Windows 7 and PYGIL Development for 4.0.3

Hope Forsmann and Dr. George Mesina

RELAP5 International Users Seminar
Date: Oct 23-24, 2012
Introduction: Windows Adaptation of 4.0.3

• Operating System Situation
• Visual Studio
• Compilation
• Execution
Windows Operating System Progression

- Windows XP support ends @ INL in Dec 2012
- Windows VISTA is unpopular
- Windows 7 is fully supported at INL and elsewhere
- Possible methods of building RELAP5-3D on Win 7
  1. Adapt old RELAP5-3D Windows XP Makefiles
  2. Apply Visual Studio (MS Makefile w/ GUI)
  3. Use RELAP5-3D Linux install scripts on CYGWIN
Selection of Installation Method

• Use Makefiles and the nmake utility for pre Windows 7 installation
  – Makefile issues
    • INL IT has no support for nmake
    • Constant issue updating non-Linux Makefiles and auxiliaries

• CYGWIN, a Linux environment for Windows platforms
  – Will allow use of Linux Makefiles, no update issue
  – executables run only within CYGWIN environment
    • unacceptable for most Windows RELAP5-3D users
    • XDR/PIB file header problems

• Visual Studio is the native Windows development environment
  – There is some support for it at INL
  – There is some support for using it with C++/Fortran
Visual Studio (VS) Solution

- A VS solution is a collection of projects and configuration options
  - Projects contain links to source files and configuration options
  - Configuration options include build order, include paths, etc…
  - Projects either build libraries (like the environmental library) or executables (such as the fluid property generators or relap5.exe)

- Generic build template
  - Version folder, e.g. r3d403ie
  - XDR = eXtended Data Representation libraries
  - RELAP3D_WIN = VS solution for RELAP5

- RELAP3D_WIN solution currently has up to 31 projects
  - relapfiles: All folders & source code files for the RELAP product
  - Project folders: one folder for each project of RELAP3D_WIN
# Visual Studio Projects

<table>
<thead>
<tr>
<th>Libraries &amp; Programs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modules</td>
<td>Created from files in envrl and relap5 folders</td>
</tr>
<tr>
<td>envrl (Environmental)</td>
<td>Creates envrl.lib from files in envrl and modules</td>
</tr>
<tr>
<td>Relap5</td>
<td>Creates relap5.exe from files in relap5, modules, and envrl.lib</td>
</tr>
<tr>
<td>sta2b</td>
<td>Creates sta2b.exe. It takes an ASCII input file and outputs an XDR format tpf-file and an ASCII .pr file</td>
</tr>
<tr>
<td>stgXYZ</td>
<td>Creates stgXYZ.exe from stgXYZ project for fluid XYZ. Used to create the fluid property table tpfXYZ.</td>
</tr>
</tbody>
</table>
Visual Studio Fluids Projects

- The following fluids project names correspond to a folder in relapfiles
- These produce a .exe executable file named after the project
- Issue with XDR tpf – Linux versions work

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Location in relapfiles directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>stgli</td>
<td>Lithium</td>
</tr>
<tr>
<td>stglipb</td>
<td>LiPb</td>
</tr>
<tr>
<td>stgms1</td>
<td>Stgms1</td>
</tr>
<tr>
<td>stgms2</td>
<td>Stgms2</td>
</tr>
<tr>
<td>stgms3</td>
<td>Stgms3</td>
</tr>
<tr>
<td>stgms4</td>
<td>Stgms4</td>
</tr>
<tr>
<td>stgn2</td>
<td>Nitrogen</td>
</tr>
<tr>
<td>stgna</td>
<td>Sodium</td>
</tr>
<tr>
<td>stgnak</td>
<td>NaK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Location in relapfiles directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>stgbipb</td>
<td>BiPb</td>
</tr>
<tr>
<td>stgblood</td>
<td>Blood</td>
</tr>
<tr>
<td>stgd20</td>
<td>D2O</td>
</tr>
<tr>
<td>stgdowa</td>
<td>DowThermA</td>
</tr>
<tr>
<td>stgglyc</td>
<td>Glycerol</td>
</tr>
<tr>
<td>stgh2</td>
<td>Hydrogen</td>
</tr>
<tr>
<td>stgh2o</td>
<td>H2O</td>
</tr>
<tr>
<td>stghe</td>
<td>Helium</td>
</tr>
<tr>
<td>stgk</td>
<td>Potassium</td>
</tr>
</tbody>
</table>
Build Order

