabilities for the Jacobian-free Newton-Krylov (JFNK) method. In MOOSE, a stiffly-stable, second-order backward difference (BDF2) formulation is used to provide second-order accurate time integration for strongly coupled physics in JFNK.

The JFNK method easily allows implicit nonlinear coupling of dependent physics under one general computational framework. Besides rapid (second-order) convergence of the iterative procedure, the JFNK method flexibly handles multiphysics problems when time scales of different physics are significantly varied during transients. The key feature of the JFNK method is combining Newton's method to solve implicit nonlinear systems with Krylov subspace iterative methods. The Krylov methods do not require an explicit form of the Jacobian, which eliminates the computationally expensive step of forming Jacobian matrices (which also may be quite difficult to determine analytically), required by Newton's method. The matrix-vector product can be approximated by the numerical differentiation of nonlinear residual functions. Therefore, JFNK readily integrates different physics into one solver framework.

2 Model Description

2.1 Fluids Flow Models

The RELAP-7 code has various flow models implemented. These include:
1). a single phase isothermal flow model (`model_type=2`),
2). a single phase nonisothermal fluid flow model (`model_type=3`),
3). a homogeneous equilibrium (HEM) two phase flow model (`model_type=32`),
4). a nonhomogeneous, nonequilibrium seven-equation two phase flow model (`model_type=7`).

2.2 Equation of State

For the **single phase flow models**, various types of equation of state can be used. These include:
1). Barotropic equation of state for isothermal flow. This is turned on by setting `type=BarotropicEquationOfState`.
2). Linear equation of state for non-isothermal single phase flow. This is turned on by setting `type = NonIsothermalEquationOfState`.
3). Stiffened gas equation of state which is turned on by setting
`type = StiffenedGasEquationOfStateVapor` for single phase vapor flow and
`type = StiffenedGasEquationOfStateLiquid` for single phase water flow.
4). Ideal gas equation of state which is turned on by setting `type=IdealGasEquationOfState`.
The equation of state for the nitrogen gas has also been implemented into the code. It can be turned on by setting `type=N2Properties`.

For the **homogeneous equilibrium flow model**, only the stiffened gas equation of state is applicable. This is turned on by setting `type = TwoPhaseStiffenedGasEOS`.

For the **seven-equation two phase flow model**, only the stiffened gas equation of state is applicable. This is turned on by setting
`type = StiffenedGasEquationOfStateLiquid` for the liquid phase,
and `type = StiffenedGasEquationOfStateVapor` for the vapor phase.

2.3 Solution Stabilization Schemes

It is well known that the continuous Galerkin finite element method is unstable when applied directly to hyperbolic systems of equations. Therefore the solution stabilization schemes are required for RELAP-7. Currently available options of solution stabilization for RELAP-7 include:
1). Streamline Upwind/Petrov Galerkin method (SUPG). The SUPG scheme works for the single phase flow only. It can be used by setting `stabilization_type = 'SUPG'`.
2). Lapidus scheme works for both the single phase flow and the two phase flow cases. It can be used by setting `stabilization_type = 'LAPIDUS'`.
3). The entropy viscosity method works for both the single phase flow and the two phase flow. This option can be used by setting `stabilization_type = 'ENTROPY_VISCOSITY'`.

2.4 Time Integration Schemes

There are two types of time integration schemes in RELAP-7 - Implicit Euler and BDF2. Implicit Euler is a first order accurate time integration scheme. This can be turned on by setting:`scheme = 'implicit-euler'` in the `Executioner` input block (explained later). BDF2 is a second order accurate time integration scheme. This is the default option for RELAP-7.

2.5 Components

A real reactor system is very complex and contains hundreds of different physical components. It is impractical to resolve the real geometry of the entire system. Instead simplified thermal hydraulic models are used to represent (via "nodalization") the major physical components and describe the major physical processes (such as fluids flow and heat transfer). There are three main types of components developed in RELAP-7: (1) one-dimensional (1-D) components describing the geometry of the reactor system, (2) zero-dimensional (0-D) components for setting boundary conditions, and (3) 0-D components for connecting 1-D components.

2.5.1 Pipe

Pipe is the most basic component in RELAP-7. It is a 1-D component which simulates thermal fluids flow in a pipe. Both a constant cross section area and a variable cross section area options are available for the Pipe component. The wall friction and heat transfer coefficients are either calculated through closure models or provided by user input. The pipe wall temperature can be provided as the wall heat transfer boundary condition. All the thermal fluids dynamic models are available in the Pipe component which includes the isothermal flow model, single-phase non-isothermal flow model, nonequilibrium 7-equation two-phase model, and the much simpler homogeneous equilibrium two-phase flow model.

2.5.2 PipeWithHeatStructure

The PipeWithHeatStructure component simulates fluids flow in a 1-D pipe coupled with 1-D or 2-D heat conduction through the pipe wall. The adiabatic, Dirichlet, or convective boundary conditions at the outer surface of the pipe wall are available. Either a plate type or cylindrical type of heat structure can be selected. Volumetric heat source within the fluids or solid materials can be added.

2.5.3 CoreChannel

The CoreChannel component is a composite component designed to simulate the coolant flow and heat conduction inside a fuel rod as well as the conjugate heat transfer between the coolant and the fuel rod. In this component, the fuel rod is divided into the same number of segments as that of the coolant flow pipe elements. Each fuel rod segment is further simulated as 1-D or 2-D heat conduction model perpendicular to the fluid flow model. Both plate type fuel rod and cylindrical fuel rod type can be simulated. The solid fuel part is able to deal with typical LWR fuel rod with complex clad/gap/fuel pellet geometries. The flow model and conjugate heat transfer model are fully coupled in contrast to loosely coupled in RELAP5.

2.5.4 Subchannel

A fully coupled subchannel channel model for the single-phase has been implemented into RELAP-7. The single-phase subchannel model includes four balance equations: mass, energy, axial momentum, and lateral momentum.

2.5.5 HeatExchanger

A Heat Exchanger component is a combination of two pipes with a solid wall in between. Similar to the CoreChannel model, the fluids flow model and conjugate heat transfer model are fully coupled. More complicated and realistic steam generator component will be developed in the future.

2.5.6 Branch

The branch model is a 0-D component representing a junction model with no volume (inertia) effects considered, and with single/multiple inlets and single/multiple outlets, of which cross section areas can be different. This model conserves the mass and energy among all connecting components.

2.5.7 VolumeBranch

The volume branch model is a 0-D component representing a joint/junction model with volume (inertia) effects considered. This model conserves the mass and energy among all connecting components.

2.5.8 SubchannelBranch

This is a 0D branch model to connect the subchannel component.

2.5.9 Pump

A simple pump model to provide a head and a reverse flow form loss coefficient (K) for either isothermal flow and non-isothermal flow. It can be driven by an user input head or through a driving component which provides shaft work.

2.5.10 Turbine

The turbine model in RELAP-7 is a simplified dynamical turbine model to simulate a reactor core isolation cooling (RCIC) turbine, which drives the RCIC pump through a common shaft.

2.5.11 SeparatorDryer

The separator dryer model in RELAP-7 separates steam and water with mechanical methods. Only an ideal separator dryer model is available in RELAP-7.

2.5.12 DownComer

The down comer component simulates a large volume to mix different streams of water and steam and to track the water level.

2.5.13 Valve

The valve component simulates the open and close behaviors of valves for incompressible flow with user given trigger and response time. The abrupt area change model is used to calculate the form loss.

2.5.14 CompressibleValve

The compressible valve component simulates the open and close behavior of valves for compressible fluid flow, including ckoking. It can be used to simulate the safety relief valves (SRV) of BWRs.

2.5.15 Check Valves

The check valve component simulates the dynamic behavior of check valves, with the form loss calculated by the abrupt area change model.

2.5.16 WetWell

The wet well component simulates the dynamic response of a BWR suppression pool and its gas space.

2.5.17 TimeDependentVolume

The time depedent volume component provides pressure, temperature, and void fraction boundary conditions as constants or time dependent functions for 1-D components. It is a pure boundary condition type of component and it does not add any entries to the global unknown vector. When acquired by its connected 1-D component, it provides a pressure, a temperature, and a void fraction boundary condition.

2.5.18 TimeDependentJunction

The time dependent junction component provides velocity and temperature boundary conditions as constants or time functions for 1-D components.

2.5.19 Tdm

Tdm sets time dependent mass flow boundary conditions for 1-D components.

2.5.20 Reactor

A virtual component that allows users to input the power for the core channel component.

2.5.21 PointKinetics

The point kinetics model is a lumped parameter neutron kinetics model to calculate the reactor power. User input reactivity or fully coupled feedback reactivity models are available.

3 Running RELAP-7

3.1 Complete Step 1 of MOOSE Environment Setup

The system environment setup for MOOSE can be found with the link:
`http://www.mooseframework.org/getting-started`

3.2 Setup Your SSH Key

SSH key allows you to establish a secure connection between your computer and GitLab. Before generating an SSH key, check to see if your system already has one by running `cat /.ssh/id_rsa.pub`. If you see a long string starting with `ssh-rsa` or `ssh-dsa`, you can skip the `ssh-keygen` step below.

To generate a new SSH key, just open your terminal and use the code below. The `ssh-keygen` command prompts you for a location and filename to store the key pair and for a password. When prompted for the location and filename you can press `enter` to use the default. It is a best practice to use a password for an SSH key but it is not required and you can skip creating a password by pressing `enter`. Note that the password you choose here can not be altered or retrieved.

```
ssh-keygen -t rsa -C "$your_email"
```

Use the code below to show your public key.

```
cat ~/.ssh/id_rsa.pub
```

Copy and paste the key to the `My SSH Key` section under the `SSH` tab in your profile. Please copy the complete key starting with `ssh-` and ending with your username and host.

To test your SSH key.

```
$ssh git@hpcgitlab.inl.gov
Welcome to GitLab, <your name here>!
Connection to hpcgitlab.inl.gov closed.
```

3.3 Checking Out the Code

```
$ cd ~/projects/
$ git clone git@hpcgitlab.inl.gov:idaholab/relap-7.git
$ cd relap-7
$ git submodule update --init
```

It is necessary to build libmesh before building any application.

```
$ cd ~/projects/relap-7/moose
$ ./update_and_rebuild_libmesh.sh
```

Once libmesh has been successfully compiled, you may now compile RELAP-7.

```
cd ~/projects/trunk/relap-7
make (add -jn to run on multiple "n" processors)
```

Once RELAP-7 has been compiled successfully, it is recommended to run the tests to make sure the version of the code you have is running correctly.

```
cd ~/projects/trunk/relap-7
./run_tests (add -jn to run "n" jobs at one time)
```

3.4 Executing RELAP-7

When first starting out with running RELAP-7, it is recommended to start from an example problem. Multiple example problems with input files are presented in this document. More examples can be found under the `/relap-7/tests` subdirectory. To demonstrate how to run RELAP-7, consider the `PWR_core_channel.i` test problem.

```
cd ~/projects/trunk/relap-7/examples/components/core_channel/
# To run with one processor
~/projects/trunk/relap-7/relap-7-opt -i PWR_core_channel.i
# To run in parallel (4 processors)
mpiexec -n 4 ../../../relap-7-opt -i PWR_core_channel.i
```

3.5 Post Processing

RELAP-7 typically writes solution data to an ExodusII file. The solution data may also be written in other formats, one being a comma separated values (CSV) file, which allows the solutionn data to be saved in a table structured format. The other being a tecplot file in either binary or ASCII format.

Several options exist for viewing ExodusII output files. One good choice is to use open-source software, Paraview (`www.paraview.org`).

3.6 Graphical User Interface

Another MOOSE based application named RAVEN provides a graphical user interface (GUI) for RELAP-7. RAVEN can be used to generate a text input fie. It is also capable of submitting the analysis and provides post processing capabilitis.

4 Input Files

RELAP-7 uses a block-structured input file. Each block is identified with square brackets. The opening brackets contain the type of the input block and the empty brackets mark the end of the block. Each block may contain subblocks.

```
[BlockName]
   <block line commands>
   [./subblock_name]
      <subblock line commands>
   [../]
[]
```

Each subblock must have an unique name when compared with all other subblocks in the current block.

Line commands are given as parameter and value pairs with an equal sign between them. They specify parameters to be used by the object being described. The parameter is a 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.

Subblocks normally contain a type line command. This line command specifies the particular type of object being described.

RELAP-7 uses SI units. This stsndardizes the model input by eliminating the possibility of errors caused by using one set of units for one model and another set of units for a different model.

The following subsections have brief descriptions of each block. More detailed descriptions can be found in the examples section.

4.1 Global Parameters

The GlobalParams block specifies the global parameters used by the code such as the initial pressure (`global_init_P`), velocity (`global_init_V`) and temperature (`global_init_T`) of the system model, the fluid flow model type (`model_type`), the stabilzation scheme type (`stabilization_type`), and the scaling factors (`scaling_factor_var`) for the primary variable, etc. The values of of global parameters are available to any other block or subblock in the input file. If a command line is missing in a block or a subblock but defined in `GlobalParams`, the block or subblock will use the parameter defined in `GlobalParams`. However, if the block or subblock has a command line , that will be used regardless of what is in GlobalParams.

The following is an example of the `GlobalParams` block:

```
[GlobalParams]
  global_init_P = 155.e5
  global_init_V = 0.
  global_init_T = 559.15
  # model_type = 2
  # model_type = 32
  # model_type = 7
  model_type = 3
  stabilization_type = 'SUPG'
  scaling_factor_var = '1e4 1e1 1e-2'
  temperature_sf = '1e-2'
[]
```

symbol indicates comments in the input file and can be located anywhere in the input file.

4.2 Equation of State

The EoS block specifies the equation of state to be used by the code. The following is an example of the EoS block:

```
[EoS]
  [./eos]
    type = NonIsothermalEquationOfState
    p_0 = 155.e5
    rho_0 = 686.
    a2 = 1.e7
    beta = .46e-3
    cv = 5.5e3
    e_0 = 3075325
    T_0 = 559.15
  [../]
```

```
[]
```

4.3 Materials

The Materials block specifies the properties of the solid materials for the code. The following is an example of the Materials block:

```
[Materials]
  [./fuel-mat]
    type = SolidMaterialProperties
    k = 2.5
    Cp = 300.
    rho = 1.032e4
  [../]
  [./gap-mat]
    type = SolidMaterialProperties
    k = 0.6
    Cp = 1.
    rho = 1.
  [../]
  [./clad-mat]
    type = SolidMaterialProperties
    k = 21.5
    Cp = 350.
    rho = 6.55e3
  [../]
[]
```

4.4 Functions

The Functions block provides the functions to be used by the code during the simulations such as reactor power as a function of time or a boundary condition pressure as a function of time, etc. The following is an example of defining pressure distribution as a function of x.

```
[Functions]
```

```
  [./p_func]
    axis = 0
    type = PiecewiseLinear
    x = '0        3'
    y = '1.05e5   1e5'
  [../]
[]
```

4.5 Components

The `Components` block specifies the components to be used in the simulations.

```
[Components]
  [./reactor]
    type = Reactor
    initial_power = 77337.69407
  [../]

  [./CCH1]
    type = CoreChannel
    eos = eos
    position = '0 0 0'
    orientation = '0 0 1'
    A = 8.7878e-5
    Dh = 0.01179
    length = 4
    n_elems = 20
    f = 0.01
    Hw = 5.33e4
    Phf = 0.029832559676
    Ts_init = 559.15
    dim_hs = 1
    fuel_type = cylinder
    name_of_hs = 'fuel gap clad'
    n_heatstruct = 3
    width_of_hs = '0.004096 0.0001 0.000552'
    elem_number_of_hs = '10 1 2'
    material_hs = 'fuel-mat gap-mat clad-mat'
    power_fraction = '1.0 0.0 0.0'
```

```
  [../]

  [./inlet]
    type = TimeDependentVolume
    input = 'CCH1(in)'
    p_bc = 155.483e5
    T_bc = 559.15
    eos = eos
  [../]

  [./outlet]
    type = TimeDependentVolume
    input = 'CCH1(out)'
    p_bc = '155.e5'
    T_bc = 559.15
    eos = eos
  [../]
[]
```

4.6 Preconditioner

The `Preconditioning` block specifies the preconditioner to be used by the preconditioned JFNK solver for the RELAP-7 code. The solution algorithm for RELAP-7 is the Jacobian-free Newton-Krylov (JFNK) method. However the Krylov methods need preconditioning to be efficient. Hence, the solvers available in RELAP-7 are preconditioned JFNK (PJFNK). Two options are available in RELAP-7 to build the preconditioning matrix, the single matrix preconditioner (SMP) and the finite difference preconditioner (FDP). The SMP option uses all the Jacobian terms derived analytically to build one preconditioning matrix. The FDP option uses numerical Jacobian by doing direct finite differences of the residual terms. The SMP option is the more efficient and is the recommended option, while the FDP option is normally slow and inefficient and is recommended to be used for small problems or for debugging purposes. The following is an example of the `Preconditioning` block:

```
[Preconditioning]
  # Uncomment one of the lines below to activate one of the blocks...
   active = 'SMP_PJFNK'
  #active = 'FDP_PJFNK'
```

```
  # The definitions of the above-named blocks follow.
  [./SMP_PJFNK]
    type = SMP
    full = true  # off diagonal blocks are used
    solve_type = 'PJFNK' # Preconditioned JFNK solver
  [../]


  [./FDP_PJFNK]
    type = FDP
    full = true
    solve_type = 'PJFNK'
  [../]
[]
```

4.7 Executioner

The `Executioner` block specifies the executioner that will be used in the simulations. The options include Transient, RavenExecutioner, etc.

```
[Executioner]
  type = Transient
  dt = 0.5
  dtmin = 1.e-5
  petsc_options_iname = '-mat_fd_type  -mat_mffd_type  -pc_type'
  petsc_options_value = '  ds               ds             lu'
  nl_rel_tol = 1e-8
  nl_abs_tol = 1e-8
  nl_max_its = 10
  l_tol = 1e-4
  l_max_its = 60
  start_time = 0.0
  num_steps = 30
  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]
```

