Tabulating NIST REFPROP Fluid Data using the Cavitation Tabulation Tool

Christopher Neal, Jeffrey Wright, Siddharth Thakur

Produced by: Streamline Numerics
Date: 8/16/2023

1 Table of Contents

2 Thermophysical Data Tabulation Tool

A Python tool is available with the `cavitation_nist` module for building the necessary liquid, vapor, and saturation input files. The Python tabulation utility has a functionality for making calls directly to *REFPROP 10* was added. This allows for the tool to tabulate as many pressure levels as the user desires. The user specifies the *REFPROP* material name, the pressure range, temperature range, and the number of data points to use over each range. The user can also pass a flag (`--log_p`) to uniformly distribute the pressure points in the log-space to ensure that an equal number of points resolve each pressure decade (this is good for large pressure ranges).

A set of `.liq, .vap`, and `.sat` file are generated using the tool, and these are the inputs used by Loci-Stream. The *Python* utility is located in the directory in the `cavitation_nist` module. Sample calls to the utility for the liquid/vapor files and the saturation files are given below.

There are two modes that the tool can be used in. One is for tabulating properties outside of a two-phase region in the thermodynamic space. The other is for tabulating the saturation curve properties of the liquid and vapor states.

Example command for liquid/vapor file tabulation:

- python cav_tab_tool.py liquidVapor --fluid_name NITROGEN --t_min 65 --t_max 300 --p_min 50000 --p_max 70e6 --log_p --num_p 200 --num_t 200

In the example above, the `liquidVapor` argument sets the tool into the mode for tabulating `.liq` and `.vap` tables. The `fluid_name` argument must match an available *REFPROP* FLD file in the REFPROP fluid database. The tabulation levels are done at constant pressure, and so the `t_min` and `t_max` arguments are the temperature range over which to tabulate properties. The `p_min` and `p_max` arguments are the inclusive bounds of the pressure range over which you want to tabulate data. The `log_p` argument converts the pressure range into a logarithmic range and distributes points equally in that space and then converts back to pressure levels. This ensures an equal number of pressure levels per decade of pressure. This is sometimes a desirable option. The `num_p` and `num_t` arguments are the number of pressure levels to tabulate and the number of temperature levels to tabulate.

Example command for saturation file tabulation:

- python cav_tab_tool.py saturation --fluid_name NITROGEN --t_min 65 --t_max 300 --num_t 150

In the example above, the `saturation` argument activates the saturation curve tabulation mode of the script. This mode has fewer arguments than the `liquidVapor` mode. The temperature range and number of points to include are the only arguments that are necessary. These arguments have the same meanings as they do for the `liquidVapor` mode of the tabulation tool that was discussed earlier.

3 Obtaining & Compiling REFPROP Dependency

This tool utilized the *REFPROP* software to perform the thermodynamic and transport property tabulations for the `cavitation_nist` module. The steps for obtaining & compiling REFPROP for use with the tool are shown below.

1. Download the *REFPROP* software using their downloader (must be done on a Windows machine)
2. Copy the REFPROP software over to your Linux machine (can be put in a location such as `/home/<User>/software/refprop/REFPROP`, where `<User>` would you be your username on the machine). **Note: do not rename the actual source code *REFPROP* folder because that might cause issues.**

3. Compile a local version of *REFPROP* on your Linux machine. First open a console and go to your `refprop` directory.
   A. `cd /home/<User>/software/refprop`
   B. Do a recursive clone of this repository (e.g. `git clone --recursive https://github.com/usnistgov/REFPROP-cmake.git`)
   C. `cd REFPROP-cmake`
   D. `mkdir build`
   E. `cd build`
   F. `cmake .. -DCMAKE_BUILD_TYPE=Release -DREFPROP_FORTRAN_PATH=/home/<User>/software/refprop/REFPROP/FORTRAN`
   G. `cmake --build .`

4. Once the compilation is complete, a `librefprop.so` file will be in the build directory. That is the shared library that contains all the *REFPROP* functions that the *Python* wrapper will call. Take that file and move it to the *REFPROP* source directory.
   A. `cp /home/<User>/software/refprop/REFPROP-cmake/build /home/<User>/software/refprop/REFPROP`

5. An environment variable `RPPREFIX` needs to be set which points to the REFPROP source directory that contains the shared library file. We suggest putting the path to the REFPROP source in your `.bashrc` file as such:
   A. `export RPPREFIX=/home/<User>/software/refprop/REFPROP`
   B. `source ~/.bashrc`

6. Clone the *REFPROP* wrappers directory into your /home/<User>/refprop directory (**Note: Not the actual source REFPROP directory**)
   A. git clone https://github.com/usnistgov/REFPROP-wrappers

7. It is also helpful to create a Python3 environment that has the *REFPROP* interface installed (portable, dependable).
    A. `cd /home/<User>/software/refprop`
    B. `python3 -m venv refprop-env`
    C. Enter that environment using: `source refprop-env/bin/activate`
    D. cd `/home/<User>/software/refprop/REFPROP-wrappers/wrappers/python/ctypes`
    E. `python setup.py install`
    F. This installs the Python wrapper to into your Python environment (You also need to install numpy using: `pip install numpy` for the wrapper example script to run).

To leave the environment just type: `deactivate`

One you are in the python environment, you can run the tabulation script and it should execute using calls to the *REFPROP* library.