TP\_PROPS Library

Jack Chessa, jfchessa@utep.edu

***Overview:***

TP\_PROPS Library is a set of Fortran90/77 routines to supply various thermophysical properties of materials employed in cryogenic propellants. Most notably Liquid Oxygen (LOX) and Liquid Methane (CH4).

***Compiling the Libraries:***

The library requires a fortran 90 compiler such as gfortran or intel fortran. The code build is rather straightforward and uses a simple makefile.

On most windows and linux based computers simply changing the F90C and F77C macros definitions in the Makefile should suffice.

***Subroutines Available:***

Currently the following subroutines are available:

* subroutine eos( T, P, mix, rho, drho, mu, dmu, kappa, dkappa, gamma, dgamma, cp, dcp ) – Computes the density, viscosity, thermal conductivity, specific heat ratio and isobaric specific heat for a given temperature, pressure and LOX, CH4 mixture, as well as the gradients of these quantities with respect to temperature (needed for calculation of the Jacobian terms) (currently, 100% LOX)
* module o2props\_mod, which contains the following subroutines (functions)
* O2\_RHO(T,P) - computes density [lbf-s2/in4]
  + O2\_MU(T,P) – computes dynamic viscosity [ lbf-s/in2]
  + O2\_CP(T,P) – computes isobaric specific heat [ in2/(s2-R) ]
  + O2\_GAMMA(T,P) - computes specific heat ratio (cp/cv) [dimensionless]
  + O2\_KAPPA(T,P) – computes thermal conductivity [in2/(s-R)]

***Units:***

These routines all use the following fundamental units:

* Length (L) = inches
* Time (T) = seconds
* Mass (M) = slinch (lbf-s2/in)
* Temperature (θ) = Rankine (·R)

With the following secondary units:

* Force (F) = lbf
* Energy (J) = lbf-in
* Pressure (P) = psia

So the derived units of interest are as follows:

* Density (M/L3) = (lbf-s2/in4)
* Dynamic viscosity (F-T/L2) = (lbf-s/in2)
* Specific heat (J/(M-θ-T)) = in2/(s2-R)
* Thermal conductivity (J/(M- θ)) = in2/(s-R)

***Adding New Properties or Materials:***

This routine is definitely a work in progress. Obviously, more materials are needed to be added and possibly the range and fidelity of the data for O2 may need modification. This can be done relatively easily with the spreadsheet and Matlab routines in the data directory. The routines work from a lookup table approach. To reasonably capture the discontinuities in the thermophysical properties at the phase change we use the following approach: Store the properties at a fixed pressure for a range of temperature. At the phase change temperature temperatures in close proximity to the phase change temperature are used to capture the discontinuity. Then a series of these data points are defined a range of pressures. Then the property at a fixed temperature and pressure can be calculated by 1) finding the pressure ranges that the pressure falls in (P1 and P2), then 2) interpolating with respect to temperature on each of these pressure curves (P1 and P2) to obtain the property values v1 and v2. Then the actual property value is interpolated between v1 and v2. At present, we use piecewise linear interpolations in both temperature and pressure. These interpolation functions are in the lookup\_mod.f90 module. The discontinuities, in general, seem better captured by this approach rather than interpolating on pressure first.

The lookup table data points are static arrays defined in each subroutine by an include statement. For example, the file o2cp.inc contains the definition of the data points for cp for O2. This allows simple modification of these data points. The data structure for these lookup tables is basically that of a compressed sparse row matrix and is described in a latter section.

Currently, only O2 has been considered. The steps employed to develop the fortran routines are described below:

1. The properties for O2 as a function of temperature at a range of pressures was extracted from the NIST fluid properties web hand book (<http://webbook.nist.gov/chemistry/fluid/>). This is good database and allows for the extraction of the fluid properties in the manner defined above by using the isobaric option and copying the data in the tab delimited format into Excel. There will need to be some units manipulations, but these are relatively easy in Excel
2. Cut and past the data matrix into a Matlab workspace variable and save it
3. Use Matlab to write the .inc files that have the correct look up table format for the fortran routines. This is done by the write\_ludata(.m) function. The Matlab script used to write the include files for O2 is the O2properties.m script.
4. Move the .inc files to the main directory.
5. If a new material is being added then a new f90 module needs to be developed. This should follow the o2props\_mode.f90 in such a similar manner that only the o2 needs to be changed to that of the new material in each subroutine.
6. Compile.