4.8 Outputs

The `Outputs` block controls the various screen and file output in the simulations.

```
[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5 Examples

5.1 Example 1: A Simple Pipe Flow Problem

5.1.1 Problem Description

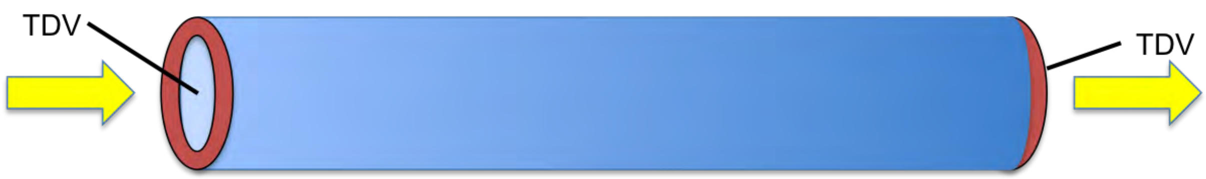

Figure 1. A simple pipe flow problem diagram

Example 1 simulates water flowing through a pipe under isothermal conditions with the following parameters:

Hydraulic Diameter = .01 m
Cross section flow area = 7.85e-5 m^2
Length = 1 m
Wall friction coefficient = .01

The boundary conditions are the following:

Inlet: Pressure = 1.05e5 Pa
Outlet: Pressure = 1e5 Pa

A 1D model can be viewed in Paraview to visualize the process better. Once Paraview is opened, on the left, select all of the variables and click ”Apply.” In the filters tab, under ”Data Analysis”, select ”Plot Over Line” and apply. When this is done, each parameter in the problem can be viewed plotted over the length of the pipe. The figure below shows pressure vs length.

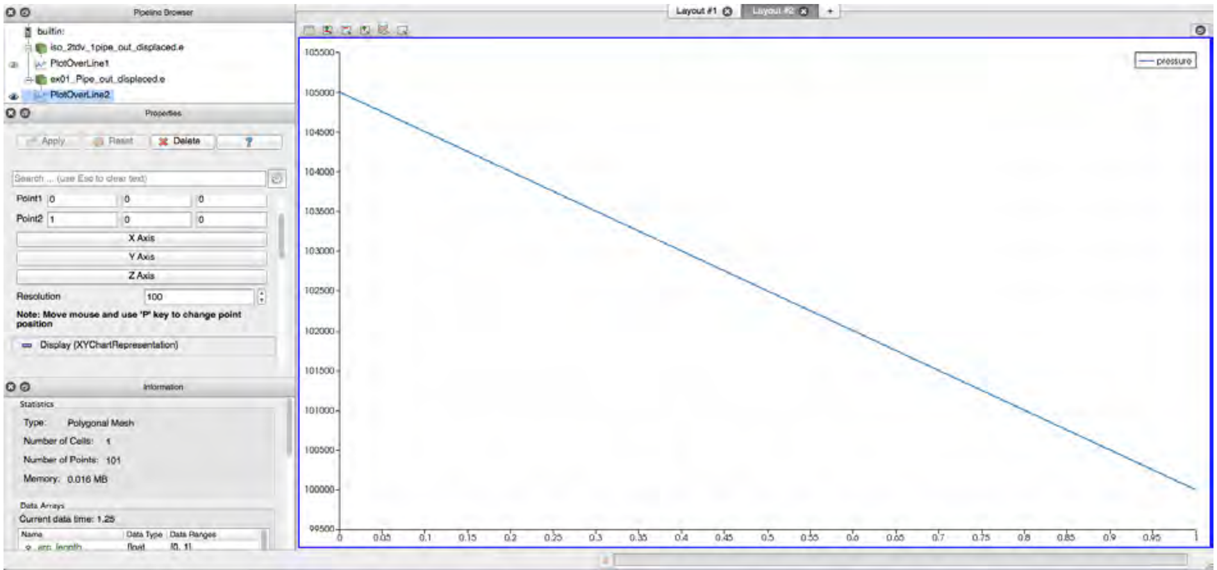

Figure 2. Pressure vs. length for the simple pipe flow problem

The pressure difference can be hand calculated to check the code calculated results using:

$$\Delta P = \frac{fL\rho u^2}{2D_h} \quad (1)$$

Where ρ is the density and u is the velocity. Both values can be found in Paraview to perform the calculation.

5.1.2 Input File

The following shows the input file to run this example problem:

```
#
# Isothermal flow through one simple pipe with B.C. set by TDVs
#

[GlobalParams]
  model_type = 2
  scaling_factor_var   = '1e+0 1e-4'
  stabilization_type = 'SUPG'
[]

[EoS]
  [./eos]
    type = BarotropicEquationOfState
    p_0 = 1.e5
    rho_0 = 1.e3
    a2 = 1.e7
  [../]
[]

[Components]
  [./pipe1]
    type = Pipe
    eos = eos
    position =    '0 0 0'
    orientation = '1 0 0'
    Dh = 0.01
    length = 1.0
    n_elems = 10
    A = 7.854e-5
    f = 0.01
  [../]

  [./inlet]
    type = TimeDependentVolume
    input = 'pipe1(in)'
    p_bc = 105e3
    eos = eos
  [../]
```

```
  [./outlet]
    type = TimeDependentVolume
    input = 'pipe1(out)'
    p_bc = 1e5
    eos = eos
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
  [../]
[]

[Executioner]
  type = Transient
  scheme = 'implicit-euler'
  dt = 1.25e-2
  dtmin = 1.e-5

  nl_rel_tol = 1e-8
  nl_abs_tol = 1e-6
  nl_max_its = 15

  l_tol = 1e-6
  l_max_its = 30

  start_time = 0.0
  end_time = 250.0
  num_steps = 500

  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
```

```
[]

[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5.1.3 Description of the Input File

The following are the destailed descriptions of the input file block by block.

The Global Parameters block:

```
[GlobalParams]
  model_type = 2
  scaling_factor_var    = '1e+0 1e-4'
  stabilization_type = 'SUPG'
[]
```

`model_type`	Fluid flow model. 2 means single phase isothermal fluid flow.
`scaling_factor_var`	Scaling factor for the primary variables ρA and $\rho u A$ respectively.
`stabilization_type`	Stabilization scheme type. Available optionsd are: `SUPG` for single phase flow, `LAPIDUS` for both single- and two-phase flow, `ENTROPY_VISCOSITY` for both single- and two-hase flow.

The Equation of State block:

```
[EoS]
  [./eos]
    type = BarotropicEquationOfState
    p_0 = 1.e5
    rho_0 = 1.e3
    a2 = 1.e7
  [../]
[]
```

`[./eos]` Subblock for eos

`type` The type of equation of state to be used. The barotropic equation of state is suitable for isothermal fluid flow model. It describes isentropic (reversible) processes, and implies a constant sound speed. It is given by $p = p_0 + a^2(\rho - \rho_0)$.

`p_0` Initial pressure value (Pa) in the equation of state calculations.

`rho_0` Initial value of density (kg/m^3) in the equation of state calculation.

`a2` a is a constant in the barotropic equation of state. The value of a can be taken roughly as the speed of sound. For this example problem with water as the fluids, using $a^2 = 1.e7$ has the same order of magnitude as that of the square of sound speed. This is the recommended value to use.

The Components Block:

```
[Components]
  [./pipe1]
    type = Pipe
    eos = eos
    # geometry
    position =    '0 0 0'
    orientation = '1 0 0'
    Dh = 0.01
    length = 1.0
    n_elems = 10
    A = 7.854e-5
```

```
    f = 0.01
  [../]

  [./inlet]
    type = TimeDependentVolume
    input = 'pipe1(in)'
    p_bc = 105e3
    eos = eos
  [../]

  [./outlet]
    type = TimeDependentVolume
    input = 'pipe1(out)'
    p_bc = 1e5
    eos = eos
  [../]
[]
```

`[./pipe1]`	Subblock for pipe1.
`type`	Component type is pipe.
`eos`	Use the eos equation of state defined in the EoS block.
`position`	The starting x, y and z coordinate of pipe1.
`orientation`	The orientation vector of pipe1. Horizontal pipe for this case.
`Dh`	Hydraulic diameter of of the pipe.
`length`	Pipe length.
`n_elems`	Number of elements in the solution of the pipe.
`A`	The cross section flow area of the pipe.
`f`	The wall friction coefficient. If this is not provided in the input file, the code built-in correlations will be used to calculate the friction coefficients (for single phase flow only).

`[./inlet]` Subblock for the component to set the inlet boundary conditions.

`type` Component type is time dependent volume.

`input` The pipe inlet end which will be connected with this component.

`p_bc` The pressure boundary condition value at the inlet of the pipe.

`eos` The equation of state to be used.

`[./outlet]` Subblock for the component to set the outlet boundary conditions.

`type` Component type is time dependent volume.

`input` The pipe outlet end which will be connected with this component.

`p_bc` The pressure boundary boundary condition value at the outlet of the pipe.

`eos` The equation of state to be used.

The Preconditioning Block:

```
[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
  [../]
[]
```

`[./SMP_PJFNK]` Subblock for SMP preconditioner and PJFNK solver. This is the recommended user opton. For advanced users, please refer to MOOSE and PETSc manuals.

`type` Indicating the preconditioning matrix type is SMP for this case.

`full` true means the entire preconditioning matrix will be assembled in the calculations.

solve_type PJFNK solver will be used for this problem.

The Executioner Block:

```
[Executioner]
  type = Transient
  scheme = 'implicit-euler'
  dt = 1.25e-2
  dtmin = 1.e-5

  nl_rel_tol = 1e-8
  nl_abs_tol = 1e-6
  nl_max_its = 15

  l_tol = 1e-6
  l_max_its = 30

  start_time = 0.0
  end_time = 250.0
  num_steps = 500

  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]
```

type Transient executioner type will be used.

scheme Implicit euler time integration will be used. If this is absent, the default option of BDF2 will be used.

dt Time step size to be used .

dtmin Minimum time step size to be used. If the time step size is below this value, the case will stop executing.

nl_rel_tol Relative tolerance for the nonlinear solve. The recommended value is 1.0E-8.

nl_abs_tol	Absolute tolerance for the nonlinear solve. The recommended value is 1.0E-6.
nl_max_its	Maximum number of nonlinear solve iterations. The recommended value is 15.
l_tol	Relative tolerance for the linear Krylov solve. The recommended value is 1.0E-6.
l_max_its	Maximum number of liner iterations. The recommended value is 30.
start_time	The start time of the simulation.
end_time	The end time of the simulation.
num_steps	The maximum number of time integration steps for the simulation. The simulation will stop either the maximum number of time steps is reached or the end_time is reach.
[./Quadrature]	The quadrature subblock. This is the recommended option.
type	Type of the quadrature rule. Advanced users can refer to the MOOSE manual for other options.
order	Order of the quadrature.

The Outputs Block:

```
[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
```

```
    perf_log = true
  [../]
[]
```

`[./out]`	Output subblock. This is the recommended option.
`type`	Exodus output file type.
`use_displaced`	Use the displaced mesh to show the results.
`output_initial`	The initial condition is output to the solution file.
`sequence`	Enable or disable sequential file output.
`append_displaced`	Append '_displaced' to the output file name.
`[./console]`	Console subblock.
`type`	Console type.
`perf_log`	Print all performance logs.

5.2 Example 2: Use Functions to Set Initial Conditions for a Simple Pipe Flow Problem

5.2.1 Problem Description

This example illustrates how to use functions to set the initial conditions. The example used here is to set the initial pressure distribution in a pipe for an isothermal flow through one simple pipe with boundary conditions set by time dependent volumes (TDV).

Functions:
Initial pressure distribution in the pipe
Axis = 0 The axis used (0, 1, or 2 for x, y, or z) if this is to be a function of position. Note that for RELAP-7 1-D component such as a pipe, we only use x coordinate in the calculations therefore we only need to set `Axis = 0`.
Type = PiecewiseLinear
x = '0 3' This is the pipe axial direction spatial coordinate relative to the starting point
y = '1.05e5 1e5'

Initial velocity distribution in the pipe
Axis = 0
Type = PiecewiseLinear
x = '0 3'
y = '3 3'

Fig. 3 shows the plot of pressure versus pipe length for this case.

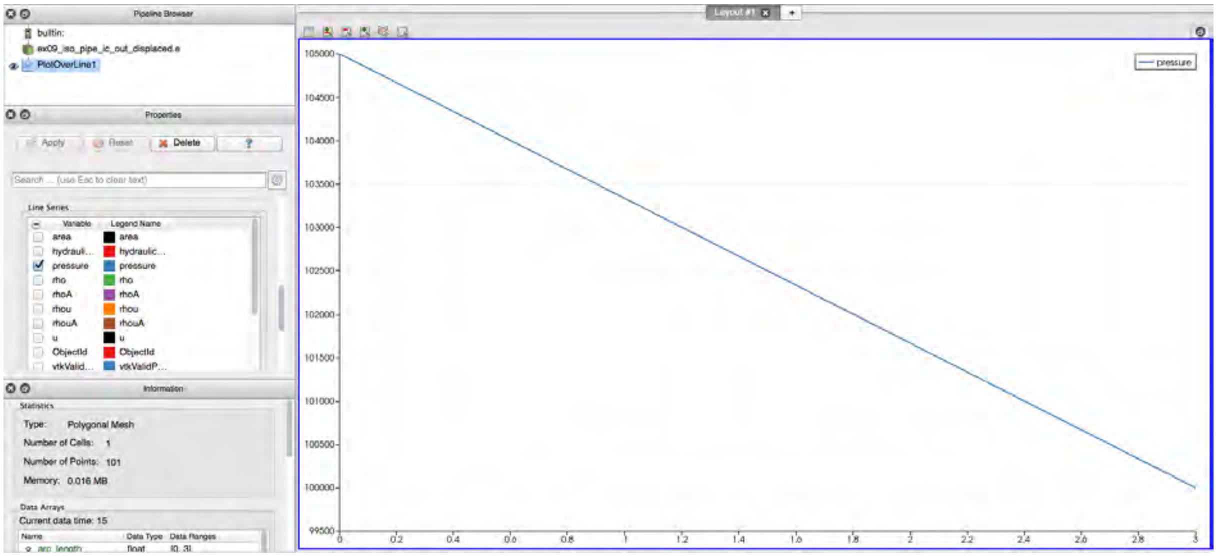

Figure 3. Pressure vs length for an isothermal pipe flow problem with initial pressure set by functions

5.2.2 Input File

The input file for this example is shown as follows:

```
#
# Isothermal flow through one simple pipe with B.C. set by TDVs and I.C. set by functions
#

[GlobalParams]
  model_type = 2
  scaling_factor_var    = '1e4 1.0'
  stabilization_type = 'SUPG'
[]

[EoS]
  [./eos]
    type = BarotropicEquationOfState
    p_0 = 1.e5
```

```
    rho_0 = 1.e3
    a2 = 1.e7
  [../]
[]

[Functions]
  # initial pressure distribution in the pipe
  [./p_func]
    axis = 0
    type = PiecewiseLinear
    x = '0        3'
    y = '1.05e5   1e5'
  [../]

  # initial velocity distribution in the pipe
  [./v_func]
    axis = 0
    type = PiecewiseLinear
    x = '0    3'
    y = '3    3'
  [../]
[]

[Components]
  [./pipe1]
    type = Pipe
    eos = eos
    position =    '0 2 0'
    orientation = '1 1 1'
    Dh = 0.01
    length = 3.0
    n_elems = 10
    A = 7.854e-5
    f = 0.01
    P_func = p_func
    V_func = v_func
  [../]

  [./inlet]
    type = TimeDependentVolume
    input = 'pipe1(in)'
```

```
    eos = eos
    p_bc = 1.05e5
  [../]

  [./outlet]
    type = TimeDependentVolume
    input = 'pipe1(out)'
    eos = eos
    p_bc = 1.0e5
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
  [../]
[]

[Executioner]
  type = Transient
  dt = 0.5
  dtmin = 1.e-5

[Executioner]
  type = Transient
  dt = 0.5
  dtmin = 1.e-5

  petsc_options_iname = '-mat_fd_type  -mat_mffd_type  -pc_type  -ksp_gmres_restart'
  petsc_options_value = '  ds              ds             lu          30'

  nl_rel_tol = 1e-8
  nl_abs_tol = 1e-8
  nl_max_its = 10

  l_tol = 1e-4
  l_max_its = 60

  start_time = 0.0
```

```
  num_steps = 30

  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]

[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5.2.3 Description of the Input File

The following will provde explanations to the input parameters that did not exist in Example 1.

The Functions Block:

```
[Functions]
  # initial pressure distribution in the pipe
  [./p_func]
    axis = 0
    type = PiecewiseLinear
    x = '0        3'
    y = '1.05e5   1e5'
  [../]

  # initial velocity distribution in the pipe
  [./v_func]
```

```
    axis = 0
    type = PiecewiseLinear
    x = '0    3'
    y = '3    3'
  [../]
[]
```

[./p_func]	Subblock for the pressure distribution function.
axis	For RELAP-7 1-D components, only the x coordinate is used in the calculations and hence `axis` is set to be 0 which allows a function to be defined as a function `x`. If `axis` is not given in the input file, the data pair of (`x`,`y`) will be taken as a time function.
type	Type of functions used to interpolate data between data points.
x	The coordinate used for the x-axis data.
y	The initial pressure distribution function data point values.

The Components Block:

```
[Components]
  [./pipe1]
    type = Pipe
    eos = eos
    position =    '0 2 0'
    orientation = '1 1 1'
    Dh = 0.01
    length = 3.0
    n_elems = 10
    A = 7.854e-5
    f = 0.01
    P_func = p_func
    V_func = v_func
  [../]

  [./inlet]
    type = TimeDependentVolume
    input = 'pipe1(in)'
    eos = eos
```

```
    p_bc = 1.05e5
  [../]

  [./outlet]
    type = TimeDependentVolume
    input = 'pipe1(out)'
    eos = eos
    p_bc = 1.0e5
  [../]
[]
```

P_func Use the pressure function defined in the Functions block to set the initial pressure value.

V_func Use the velocity function defined in the Functions block to set the initial velocity value.

The Executioner Block:

