

TRACE/SNAP User Workshop
March 26th - 29th, 2018

System Source Computer Training Center
Columbia, Maryland

SNAP Variables, Job Streams and Post-Processing

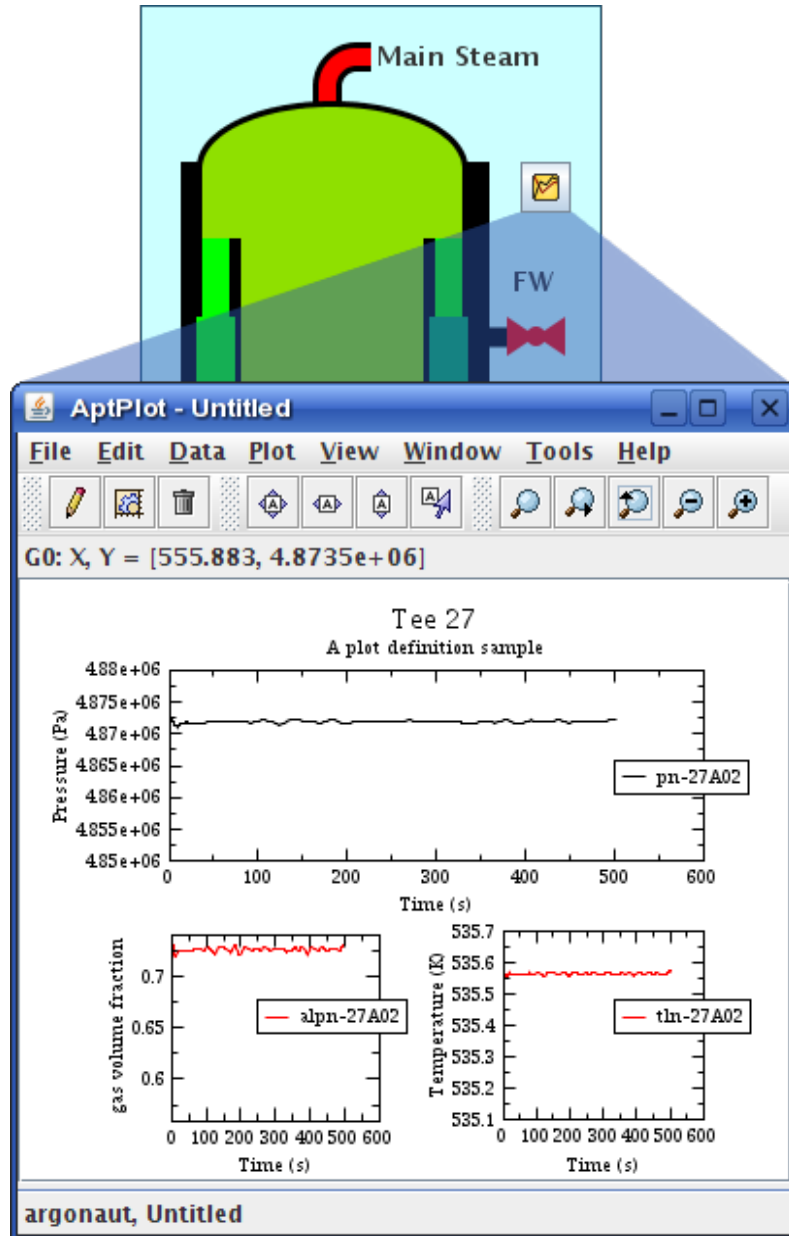
Applied Programming Technology, Inc.



Animation Models

- 2D & 3D Visualization of the results provides insight into the model.
- Interactive or Replay Modes.
- Replay proportional to realtime
- The views can be built from the 2D views of the pre-processor model
- Can include data from multiple runs, different codes or experimental results in the same animation.
 - The Time data obtained from the master run
 - The slave run data is interpolated to match the master time steps
- Color Maps (aka Data Ranges) map data values.
- Python Data Sources – Custom Data Source
- Easily Add Customized Display Elements

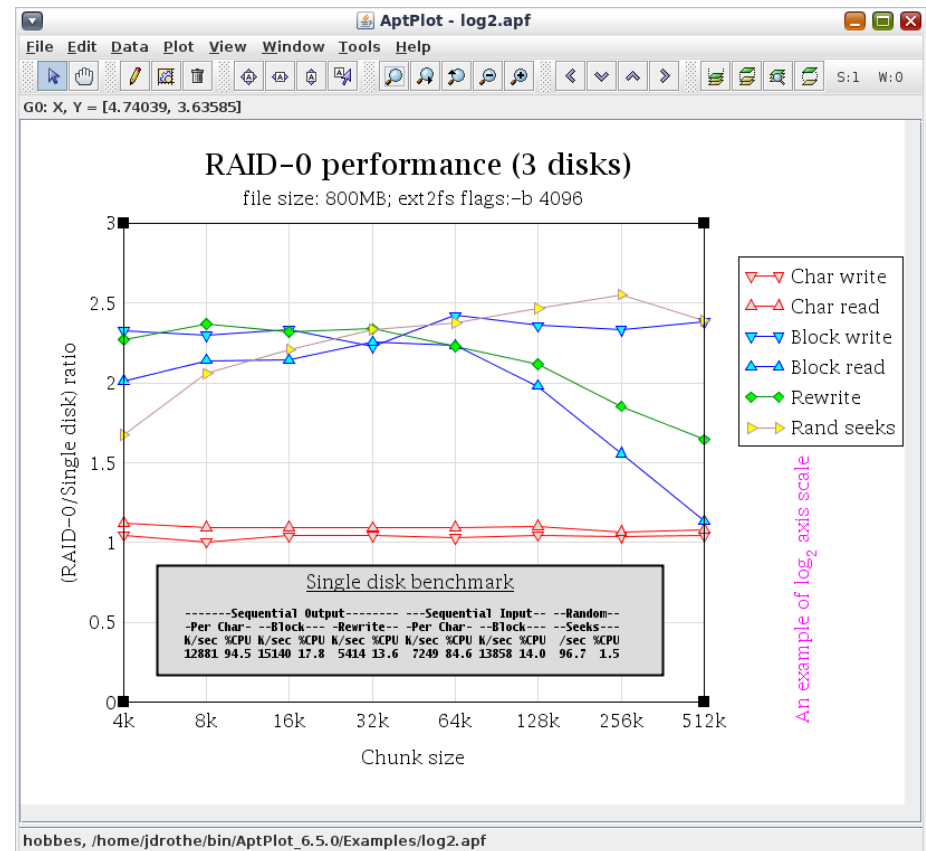
Plot Definitions



- Plot definitions saved with Animation Model
- Defines formatting with a parameter file saved from AptPlot
- Maps channels directly to Graphs and Sets
- Add custom batch commands to tweak the graph
- Drag & Drop Plot Definitions onto Views to create plot launchers.

