DYNAMIC ANALYSIS USING THE SCRIPTING MODULE

FluidFlow is a steady-state pipe network analysis program, solving for flows and pressures throughout a pipe system. Its solution is essentially a “snap-shot” in time of the system operating under the conditions pertaining at that instant.

An engineer may wish to optimize the network design by altering pumps speeds, pipe sizes, etc. or view the performance of the network under different conditions such as different demands. Setting the new parameters can be done manually through the input palette but, for a large number of iterations, both the setting of the parameters and the recording of results can become tedious. FluidFlow offers a mechanism to automate the parameter setting, the network calculation, and the recording of results.

The automation is by means of a Pascal- or Basic-type scripting language and the recording of results by means of graphical, tabular and Excel outputs. However, since the scripting language is a comprehensive programming language, it is possible to iteratively solve the network for different parameter values where the parameter values are calculated on a statistical basis or where the parameter values for one iteration are calculated from the results of a previous iteration. This is a particularly powerful feature where we want to model the performance of a network over a time, for example the falling of the liquid levels in a reservoir over time or pump failures.

The Pascal dialect supported in FluidFlow is somewhere between standard Pascal and Delphi. Although the network, output, and elements are exposed as objects, the script user cannot define object classes. Otherwise most other standards types (integer, floating point, string, arrays, and variants) are supported; most of the control structures (including try blocks); and a fairly comprehensive library of standard routines. In general terms, any parameter that can be set through the input palette can be set through a script command.

One of the more useful features of the scripting language is the ability for a script to read values from an Excel spreadsheet and output results to an Excel spreadsheet. Reading values from a spreadsheet is a particularly convenient way of iterating through a large number of parameter values, and outputting to a spreadsheet enables subsequent post-processing of results with Excel.

The FluidFlow installation files come with a number of simple examples of scripting showing how to input from Excel, output to Excel, read and write files, construct result graphs, and access the various exposed objects. However, to write a script to animate a realistic network is a fairly substantial undertaking.

In summary, the scripting module not only automates the setting of parameters such as pump status (on or off) but can also be used to animate a network. It allows an engineer to examine the flows and pressures in a network under various conditions and to examine the behaviour of a network over time under varying demands and failure scenarios.

The applications of the Scripting module are extensive, from the simple - such as sizing an orifice plate to generate a desired pressure drop, to something more complicated - such as simulating the interaction between variable speed pumps and associated control valve positions. Some examples are listed below:
**Aging of a cooling water system:**

Assume a cooling water system will deposit 0.07 mm of scale each year. Heat transfer considerations dictate that once the flow drops below 95% of its clean initial value the system will need cleaning. Looping through a sequence of calculations with increasing levels of scale will calculate the time interval between cleaning.

**Pipe Sizing:**

From past operating experience on similar plants we know that operating with pipe velocities between 1.5 and 3.5 m/s has been successful. Therefore we require a script which highlights all pipes whose velocity falls outside this range.

**Valve Closure and Pressure Loss**

Calculate the pressure drop across a valve with constant flowrate as the valve closes in 2.5% increments and chart the results.

**Troubleshooting of Existing Plant:**

The condition of existing plants is often uncertain. Pipes may be slimed or roughened compared to new conditions; orifice plates may be eroded; valves may be in positions different to those shown by the tell-tales and pump impellers may be worn. Physical investigation such as dismantling pumps is expensive, time consuming and may require shutdown. However, it is often possible to obtain reliable measured flows and pressures at some locations around the plant. Scripting allows the physical investigation to be replaced by a virtual investigation using measured flows and pressures as a guide.

A simple script could run the model with different impeller diameters in order to simulate impeller wear. Model results would then be compared to known data to see if there was any relation. The same approach could be applied to pipe roughening. The objective would be to “calibrate” the model to agree with known site conditions and then use the conditions in the model as a guide to the cause of those conditions.

**Hazard and Operability Safety Studies:**

Determine whether a given/unsafe pressure is exceeded if a pump speed is increased to maximum, or a valve is closed etc. Failure conditions could be simulated such as control valves failing closed.
**Pump Operation**

Controlling pump stops and starts based on system performance or from externally defined conditions:- The example shown below shows a system of pumps and transfer tanks with relevant output to Excel.

The results displayed on the flowsheet show the steady-state condition assuming water depths in each tank are 1.5m. Since pumps are not matched, tank levels vary over time with pumps starting and stopping based on tank high and low set levels. FluidFlow3 simulates this via a purpose-programmed ‘script’ based on a selected time interval and level changes in the tanks. Pumps stop and start accordingly.

Automatic transfer of any input or calculated data to Excel is possible, for instance pump operation times. The chart shows the variation of tank depths over time.

The advantages of a script such as this would be the ability to size tanks efficiently and to determine the number pump starts per hour. The specific requirement of this script was to estimate the “reserve” of water in the system should a pump or pipe fail.

**Accutech offers a Scripting service.**

*Based on your steady-state model and your specification we can write your script for you. Contact us for a quote.*