```
[Executioner]
  type = Transient
  dt = 0.5
  dtmin = 1.e-5

[Executioner]
  type = Transient
  dt = 0.5
  dtmin = 1.e-5

  petsc_options_iname = '-mat_fd_type  -mat_mffd_type  -pc_type'
  petsc_options_value = '  ds              ds              lu'

  nl_rel_tol = 1e-8
  nl_abs_tol = 1e-8
  nl_max_its = 10

  l_tol = 1e-4
  l_max_its = 60

  start_time = 0.0
  num_steps = 30
```

```
  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]

  nl_rel_tol = 1e-8
  nl_abs_tol = 1e-8
  nl_max_its = 10

  l_tol = 1e-4
  l_max_its = 60

  start_time = 0.0
  num_steps = 30

  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]
```

`petsc_options_iname` Names of PETSc name and value pairs. `-mat_fd_type = ds` is an option to control how the Jacobian matrix is approximated by using the finite differencing method. `-mat_mffd_type = ds` is an option to control how matrix-free operation is approximated by using the finite difference method. Users are recommended to refer to the PETSc manual for other options.

`petsc_options_value` Values of PETSc name/value paris.

5.3 Example 3: Two Phase Flow Through a Pipe - HEM Model

5.3.1 Problem Description

Two phase flow with HEM model through one simple pipe with boundary conditions set by time dependent volumes and initial conditions set by functions. The initial conditions are close to steady state solutions; therefore the simulation quickly converges to the steady state.

The following is the description of the functions defined in the `Functions` block:
Initial pressure distribution in the pipe
Axis = 0
Type = PiecewiseLinear
x = ’0 1’
y = ’7.01e6 7e6’

Initial velocity distribution in the pipe
Axis = 0
Type = PiecewiseLinear
x = ’0 1’
y = ’1.35 2.1’

Initial temperature distribution in the pipe
Axis = 0
Type = PiecewiseLinear
x = ’0 0.04 1’
y = ’517 517.343 517.253’

Initial void fraction distribution in the pipe
axis = 0
type = PiecewiseLinear
x = ’0 0.04 1’
y = ’0 0. 0.37’

The parameters and functions used for the pipe component are the following:
Position = ’0 0 0’
Orientation = ’1 0 0’

Area = $7.854e-5m^2$
Diameter = 0.01 m
Length = 1. m
Number of elements = 100
Friction coefficient = 0.1
Convective heat transfer coefficient = 1e4
Heat flux perimeter = 0.031416
Wall temperature = 528 K
The name of EOS to use = two_phase_eos
Initial condition determined by functions which are close to SS solutions
Function which defines pressure as a function of x = p_func
Function which defines velocity as a function of x = v_func
Function which defines temperature as a function of x = temp_func
Function which defines vapor volume fraction as a function of x = void_func

The inlet boundary conditions:
Vapor volume fraction = 0.
The name of EOS to use = two_phase_eos

The outlet boundary conditions:
Vapor volume fraction = 0.5
The name of EOS to use = two_phase_eos

5.3.2 Input File

```
#
# Two phase flow with HEM model through one simple pipe with B.C. set by TDVs and
# I.C. set by functions. The I.C.s are close to steady state solutions; therefore
# the simulation quickly converges to the steady state.
#

[GlobalParams]
  model_type = 32
  stabilization_type = 'NONE'
  scaling_factor_var   = '1e+4 1 1e-2'
  gravity = '0.0, 0.0, -9.8'
[]

[EoS]
  [./two_phase_eos]
    type = TwoPhaseStiffenedGasEOS
  [../]
[]

[Functions]

  # initial pressure distribution in the pipe
  [./p_func]
    axis = 0
    type = PiecewiseLinear
    x = '0        1'
    y = '7.01e6   7e6'
  [../]

  # initial velocity distribution in the pipe
  [./v_func]
    axis = 0
    type                      = PiecewiseLinear
    x                         = '0     1'
    y                         = '1.35  2.1'
  [../]

  # initial temperature distribution in the pipe
```

```
  [./temp_func]
    axis = 0
    type                        = PiecewiseLinear
    x                           = '0     0.04     1'
    y                           = '517   517.343  517.253'
  [../]

  # initial void fraction distribution in the pipe
  [./void_func]
    axis = 0
    type                        = PiecewiseLinear
    x                           = '0   0.04  1'
    y                           = '0   0.    0.37'
  [../]
[]

[Components]
  [./pipe1]
    type = Pipe
    position =    '0 0 0'
    orientation = '1 0 0'
    A = 7.854e-5
    Dh = 0.01
    length = 1.
    n_elems = 100
    f = 0.1
    Hw = 1e4
    Phf = 0.031416000000
    Tw = 528
    eos = two_phase_eos

    # initial condition determined by functions which are close to steady state solutions
    P_func = p_func
    V_func = v_func
    T_func = temp_func
    volume_fraction_vapor_func = void_func
  [../]

  [./inlet]
    type = TimeDependentVolume
    input = 'pipe1(in)'
```

```
    p_bc = 7.01e6
    T_bc = 517.
    volume_fraction_vapor_bc = 0.
    eos = two_phase_eos
  [../]

  [./outlet]
    type = TimeDependentVolume
    input = 'pipe1(out)'
    p_bc = 7.0e6
    T_bc = 517
    volume_fraction_vapor_bc = 0.5
    eos = two_phase_eos
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
    petsc_options_iname = '-mat_fd_type   -mat_mffd_type  -pc-type'
    petsc_options_value = 'ds  ds lu'
  [../]
[]

[Executioner]
  type = Transient
  scheme = 'implicit-euler'

  dt =  1e-1
  dtmin = 1.e-3

  nl_rel_tol = 1e-9
  nl_abs_tol = 1e-9
  nl_max_its = 50

  l_tol = 1e-3
  l_max_its = 60

  start_time = 0.0
```

```
  num_steps = 50

  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]

[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5.3.3 Description of the Input File

The following are the detailed explanations of the input parameters that did not exit in the previous examples.

The Global Parameters Block:

```
[GlobalParams]
  model_type = 32
  stabilization_type = 'NONE'
  scaling_factor_var   = '1e+4 1 1e-2'
  gravity = '0.0, 0.0, -9.8'
[]
```

`model_type` `model_type=32` indicates that the HEM two-phase flow model will be used for this problem.

`gravity` The gravity acceleration vector.

The Equation of State Block:

```
[EoS]
  [./two_phase_eos]
    type = TwoPhaseStiffenedGasEOS
  [../]
[]
```

`type` Two phase stiffened gas equation of state will be used for this problem.

The Components Block:

```
[Components]
  [./pipe1]
    type = Pipe
    position =    '0 0 0'
    orientation = '1 0 0'
    A = 7.854e-5
    Dh = 0.01
    length = 1.
    n_elems = 100
    f = 0.1
    Hw = 1e4
    Phf = 0.031416000000
    Tw = 528
    eos = two_phase_eos
    P_func = p_func
    V_func = v_func
    T_func = temp_func
    volume_fraction_vapor_func = void_func
  [../]

  [./inlet]
    type = TimeDependentVolume
    input = 'pipe1(in)'
    p_bc = 7.01e6
    T_bc = 517.
    volume_fraction_vapor_bc = 0.
```

```
    eos = two_phase_eos
  [../]

  [./outlet]
    type = TimeDependentVolume
    input = 'pipe1(out)'
    p_bc = 7.0e6
    T_bc = 517
    volume_fraction_vapor_bc = 0.5
    eos = two_phase_eos
  [../]
[]
```

`Hw`	The wall convective heat transfer coefficient.
`Phf`	Heat flux perimeter.
`Tw`	Initial pipe wall temperature.
`T_bc`	Temperature values at the pipe boundaries.
`volume_fraction_vapor_bc`	Vapor phase volume fraction at the pipe boundaries.

5.4 Example 4: Two Phase Flow Through a Pipe - 7-Equation Model

5.4.1 Problem Description

This example shows a two phase flow with the 7-equation model through one simple pipe with boundary conditions set by Inlet and Outlet components at the pipe inlet and outlet ends respectively. The initial conditions are set by functions. The initial conditions are set to be close to steady state solutions; therefore the simulation quickly converges to steady state solutions.

5.4.2 Input File

The input file for this pipe flow case with the 7-equation two-phase flow model is shown below:

```
[GlobalParams]
  model_type = 7
  global_init_T = 517.252
  global_init_P = 7.0e6
  global_init_V = 0
  global_init_volume_fraction_vapor = 0.95
  scaling_factor_var_2phase = '1
                               1e1 1e1 1e-3
                               1e1 1e1 1e-3'
  phase_interaction = true
  pressure_relaxation = true
  velocity_relaxation = true
  interface_transfer = true
  wall_mass_transfer = true
  stabilization_type = LAPIDUS
  specific_interfacial_area_max_value = 1700
[]

[Functions]

  # initial pressure distribution in the pipe
  [./p_func]
    axis = 0
    type = PiecewiseLinear
    x = '0        3.66'
```

```
    y = '7.005e6  7e6'
  [../]

  # initial velocity distribution in the pipe
  [./v_func]
    axis = 0
    type = PiecewiseLinear
    x = '0     3.66'
    y = '6     6.6'
  [../]

  # initial temperature distribution in the pipe
  [./temp_func]
    axis = 0
    type = PiecewiseLinear
    x = '0           3.66'
    y = '517.252     517.817'
  [../]

  # initial void fraction distribution in the pipe
  [./void_func]
    axis = 0
    type = PiecewiseLinear
    x = '0      3.66'
    y = '0.95   0.956'
  [../]

[]

[EoS]
  [./eos_liquid]
    type = StiffenedGasEquationOfStateLiquid
    gamma = 2.35
    q = -1167e3
    q_prime = 0
    p_inf = 1.e9
    cv = 1816
  [../]

  [./eos_vapor]
    type = StiffenedGasEquationOfStateVapor
```

```
    gamma = 1.43
    q = 2030e3
    q_prime = -23e3
    p_inf = 0
    cv = 1040
  [../]
[]

[Components]
  [./pipe]
    type = Pipe
    eos_liquid = eos_liquid
    eos_vapor = eos_vapor
    position = '0 0 0'
    orientation = '0 0 1'
    A = 1.907720E-04
    Dh = 1.698566E-02
    length = 3.66
    f = 1.698566E-02
    Hw_liquid = 1000.0
    Hw_vapor = 1000.0
    Phf = 0.0489623
    Tw = 550
    n_elems = 50

    P_liquid_func = p_func
    P_vapor_func = p_func
    V_liquid_func = v_func
    V_vapor_func = v_func
    T_liquid_func = temp_func
    T_vapor_func = temp_func
    volume_fraction_vapor_func = void_func
  [../]

  [./inlet]
    type = Inlet
    input = 'pipe(in)'
    p_liquid = 7.005e6
    p_vapor = 7.005e6
    T_liquid = 517.252
    T_vapor = 517.252
```

```
    volume_fraction_vapor = 0.95
    eos_liquid = eos_liquid
    eos_vapor = eos_vapor
  [../]

  [./outlet]
    type = Outlet
    input = 'pipe(out)'
    p_vapor = 7.0e6
    p_liquid = 7.0e6
    eos_liquid = eos_liquid
    eos_vapor = eos_vapor
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
  [../]
[]

[Executioner]
  type = Transient
  dt = 1e-1
  dtmin = 1.e-5
  nl_rel_tol = 1e-10
  nl_abs_tol = 1e-8
  nl_max_its = 30
  l_tol = 1e-3
  l_max_its = 30

  petsc_options_iname = '-mat_mffd_type  -ksp_gmres_restart'
  petsc_options_value = 'ds             300'

  start_time = 0.0
  num_steps = 100

  [./Quadrature]
    type = TRAP
```

```
    order = FIRST
  [../]
[]


[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5.4.3 Description of the Input File

The following provides the explanations of the input parameters that did not exist in the previous examples:

The Global Parameters Block:

```
[GlobalParams]
  model_type = 7
  global_init_T = 517.252
  global_init_P = 7.0e6
  global_init_V = 0
  global_init_volume_fraction_vapor = 0.95
  scaling_factor_var_2phase = '1
                               1e1 1e1 1e-3
                               1e1 1e1 1e-3'
  phase_interaction = true
  pressure_relaxation = true
  velocity_relaxation = true
  interface_transfer = true
```

```
  wall_mass_transfer = true
  stabilization_type = LAPIDUS
  specific_interfacial_area_max_value = 1700
[]
```

model_type = 7	Turn on the 7-equation two phase flow model.
scaling_factor_var_2phase	Scaling factors for each variable of the 7-equation two phase flow model.
phase_interaction	Option to turn on/off the interaction between the two phases.
pressure_relaxation	Option to turn on/off the pressure relaxation terms.
velocity_relaxation	Option to turn on/off the velocity relaxation terms.
interface_transfer	Option to turn on/off the interface mass and heat transfer.
wall_mass_transfer	Option to turn on/off wall boiling model to calculate the interfacial mass transfer near the wall.
specific_interfacial_area_max_value	The maximum value of the specific interfacial area.

The Equation of State Block:

```
[EoS]
  [./eos_liquid]
    type = StiffenedGasEquationOfStateLiquid
    gamma = 2.35
    q = -1167e3
    q_prime = 0
    p_inf = 1.e9
    cv = 1816
  [../]

  [./eos_vapor]
    type = StiffenedGasEquationOfStateVapor
    gamma = 1.43
```

```
    q = 2030e3
    q_prime = -23e3
    p_inf = 0
    cv = 1040
  [../]
[]
```

`[./eos_liquid]`	Subblock for the liquid phase equation of state.
`type`	Stiffened gas equation of state (SGEOS) for liquid.
`gamma`	The γ value in SGEOS calculations for liquid.
`q`	The `q` value in SGEOS calculations for liquid.
`q_prime`	The q' value in SGEOS calculations for liquid.
`p_inf`	The p_∞ value in SGEOS calculations for liquid.
`cv`	The specific heat value in SGEOS calculations for liquid.

The Components Block

```
[Components]
  [./pipe]
    type = Pipe
    eos_liquid = eos_liquid
    eos_vapor = eos_vapor
    position = '0 0 0'
    orientation = '1 0 0'
    A = 1.907720E-04
    Dh = 1.698566E-02
    length = 3.66
    f = 1.698566E-02
    Hw_liquid = 1000.0
    Hw_vapor = 1000.0
    Phf = 0.0489623
    Tw = 550
    n_elems = 50
```

```
    P_liquid_func = p_func
    P_vapor_func = p_func
    V_liquid_func = v_func
    V_vapor_func = v_func
    T_liquid_func = temp_func
    T_vapor_func = temp_func
    volume_fraction_vapor_func = void_func
  [../]

  [./inlet]
    type = Inlet
    input = 'pipe(in)'
    p_liquid = 7.005e6
    p_vapor = 7.005e6
    T_liquid = 517.252
    T_vapor = 517.252
    volume_fraction_vapor = 0.95
    eos_liquid = eos_liquid
    eos_vapor = eos_vapor
  [../]

  [./outlet]
    type = Outlet
    input = 'pipe(out)'
    p_vapor = 7.0e6
    p_liquid = 7.0e6
    eos_liquid = eos_liquid
    eos_vapor = eos_vapor
  [../]
[]
```

`eos_liquid`	Defines the name of the equation of state to use for the liquid phase.
`eos_vapor`	Defines the name of the equation of state to use for the vapor phase.
`Hw_liquid`	Convective heat transfer coefficient for the liquid phase.
`Hw_vapor`	Convective heat transfer coefficient for the vapor phase.

`P_liquid_func`	Function which defines liquid phase pressure as a function of `x`.
`P_vapor_func`	Function which defines vapor phase pressure as a function of `x`.
`T_liquid_func`	Function which defines liquid phase temperature as a function of `x`.
`T_vapor_func`	Function which defines vapor phase temperature as a function of `x`.
`volume_fraction_vapor_func`	Function which defines vapor phase volume fraction as a function of `x`.
`p_liquid`	Prescribed pressure value at the pipe boundary for the liquid phase.
`p_vapor`	Prescribed pressure value at the pipe boundary for the vapor phase.
`T_liquid`	Prescribed temperature value at the pipe boundary for the liquid phase.
`T_vapor`	Prescribed temperatur value at the pipe boundary for the vapor phase.
`volume_fraction_vapor`	Prescribed vapor phase volume fraction.

5.5 Example 5: A Core Channel Problem

5.5.1 Problem Description

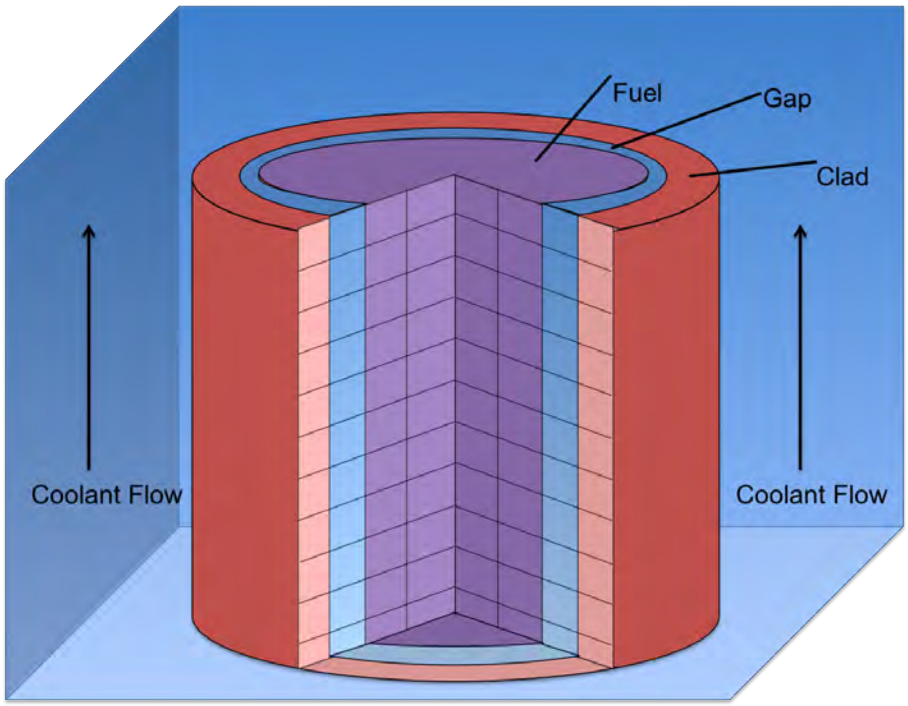

Figure 4. Diagram of a core channel problem

This example simulates single phase fluids flow and heat transfer in a core channel, as schematically shown in Fig. 4. This problem similates fluids flow in one subchannel with a single fuel rod as the heat source. The boundary conditions are applied to the ends of the core channel. The reactor power = 77337.69407 W is used for the core channel. Other parameters used are the following:

Fluids Flow:
Cross section flow area = 8.7878e-5 m^2
Hydraulic Diameter = 0.01179 m
Length = 4 m
Friction coefficient = 0.01
Convective heat transfer coefficient = 5.33e4 $W/m^2 * K$

Heat flux perimeter = 0.029832559676 m
Number of elements = 20

Heat Structure:
Dimension of mesh used for heat conduction = 1
Geometry type of fuel = cylinder
Heat structure names = 'fuel gap clad'
Number of heat structures = 3
Width of each heat structure = '0.004096 0.0001 0.000552' m
Number of elements of each heat structure = '10 1 2'
Name of materials used in the heat structure = 'fuel-mat gap-mat clad-mat'
Fraction of reactor power goes into heat structure = '1.0 0.0 0.0'
Initial Solid Temperature = 559.15 K

The boundary conditions are the following:

Inlet:
Pressure = 155.483e5 Pa
Temperature = 559.15 K

Outlet:
Pressure = 155.e5Pa
Temperature = 559.15 K

The solid material properties (k = Thermal conductivity, Cp = heat capacity, rho = Density) are:

Fuel:
k = 2.5
Cp = 300
rho = 1.032e4

Gap:
k = 0.6
Cp = 1.
rho = 1.

Clad:
k = 21.5
Cp = 350.

rho = 6.55e3

5.5.2 Input File

The follwoing shows the input file for the Core Channel example. Please note that the mesh used for this problem does not correspond to what is shown in Fig. 4.