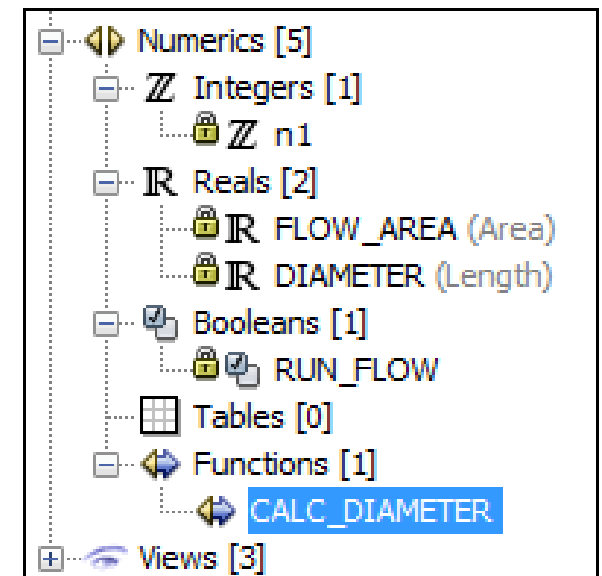
AptPlot

- Pure- Java WYSIWYG Plotting Application
- AptPlot is a descendant of AcGrace & Xmgr5
- Creates Publishing-Quality plots in several output formats:
 - PDF, JPG, PNG, EMF, TIFF, PostScript
- Analysis Code Support Plug-in (ACS)
- All plot properties may be set through batch scripting
all plot properties. AptPlot file format consists entirely of batch commands.



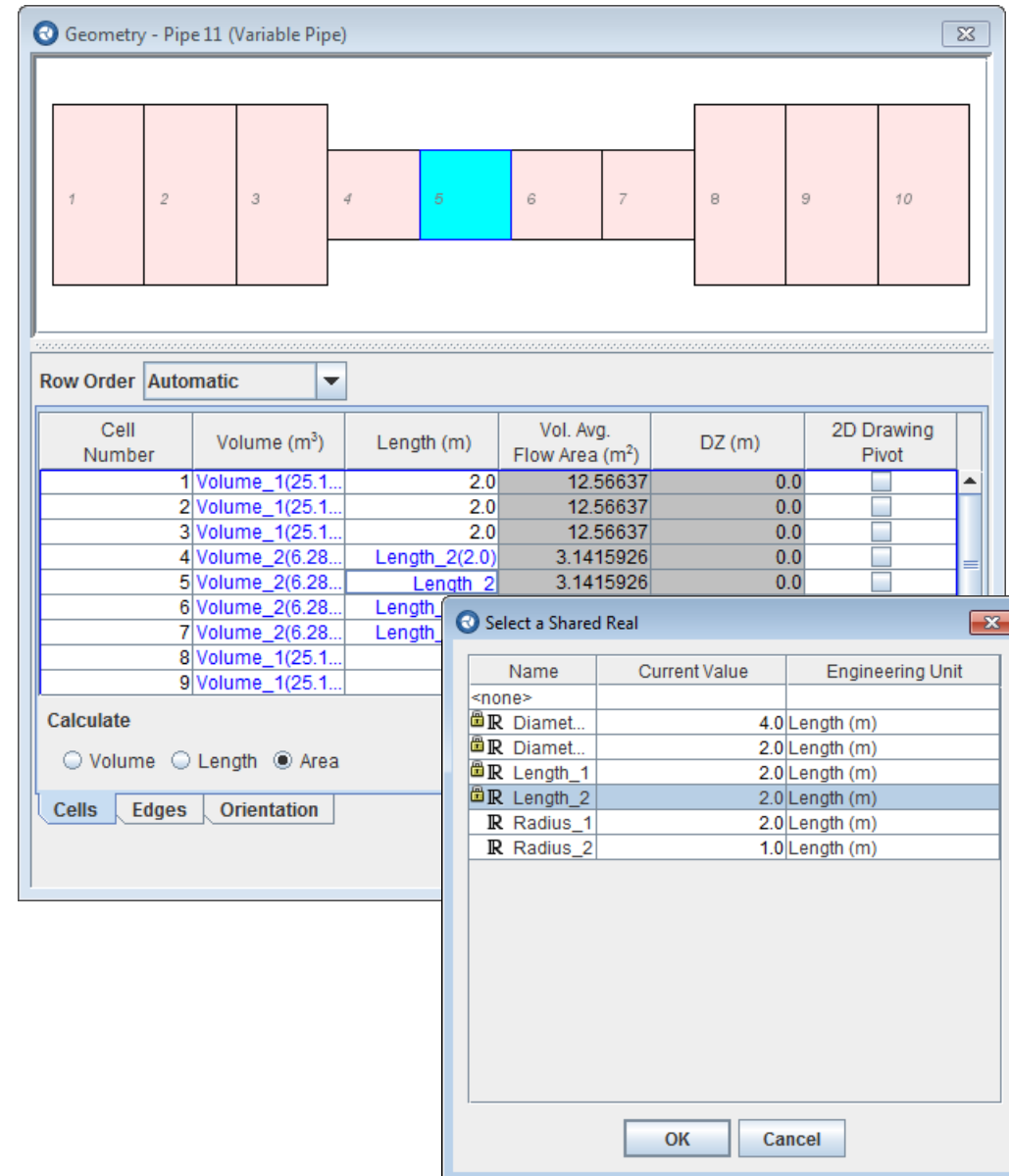
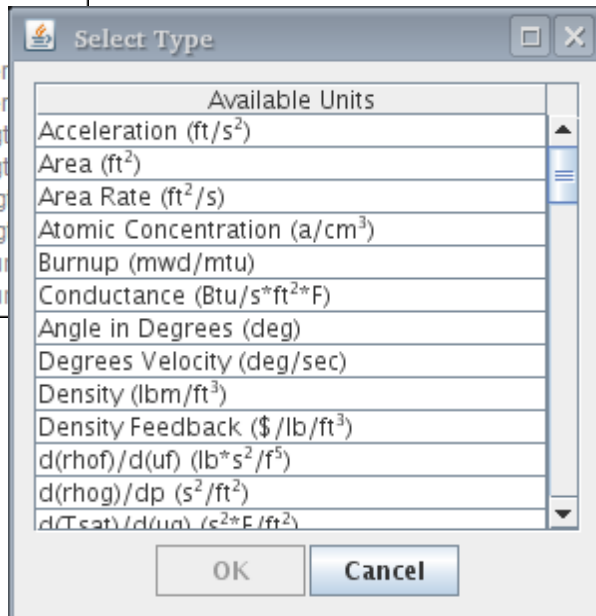
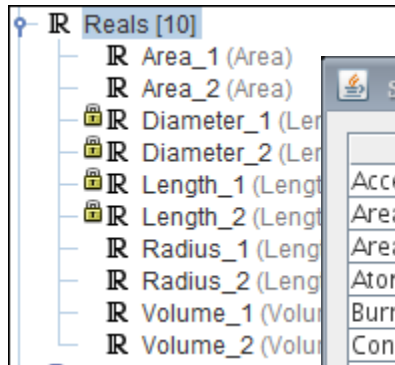
SNAP Variables and Functions

- Variables may be used in place of entering an explicit value for component data.
- Python, MATLAB or MathCAD functions can be implemented to calculate variable values.
- Variables are either Interactive or Calculated.
 - Interactive variable values are entered directly.
 - Only interactive variables may be used as independent variables in job streams.
 - Calculated variables are assigned values by functions.
- Variable Types:
Boolean, Integer, Real, String, Table
- Real variables have engineering units (Area, Length, etc.).
- Variables can be assigned to properties that have the same units.