• Build order is one of the configuration properties
• Projects that depend on modules or libraries must be built after
• Build order
  – XDR
  – cPibAccess
  – Modules
  – Pib
  – Envrl
  – All fluid projects
  – Relap5
• Executable product installation includes licensing in the build order
Configuration for C-Language coding

• C Libraries
  – The cPibAccess, license, and XDR projects are static C libraries.
  – The libraries use Multi-threaded (/MT) Runtime Libraries and do not use Precompiled Headers.
  – All other options are default selections for a static C library project.

• C Executable programs (E.g. licensing)
  – The executable programs are built using the C static libraries.
  – The executables use Multi-threaded (/MT) Runtime Libraries and do not use Precompiled Headers.
  – All other options are default selections for a C application (.exe) project.
Configuration for Fortran coding

- FORTRAN Libraries: Modules, envrl.lib, relap.lib, and pib.lib
  - All four are static libraries
- FORTRAN Executable programs (E.g. fluids, relap5.exe)
- All of the FORTRAN programs use the following compiler options:
  - Preprocess Source File is turned on and the preprocessor definitions are located in a file called define.inc RELAP3D_WIN directory.
  - The define.inc file is included in the Additional Options: @..\define.inc.
  - Data values are set to Default Integer KIND = 4, Default Real KIND = 8 (/real_size:64), and Default Double Precision KIND = 8.
  - Run-time Check Array and String bounds are set to Yes (/check:bounds)
Building a RELAP5-3D Executable from Source

• Need the following:
  – VS 2010 or newer
    • *Note*: VS projects are not backwards compatible
  – INTEL Parallel Studio XE 2011
    • Or compatible FORTRAN and C++ compiler
  – A copy of the RELAP5-3D source code
  – A copy of the RELAP3D_WIN solution
  – A copy of the windows XDR libraries
Status

- Windows 7 installation complete for version 4.0.3 using
  - Visual Studio 9.0 (VS 2010)
  - Intel Fortran/C++ 12.0
- A generic build solution is available for those with source code
- The standard installation problems run with no errors
PYGMALION  A RELAP5-3D User Aid Program

- PYGI (Pygmalion) purpose
- PYGI Upgrade Goals
- Current Status
PYGMALION Purpose

- Primary purpose is to move information from a steady-state restart-plot file into a new input file
  - **Control Volume data moved**: pressure, liquid specific internal energy, vapor specific internal energy, and void fraction for each control volume
  - **Junction data moved**: PYGI writes liquid and vapor velocities as the initial condition
  - **Component data moved**: Pump, compressor and turbine info
  - **Control variables data moved**: control variable calculated value
- PYGI is not restricted to steady-state data
  - Used to create a new starting deck, partway through a transient, for a parameter study
Necessity for Upgrade and Goals

- PYGI was designed to work with a **combined** restart-plot file
  - Works for versions up to RELAP5-3D/2.4
- Must be backward compatible to versions 2.4 and lower
  - Must still read old-style restart-plot file
- PYGI does NOT work with RELAP5-3D products from 3.0 up
  - Conversion for Fortran 95 split restart-plot file into two separate files: restart file and plot file
  - The data PYGI needs resides in the plot file
- For RELAP5-3D/3.0+, PYGI must read the plot file
  - The plot file has 3 different formats:
    - ASCII
    - Machine-dependent Binary
    - Machine Independent Binary (XDR)
Under the Hood

• Read method for RELAP5-3D input file remains same

• Four read methods for “steady-state” plot data
  – Original combined restart-plot file (BACKWARD COMPATIBLE)
  – ASCII plot file
  – Machine-dependent plot file
  – Machine Independent plot file
    • Machine Independent plot files, PYGI uses the PIB modules from RELAP5-3D distribution
    • PIB library is RELAP5-3D specific interface to industry standard machine independent XDR library