```
[GlobalParams]
  global_init_P = 155.e5
  global_init_V = 0.
  global_init_T = 559.15
  model_type = 3
  stabilization_type = 'SUPG'
  scaling_factor_var = '1e4 1e1 1e-2'
  temperature_sf = '1e-2'
[]

[EoS]
  [./eos]
    type = NonIsothermalEquationOfState
    p_0 = 155.e5
    rho_0 = 686.0
    a2 = 1.e7
    beta = .46e-3
    cv = 5.5e3
    e_0 = 3075325
    T_0 = 559.15
  [../]
[]

[Materials]
  [./fuel-mat]
    type = SolidMaterialProperties
    k = 2.5
    Cp = 300.
    rho = 1.032e4
  [../]
  [./gap-mat]
```

```
    type = SolidMaterialProperties
    k = 0.6
    Cp = 1.
    rho = 1.
  [../]
  [./clad-mat]
    type = SolidMaterialProperties
    k = 21.5
    Cp = 350.
    rho = 6.55e3
  [../]
[]

[Components]
  [./reactor]
    type = Reactor
    initial_power = 77337.69407
  [../]

  [./CCH1]
    type = CoreChannel
    eos = eos
    position = '0 0 0'
    orientation = '0 0 1'
    A = 8.7878e-5
    Dh = 0.01179
    length = 4
    n_elems = 20
    f = 0.01
    Hw = 5.33e4
    Phf = 0.029832559676
    Ts_init = 559.15
    dim_hs = 1
    fuel_type = cylinder
    name_of_hs = 'fuel gap clad'
    n_heatstruct = 3
    width_of_hs = '0.004096 0.0001 0.000552'
    elem_number_of_hs = '10 1 2'
    material_hs = 'fuel-mat gap-mat clad-mat'
    power_fraction = '1.0 0.0 0.0'
  [../]
```

```
  [./inlet]
    type = TimeDependentVolume
    input = 'CCH1(in)'
    p_bc = 155.483e5
    T_bc = 559.15
    eos = eos
  [../]

  [./outlet]
    type = TimeDependentVolume
    input = 'CCH1(out)'
    p_bc = '155.e5'
    T_bc = 559.15
    eos = eos
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
  [../]
[]

[Executioner]
  type = Transient
  scheme = 'implicit-euler'
  dt = 1.e-2 #1.e-5
  dtmin = 1.e-5

  nl_rel_tol = 1e-9
  nl_abs_tol = 1e-8
  nl_max_its = 20

  l_tol = 1e-3
  l_max_its = 30

  start_time = 0.0
  num_steps = 50
```

```
  [./Quadrature]
     type = TRAP
     order = FIRST
  [../]
[]

[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5.5.3 Description of the Input File

The Global Parameter Block:

```
[GlobalParams]
  global_init_P = 155.e5
  global_init_V = 0.
  global_init_T = 559.15
  model_type = 3
  stabilization_type = 'SUPG'
  scaling_factor_var = '1e4 1e1 1e-2'
  temperature_sf = '1e-2'
[]
```

`temperature_sf` Scaling factor for the temperature variable in solid materials.

The Materials Block:

```
[Materials]
  [./fuel-mat]
    type = SolidMaterialProperties
    k = 2.5
    Cp = 300.
    rho = 1.032e4
  [../]
  [./gap-mat]
    type = SolidMaterialProperties
    k = 0.6
    Cp = 1.
    rho = 1.
  [../]
  [./clad-mat]
    type = SolidMaterialProperties
    k = 21.5
    Cp = 350.
    rho = 6.55e3
  [../]
[]
```

`k` Thermal conductivity.

`Cp` Specific heat.

`rho` Density of solid materials.

```
[Components]
  [./reactor]
    type = Reactor
    initial_power = 77337.69407
  [../]

  [./CCH1]
    type = CoreChannel
    eos = eos
    position = '0 0 0'
    orientation = '0 0 1'
    A = 8.7878e-5
```

```
    Dh = 0.01179
    length = 4
    n_elems = 20
    f = 0.01
    Hw = 5.33e4
    Phf = 0.029832559676
    Ts_init = 559.15
    dim_hs = 1
    fuel_type = cylinder
    name_of_hs = 'fuel gap clad'
    n_heatstruct = 3
    width_of_hs = '0.004096 0.0001 0.000552'
    elem_number_of_hs = '10 1 2'
    material_hs = 'fuel-mat gap-mat clad-mat'
    power_fraction = '1.0 0.0 0.0'
  [../]

  [./inlet]
    type = TimeDependentVolume
    input = 'CCH1(in)'
    p_bc = 155.483e5
    T_bc = 559.15
    eos = eos
  [../]

  [./outlet]
    type = TimeDependentVolume
    input = 'CCH1(out)'
    p_bc = '155.e5'
    T_bc = 559.15
    eos = eos
  [../]
[]
```

`[./reactor]`	Subblock for the reactor component.
`initial_power`	Prescribed initial reactor power.
`[./CCH1]`	Subblock for the core channel component.

`Hw`	Wall heat transfer coefficient.
`Ts_init`	Prescribed initial temperature of the solid materials.
`dim_hs`	The dimension of the mesh used for the heat conduction calculations in the heat structure. The options are `dim_hs=1` for 1D heat conduction calculations or `dim_hs=2` for 2D heat conduction calculations.
`fuel_type`	Geometry type of the fuel. The available options are `fuel_type = cylinder` or `fuel_type = plate`.
`name_of_hs`	Prescribed heat structure names.
`n_heatstruct`	Prescribed number of heat structures.
`width_of_hs`	Width of each heat structure.
`elem_number_of_hs`	Number of elements of each heat structure.
`material_hs`	Name of the materials (defined in the Materials block) used in the heat structure.
`power_fraction`	The fraction of reactor power that goes into each heat structure.

5.6 Example 6: A Two Pipes Flow Problem

5.6.1 Problem Description

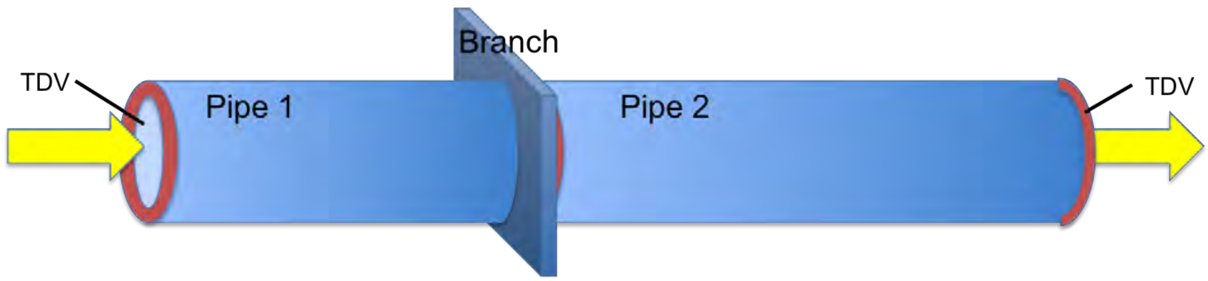

Figure 5. Diagram of a two pipes flow problem

This example demonstrates flow through two pipes with different parameters connected by a branch. There is no heat exchange through the pipe walls for this case. The following are the input parameters for each component:

Pipe 1:
Hydraulic Diameter = .01 m
Area = 1.0e-4 m^2
Length = 1 m
Friction coefficient = .01
Convective heat transfer coefficient = 0

Pipe 2:
Hydraulic Diameter = .02 m
Area = 4e-4 m^2
Length = 1.5 m
Friction = .02
Convective Heat Transfer Coefficient = 0

Branch:
Form loss coefficients = '5 5'
Area = 1.5e-4 m^2

Initial pressure = 1e5 Pa

Inlet:
Pressure = 1.01e5 Pa
Temperature = 300 K

Outlet:
Pressure = 1e5 Pa

The following figure shows the results of the pressure versus pipes length for this case.

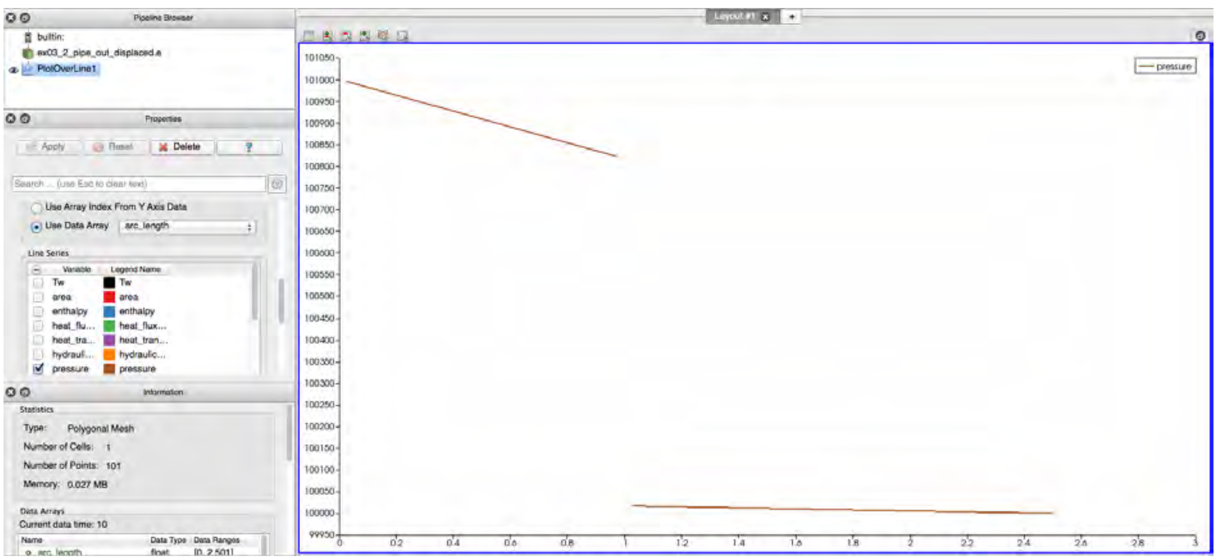

Figure 6. Pressure vs. length for the two pipes flow problem

The results of the pressure difference can be checked by doing hand calculations by summing ΔP using:

$$\Delta P = \frac{fL\rho u^2}{2D_h} \tag{2}$$

and:

$$\Delta P = \frac{k\rho u^2}{2} \tag{3}$$

5.6.2 Input File

The input file for this example problem is listed as follows:

```
[GlobalParams]
  model_type = 3
  gravity = '0 0 0'
  stabilization_type = 'SUPG'
  scaling_factor_var = '1.e3 1.e-1 1.e-3'
  global_init_T = 300
  global_init_P = 1.0e5
  global_init_V = 1
[]

[EoS]
   [./eos]
     type = NonIsothermalEquationOfState
     p_0 = 1.e5
     rho_0 = 1.e3
     a2 = 1.e7
     beta = .46e-3
     cv = 4.18e3
     e_0 = 1.254e6
     T_0 = 300
   [../]
[]

[Components]
  [./pipe1]
    type = Pipe
    eos = eos
    position = '0 0 0'
    orientation = '1 0 0'
    length = 1
    n_elems = 25
    A = 1.0e-4
    Dh = 0.01
    f = .01
    Hw = 0.
  [../]
```

```
  [./Branch1]
    type = Branch
    inputs = 'pipe1(out)'
    outputs = 'pipe2(in) '
    K = '5  5'
    Area = 1.5e-4
    Initial_pressure = 1e5
    eos = eos
  [../]

  [./pipe2]
    type = Pipe
    eos = eos
    position = '1.001 0 0'
    orientation = '1 0 0'
    length = 1.5
    n_elems = 25
    A = 4e-4
    Dh = 0.02
    f = .02
    Hw = 0.
  [../]

  [./inlet]
    type = TimeDependentVolume
    input = 'pipe1(in)'
    p_bc = 1.01e5
    T_bc = 300
    eos = eos
  [../]

  [./outlet]
    type = TimeDependentVolume
    input = 'pipe2(out)'
    p_bc = 1e5
    eos = eos
  [../]
[]

[Preconditioning]
```

```
 [./SMP_PJFNK]
   type = SMP
   full = true
   solve_type = 'PJFNK'
 [../]
[]

[Executioner]
  type = Transient
  scheme = 'implicit-euler'

  dt = 1e-2
  dtmin = 1e-5

  nl_rel_tol = 1e-8
  nl_abs_tol = 1e-6
  nl_max_its = 30

  l_tol = 1e-4
  l_max_its = 30

  petsc_options_iname = '-mat_fd_coloring_err -mat_fd_type -mat_mffd_type -pc_type'
  petsc_options_value = '1.e-10                ds              ds              lu'

  start_time = 0.0
  end_time = 10.0
  num_steps = 5000

  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]

[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
```

```
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5.6.3 Description of the Input File

The only input parameters that need explanations are those in the `Branch` subblock.

```
  [./Branch1]
    type = Branch
    inputs = 'pipe1(out)'
    outputs = 'pipe2(in) '
    K = '5  5'
    Area = 1.5e-4
    Initial_pressure = 1e5
    eos = eos
  [../]
```

`[./Branch]`	Subblock for the `Branch` component.
`type`	The component type is `Branch`.
`inputs`	Inputs of this branch connects to the outlet end of `pipe1`.
`outputs`	Outputs of this branch connects to the inlet end of `pipe2`.
`K`	Form loss coefficients at the branch inputs and outputs.
`Area`	The reference area of this branch.
`Initial_pressure`	Prescribed initial pressure for the branch.
`eos`	The name of the equation of state to be used for this branch.

5.7 Example 7: A Volume Branch Case - Three Pipes In and Two Pipes Out

5.7.1 Problem Description

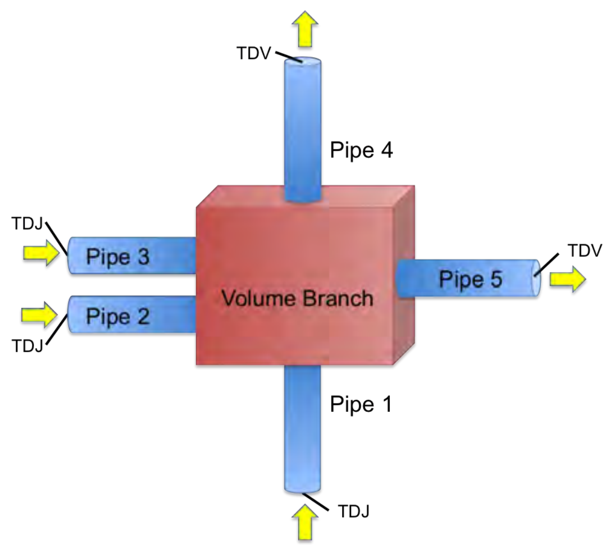

Figure 7. Diagram of a volume branch case with three pipes flowing in and two pipes flowing out

This example shows a volume branch case which connects three pipes with water flowing into them and two pipes with water flowing out of them. The pipe sizes are all the same, but the boundary conditions vary. The volume branch model includes the volume effects of a junction. The input parameters used in this example are listed in the following:

Pipes:
Area = 3.14e-4 m^2
Diameter = 0.02 m
Length = 1 m
Number of elements in pipe = 10
Friction coefficient = 0.01

Branch1:
Type = VolumeBranch
inputs = ’pipe1(out) pipe2(out) pipe3(out)’
outputs = ’pipe4(in) pipe5(in)’
Form loss coefficients = ’0.01 0.01 0.01 0.01 100’
Area = $3.14e-2m^2$
volume = $3.14e-2m^3$
Initial Temperature = 628.15 K

Inlet 1:
type = TimeDependentJunction
Velocity = 1.0 m/s
Temperature = 628.15 K

Inlet 2:
type = TimeDependentJunction
Velocity = 1.0 m/s
Temperature = 628.15 K

Inlet 3:
type = TimeDependentJunction
Velocity = 10.0 m/s
Temperature = 528.15 K

Outlet 1:
type = TimeDependentVolume
Pressure = 1.0e5 Pa
Temperature = 628.15 K

Outlet 2:
type = TimeDependentVolume
Pressure = 1.5e5 Pa
Temperature = 628.15 K

5.7.2 Input File

```
[GlobalParams]
  global_init_P = 1.5e5
  global_init_V = 1
  global_init_T = 628.15
  model_type = 3
  stabilization_type = 'SUPG'
  scaling_factor_var = '1e4 1 1e-3'
[]

[EoS]
  active = 'eos'
  [./eos]
    type = NonIsothermalEquationOfState
    p_0 = 1e5
    rho_0 = 865.51
    a2 = 5.7837e6
    beta = 2.7524e-4
    cv =  1272.0
    e_0 = 7.9898e5
    T_0 = 628.15
  [../]
[]