Real Variables

- Double Precision Floating Point with engineering units
- Values converted automatically when model units toggle between SI and British.



Integer & Boolean Variables

- Integers
 - Single Value
 - Enumerated
 - Select the value from a predefined list.
- Booleans
 - Useful for controlling the execution of functions

The screenshot shows a configuration window for a variable named 'n1'. It has a 'General' tab and a 'Show Disabled' checkbox. The fields are: Name (n1), Description (<none>), Type (Enumeration selected, Single Value unselected), Enumerated Values ([4] Option_1(7), Option_2(1...)), and Value ([7] Option_1 selected).

| Field | Value |
|-------------------|--------------------------------|
| Name | n1 |
| Description | <none> |
| Type | Enumeration |
| Enumerated Values | [4] Option_1(7), Option_2(1... |
| Value | [7] Option_1 |

The screenshot shows a 'Define Enum Entries' dialog with a table of entries. The table has two columns: 'Name' and 'Value'. The entries are: Option_1 (7), Option_2 (13), Option_3 (11), and Option_4 (2). There are 'OK' and 'Cancel' buttons at the bottom.

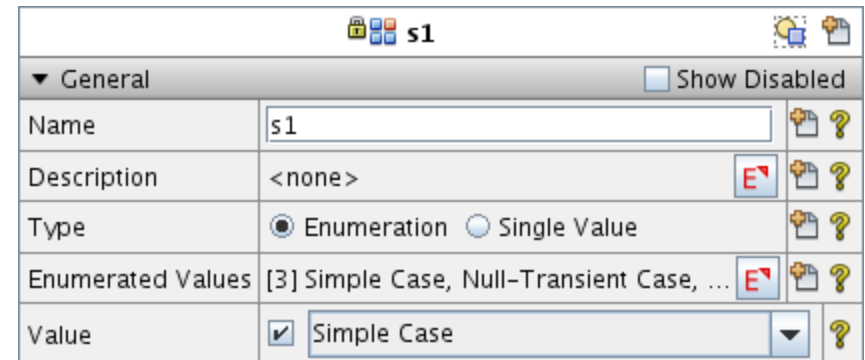
| Name | Value |
|----------|-------|
| Option_1 | 7 |
| Option_2 | 13 |
| Option_3 | 11 |
| Option_4 | 2 |

The screenshot shows a configuration window for a variable named 'ENABLER'. It has a 'General' tab and a 'Show Disabled' checkbox. The fields are: Name (ENABLER), Description (<none>), Interactive Variable (True selected, False unselected), Value (True selected, False unselected), and Parametric (On unselected, Off selected).

| Field | Value |
|----------------------|---------|
| Name | ENABLER |
| Description | <none> |
| Interactive Variable | True |
| Value | True |
| Parametric | Off |

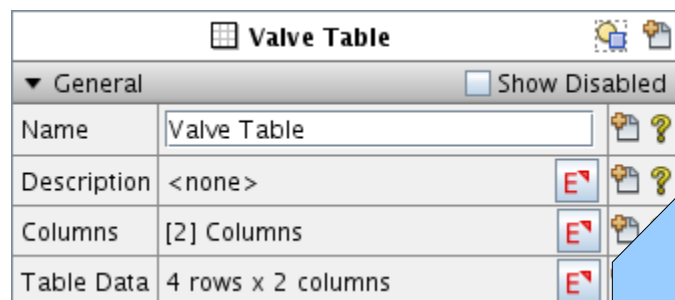
String & Table Variables

- Strings
 - Single Value
 - Enumerated
 - Select the value from a predefined list.
- Tables
 - Columns Define Data Types
 - Real, Integer, Boolean, String
 - Individual Table Values Accessible through functions



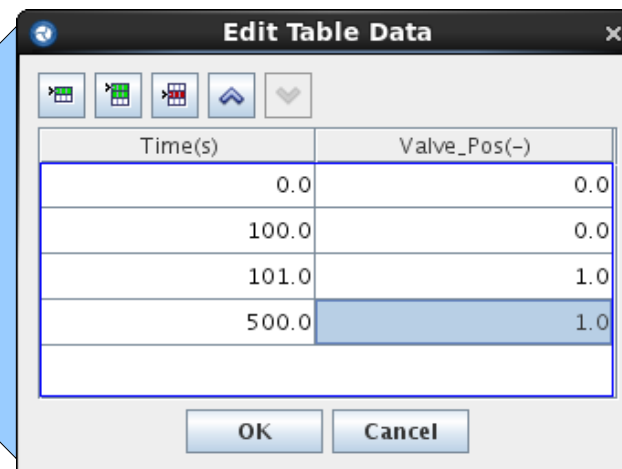
The screenshot shows a configuration window for a variable named 's1'. It has a 'General' tab and a 'Show Disabled' checkbox. The fields are: Name (s1), Description (<none>), Type (Enumeration selected, Single Value unselected), Enumerated Values ([3] Simple Case, Null-Transient Case, ...), and Value (Simple Case selected from a dropdown menu).

| s1 | |
|--|---|
| ▼ General <input type="checkbox"/> Show Disabled | |
| Name | s1 |
| Description | <none> |
| Type | <input checked="" type="radio"/> Enumeration <input type="radio"/> Single Value |
| Enumerated Values | [3] Simple Case, Null-Transient Case, ... |
| Value | <input checked="" type="checkbox"/> Simple Case |



The screenshot shows a configuration window for a variable named 'Valve Table'. It has a 'General' tab and a 'Show Disabled' checkbox. The fields are: Name (Valve Table), Description (<none>), Columns ([2] Columns), and Table Data (4 rows x 2 columns).

| Valve Table | |
|--|--------------------|
| ▼ General <input type="checkbox"/> Show Disabled | |
| Name | Valve Table |
| Description | <none> |
| Columns | [2] Columns |
| Table Data | 4 rows x 2 columns |



The screenshot shows a dialog box titled 'Edit Table Data' with a table of data. The table has two columns: 'Time(s)' and 'Valve_Pos(-)'. The data is as follows:

| Time(s) | Valve_Pos(-) |
|---------|--------------|
| 0.0 | 0.0 |
| 100.0 | 0.0 |
| 101.0 | 1.0 |
| 500.0 | 1.0 |