• Steady-state data from each read method is transferred into a single common set of arrays
  – This unifies the real and integer “kind parameter”
  – The output file is written from these arrays
PYGMALION Flowchart

Read RELAP5-3D input file

Determine “steady-state” file-type

Read “steady-state” data

restart-plot Transfer

ASCII Transfer

Machine dependent Transfer

Machine Independent Transfer

Copy old input file substituting steady-state data into new RELAP5-3D input file
Invoke PYGI from the Command Line Prompt

- Example PYGMALION command line (for RELAP5-3D version 2.4).

  ```
  prompt:> PYGI -r restartPlot.plt -t time < origInput.i > newInput.i
  ```

<table>
<thead>
<tr>
<th>Flag</th>
<th>Input Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>origInput.i</td>
<td>original input file used to create “steady-state”</td>
</tr>
<tr>
<td>&gt;</td>
<td>newInput.i</td>
<td>new input file with “steady-state” data</td>
</tr>
<tr>
<td>-r</td>
<td>file</td>
<td>“steady-stated” (restart-)plot file</td>
</tr>
<tr>
<td>-t</td>
<td>val</td>
<td>floating point time of data for creating new input file</td>
</tr>
<tr>
<td>-e</td>
<td>frac</td>
<td>fraction of maximum normalized truncation error</td>
</tr>
</tbody>
</table>
| -F   | fmt = a, m, or b | a for ASCII  
m for machine dependent  
b for machine independent (XDR) |
PYGI Command Line Prompt (continued)

- Example PYGMALION command line (for RELAP5-3D version 4.0)

  `prompt: > PYGI -r restartPlot.r -t time -B -C -F a < origInput.i > newInput.i`

More command line options

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-B</td>
<td>Forces PYGI to examine records that begin with a blank</td>
</tr>
<tr>
<td>-C</td>
<td>Prevents control variable initial conditions from changing</td>
</tr>
<tr>
<td>-m</td>
<td>PYGI usage</td>
</tr>
</tbody>
</table>

Obsolete options

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-l</td>
<td>Indicates the original input deck is from RELAP5/MOD1</td>
</tr>
<tr>
<td>-O</td>
<td>Output (new input file) formatted for RELAP5/MOD1</td>
</tr>
</tbody>
</table>
prompt: > PYG1 -r ctest1.plt -t 1.0 -F m < ctest1.i > ctest1m.i
scanning old deck
attempting to find i c data for:
  26 control volumes
  30 junctions
  1 pumps
  0 compressors
  0 turbines
  0 motor valves
  13 control variables

the rstplt file provides data for:
  30 control volumes
  31 junctions
  1 pumps
  0 compressors
  0 turbines
  2 valves
  16 control variables
  1395 time  1

processing data from plot record
  at  1.00  seconds

80 card replacements attempted

80 card replacements completed

0 $pygmsg messages written into newdk
Thank You!
Testing and Comparison

- Pygmalion input files must pass RELAP5-3D Input Processing inspection and run

- Successfully tested with several sample input problems including:
  - Edward’s Pipe
  - Typical PWR
  - cstest1

- Effect of plot-file format
  - Some values are rounded in ASCII format and truncated in machine Independent XDR format
  - Resulting new input files differ in some of the last significant digits
    - This is to be expected
    - Neither format is clearly better
Conclusions

- RELAP5-3D/4.0.3 builds and runs on Windows 7 Environments
- The build procedure employs MS Visual Studio and Intel Fortran 12
- Issue with XDR formatted tpf fluid files

- PYGMALION has been upgraded to handle:
  - Original combined restart-plot file (BACKWARD COMPATIBLE)
  - ASCII plot file
  - Machine-dependent plot file
  - Machine Independent plot file
- It successfully builds correct RELAP5-3D input files
- The format of the plot file affects the last significant digit of the data placed into the new input file
- It runs on Linux Operating Systems