[Components]

  [./pipe1]
    type = Pipe
    eos = eos
    position = '0 0 0'
    orientation = '0 0 1'
    A = 3.14e-4
    Dh = 0.02
    length = 1
    n_elems = 10
    f = 0.01
    Hw = 0
  [../]
```

```
[./pipe2]
  type = Pipe
  eos = eos
  position = '-1.5 0 1.25'
  orientation = '1 0 0'
  A = 3.14e-4
  Dh = 0.02
  length = 1
  n_elems = 10
  f = 0.01
  Hw = 0
[../]

[./pipe3]
  type = Pipe
  eos = eos
  position = '-1.5 0 1.75'
  orientation = '1 0 0'
  A = 3.14e-4
  Dh = 0.02
  length = 1
  n_elems = 10
  f = 0.01
  Hw = 0
[../]

[./pipe4]
  type = Pipe
  eos = eos
  position = '0 0 2'
  orientation = '0 0 1'
  A = 3.14e-4
  Dh = 0.02
  length = 1
  n_elems = 10
  f = 0.01
  Hw = 0
[../]

[./pipe5]
```

```
  type = Pipe
  eos = eos
  # geometry
  position = '0.5 0 1.5'
  orientation = '1 0 0'
  A = 3.14e-4
  Dh = 0.02
  length = 1
  n_elems = 10
  f = 0.01
  Hw = 0
[../]

[./branch1]
  type = VolumeBranch
  eos = eos
  center = '0 0 1.5'
  inputs = 'pipe1(out) pipe2(out) pipe3(out)'
  outputs = 'pipe4(in) pipe5(in)'
  K = '0.01 0.01 0.01 0.01 100'
  Area = 3.14e-2
  volume = 3.14e-2
  initial_T = 628.15
[../]

[./inlet1]
  type = TimeDependentJunction
  input = 'pipe1(in)'
  eos = eos
  v_bc = 1.0
  T_bc = 628.15
[../]

[./inlet2]
  type = TimeDependentJunction
  input = 'pipe2(in)'
  eos = eos
  v_bc = 1.0
  T_bc = 628.15
[../]
```

```
  [./inlet3]
    type = TimeDependentJunction
    input = 'pipe3(in)'
    eos = eos
    v_bc = 10.0
    T_bc = 528.15
  [../]

  [./outlet1]
    type = TimeDependentVolume
    input = 'pipe4(out)'
    eos = eos
    p_bc = '1.0e5'
    T_bc = 628.15
  [../]

  [./outlet2]
    type = TimeDependentVolume
    input = 'pipe5(out)'
    eos = eos
    p_bc = '1.5e5'
    T_bc = 628.15
  [../]
[]

[Preconditioning]

  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
  [../]

[]


[Executioner]
  type = Transient
  scheme = 'implicit-euler'

  dt = 0.2
```

```
  dtmin = 1e-10

  # setting time step range
  [./TimeStepper]
    type = FunctionDT
    time_t = '0    0.05  0.051    0.3   0.31    2   2.1   1e5'
    time_dt ='1.e-2  1.e-2 5e-2    5e-2  2e-1  0.2  0.5     0.5 '
  [../]

  nl_rel_tol = 1e-6
  nl_abs_tol = 1e-7
  nl_max_its = 30

  l_tol = 1e-6
  l_max_its = 100

  petsc_options_iname = '-pc_type  -ksp_gmres_restart'
  petsc_options_value = 'lu         300'

  start_time = 0.0
  num_steps = 500

  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]

[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
```

[]

5.7.3 Description of the Input File

```
[./branch1]
  type = VolumeBranch
  eos = eos
  center = '0 0 1.5'
  inputs = 'pipe1(out) pipe2(out) pipe3(out)'
  outputs = 'pipe4(in) pipe5(in)'
  K = '0.01 0.01 0.01 0.01 100'
  Area = 3.14e-2
  volume = 3.14e-2
  initial_T = 628.15
[../]
```

`center` The `(x,y,z)` coordinate of the center of the volume branch.

`volume` The volume of VolumeBranch component.

```
[./TimeStepper]
  type = FunctionDT
  time_t = '0     0.05  0.051     0.3    0.31     2    2.1    1e5'
  time_dt ='1.e-2  1.e-2 5e-2     5e-2   2e-1   0.2   0.5      0.5 '
[../]
```

`[./TimeStepper]` Subblock to define time step size as a function of simulation time.

`type` FunctionDT type to be used.

`time_t` The value of time `t`.

`time_dt` The value of time step.

5.8 Example 8: A Simple Pipe Loop with Pump

5.8.1 Problem Description

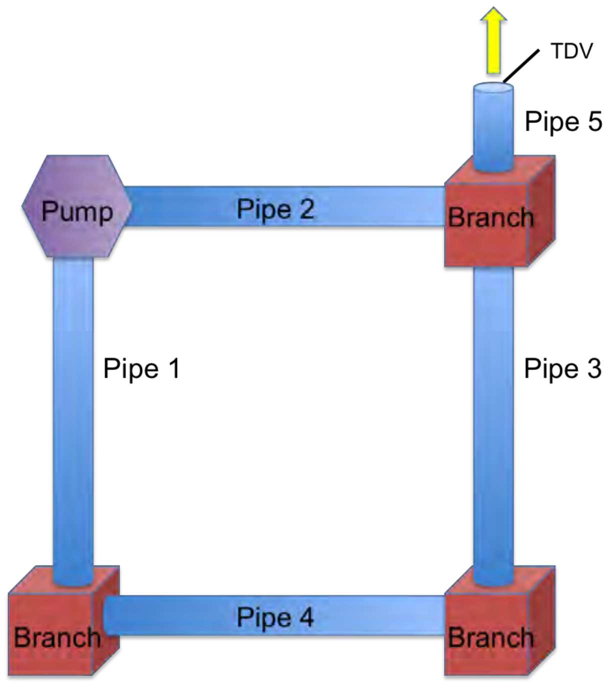

Figure 8. Diagram of a simple loop of pipes connected by branches and a pump

This example shows a loop made up of 5 pipes, 3 branches and a pump. The pump acts as a junction, connecting pipes 1 and 2, and pipe 5 is used as an outlet to help control the pressure. The input parameters used for the Pump model are the following:

Pump head = 1.0
Form loss coefficient (in the reverse direction) = '10. 10.'
Area = $0.785398163e-3m^2$
Initial pressure = 1.e5 Pa

5.8.2 Input File

```
[GlobalParams]
  model_type = 3
  stabilization_type = 'SUPG'
  scaling_factor_var = '1e4 1e1 1e-2'
[]

[EoS]
  [./eos]
    type = NonIsothermalEquationOfState
    p_0 = 1.e5
    rho_0 = 1.e3
    a2 = 1.e7
    beta = -.46e-3
    cv = 4.18e3
    e_0 = 1.254e6
    T_0 = 300
  [../]
[]

[Components]

  [./pipe1]
    type = Pipe
    eos = eos
    position = '0 0 0'
    orientation = '1 0 0'
    A = 0.785398163e-4
    Dh = 0.01
    length = 1
    n_elems = 10
    f = 0.01
    Hw = 0.0
  [../]

  [./pipe2]
    type = Pipe
    eos = eos
    position = '1 0 0'
```

```
  orientation = '0 1 0'
  A = 0.785398163e-4
  Dh = 0.01
  length = 1
  n_elems = 10
  f = 0.01
  Hw = 0.0
[../]

[./pipe3]
  type = Pipe
  eos = eos
  position = '1 1 0'
  orientation = '-1 0 0'
  A = 0.785398163e-4
  Dh = 0.01
  length = 1
  n_elems = 10
  f = 0.01
  Hw = 0.0
[../]

[./pipe4]
  type = Pipe
  eos = eos
  # geometry
  position = '0 1 0'
  orientation = '0 -1 0'
  A = 0.785398163e-4
  Dh = 0.01
  length = 1
  n_elems = 10
  f = 0.01
  Hw = 0.0
[../]

[./pipe5]
  type = Pipe
  eos = eos
  position = '1 1 0'
  orientation = '0 1 0'
```

```
  A = 0.785398163e-4
  Dh = 0.01
  length = .25
  n_elems = 10
  f = 0.01
  Hw = 0.0
[../]

[./pump]
  type = Pump
  eos = eos
  inputs = 'pipe1(out)'
  outputs = 'pipe2(in)'
  Head = 1.0
  K_reverse = '10. 10.'
  Area = 0.785398163e-3
  Initial_pressure = 1.e5
[../]

[./branch2]
  type = Branch
  eos = eos
  inputs = 'pipe2(out)'
  outputs = 'pipe3(in) pipe5(in)'
  K = '3. 3. 3.'
  Area = 0.785398163e-3
  Initial_pressure = 1.e5
[../]

[./branch3]
  type = Branch
  eos = eos
  inputs = 'pipe3(out)'
  outputs = 'pipe4(in)'
  K = '3. 3.'
  Area = 0.785398163e-3
  Initial_pressure = 1.e5
[../]

[./branch4]
  type = Branch
```

```
    eos = eos
    inputs = 'pipe4(out)'
    outputs = 'pipe1(in)'
    K = '3. 3.'
    Area = 0.785398163e-3
    Initial_pressure = 1.e5
  [../]

  [./TDV]
    type = TimeDependentVolume
    input = 'pipe5(out)'
    p_bc = '1.e5'
    T_bc = 300.0
    eos = eos
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
  [../]
[]

[Executioner]
  type = Transient
  scheme = 'implicit-euler'
  dt = 1.e-4
  dtmin = 1e-7

  nl_rel_tol = 1e-6
  nl_abs_tol = 1e-8
  nl_max_its = 10

  l_tol = 1e-8
  l_max_its = 60

  petsc_options_iname = '-ksp_gmres_restart'
  petsc_options_value = '30'
```

```
  start_time = 0.0
  num_steps = 50

  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]

[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5.8.3 Description of the Input File

```
  [./pump]
    type = Pump
    eos = eos
    inputs = 'pipe1(out)'
    outputs = 'pipe2(in)'
    Head = 1.0
    K_reverse = '10. 10.'
    Area = 0.785398163e-3
    Initial_pressure = 1.e5
  [../]
```

`[./pump]` Subblock for the pump component.

`type`	Component type is a pump.
`eos`	Name of equation of state to be used by this pump component.
`inputs`	The inlet of the pump connects to outlet of `pipe1`.
`outputs`	The outlet of the pump connects to the inlet of `pipe2`.
`Head`	Prescribed pump head.
`K_reverse`	Form loss coefficient in reverse flow direction.
`Area`	Reference area for the pump.
`Initial_pressure`	Prescribed initial pressure for this pump component.

5.9 Example 9: A Heat Exchanger Problem

5.9.1 Problem Description

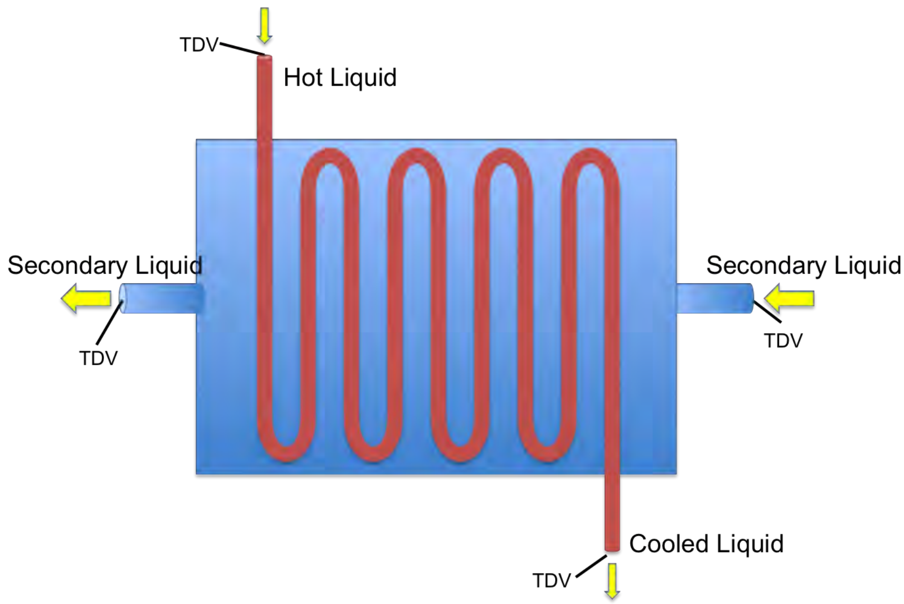

Figure 9. Diagram of a heat exchanger problem

This examples illustrates how to run a case with the heat exchanger component. The heat exchanger has two inlets and two outlets, primary and secondary respectively. The primary side inlet lets in hot water, while the secondary side inlet lets in liquid at a cooler temperature. This helps cool the primary side liquid, so the primary side outlet temperature is somewhat lower while the secondary side outlet temperature is somewhat higher. This example problem is for subcooled water only. The input parameters used for this case are listed in the following:

Materials:
Thermal Conductivity = 100.0
Density = 100.0
Specific Heat = 100.0

Heat Exchanger:

Area of primary pipe = $0.785398163e-4m^2$
Area of secondary pipe = $0.785398163e-4m^2$
Diameter of primary pipe = 0.01 m
Diameter of secondary pipe = 0.01 m
Length = 1 m
Primary convective heat transfer coefficient = 1.e4
Secondary convective heat transfer coefficient = 1.e4
Primary heat flux perimeter= 0.031415926520
Secondary heat flux perimeter = 0.031415926520
Friction coefficient = 0.01
Initial wall temperature = 300.0 K
Dimension of mesh used for wall = 1
Thickness of the wall between primary and secondary loop = 0.001 m

Primary inlet TDV:
Pressure = 1.05e5 Pa
Temperature = 400.0 K

Primary outlet TDV:
Pressure = 1.e5 Pa
Temperature = 300.0 K

Secondary inlet TDV:
Pressure = 1.05e5 Pa
Temperature = 300.0 K

Secondary outlet TDV:
Pressure = 1.e5 Pa
Temperature = 300.0 K

5.9.2 Input File

```
[GlobalParams]
  global_init_P = 1.e5
```

```
  global_init_V = 0.
  global_init_T = 300.
  model_type = 3
  stabilization_type = 'SUPG'
  scaling_factor_var = '1e4 1e1 1e-2'
  temperature_sf = '1e-2'
[]

[EoS]
  [./eos]
    type = NonIsothermalEquationOfState
    p_0 = 1.e5
    rho_0 = 1.e3
    a2 = 1.e7
    beta = .46e-3
    cv = 4.18e3
    e_0 = 1.254e6
    T_0 = 300
  [../]
[]

[Materials]
  [./wall-mat]
    type = SolidMaterialProperties
    k = 100.0
    rho = 100.0
    Cp = 100.0
  [../]
[]

[Components]

  [./HX]
    type = HeatExchanger
    eos = eos
    eos_secondary = eos
    position = '0 0 0'
    orientation = '1 0 0'
    A = 0.785398163e-4
    A_secondary = 0.785398163e-4
    Dh = 0.01
```

```
  Dh_secondary = 0.01
  length = 1
  n_elems = 10
  Hw = 1.e4
  Hw_secondary = 1.e4
  Phf = 0.031415926520
  Phf_secondary = 0.031415926520
  f = 0.01
  f_secondary = 0.01
  Twall_init = 300.0
  dim_wall = 1
  wall_thickness = 0.001
  material_wall = wall-mat
  n_wall_elems = 2
[../]

[./primary_inlet_TDV]
  type = TimeDependentVolume
  input = 'HX(primary_in)'
  p_bc = 1.05e5
  T_bc = 400.0
  eos = eos
[../]

[./primary_outlet_TDV]
  type = TimeDependentVolume
  input = 'HX(primary_out)'
  p_bc = '1.e5'
  T_bc = 300.0
  eos = eos
[../]

[./secondary_inlet_TDV]
  type = TimeDependentVolume
  input = 'HX(secondary_in)'
  p_bc = 1.05e5
  T_bc = 300.0
  eos = eos
[../]

[./secondary_outlet_TDV]
```

```
    type = TimeDependentVolume
    input = 'HX(secondary_out)'
    p_bc = '1.e5'
    T_bc = 300.0
    eos = eos
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
    line_search = basic
  [../]
[]

[Executioner]
  type = Transient
  scheme = 'implicit-euler'
  dt = 1.e-3
  dtmin = 1e-7

  nl_rel_tol = 1e-6
  nl_abs_tol = 1e-8
  nl_max_its = 20

  l_tol = 1e-6
  l_max_its = 300

  petsc_options_iname = '-ksp_gmres_restart -pc_type'
  petsc_options_value = '300 lu'

  start_time = 0.0
  num_steps = 1000

  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]
```

```
[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5.9.3 Description of the Input File

The input parameters for the heat exchanger component needs explanantions.