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

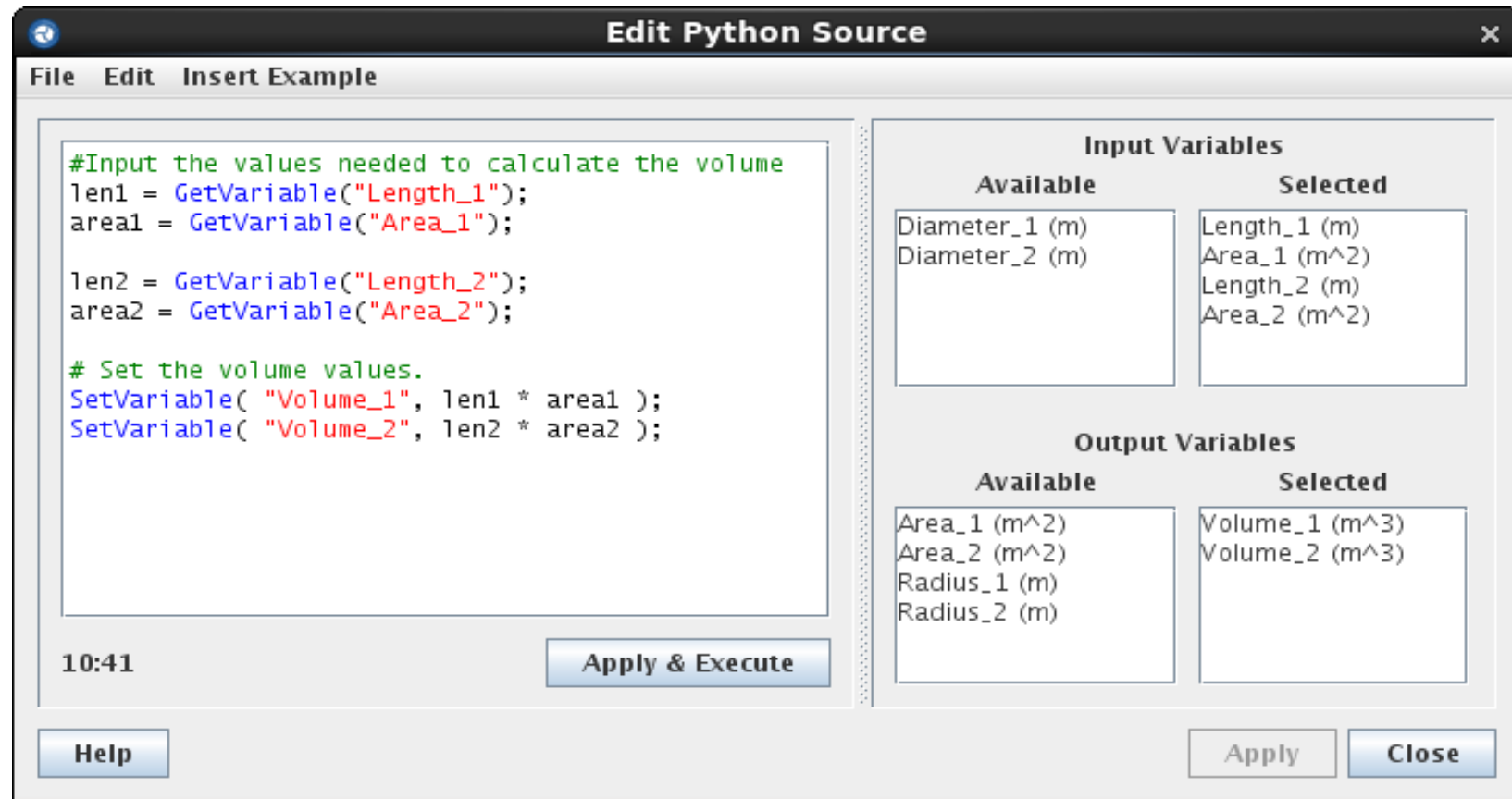
Functions

- Python and MATLAB
- Operate on SNAP Variables
 - Input: Interactive and Non-Interactive Variables
 - Output: Non-Interactive Variables
- Customizable execution order
- Enabling Booleans control execution
- Python - function source embedded in .med file
- MATLAB - reference to .m file
- Application Options- units, console, application window (Mathcad only)

Python Interface

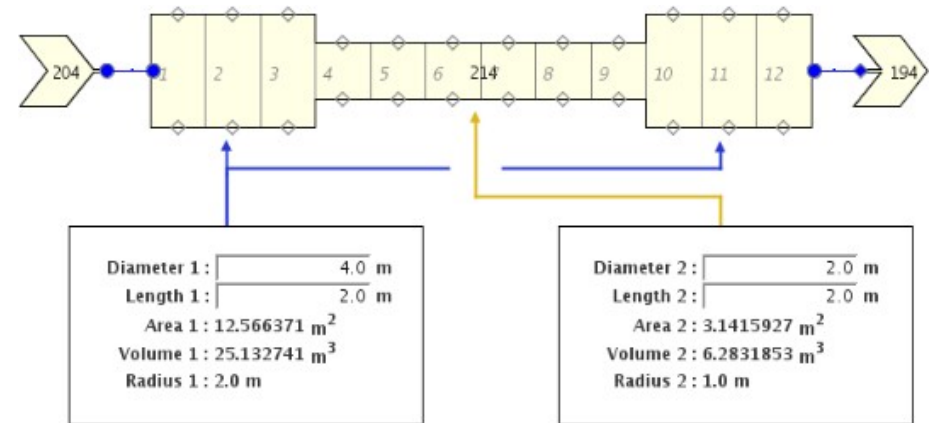
- Python code interpreted via Jython
- Python Function editor used to modify source code
- Input, output variable lists inferred upon function execution
- Custom Python module for accessing and modifying variables
- Python Interface Methods
 - GetVariable("name")
 - SetVariable("name", value)
 - GetTable("name")
 - GetRowCount()
 - GetColumnCount()
 - GetValueAt(row, column)
 - SetValueAt(row, column, value)

Sample Python Function



Variables in Views

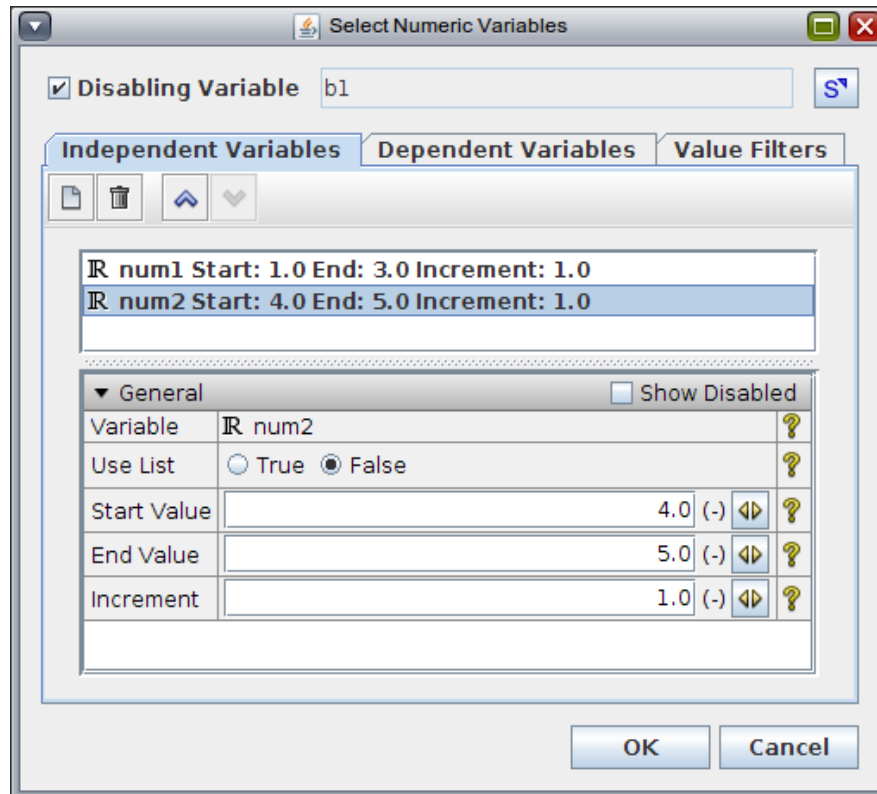
- Booleans, Integers, Reals & Strings
- Drag and drop, or use “Add to View” right-click menu item
- Interactive variables can be set to editable
- Changing a value executes all SNAP Functions
- Enumerated Integers and Strings support drop down lists and radio buttons.
- Values updated after function execution



