Numerics Capabilities Exercise

Objectives

- Practice adding Numerics variables to manage transient model differences.

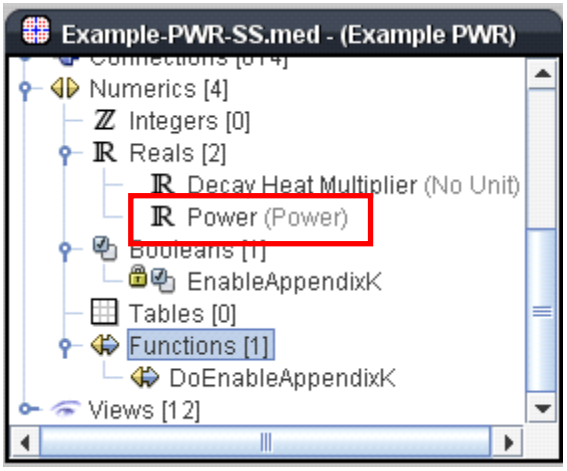
Preliminary Setup (Open Model)

Open the 'Day4/Morning/PWR1_Numerics' folder and double click on 'PWR1-SS-N.med' to open the example PWR model.

The "Exercise Key" included in the workbook may be useful to help locate the various parts of the SNAP Model Editor that are referred to in this exercise.

Add a Numerics Variable for Power

The model is currently configured to run under best estimate conditions at a core power of 2300 MW. However, for the SBLOCA simulation (to be performed in a later exercise) we would like to include 'Appendix K' assumptions. One of these assumptions is that power is increased to 102% (2300*1.02=2346 MW). Another assumption is that a decay heat multiplier of 1.2 is used. The decay heat multiplier Numerics variable has already been added to the model. Add a Power variable via Numerics by doing the following:



1. In the Navigator Window, locate and expand the Numerics section by clicking on the ⊢ in front of ⊢ Numerics.
2. Right click on ⊢ Reals and select 'New' to create a new Numerics variable.
3. Numerics variables have units associated with them. In this case we want to configure the power, so choose 'Power (W)' in the 'Select Type' dialog that appears, and click OK.
4. In the Properties Window set Name | Power, Interactive Variable | ○ True ◉ False, and Initial Value | 2.3E9 (W).

> Interactive variables are variables that will be modified manually. If the variables will be modified using Python code (or Matlab or Mathcad code), then 'Interactive Variable' should be set to false. In this case, the value will be modified via Python.

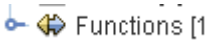
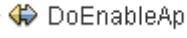
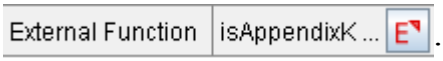
5. In the Navigator Window, locate and expand ⊢ Power Components [1]. Then expand ⊢ Powers [1] and click on ⊢ Power 999. In the Properties Window, click on the icon ◂▸ next to Initial Power | 2.3E9 (W) ◂▸. This brings up the dialog for selecting a Numerics variable. Select 'Power' and click OK. The initial power field should now look like Initial Power | Power(2.3E9) (W) ◂▸.

As noted, Numerics variables are assigned units. When you select a numerics variable, the list is filtered such that only Numerics variables that have the correct units are displayed. In this case, the unit is (W)atts.

Numerics can also be used to set values in many tables. However there is no  icon to click. One way to use a Numerics variable is click in the name field of the Numerics variable, hit 'Ctrl-c' to copy the name, and 'Ctrl-v' to paste this into a table value. Some tables don't accept Numerics values yet. If there are places you need to apply Numerics that you can't, visit the SNAP website and submit a trouble report.