```
  [./HX]
    type = HeatExchanger
    eos = eos
    eos_secondary = eos
    position = '0 0 0'
    orientation = '1 0 0'
    A = 0.785398163e-4
    A_secondary = 0.785398163e-4
    Dh = 0.01
    Dh_secondary = 0.01
    length = 1
    n_elems = 10
    Hw = 1.e4
    Hw_secondary = 1.e4
    Phf = 0.031415926520
    Phf_secondary = 0.031415926520
    f = 0.01
    f_secondary = 0.01
    Twall_init = 300.0
    dim_wall = 1
```

```
    wall_thickness = 0.001
    material_wall = wall-mat
    n_wall_elems = 2
  [../]
```

`[./HX]`	Subblock for the heat exchanger component.
`eos`	Name of the equation of state to be used by the fluids in the primary side of the heat exchanger.
`eos_secondary`	Name of the equation of state to be used by the fluids in the secondary side of the heat exchanger.
`position`	The `x,y,z` coordinate of the start of heat exchanger.
`orientation`	The orientation vector of heat exchanger.
`A`	The flow area of the primary side of the heat exchanger.
`A_secondary`	The flow area of the secondary side of the heat exchanger.
`Dh`	Hydraulic dimater of the primary side of the heat exchanger.
`Dh_secondary`	Hydraulic diameter of the secondary side of the heat exchanger.
`length`	Length of heat exchanger.
`n_elems`	The number of elements in the heat exchanger.
`Hw`	The convective heat transfer coefficient on the primary side of the heat exchanger.
`hw_secondary`	The convective heat transfer coefficient on the secondary side of the heat exchanger.
`Phf`	Heat flux perimeter of the primary side of the heat exchanger.
`Phf_secondary`	Heat fluc perimeter of the secondary side of the heat exchanger.
`f`	Wall friction coefficient on the primary side of the heat exhanger.

`f_secondary` Wall friction ceofficient on the secondary side of the heat exchanger.

`Twall_init` Prescribed initial temperature of the solid materials between the fluids.

`dim_wall` The dimension of the mesh used in the heat conduction calculations of the wall in the heat exchanger.

`wall_thickness` The wall thickness.

`material_wall` Name of the material used in the wall.

`n_wall_elems` Number of elements in the wall.

5.10 Example 10: A Loop With Core Channel and Heat Exchanger

5.10.1 Problem Description

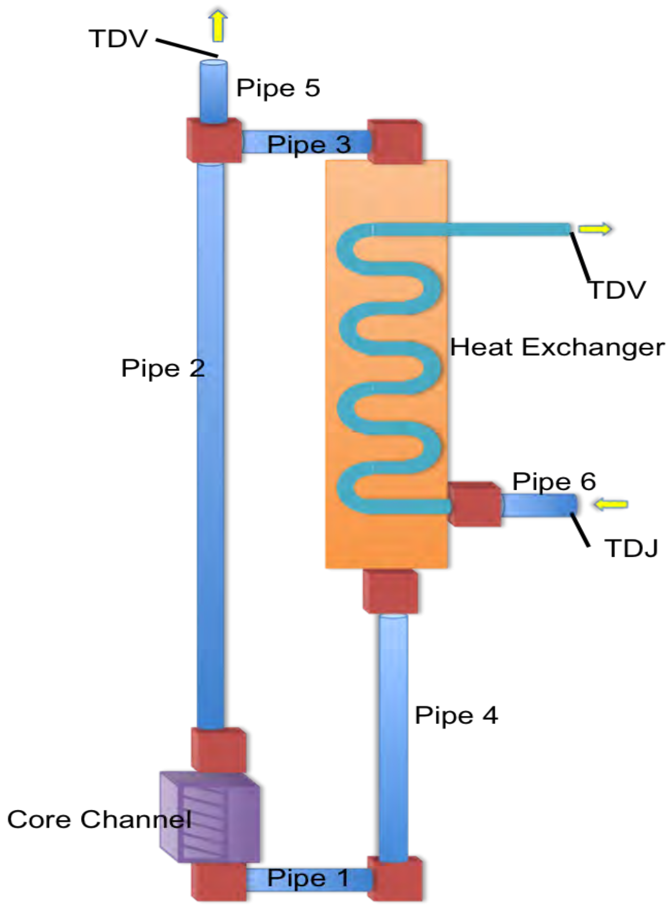

Figure 10. Diagram of a loop with core channel and heat exchanger

This example is a simple loop whcich consists of 6 pipes, a core channel, and a heat exchanger. Each component differs in length, but has the same area and diameter. The core channel is used to heat up the liquid, while the heat exchanger is used to cool it down. Pipe 5 is used to control the system pressure. The input parameters are listed in the following:

Materials:
Fuel
Thermal conductivity = 29.3 W/m K
Specific heat = 191.67 J/kg K
Density = $1.4583e4kg/m^3$

Clad
Thermal conductivity = 26.3 W/m K
Specific heat = 638 J/kg K
Density = $7.646e3kg/m^3$

Wall
Thermal conductivity = 26.3 W/m K
Specific heat = 638 J/kg K
Density = $7.646e3kg/m^3$

Reactor's initial power = 5.129e7

Pipe1:
Area = $0.44934m^2$
Diameter = 2.972e-3 m
Length = 1 m
Friction Coefficient = 0.001
Hw = 0.0

Pipe 2 length = 5.18 m
Pipe 3 length = 1 m
Pipe 4 length = 2.27 m
Pipe 5 length = 0.02 m
Pipe 6 length = 0.3 m

Core Channel:
Length = 0.8 m
Friction coefficient = 0.022
Convective heat transfer coefficient = 1.6129e5
Heat flux perimeter = 497.778

Initial solid temperature = 628.15 K
Width of each heat structure = ’0.00348 0.00052’

Heat Exchanger:
Area of primary pipe = 0.44934m^2
Area of secondary pipe = 0.44934m^2
Diameter of primary pipe = 0.0186 m
Diameter of secondary pipe = 0.014 m
Length = 3.71 m
Primary convective heat transfer coefficient = 1.6129e5
Secondary convective heat transfer coefficient = 1.6129e5
Primary heat flux perimeter= 327.568
Secondary heat flux perimeter = 327.568
Friction coefficient = 0.022
Initial wall temperature = 628.15 K
Dimension of mesh used for wall = 1
Thickness of the wall between primary and secondary loop = 0.0044 m

Branches:
Form loss coefficients = ’0.5 0.5’
Area = 0.44934m^2
Initial Pressure = 1e5 Pa (Branch 1 = 2e5 Pa)

Inlet:
Type = TimeDependentJunction
Input = ’pipe6(in)’
Velocity = 0.5 m/s
Temperature = 628.15 K

Outlet 1:
Type = TimeDependentVolume
Input = ’IHX(secondary_out)’
Pressure = 1.0e5 Pa
Temperature = 761.15 K

Outlet 2:
Type = TimeDependentVolume
Input = ’pipe5(out)’ Pressure = 1e5 Pa

Temperature = 783.15 K

5.10.2 Input File

```
[GlobalParams]
  model_type = 3
  global_init_P = 1.0e5
  global_init_V = 0.5
  global_init_T = 628.15
  stabilization_type = 'NONE'
[]

[EoS]
  [./eos]
    type = NonIsothermalEquationOfState
    p_0 = 1e5
    rho_0 = 865.51
    a2 = 5.7837e6
    beta = 2.7524e-4
    cv =  1272.0
    e_0 = 7.9898e5
    T_0 = 628.15
  [../]
[]

[Materials]
  [./fuel-mat]
    type = SolidMaterialProperties
    k = 29.3
    Cp = 191.67
    rho = 1.4583e4
  [../]
  [./clad-mat]
    type = SolidMaterialProperties
    k = 26.3
    Cp = 638
    rho = 7.646e3
  [../]
```

```
  [./wall-mat]
    type = SolidMaterialProperties
    k = 26.3
    rho = 7.646e3
    Cp = 638
  [../]
[]

[Components]

  [./reactor]
    type = Reactor
    initial_power = 5.1296e7
  [../]

  [./pipe1]
    type = Pipe
    eos = eos
    position = '0 1 0'
    orientation = '0 -1 0'
    A = 0.44934
    Dh = 2.972e-3
    length = 1
    n_elems = 5
    f = 0.001
    Hw = 0.0
  [../]

  [./CH1]
    type = CoreChannel
    eos = eos
    position = '0 0 0'
    orientation = '0 0 1'
    A = 0.44934
    Dh = 2.972e-3
    length = 0.8
    n_elems = 10
    f = 0.022
    Hw = 1.6129e5
    Phf = 497.778852000000
```

```
  dim_hs = 1
  name_of_hs = 'fuel clad'
  Ts_init = 628.15
  n_heatstruct = 2
  fuel_type = cylinder
  width_of_hs = '0.00348  0.00052'
  elem_number_of_hs = '4 1'
  material_hs = 'fuel-mat clad-mat'
  #peak_power = '1.3366e8 0.'
  power_fraction = '1.0 0.0'
[../]

[./pipe2]
  type = Pipe
  eos = eos
  position = '0 0 0.8'
  orientation = '0 0 1'
  A = 0.44934
  Dh = 2.972e-3
  length = 5.18
  n_elems = 5
  f = 0.001
  Hw = 0.0
[../]

[./pipe3]
  type = Pipe
  eos = eos
  position = '0 0 5.98'
  orientation = '0 1 0'
  A = 0.44934
  Dh = 2.972e-3
  length = 1
  n_elems = 5
  f = 0.001
  Hw = 0.0
[../]

[./IHX]
  type = HeatExchanger
  eos = eos
```

```
  eos_secondary = eos
  position = '0 1.0 5.98'
  orientation = '0 0 -1'
  A = 0.44934
  A_secondary = 0.44934
  Dh = 0.0186
  Dh_secondary = 0.014
  length = 3.71
  n_elems = 10
  Hw = 1.6129e5
  Hw_secondary = 1.6129e5
  Phf = 327.568860000000
  Phf_secondary = 327.568860000000
  f = 0.022
  f_secondary = 0.022
  Twall_init = 628.15
  wall_thickness = 0.0044
  dim_wall = 1
  material_wall = wall-mat
  n_wall_elems = 2
[../]

[./pipe4]
  type = Pipe
  eos = eos
  position = '0 1.0 2.27'
  orientation = '0 0 -1'
  A = 0.44934
  Dh = 2.972e-3
  length = 2.27
  n_elems = 5
  f = 0.001
  Hw = 0.0
[../]

[./pipe5]
  type = Pipe
  eos = eos
  position = '0 0 5.98'
  orientation = '0 0 1'
  A =  0.44934
```

```
  Dh = 2.972e-3
  length = 0.02
  n_elems = 2
  f = 10
  Hw = 0.0
[../]

[./Branch1]
  type = Branch
  inputs = 'pipe1(out)'
  outputs = 'CH1(in) '
  K = '0.5 0.5'
  Area = 0.44934
  Initial_pressure = 2e5
  eos = eos
[../]

[./Branch2]
  type = Branch
  inputs = 'CH1(out) '
  outputs = 'pipe2(in)'
  K = '0.5 0.5'
  Area =  0.44934
  Initial_pressure = 1.0e5
  eos = eos
[../]

[./Branch3]
  type = Branch
  inputs = 'pipe2(out)'
  outputs = 'pipe3(in) pipe5(in)'
  K = '0.0 0.0 0.0'
  Area =   0.44934
  Initial_pressure = 1e5
  eos = eos
[../]

[./Branch4]
  type = Branch
  inputs = 'pipe3(out)'
  outputs = 'IHX(primary_in)'
```

```
  K = '0.1 0.1'
  Area = 0.44934
  Initial_pressure = 1e5
  eos = eos
[../]

[./Branch5]
  type = Branch
  inputs = 'IHX(primary_out)'
  outputs = 'pipe4(in)'
  K = '0.0 0.0'
  Area = 0.44934
  Initial_pressure = 1e5
  eos = eos
[../]

[./Branch6]
  type = Branch
  inputs = 'pipe4(out)'
  outputs = 'pipe1(in)'
  K = '0.0 0.0'
  Area = 0.44934
  Initial_pressure = 1e5
  eos = eos
[../]

[./pipe6]
  type = Pipe
  eos = eos
  position = '0 1.5 2.27'
  orientation = '0 -1 0'
  A = 0.44934
  Dh = 2.972e-3
  length = 0.3
  n_elems = 5
  f = 0.001
  Hw = 0.0
[../]

[./Branch7]
  type = Branch
```

```
    inputs = 'pipe6(out)'
    outputs = 'IHX(secondary_in)'
    K = '0.0 0.0'
    Area = 0.44934
    Initial_pressure = 1e5
    eos = eos
  [../]

  [./inlet_TDJ]
    type = TimeDependentJunction
    input = 'pipe6(in)'
    v_bc = 0.5
    T_bc = 628.15
    eos = eos
  [../]

  [./outlet_TDV]
    type = TimeDependentVolume
    input = 'IHX(secondary_out)'
    p_bc = '1.0e5'
    T_bc = 761.15
    eos = eos
  [../]

  [./TDV_p]
    type = TimeDependentVolume
    input = 'pipe5(out)'
    p_bc = '1e5'
    T_bc = 783.15
    eos = eos
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
  [../]
[]
```

```
[Executioner]
  type = Transient
  scheme = 'implicit-euler'
  dt = 1e-2
  dtmin = 1e-10
  nl_rel_tol = 1e-8
  nl_abs_tol = 1e-8
  nl_max_its = 30
  l_tol = 1e-6
  l_max_its = 300
  start_time = 0.0
  num_steps = 500
  end_time = 10.
  petsc_options_iname = '-pc_type -mat_fd_type -mat_mffd_type -ksp_gmres_restart'
  petsc_options_value = '   lu        ds           ds               300'

  [./Quadrature]
     type = TRAP
     order = FIRST
  [../]
[]

[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5.11 Example 11: A Model Pressurized Water Reactor Problem

5.11.1 Problem Description

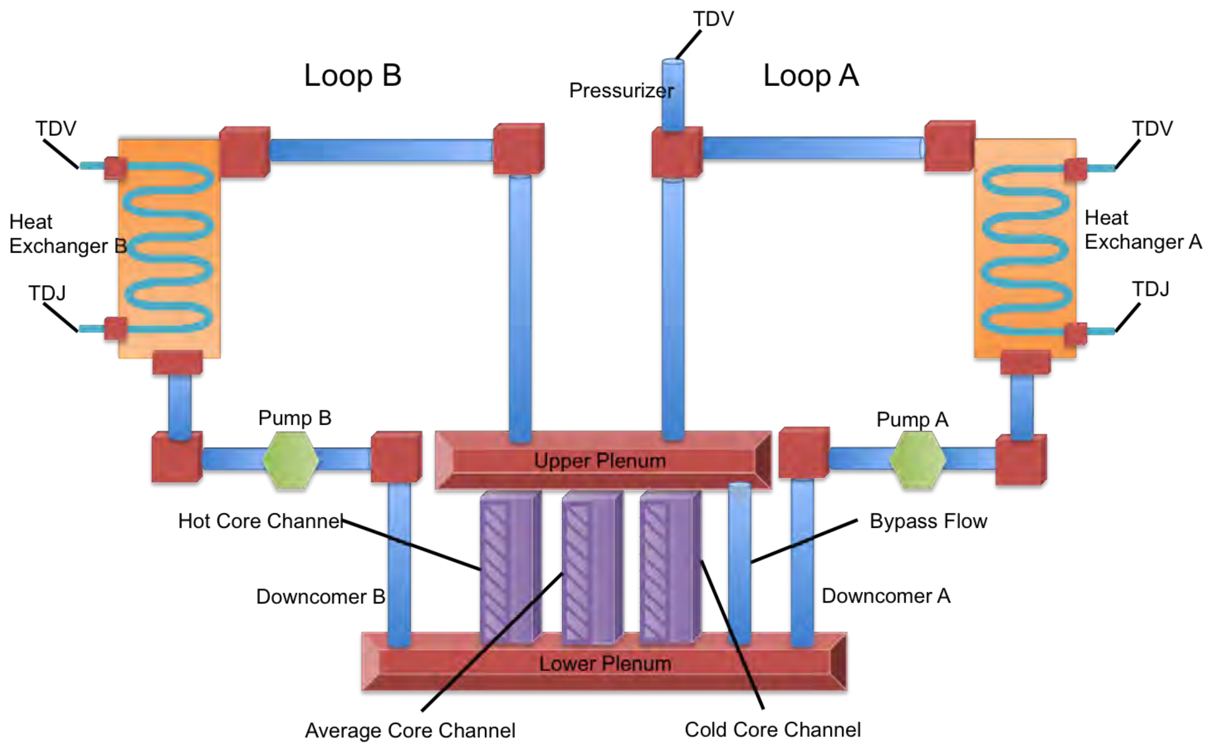

Figure 11. Diagram of a model pressurized water reactor problem

This example shows a simplified PWR plant model. It is made up of thress sections - Loop A, Loop B and a reator vessel model. The reactor vessel region contains three parallel core channels, a bypass flow channel, and an upper and lower plenum. The three core channels represent all the cooling channels and fuel rods in the high power region, average power region and low power region of the reactor core respectively. The upper and lower plenum are modeled with volume branches. The two loops have a Hot Leg, a Heat Exchanger and its secondary side pipes, the Cold Leg and a primary Pump. The heat exchanger secondary side pipes are modeled with subcooled water since the steam generator model is not available yet. Loop A contains a time dependent volume component that works as a pressurizer to help regulate the system pressure.