Parametric Job Streams

- Submit Sets of Calculations
 - Interactive variables modified over a range of values
 - Parametric model nodes generate an input file for each combination of parametric values
- Multiple Types Supported:
 - Numeric Combination
 - Tabular Parametric
- Cases May be Filtered
 - Conditional Logic
 - Explicitly Included/Excluded

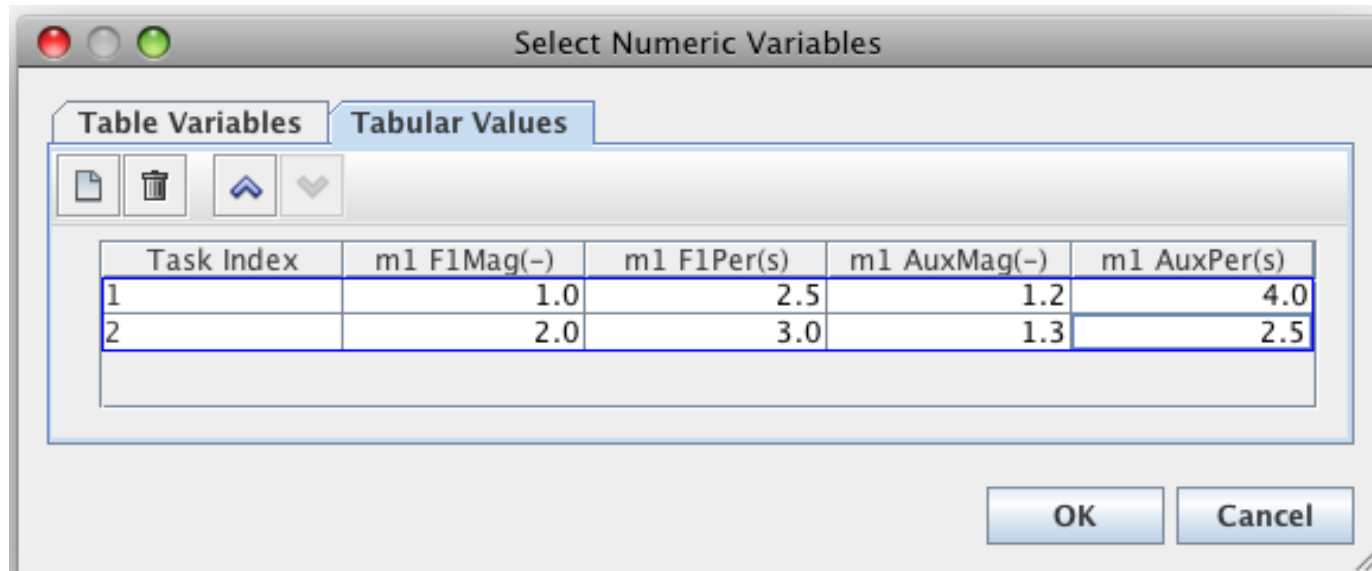
Numeric Combination



- Builds a combination of values from each of the selected independent variables.
- Either an array of values or a start, end and increment set is defined for each variable.
- Each combination of values corresponds to a parametric case.
 - For example: x has 3 values, y has 7 values, the parametric set will contain 21 cases.

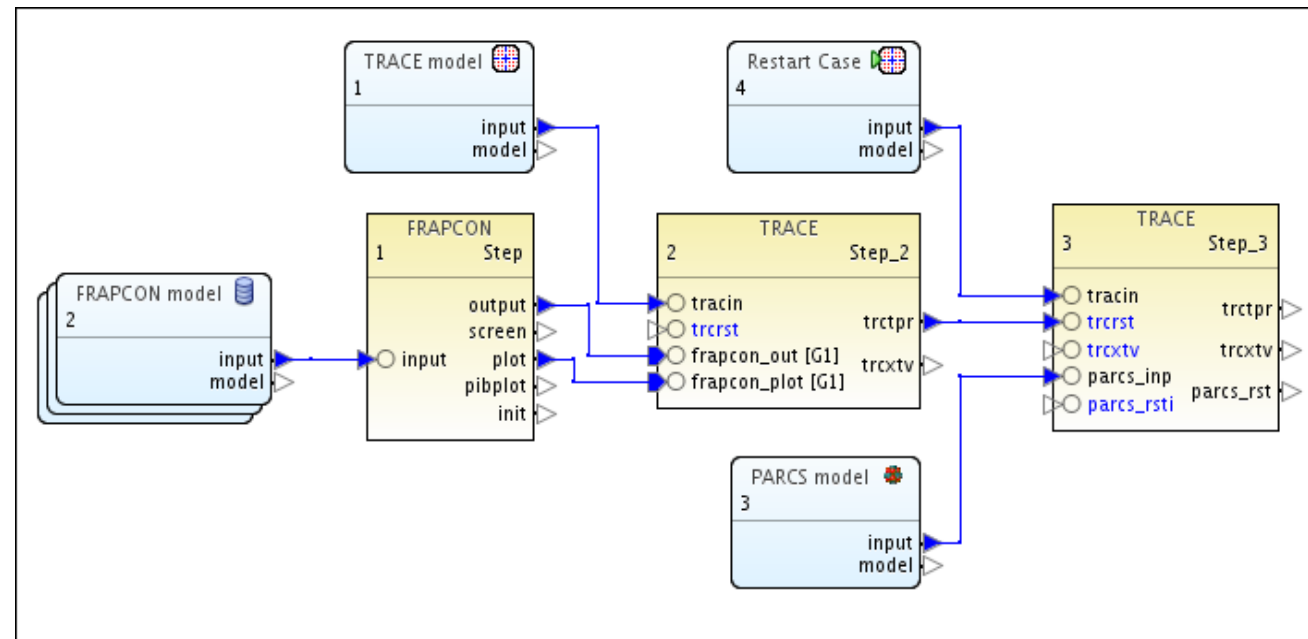
Tabular Parametric

- An explicit table of variables.
- Each row corresponds to a parametric case.
- Each independent variable must have a value in every row.
- The order of the rows can be changed.
- Supports integer, Boolean, real and String variables.



Engineering Templates

- Used to construct complex Job Streams using multiple source models.
- Underlying Source Models are queried for available SNAP variables. These models can be manipulated through variables.
- 'Global' engineering template variables can also be mapped to variables in the underlying models.