Configure the Numerics Variable via Python

The model already contains a Numerics Python function that is used to modify Appendix K parameters, the function uses a Numerics 'Boolean' value to determine whether best estimate conditions or Appendix K conditions should be used. The coding to update the 'Decay Heat Multiplier' is already included in the Python function. The coding to update the Power for Appendix K conditions needs to be added. To do this, do the following:

1. In the Navigator Window, under 'Numerics', expand  .
2. Click on  . In the Properties Window, Expand  . A dialog box containing the Python script for this Numerics function should appear. The following Python script is defined:

```
isAppendixK = GetVariable("EnableAppendixK")

decay_heat_multiplier = 1.0

if isAppendixK == True:
    decay_heat_multiplier = 1.2

SetVariable("Decay_Heat_Multiplier",
decay_heat_multiplier)
```

This Python script does the following: The first line of code reads the Numerics 'EnableAppendixK' Boolean value. Then the best estimate values are set in local Python variable (not Numerics variables). Next an 'if' statement checks whether the Numerics Boolean 'EnableAppendixK' is True. If so it updates the best estimate values to Appendix K values. Finally the Numerics values are updated.

3. Underneath 'decay_heat_multiplier = 1.0' add the line (to set best estimate power):

power = 2.3E9

```
isAppendixK = GetVariable("EnableAppendixK")

decay_heat_multiplier = 1.0
power = 2.3E9

if isAppendixK == True:
    decay_heat_multiplier = 1.2

SetVariable("Decay_Heat_Multiplier",
decay_heat_multiplier)
```

Both 'decay_heat_multiplier' and 'power' are local Python variables, not Numerics variables. To set the actual Numerics variables, the 'SetVariable' function must be called as shown at the bottom of the script. The SNAP menu includes an 'Insert Example' option that shows you how to get and set Numerics variables from Python.

4. On the line after 'decay_heat_multiplier = 1.2', type 4 spaces followed by:

power = power*1.02

```
isAppendixK = GetVariable("EnableAppendixK")

decay_heat_multiplier = 1.0
power = 2.3E9

if isAppendixK == True:
    decay_heat_multiplier = 1.2
    power = power*1.02

SetVariable("Decay_Heat_Multiplier",
decay_heat_multiplier)
```

Indentation is used in Python to indicate code blocks that belong to 'if' statements, loops, or other code sections. The size of the indentation doesn't matter other than that the statements within a code block have to align, and an unindent indicates one is exiting the code block. Since the 'decay_heat_multiplier = 1.2' statement, inside the if statement, is preceded by 4 spaces, the 'power = power*1.02' statement must also be preceded by 4 spaces to be part of the 'if' block.

5. Below 'SetVariable("Decay Heat Multiplier", decay_heat_multiplier)' add:

SetVariable("Power", power)

```
isAppendixK = GetVariable("EnableAppendixK")

decay_heat_multiplier = 1.0
power = 2.3E9

if isAppendixK == True:
    decay_heat_multiplier = 1.2
    power = power*1.02

SetVariable("Decay_Heat_Multiplier",
decay_heat_multiplier)
SetVariable("Power",power)
```

6. Click on 

to save the changes then click on 

.

7. To enable the Appendix K values, in the Navigator Window, locate and click on "Numerics → Booleans", click on EnableAppendixK, and in the Properties Window set Value | True False. This causes the Numerics function to execute and update the Numerics values to 'Appendix K' values.
8. Save the Model as "PWR1-SS-Numerics.med"

Run a Brief Steady-State Calculation to Verify Power

It is often useful to run a quick simulation when the model is changed to verify that the model is behaving as expected. The model is set up to run a 20 second calculation. A job stream to submit the calculation is included in the model that we are using here. To examine a brief steady-state run do the following:

1. Locate and click on the "Job Stream" tab at the bottom of the View Window.
2. Lock the View Window by clicking on the padlock located on the left-hand side of the Toolbar.
3. Click on the "Execute" button in the View Window. Click on the OK button in the "Submit Stream" popup dialog window.
4. The Job Status window will appear and the calculation will proceed. When the calculation is finished, the graphics file will be demultiplexed (demultiplexing arranges the graphics file in parameter versus time pairs that makes it faster to plot the calculated parameters).
5. After the graphics file is demultiplexed, locate and click on the AptPlot icon at the top of the "Job Status" window.
6. The AptPlot window will appear along with the "Read TRACE" dialog box.
7. Locate and click on the Power component power-999 in the left-hand window of the dialog box. In the right-hand window, locate and click on rpower-999 then click on the "Plot" button at the bottom of the window.
8. Verify that the calculated power is at 2346 MW.