5.11.2 Input File

```
[GlobalParams]
  model_type = 3
  global_init_P = 15.17e6
  global_init_V = 1.
  global_init_T = 564.15
  scaling_factor_var = '1.e-1 1.e-5 1.e-8'
  temperature_sf = 1e-2
  stabilization_type = 'SUPG'
[]

[EoS]
  [./eos]
    type = NonIsothermalEquationOfState
    p_0 = 15.17e6
    rho_0 = 738.350
    a2 = 1.e7
    beta = .46e-3
    cv = 5.832e3
    e_0 = 3290122.80
    T_0 = 564.15
  [../]
[]

[Materials]
  [./fuel-mat]
    type = SolidMaterialProperties
    k = 3.65
    Cp = 288.734
    rho = 1.0412e2
  [../]

  [./gap-mat]
    type = SolidMaterialProperties
    k = 1.084498
    Cp = 1.0
    rho = 1.0
  [../]
```

```
  [./clad-mat]
    type = SolidMaterialProperties
    k = 16.48672
    Cp = 321.384
    rho = 6.6e1
  [../]

  [./clad3-mat]
    type = SolidMaterialProperties
    k = 16.48672
    Cp = 6.6e3
    rho = 6.6e1
  [../]

  [./wall-mat]
    type = SolidMaterialProperties
    k = 100.0
    rho = 100.0
    Cp = 100.0
  [../]
[]

[Components]
  active = 'reactor
            CH1 CH2 CH3 bypass_pipe LowerPlenum UpperPlenum
            DownComer-A
            pipe1-HL-A pipe2-HL-A
            HX-A
            pipe1-CL-A pipe2-CL-A
            Pump-A
            pipe1-SC-A pipe2-SC-A
            MassFlowRateIn-SC-A PressureOutlet-SC-A
            Branch1-A Branch2-A Branch3-A Branch4-A Branch5-A Branch6-A
            pipe-to-Pressurizer Pressurizer
            DownComer-B
            pipe1-HL-B pipe2-HL-B
            HX-B
            pipe1-CL-B pipe2-CL-B
            Pump-B
            pipe1-SC-B pipe2-SC-B
```

```
          MassFlowRateIn-SC-B PressureOutlet-SC-B
          Branch1-B Branch2-B Branch3-B Branch4-B Branch5-B Branch6-B
          '

[./reactor]
  type = Reactor
  initial_power = 2.77199979e9
[../]

#Vessel region components
[./CH1]
  type = CoreChannel
  eos = eos
  position = '0 -1.2 0'
  orientation = '0 0 1'
  A = 1.161864
  Dh = 0.01332254
  length = 3.6576
  n_elems = 8
  f = 0.01
  Hw = 5.33e4
  Phf = 321.341084980423
  Ts_init = 564.15
  dim_hs = 1
  n_heatstruct = 3
  name_of_hs = 'FUEL GAP CLAD'
  fuel_type = cylinder
  width_of_hs = '0.0046955  0.0000955  0.000673'
  elem_number_of_hs = '3 1 1'
  material_hs = 'fuel-mat gap-mat clad-mat'
  power_fraction = '3.33672612e-1 0 0'
[../]

[./CH2]
  type = CoreChannel
  eos = eos
  position = '0 0 0'
  orientation = '0 0 1'
  A = 1.549152542
  Dh = 0.01332254
  length = 3.6576
```

```
  n_elems = 8 #16
  f = 0.01
  Hw = 5.33e4
  Phf = 428.454929876871
  Ts_init = 564.15
  dim_hs = 1
  n_heatstruct = 3
  name_of_hs = 'FUEL GAP CLAD'
  fuel_type = cylinder
  width_of_hs = '0.0046955  0.0000955  0.000673'
  elem_number_of_hs = '3 1 1'
  material_hs = 'fuel-mat gap-mat clad-mat'
  power_fraction = '3.69921461e-1 0 0'
[../]

[./CH3]
  type = CoreChannel
  eos = eos
  position = '0 1.2 0'
  orientation = '0 0 1'
  A = 1.858983051
  Dh = 0.01332254
  length = 3.6576
  n_elems = 8
  f = 0.01
  Hw = 5.33e4
  # aw = 276.5737513
  Phf = 514.145916018189
  Ts_init = 564.15
  dim_hs = 1
  n_heatstruct = 3
  name_of_hs = 'FUEL GAP CLAD'
  fuel_type = cylinder
  width_of_hs = '0.0046955  0.0000955  0.000673'
  elem_number_of_hs = '3 1 1'
  material_hs = 'fuel-mat gap-mat clad3-mat'
  #peak_power = '3.401687e8 0. 0.'
  power_fraction = '2.96405926e-1 0 0'
[../]

[./bypass_pipe]
```

```
  type = Pipe
  eos = eos
  position = '0 1.5 0'
  orientation = '0 0 1'
  A = 1.589571014
  Dh = 1.42264
  length = 3.6576
  n_elems = 5
  f = 0.001
  Hw = 0.0
[../]

[./LowerPlenum]
  type = Branch
  eos = eos
  inputs = 'DownComer-A(out) DownComer-B(out)'
  outputs = 'CH1(in) CH2(in) CH3(in) bypass_pipe(in)'
  K = '0.2 0.2 0.2 0.2 0.4 40.0'
  Area = 3.618573408
  Initial_pressure = 151.7e5
[../]

[./UpperPlenum]
  type = Branch
  eos = eos
  inputs = 'CH1(out) CH2(out) CH3(out) bypass_pipe(out)'
  outputs = 'pipe1-HL-A(in) pipe1-HL-B(in)'
  K = '0.5 0.5 0.5 80.0 0.5 0.5'
  Area = 7.562307456
  Initial_pressure = 151.7e5
[../]

#Loop A components
[./DownComer-A]
  type = Pipe
  eos = eos
  position = '0 2.0 4.0'
  orientation = '0 0 -1'
  A = 3.6185734
  Dh = 1.74724302
  length = 4
```

```
  n_elems = 3
  f = 0.001
  Hw = 0.
[../]

[./pipe1-HL-A]
  type = Pipe
  eos = eos
  position = '0 0.5 4.0'
  orientation = '0 0 1'
  A = 7.562307456
  Dh = 3.103003207
  length = 4.
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]

[./pipe2-HL-A]
  type = Pipe
  eos = eos
  position = '0 0.5 8.0'
  orientation = '0 1 0'
  A = 2.624474
  Dh = 1.828
  length = 3.5
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]

[./pipe1-CL-A]
  type = Pipe
  eos = eos
  position = '0 3.0 4.0'
  orientation = '0 -1 0'
  A = 2.624474
  Dh = 1.828
  length = 1.
  n_elems = 3
  f = 0.001
```

```
  Hw = 0.0
[../]

[./pipe2-CL-A]
  type = Pipe
  eos = eos
  position = '0 4 4.0'
  orientation = '0 -1 0'
  A = 2.624474
  Dh = 1.828
  length = 0.8
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]

[./pipe1-SC-A]
  type = Pipe
  eos = eos
  position = '0 5.2 4.0'
  orientation = '0 -1 0'
  A = 2.624474
  Dh = 1.828
  length = 1.
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]

[./pipe2-SC-A]
  type = Pipe
  eos = eos
  position = '0 4.2 8.0'
  orientation = '0 1 0'
  A = 2.624474
  Dh = 1.828
  length = 1.
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]
```

```
[./Branch1-A]
  type = Branch
  eos = eos
  inputs = 'pipe1-HL-A(out)'
  outputs = 'pipe2-HL-A(in) pipe-to-Pressurizer(in)'
  K = '0.5 0.7 80.'
  Area = 7.562307456
  Initial_pressure = 151.7e5
[../]

[./Branch2-A]
  type = Branch
  eos = eos
  inputs = 'pipe1-CL-A(out)'
  outputs = 'DownComer-A(in)'
  K = '0.5 0.7'
  Area = 3.6185734
  Initial_pressure = 151.7e5
[../]

[./Branch3-A]
  type = Branch
  eos = eos
  inputs = 'pipe2-HL-A(out)'
  outputs = 'HX-A(primary_in)'
  K = '0.5 0.7'
  Area = 2.624474
  Initial_pressure = 151.7e5
[../]

[./Pump-A]
  type = IdealPump
  eos = eos
  inputs = 'pipe2-CL-A(out)'
  outputs = 'pipe1-CL-A(in)'
  Area = 2.624474
  mass_flow_rate = 8801.1
  Initial_pressure = 151.7e5
[../]
```

```
[./HX-A]
  type = HeatExchanger
  eos = eos
  eos_secondary = eos
  position = '0 4. 8.'
  orientation = '0 0 -1'
  A = 5.
  A_secondary = 5.
  Dh = 0.01
  Dh_secondary = 0.01
  length = 4.
  n_elems = 10
  Hw = 1.e4
  Hw_secondary = 1.e4
  Phf = 2695.100000000000
  Phf_secondary = 2695.100000000000
  f = 0.01
  f_secondary = 0.01
  dim_wall = 1
  Twall_init = 564.15
  wall_thickness = 0.001
  n_wall_elems = 2
  material_wall = wall-mat
[../]

[./Branch4-A]
  type = Branch
  eos = eos
  inputs = 'pipe1-SC-A(out)'
  outputs = 'HX-A(secondary_in)'
  K = '0.5 0.7'
  Area = 2.624474e2
  Initial_pressure = 151.7e5
[../]

[./Branch5-A]
  type = Branch
  eos = eos
  inputs = 'HX-A(secondary_out)'
  outputs = 'pipe2-SC-A(in)'
  K = '0.5 0.7'
```

```
  Area = 2.624474e2
  Initial_pressure = 151.7e5
[../]

[./Branch6-A]
  type = Branch
  eos = eos
  inputs = 'HX-A(primary_out)'
  outputs = 'pipe2-CL-A(in)'
  K = '0.5 0.7'
  Area = 2.624474e2
  Initial_pressure = 151.7e5
[../]

[./MassFlowRateIn-SC-A]
  type = TimeDependentJunction
  input = 'pipe1-SC-A(in)'
  v_bc = 4.542
  T_bc = 537.15
  eos = eos
[../]

[./PressureOutlet-SC-A]
  type = TimeDependentVolume
  input = 'pipe2-SC-A(out)'
  p_bc = '151.7e5'
  T_bc = 564.15
  eos = eos
[../]

#Loop B components
[./DownComer-B]
  type = Pipe
  eos = eos
  position = '0 -2.0 4.0'
  orientation = '0 0 -1'
  A = 3.6185734
  Dh = 1.74724302
  length = 4
  n_elems = 3
  f = 0.001
```

```
  Hw = 0.
[../]

[./pipe1-HL-B]
  type = Pipe
  eos = eos
  position = '0 -0.5 4.0'
  orientation = '0 0 1'
  A = 7.562307456
  Dh = 3.103003207
  length = 4.
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]

[./pipe2-HL-B]
  type = Pipe
  eos = eos
  position = '0 -0.5 8.0'
  orientation = '0 -1 0'
  A = 2.624474
  Dh = 1.828
  length = 3.5
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]

[./pipe1-CL-B]
  type = Pipe
  eos = eos
  position = '0 -3.0 4.0'
  orientation = '0 1 0'
  A = 2.624474
  Dh = 1.828
  length = 1.
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]
```

```
[./pipe2-CL-B]
  type = Pipe
  eos = eos
  position = '0 -4.0 4.0'
  orientation = '0 1 0'
  A = 2.624474
  Dh = 1.828
  length = 0.8
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]

[./pipe1-SC-B]
  type = Pipe
  eos = eos
  position = '0 -5.2 4.0'
  orientation = '0 1 0'
  A = 2.624474
  Dh = 1.828
  length = 1.
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]

[./pipe2-SC-B]
  type = Pipe
  eos = eos
  position = '0 -4.2 8.0'
  orientation = '0 -1 0'
  A = 2.624474
  Dh = 1.828
  length = 1.
  n_elems = 3
  f = 0.001
  Hw = 0.0
[../]

[./Branch1-B]
```

```
  type = Branch
  eos = eos
  inputs = 'pipe1-HL-B(out)'
  outputs = 'pipe2-HL-B(in)'
  K = '0.5 0.7'
  Area = 7.562307456
  Initial_pressure = 151.7e5
[../]

[./Branch2-B]
  type = Branch
  eos = eos
  inputs = 'pipe1-CL-B(out)'
  outputs = 'DownComer-B(in)'
  K = '0.5 0.7'
  Area = 3.6185734
  Initial_pressure = 151.7e5
[../]

[./Branch3-B]
  type = Branch
  eos = eos
  inputs = 'pipe2-HL-B(out)'
  outputs = 'HX-B(primary_in)'
  K = '0.5 0.7'
  Area = 2.624474
  Initial_pressure = 151.7e5
[../]

[./Pump-B]
  type = IdealPump
  eos = eos
  inputs = 'pipe2-CL-B(out)'
  outputs = 'pipe1-CL-B(in)'
  Area = 2.624474
  mass_flow_rate = 8801.1
  Initial_pressure = 151.7e5
[../]

[./HX-B]
  type = HeatExchanger
```

```
  eos = eos
  eos_secondary = eos
  position = '0 -4. 8.'
  orientation = '0 0 -1'
  A = 5.
  A_secondary = 5.
  Dh = 0.01
  Dh_secondary = 0.01
  length = 4.
  n_elems = 10
  Hw = 1.e4
  Hw_secondary = 1.e4
  Phf = 2695.100000000000
  Phf_secondary = 2695.100000000000
  f = 0.01
  f_secondary = 0.01
  dim_wall = 1
  Twall_init = 564.15
  wall_thickness = 0.001
  material_wall = wall-mat
  n_wall_elems = 2
  disp_mode = -1.0
[../]

[./Branch4-B]
  type = Branch
  eos = eos
  inputs = 'pipe1-SC-B(out)'
  outputs = 'HX-B(secondary_in)'
  K = '0.5 0.7'
  Area = 2.624474e2
  Initial_pressure = 151.7e5
[../]

[./Branch5-B]
  type = Branch
  eos = eos
  inputs = 'HX-B(secondary_out)'
  outputs = 'pipe2-SC-B(in)'
  K = '0.5 0.7'
  Area = 2.624474e2
```

```
    Initial_pressure = 151.7e5
  [../]

  [./Branch6-B]
    type = Branch
    eos = eos
    inputs = 'HX-B(primary_out)'
    outputs = 'pipe2-CL-B(in)'
    K = '0.5 0.7'
    Area = 2.624474e2
    Initial_pressure = 151.7e5
  [../]

  [./MassFlowRateIn-SC-B]
    #type = TDM
    type = TimeDependentJunction
    input = 'pipe1-SC-B(in)'
#   massflowrate_bc = 8801.1
    v_bc = 4.542
    T_bc = 537.15
    eos = eos
  [../]

  [./PressureOutlet-SC-B]
    type = TimeDependentVolume
    input = 'pipe2-SC-B(out)'
    p_bc = '151.7e5'
    T_bc = 564.15
    eos = eos
  [../]

  # Pressurizer
  [./pipe-to-Pressurizer]
    type = Pipe
    eos = eos
    position = '0 0.5 8.0'
    orientation = '0 0 1'
    A = 2.624474
    Dh = 1.828
    length = 0.5
    n_elems = 3
```

```
    f = 10.
    Hw = 0.0
  [../]

  [./Pressurizer]
    type = TimeDependentVolume
    input = 'pipe-to-Pressurizer(out)'
    p_bc = '151.7e5'
    T_bc = 564.15
    eos = eos
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
  [../]
[]

[Executioner]
  type = Transient
  scheme = 'implicit-euler'

  #use this parameter to have a better initial guess from previous two time steps, therefo
  [./Predictor]
    type = SimplePredictor
    scale = 0.6
  [../]

  dt = 0.1
  dtmin = 1e-7
  nl_rel_tol = 1e-9
  nl_abs_tol = 1e-8
  nl_max_its = 30
  l_tol = 1e-4
  l_max_its = 300
  start_time = 0.0
  end_time = 50
```

```
  num_steps = 250
  petsc_options_iname = '-mat_fd_type  -mat_mffd_type -ksp_gmres_restart  -pc_type'
  petsc_options_value = 'ds            ds                300                lu'

  [./Quadrature]
     type = TRAP
     order = FIRST
  [../]
[]

[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]
[]
```

5.11.3 Description of the Input File

Most of the input parameters have been explained in the previous examples. Only a few need some explanations here.

```
  [./Pump-A]
    type = IdealPump
    eos = eos
    inputs = 'pipe2-CL-A(out)'
    outputs = 'pipe1-CL-A(in)'
    Area = 2.624474
    mass_flow_rate = 8801.1
    Initial_pressure = 151.7e5
  [../]
```

`type` The ideal pump component is used here. Please note that Ideal Pump component is an interim model. It merely provides a prescribed mass flow rate to the pipes with this component. This component will eventually be replaced with a realistic pump model.

`eos` Name of equation of state to be used by IdealPump.

`inputs` The inlet of the ideal pump connects to the outlet of `pipe2-CL-A`.

`outputs` The outlet of the ideal pump connects to the inlet of `pipe1-CL-A`.

`Area` The reference flow area of the ideal pump.

`mass_flow_rate` The prescribed mass flow rate generated by the ideal pump.

`Initial_pressure` The prescribed initial preesure of the ideal pump.

```
[./Predictor]
  type = SimplePredictor
  scale = 0.6
[../]
```

`[./Predictor]` Subblock for predictor. The predictor provides a better initil guess from the solution of previous two old time steps. It reduces the number of nonlinear iterations.

`type` The SimplePredictor uses an algorithm that will predict the next solution based on previous solutions. The algorithm is: $sol^{n+1} = sol^n + \frac{sol^n - sol^{n-1}}{\delta t^n} * \delta t^{n+1} * scale$.

`scale` The scale factor used in the SimplePredictor calculation.

5.12 Example 12: A Simplified Primary System Model of a Boiling Water Reactor

5.12.1 Problem Description

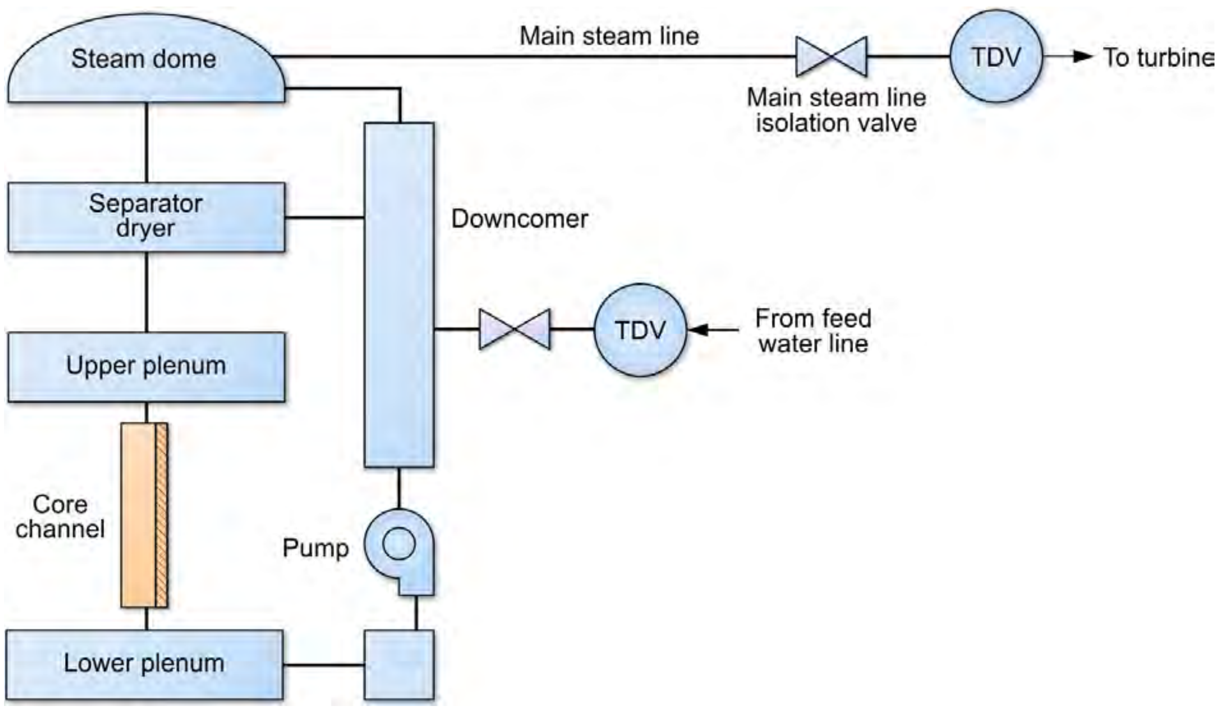

Figure 12. Diagram of a simplified primary system of a BWR model

This example is a simplified primary system of a boiling water reactor. The model consists of down comer, lower plenum, reactor core, upper plenum, separator dryer, steam dome, main steam line, feedwater line and the primary pump model. One core channel is used to represent the entire reactor core. The lower plenum, upper lenum and steam dome are modeled with volume branch models. External to the reactor vessel, the main steam line is connected to the steam dome. A time dependent volume is attached to the main steam line to provide the necessary boundary conditions for the steam flow. A feedwater line is connected to the downcomer model. A time dependent volume is attached to the feedwater line to provide the necessary boundary conditions for the feedwater. Notably missing from this simplified BWR model are the jet pumps and the recirculation loops that allow the operator to vary coolant flow rate through the core and change the reactor power. Instead, a pump model is used to represent the functions of the jet pumps and the recirculation loops.