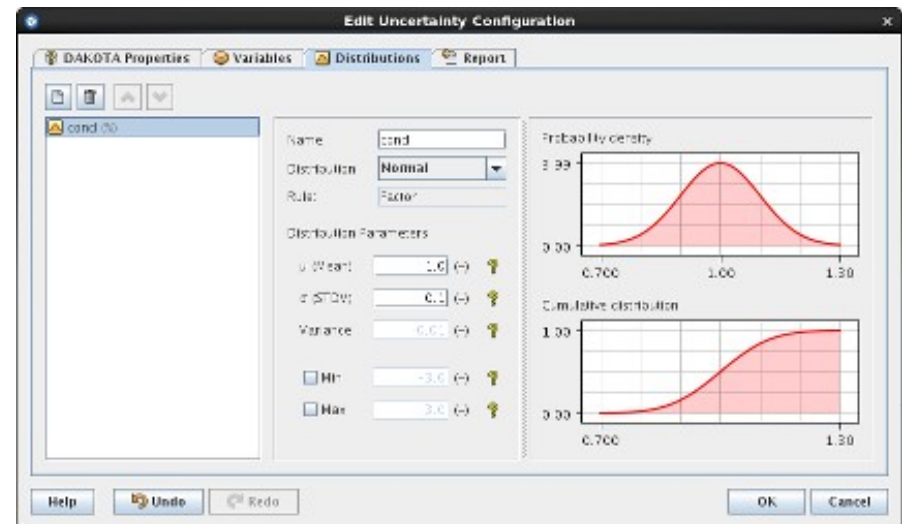


Exercises

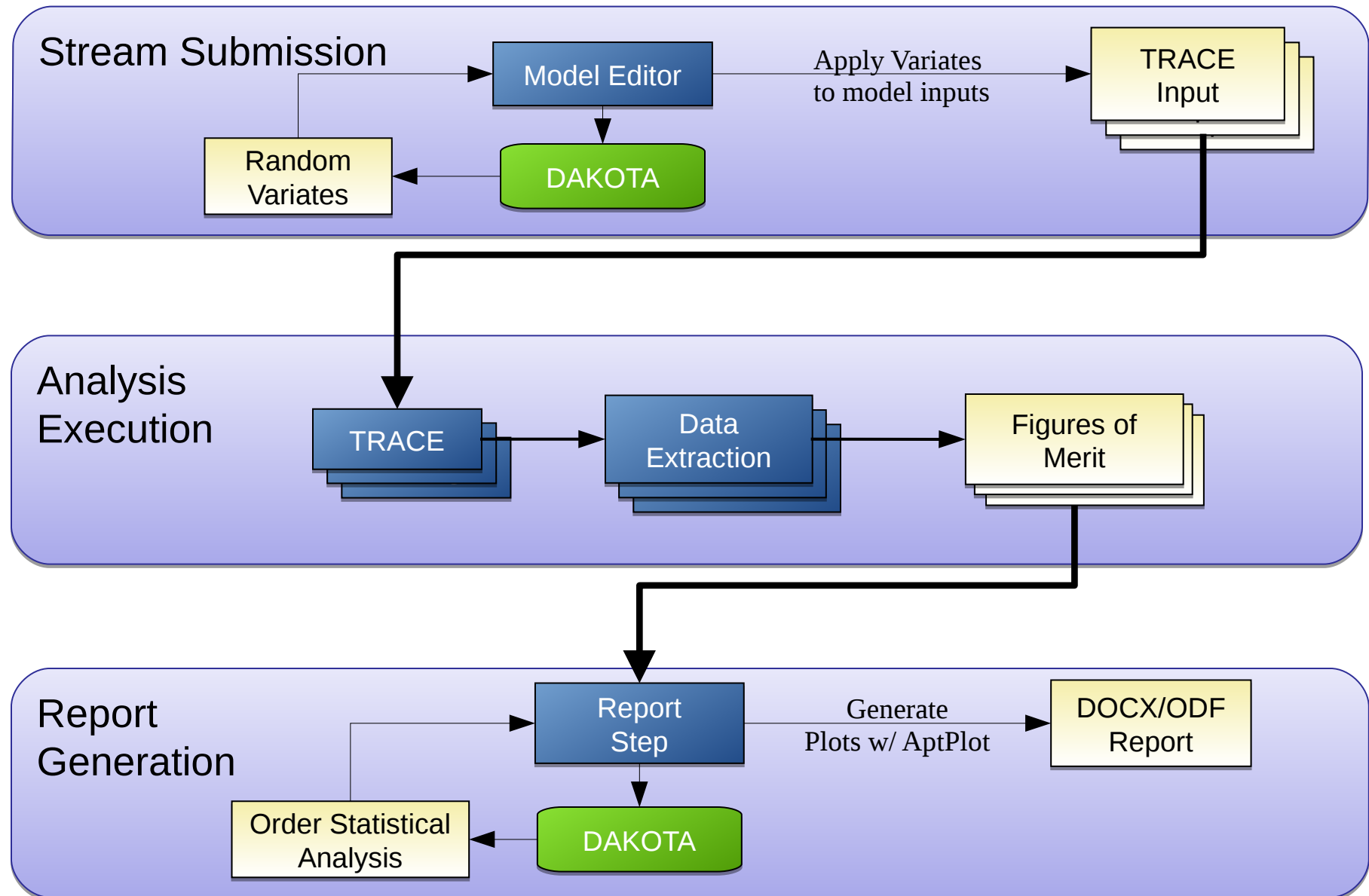
- Exercise 12. SNAP Variables and Parametrics
- Exercise 13. Animating a Model
- Exercise 14. Interactive Controls
- Exercise 15. AptPlot in Job Streams
- Exercise 16. Tabular Parametric and Axial Plotting

Uncertainty Quantification Stream Type

- Assign distributions to sensitivity coefficients and/or variables for UQ Analysis
- Executes DAKOTA to generate randomized input values and calculate results:
 - PDF Types: Normal, Log-normal, Uniform, Triangular, Beta.
 - Monte-Carlo and Latin Hypercube Sampling
 - Random variate can be applied as a factor, an additive value (with engineering units) or a replacement value.
 - Apply to SNAP variables, TRACE Sensitivity Coefficients, or TRACE Tabular Data (Fill, Break, General Table).
- 'Data Extract' Job Step retrieves "Figure of Merit" data from plot files via AptPlot.
- DOCX or ODF Formatted Generated Results Report.
 - Includes AptPlot generated plots of results.



