Pygmalion Modifications

Hope Forsmann

December, 2014

Pygi modifications correct design flaws and support changes to RELAP5-3D.

Background

PYGMALION (or Pygi for short) is a RELAP5-3D user aid program developed at the INL. The Pygi utility program creates a copy of a RELAP5 input file with updated initial condition information. The initial conditions are obtained by performing steady-state calculations, the final results of which are written for subsequent use to an output file, restart file for RELAP5-3D or restart/plot for older RELAP5 versions. Pygi accesses the output file, obtains the final conditions for each component of the system model, and replaces the appropriate cards in the original input file with cards containing the new conditions. The new input file then accurately represents the hydrodynamic state of the problem as it was at the end of the steady-state initialization run. Although it is typically used to create decks with steady-state initial conditions, it can also replace the initial conditions with conditions from any plot time from a transient calculation.

Discussion

Some modifications were made to the utility program to address user problems. The first problem resulted in Pygi writing error messages to the updated input file when the input and plot file had more than 40 volumes associated with a component. A design flaw, which originally was intended to decrease memory usage, was found. In practice, it limited the number of volumes to 40, however RELAP5-3D supports up to 99 volumes. Since memory is less of a concern with today's computers and in order to simplify the program, an upper limit of 100 is now used at all times.

The second problem was due to changes in the RELAP5-3D Fortran 90 machine dependent binary plot file format. The binary format written by RELAP5-3D was altered per customer request and Pygi was not updated to match those changes. Pygi and RELAP5-3D are now in sync and Pygi can correctly process the machine dependent binary plot file format for Fortran 90 versions of the code (V 4.x.x and newer).

Pygi has not been through a formal verification and validation process. Therefore, the user must verify that Pygi has correctly transferred the final conditions to the input file. This verification can be accomplished by running a null transient, in which all boundary conditions are held constant, with the updated input file. If the hydrodynamic conditions remain steady during the null transient, Pygi has been successfully used to generate a steady-state input deck. If the hydrodynamic conditions are not acceptably steady, Pygi can be

used again or transients can be initiated with restarts. The use of Pygi is more convenient, but restarts are more accurate for some cases. Engineering judgment is required to determine what is acceptably steady