5.12.2 Input File

```
[GlobalParams]
  model_type = 32
  global_init_P = 7.e6
  global_init_V = 3.
  global_init_T = 517.
  scaling_factor_var   = '1e-3 1e-4 1e-8'
  temperature_sf = '1e-4'
  gravity = '0 0 -9.8'
  stabilization_type = 'LAPIDUS'
[]

[EoS]

  [./two_phase_eos]
    type = TwoPhaseStiffenedGasEOS
  [../]

  [./vapor_phase_eos]
    type = StiffenedGasEquationOfStateVapor
  [../]

  [./liquid_phase_eos]
    type = StiffenedGasEquationOfStateLiquid
  [../]

[]

[Materials]
  [./fuel-mat]
    type = SolidMaterialProperties
    k = 3.7
    Cp = 3.e2
    rho = 10.42e3
```

```
  [../]
  [./gap-mat]
    type = SolidMaterialProperties
    k = 0.7
    Cp = 5e3
    rho = 1.0
  [../]
  [./clad-mat]
    type = SolidMaterialProperties
    k = 16
    Cp = 356.
    rho = 6.551400e3
  [../]
[]

[Components]
  [./reactor]
    type = Reactor
    initial_power = 3293.0e6
  [../]

  [./ch1]
    type = CoreChannel
    eos = two_phase_eos
    position = '0  0.0  5.28'
    orientation = '0 0 1'
    A = 7.8
    Dh = 1.3597E-02
    length = 3.66
    n_elems = 100
    f = 0.2 #0.05
    Hw = 5.0e4
    Phf = 1836.8430600
    Ts_init = 517.
    fuel_type = cylinder
    dim_hs = 1
    n_heatstruct = 3
    name_of_hs =  'FUEL GAP CLAD'
    width_of_hs = '6.057900e-3  1.524000e-4  9.398000e-4'
    elem_number_of_hs = '5 1 2'
    material_hs = 'fuel-mat gap-mat clad-mat'
```

```
  power_fraction = '1.0 0.0 0.0'
  stabilization_type = 'NONE'
  model_type = 32
[../]

[./pipe6]
  type = Pipe
  position =    '0.0 0.0 10.82'
  orientation = '0 0 1'
  A = 3.93
  Dh = 1.0
  length = 2.72
  n_elems = 40
  f = 0.1
  eos = two_phase_eos
  model_type = 32
  stabilization_type = 'NONE'
[../]

[./pipe7]
  type = Pipe
  position = '0.0  0.0  15.42'
  orientation = '0  0  1'
  A = 3.93
  Dh = 1.0
  length = 0.1
  n_elems = 5
  f = 0.1
  eos = vapor_phase_eos
  model_type = 3
[../]

[./pipe8]
  type = Pipe
  position = '0.0  2.0  14.48'
  orientation = '0  1  0'
  A = 3.93
  Dh = 1.0
  length = 0.5
  n_elems = 5
  f = 0.1
```

```
  eos = liquid_phase_eos
  model_type = 3
  stabilization_type = 'SUPG'
[../]

[./pipe9]
  # main steam line coming out of dome
  type = Pipe
  position = '0.0  3  18.92'
  orientation = '0 1 0'
  A = 1.32
  Dh = 1.0
  length = 1.0
  n_elems = 5
  f = 0.1
  eos = vapor_phase_eos
  model_type = 3
[../]

[./pipe10]
  type = Pipe
  #position = '0.0  5.0  10.51'
  position = '0.0  2.75  2.10'
  orientation = '0  0  -1'
  A = 8.55
  Dh = 1.0
  #length = 6.42
  length = 0.5
  n_elems = 5
  f = 0.1
  eos = liquid_phase_eos
  model_type = 3
  stabilization_type = 'SUPG'
[../]

[./pipe11]
  type = Pipe
  position = '0.0  2.75  1.60'
  orientation = '0  0  -1'
  A = 8.55
  Dh = 1.0
```

```
    length = 0.5
    n_elems = 5
    f = 0.1
    eos = liquid_phase_eos
    model_type = 3
    stabilization_type = 'SUPG'
  [../]

  [./pipe14]
    # main steam line to MIV
    type = Pipe
    position = '0.0  4.0  18.92'
    orientation = '0 1 0'
    A = 1.32
    Dh = 1.0
    length = 1.0
    n_elems = 5
    f = 0.0
    eos = vapor_phase_eos
    model_type = 3
  [../]

 [./MainIsolationValve]
    type = VolumeBranch
    eos = vapor_phase_eos
    center = '0.0  5.0  18.92'
    inputs = 'pipe14(out)'
    outputs = 'pipe_steam_turbine(in)'
    K = '0.0 0.0'
    volume = 1.32
    Area = 1.32
    initial_T = 517.0
    scale_factors = '1.0E-4  1.0E-8 1.0'  # rho, rhoE
  [../]

  [./pipe_steam_turbine]
    # main steam line to TDV
    type = Pipe
    position = '0.0  5  18.92'
    orientation = '0 1 0'
    A = 1.32
```

```
  Dh = 1.0
  length = 1.0
  n_elems = 5
  f = 0.0
  eos = vapor_phase_eos
  model_type = 3
[../]

[./Pump]
  type = Pump
  eos = liquid_phase_eos
  inputs = 'pipe10(out)'
  outputs = 'pipe11(in)'
  Area = 3.0
  Initial_pressure = 7.3e6
  #mass_flow_rate = 12915.0
  Head = 40.0
  K_reverse = '10. 10.'
[../]

[./SeparatorDryer]
  type = SeparatorDryer
  eos = two_phase_eos
  center = '0.0 0.0 14.48'
  inputs = 'pipe6(out)'
  outputs = 'pipe7(in)  pipe8(in)'
  K = '1.0  1.0  5.0'
  volume = 19.30
  Area = 10.27
  initial_T = 517.0
  scale_factors = '1.0E-3  1.0E-8  1.0E-0'
[../]

[./lowerplenum]
  type = VolumeBranch
  eos = two_phase_eos
  center = '0.0  0.0  2.64'
  inputs = 'pipe11(out)'
  outputs = 'ch1(in)'
  K = '1.0 20.0'
  volume = 61.48
```

```
  Area = 11.64
  initial_T = 517.0
  scale_factors = '1.0E-3  1.0E-8  1.0E-0'
[../]

[./upperplenum]
  type = VolumeBranch
  eos = two_phase_eos
  center = '0.0  0.0  9.88'
  inputs = 'ch1(out)'
  outputs = 'pipe6(in)'
  K = '3.0 1.0'
  volume = 26.99
  Area = 14.36
  initial_T = 517.0
  scale_factors = '1.0E-3  1.0E-8  1.0E-0'
[../]

[./Dome]
  type = VolumeBranch
  eos = vapor_phase_eos
  center = '0.0  0.0  18.92'
  inputs = 'pipe7(out)'
  outputs = 'pipe9(in)'
  K = '1.0 1.0'
  volume = 178.19
  Area = 26.19
  scale_factors = '1.0E-3  1.0E-8  1.0E-0'
[../]

[./DownComer]
  type = DownComer
  eos = liquid_phase_eos
  center = '0.0  2.75  9.81'
  inputs = 'pipe8(out) pipe_feedwater3(out)'
  outputs = 'pipe10(in)'
  K = '1.0 10.0 1.0'
  volume = 201.3
  Area = 15
  initial_level = 13.42
  initial_T = 517.0
```

```
  dome_component = 'Dome'
  dome_eos = vapor_phase_eos
  scale_factors = '1.0E-4  1.0E-9  1.0E-0'
[../]

[./SteamLineBranch]
  type = VolumeBranch
  eos = vapor_phase_eos
  center = '0.0  4.0  18.92'
  inputs = 'pipe9(out)'
  outputs = 'pipe14(in)'
  K = '0.0 0.0'
  volume = 2.64
  Area = 1.32
  initial_T = 517.0
  scale_factors = '1.0E-4 1.0E-8 1.0'
[../]

[./pipe_feedwater1]
  #feedwater line from TDV
  type = Pipe
  position = '0.0  6.0  12.52' #'0.0  7.0  12.52'
  orientation = '0  -1  0'
  A = 1.32
  Dh = 1.0
  length = 1.0
  n_elems = 5
  f = 0.01 #1
  eos = liquid_phase_eos
  model_type = 3
  stabilization_type = 'SUPG'
[../]

[./FeedWaterValve]
  type = Valve
  eos = liquid_phase_eos
  center = '0.0  5.0  12.52'
  inputs = 'pipe_feedwater1(out)'
  outputs = 'pipe_feedwater2(in)'
  K = '0.0 0.0'
  volume = 1.32
```

```
  Area = 1.32
  initial_T = 517.0
  initial_status = open
  trigger_time = 1.0E5
  response_time = 1.1E5
  scale_factors = '1.0E-4  1.0E-8'
[../]

[./pipe_feedwater2]
  #feedwater line from feed water valve
  type = Pipe
  position = '0.0  5.0  12.52' #'0.0  7.0  12.52'
  orientation = '0  -1  0'
  A = 1.32
  Dh = 1.0
  length = 1.0
  n_elems = 5
  f = 0.01
  eos = liquid_phase_eos
  model_type = 3
  stabilization_type = 'SUPG'
[../]

[./branch_feedwater_line]
  type = VolumeBranch
  eos = liquid_phase_eos
  center = '0.0  4.0  12.52'
  inputs = 'pipe_feedwater2(out)'
  outputs = 'pipe_feedwater3(in)'
  K = '0  0'
  volume = 1.32
  Area = 1.32
  initial_T = 517.0
  scale_factors = '1.0E-4 1.0E-8 1.0'
[../]

[./pipe_feedwater3]
  #feedwater line to downcomer
  type = Pipe
  position = '0.0  4.0  12.52' #'0.0  6.0  12.52'
  orientation = '0  -1  0'
```

```
    A = 1.32
    Dh = 1.0
    length = 1.0
    n_elems = 5
    f = 0.01
    eos = liquid_phase_eos
    model_type = 3
    stabilization_type = 'SUPG'
  [../]

  [./inlet]
    type = TimeDependentVolume
    input = 'pipe_feedwater1(in)'
    p_bc = 7.1e6
    T_bc = 508.
    eos = liquid_phase_eos
  [../]

  [./outlet1]
    type = TimeDependentVolume
    input = 'pipe_steam_turbine(out)'
    p_bc = 7.0e6
    T_bc = 517
    eos = vapor_phase_eos
    weak_bc = false
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
    solve_type = 'PJFNK'
  [../]

[]

[Executioner]
  type = Transient
  dtmin = 1.e-7
  [./TimeStepper]
```

```
    type = SolutionTimeAdaptiveDT
    dt = 0.1
  [../]
  petsc_options_iname = '-ksp_gmres_restart -pc_type'
  petsc_options_value = '30 lu'

  nl_rel_tol = 1e-8
  nl_abs_tol = 1e-6
  nl_max_its = 15

  l_tol = 1e-5
  l_max_its = 100

  start_time = 0.0
  end_time = 400
  num_steps = 500000

  [./Quadrature]
    type = TRAP
    order = FIRST
  [../]
[]

[Outputs]
  [./out]
    type = Exodus
    use_displaced = true
    output_initial = true
    sequence = false
    append_displaced = true
    interval = 50
  [../]

  [./console]
    type = Console
    perf_log = true
  [../]

  [./checkpoint]
    type = Checkpoint
    num_files = 1
```

```
  [../]
[]
```

5.12.3 Description of the Input File

```
[./SeparatorDryer]
  type = SeparatorDryer
  eos = two_phase_eos
  center = '0.0 0.0 14.48'
  inputs = 'pipe6(out)'
  outputs = 'pipe7(in)  pipe8(in)'
  K = '1.0  1.0  5.0'
  volume = 19.30
  Area = 10.27
  initial_T = 517.0
  scale_factors = '1.0E-3  1.0E-8  1.0E-0'
[../]
```

`[./SeparatorDryer]`	Subblock for separator dryer component.
`type`	Component type is SeparatorDryer.
`eos`	Name of equation of state to be used by the separator dryer component.
`center`	The `(x,y,z)` coordinate of the center of the separator dryer component.
`inputs`	The inlet of the separator dryer component connects with the outlet of `pipe6`.
`outputs`	The outlets of the separator dryer component connects with the inlet of `pipe7` and `pipe8`.
`K`	The form loss coefficients at the junctions with the pipes.
`volume`	The volume of the separator dryer component.
`Area`	The flow area of separator dryer component.
`initial_T`	The initial temperature of the separator dryer component.

scale_factors — The scaling factors for the primary variables of the separator dryer component.

```
[./DownComer]
  type = DownComer
  eos = liquid_phase_eos
  center = '0.0  2.75  9.81'
  inputs = 'pipe8(out) pipe_feedwater3(out)'
  outputs = 'pipe10(in)'
  K = '1.0 10.0 1.0'
  volume = 201.3
  Area = 15
  initial_level = 13.42
  initial_T = 517.0
  dome_component = 'Dome'
  dome_eos = vapor_phase_eos
  scale_factors = '1.0E-4  1.0E-9  1.0E-0'
[../]
```

`[./DownComer]`	Subblock for the down comer component.
`type`	The component type is DownComer.
`eos`	Name od the equation of state to be used by the down comer component.
`center`	The `(x,y,z)` coordinate of the center of the down comer component.
`inputs`	The inlet of down comer connects with the outlet of `pipe8` and `pipe_feedwater3`.
`outputs`	The outlet of down comer connects with the inlet of `pipe10`.
`K`	The form loss coefficients at the junctions with the pipes.
`volume`	The volume of the down comer component.
`Area`	The reference flow area of the downcomer component.
`initial_level`	The initial liquid level of the down comer component.

initial_T The initial temperature of the down comer component.

dome_component Name of the steam dome component to provide the dome pressure to down comer.

dome_eos Name of the equation of state to be used by the steam dome component.

scale_factors The saling factors for the primary variables of the down comer component.

```
[./FeedWaterValve]
  type = Valve
  eos = liquid_phase_eos
  center = '0.0  5.0  12.52'
  inputs = 'pipe_feedwater1(out)'
  outputs = 'pipe_feedwater2(in)'
  K = '0.0 0.0'
  volume = 1.32
  Area = 1.32
  initial_T = 517.0
  initial_status = open
  trigger_time = 1.0E5
  response_time = 1.1E5
  scale_factors = '1.0E-4  1.0E-8'
[../]
```

[./FeedWater] Subblock for the valve component.

type The component type is Valve.

eos Name od the equation of state to be used by the valve component.

center The (x,y,z) coordinate of the center of the valve component.

inputs The inlet of valve connects with the outlet of pipe_feedwater1.

outputs The outlet of valve connects with the inlet of pipe_feedwater2.

K The form loss coefficients at the junctions with the pipes.

`volume` The volume of the valve component.

`Area` The reference flow area of the valve component.

`initial_T` The initial temperature of the valve component.

`initial_status` The initial status of the valve is at the open position.

`trigger_time` The time at which the valve open/close action is triggered.

`response_time` The time at which the valve open/close action is complete.

`scale_factors` The saling factors for the primary variables of the valve component.

```
[./checkpoint]
  type = Checkpoint
  num_files = 1
[../]
```

`[./checkpoint]` Subblock for check point. This is used to create restart files for subsequent runs.

`num_files` Number of restart files to save.

References

[1] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, vol. 193, pp. 357–397, Jan. 2004. `http://dx.doi.org/10.1016/j.jcp.2003.08.010`.

[2] P. N. Brown and A. C. Hindmarsh, "Matrix-free methods for stiff systems of ODEs," *SIAM J. Numer. Anal.*, vol. 23, pp. 610–638, June 1986. `http://www.jstor.org/stable/2157527`.

[3] B. S. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey, "`libMesh`: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations," *Engineering with Computers*, vol. 22, no. 3–4, pp. 237–254, 2006. `http://dx.doi.org/10.1007/s00366-006-0049-3`.

[4] S. Balay, W. D. Gropp, L. Curfman-McInnes, and B. F. Smith, "Efficient management of parallelism in object oriented numerical software libraries," in *Modern Software Tools in Scientific Computing* (E. Arge, A. M. Bruaset, and H. P. Langtangen, eds.), pp. 163–202, Birkhäuser Press, 1997.

[5] M. Heroux *et al.*, "An overview of Trilinos," Tech. Rep. SAND2003-2927, Sandia National Laboratories, 2003.

[6] H. Zhang, L. Zou, D. Andrs, H. Zhao, and R. C. Martineau, "Point Kinetics Calculations with Fully Coupled Thermal Fluids Reactivity Feedback," in *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2013)*, (Sun Valley, Idaho, USA), May 5–9, 2013.

[7] L. Zou, J. Peterson, H. Zhao, H. Zhang, D. Andrs, and R. C. Martineau, "Solving Multi-Mesh Flow and Conjugate Heat Transfer Problems with RELAP-7," in *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2013)*, (Sun Valley, Idaho, USA), May 5–9, 2013.

[8] R. D. Falgout and U. M. Yang, "HYPRE: A Library of High Performance Preconditioners," in *International Conference on Computational Science*, pp. 632–641, 2002.

v1.28

Idaho National Laboratory