Uncertainty Quantification Job Stream



Model Notebooks

- Supported Formats
 - ODF (Open Document Format)
 - DOCX (Microsoft Word)
- Model Summary Data
- Validation Report Results
- Hyperlinks Across Connections
- Embedded Attribute Level Documentation
 - Model Notes Included Directly
 - Notes can include External Hyperlinks
- Generated Plots for Select Data
- Optional Ownership/Reviewer Data
- Optional Classification Markings
- Optional Sub-System Sections

Export Model Notebook - StandPipe5.med

General Sub-Systems

Title Page: <none>

Front Matter: One Front Matter Note Specified

Classification: ☒ DEMO

Header: ☒ StandPipe Model

Footer: ☒ Advanced Practices Demo

Page Styles: ☐ Left/Right ☒ Single Page

Misc. Mathcad Output Format: Rich Text Format (.rtf)


☒ Include Section Titles ☒ Include Input Listing

☒ Include User Numerics ☒ Include Owner/Reviewer Listing

☒ Open Exported Notebook ☒ Include Component Images

Help Export Cancel

3.20 Pipe 81 (guide tubes - cell 2)

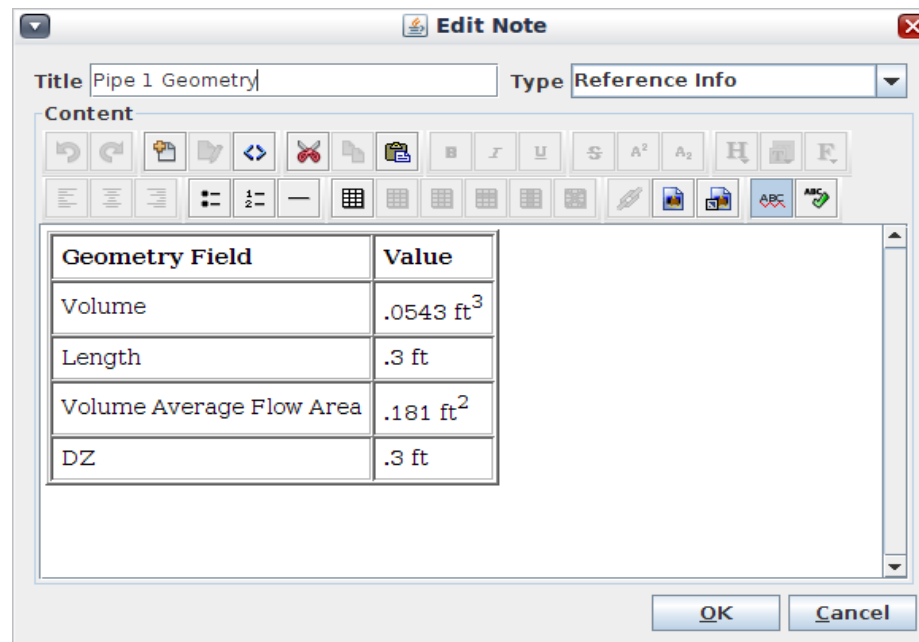


| Name | Value | Refs |
|---------------------|---|------|
| Component Geometry | Cells: 8, Functions: 9 - Valid Values [below] | - |
| Initial Conditions | Valid Values [below] | - |
| Friction | Valid Values [below] | - |
| Fluid Power Options | Not Modeled | - |
| Critical Heat Flux | [1] AECL_IPPE | - |
| Wall Roughness | 0.0m | - |
| Pipe Type | [0] No Accumulator | - |
| Number of Pipes | 1 | - |

Patch Bottom TRACE Sample UNCLASSIFIED 97

Model Notes

- Supports HTML formatting (HTML 3.2)
 - tables, external links, lists, etc.
- Supports linking in or pasting in images.
- Built in spell checking.
- Custom notes can be assigned to any number of components and/or individual attributes.



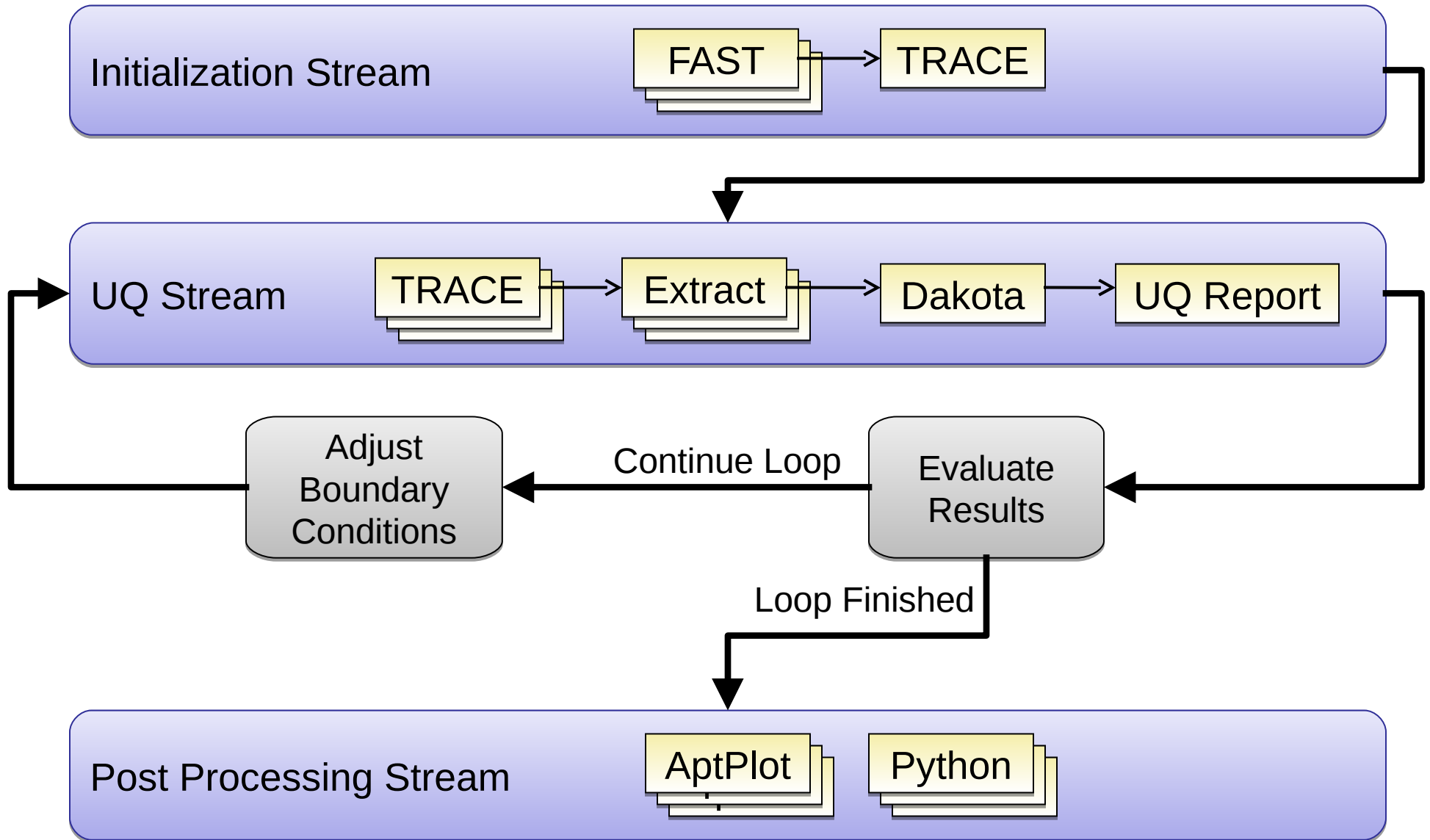
Model Notes – Model Note Viewer

- Toggle between a components view and notes view.
- Components View
 - Lists every component and its attributes.
 - Lists the notes referenced by each attribute.
 - Add / remove / edit notes.
 - Filter list by category, component, and attribute.
- Notes View
 - Lists every note.
 - Filter list by note type, title, contents.
 - Filter by connected component category, component, and attribute.
 - Add / remove / edit notes.
 - Add / remove references from component attributes.

Job Stream Sequences

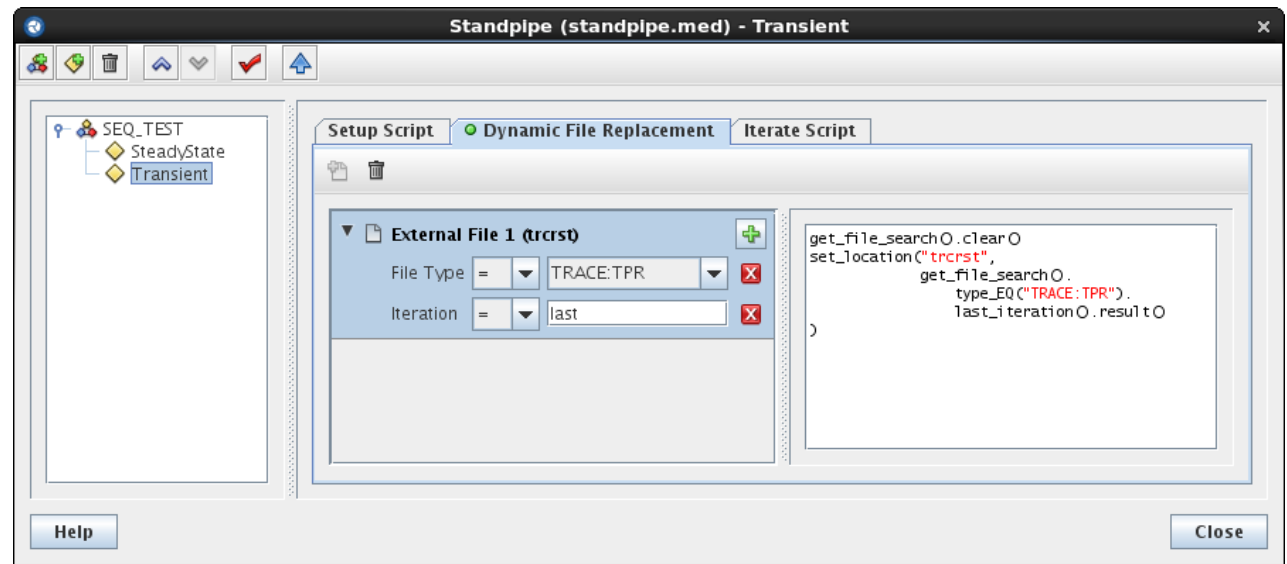
- A set of Job Streams that are executed in order.
- Supports looping with “Setup” and “Iterate” Python scripts
 - Setup – Once per Job Stream
 - Iterate – Once per Job Stream execution (i.e. loop iteration)
 - Retrieve task and file keyword values from completed streams
 - Set SNAP Variables to direct calculation
- Use outputs from one stream as inputs to the next stream in the Sequence using Dynamic File Replacement
- Packages all necessary local files during submission
 - Includes accessory plug-in files such as PARCS Cross-section files
 - MED files.
- Stream Sequences dialog accessible through the Job Streams node in the Navigator.

Job Stream Sequences and Looping



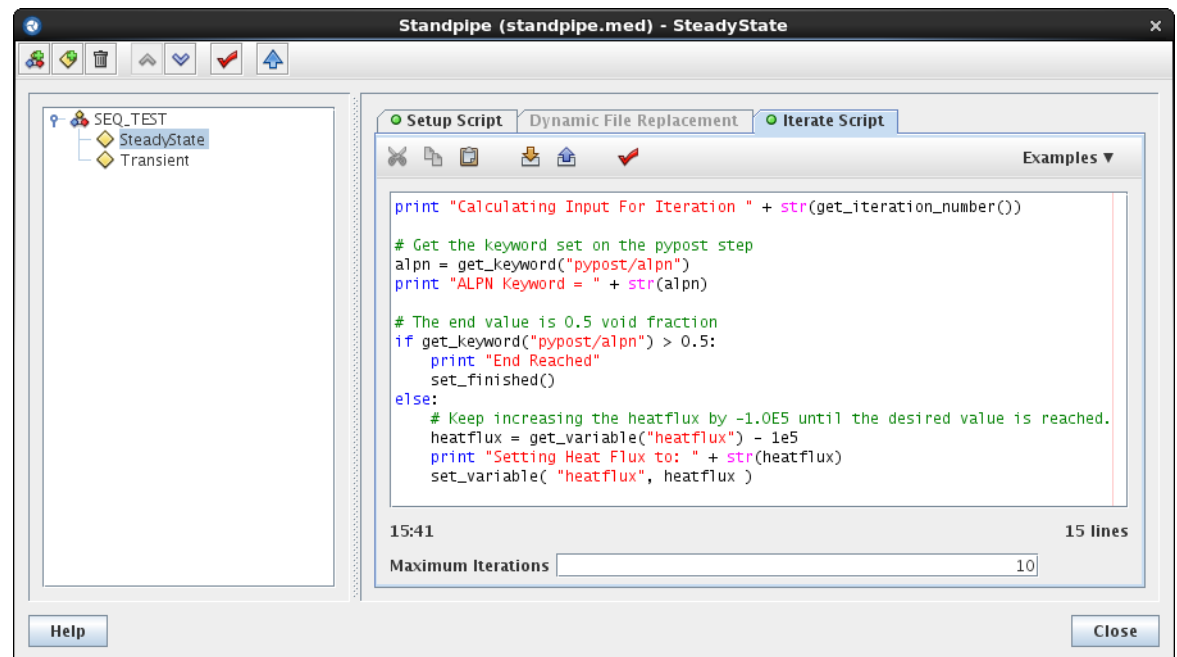
Sequences – Dynamic File Replacement

- Searches the output files from all preceding sequence job streams to define an external file or file set in the selected job stream.
- Add search criteria to limit the results to the desired output file.
 - Criteria include: Stream, Iteration, Step Name, Step Type, File Name, File Type, File Label, Task Keywords, Step Keywords
- Reports an error if the selection criteria resolves to more than one file for a single-file target.



Sequences – Looping

- Python Iteration script controls looping.
The script can steer the calculation by setting variable values and is used to end the loop.
- Maximum Iterations – Prevent Infinite Loops
- Built-in Python Methods
 - Read keywords from preceding streams
 - Set model variables prior to stream execution
 - File searching and replacement
 - End the loop



Exercises

- Exercise 17. Uncertainty Quantification with TRACE and DAKOTA
- Exercise 18. Creating a TRACE Model Notebook
- Exercise 19. Job Stream Sequences

Optional Exercises

- Exercise 20. AptPlot Commands
- Exercise 21. AptPlot Scripting