Development of a RELAP5-3D Property Library for Use by Other Computer Codes

C. B. Davis, Y. J. Choi, J. H. Forsmann

2015 IRUG Meeting

Idaho Falls, ID

August 13-14, 2015

Outline

- Background
- Input and output
- Automatic testing
- Results
- Conclusions and recommendations

Background

- RELAP5-3D contains more than 25 working fluids
  - Light and heavy water, gases (N_2, CO_2, etc.), liquid metals (Na, NaK, Bi-Pb, etc.), molten salts (FLiBe, FLiNaK, etc.), etc.
- “Exact” fluid properties calculated from an equation of state are contained in external *tpf* files for each working fluid
  - The *tpf* files are machine-independent binary files written in XDR format
  - Portable across different computer platforms
- To achieve fast running speed, RELAP5-3D obtains approximate fluid properties for internal calculations by interpolation
- The combination of a large number of working fluids, portable *tpf* files, and fast-running interpolators means that the RELAP5-3D data would be very useful to engineers and other computer programs if they were readily available

Background (cont'd)

- The purpose of this task was to develop an easily accessible library and driver programs that can determine fluid properties for any of the working fluids simulated by RELAP5-3D
- The following items were produced during this task
  - Property library (*polate.a*)
  - Interactive driver program (*polate.x*)
  - Driver program for automatic testing (*polated.x*)
  - Subroutine that can be called by other computer programs (*polates.F*)
  - Interpolating subroutine (*stpuph.F*) that can calculate metastable states for fluid *h2on* with independent variables of pressure (*P*) and specific enthalpy (*h*)

Background (cont'd)

- The *polate* library and driver routines are not used by RELAP5-3D but are now automatically placed in a new directory under the parent directory (i.e. *r3d433is/polate*)

Inputs for the polate routines

- Inputs identify the *getstate* subroutine to be used (*istate*), the fluid property number (*fnum*), two thermodynamic variables, a phase identifier (*itin*), and a logical flag (*verify*)
- *istate* is related to the name of the *getstate* subroutine

istate	Subroutine	First variable	Second variable
1	*getstate1*	Quality	Temperature (K)
2	*getstate2*	Quality	Pressure (Pa)
3	*getstate3*	Temperature (K)	Pressure (Pa)
5	*getstate5*	Specific enthalpy (J/kg)	Pressure (Pa)
6	*getstate6*	Specific internal energy (J/kg)	Pressure (Pa)

- *fnum* is the fluid number
  - For example, 1 = *h2o*, 12 = *h2on*, 18 = *h2o95, 28 = r134a*

Inputs for the polate routines (cont’d)

- Phase identifier (*itin*)
  - 1 = liquid
  - 2 = two-phase
  - 3 = vapor
  - 4 = supercritical
- The phase identifier can be used to force the calculation of equilibrium two-phase properties or metastable properties
- If *verify* is true, numerical derivatives of state properties are calculated and compared with analytical derivatives
  - Normally used only during automatic testing
  - The numerical derivatives are finite difference approximations obtained by making two calls to *polate*, one with a nominal independent variable (i.e. *P*, *h*, etc.) and one in which the nominal value is perturbed by a factor of 1.000001

Outputs for the polate routines

- The *s* array, which contains thermodynamic and transport properties
  - The entire *s* array is generally not consistent with the definitions contained in *mnemod.F90*
    - Some variables are either not updated or contain the results of intermediate calculations
    - The *polate* routines place NaN, Not a Number, in the positions that are not consistent with *mnemod.F90* to prevent the use of invalid thermodynamic data
- The output fluid state, *it*
  - Has the same meaning as *itin*
- A logical flag, *err*, that is set to true when an error has occurred
  - For example, if one of the input thermodynamic properties exceeds the bounds in the *tpf* file

Outputs for the polate routines (cont'd)

- Analytical property derivatives that are functions of basic thermodynamic data
  - *istate* = 1
    - $\frac{dP^s}{dT}, \left(\frac{dh_f}{dT}\right), \left(\frac{dh_g}{dT}\right)$
  - *istate* = 2
    - $\frac{dT^s}{dP}, \left(\frac{dh_f}{dP}\right), \left(\frac{dh_g}{dP}\right)$
  - *istate* = 5
    - $\left(\frac{\partial\rho}{\partial h}\right)_P, \left(\frac{\partial T}{\partial h}\right)_P, \left(\frac{\partial\rho}{\partial P}\right)_h, \left(\frac{\partial T}{\partial P}\right)_h$
  - *istate* = 6
    - $\left(\frac{\partial\rho}{\partial U}\right)_P, \left(\frac{\partial T}{\partial U}\right)_P, \left(\frac{\partial\rho}{\partial P}\right)_U, \left(\frac{\partial T}{\partial P}\right)_U$
- Corresponding numerical derivatives are calculated if *verify* = true

Outputs for the polate routines (cont'd)

- For *istate* = 5 or 6, property derivatives can be calculated for equilibrium single-phase states, metastable single-phase states, or equilibrium two-phase states

Testing

- Testing procedures were developed that use *polated.x* to perform automatic testing during code installation
- These testing procedures used Linux scripts, input files, and AptPlot scripts
- Testing was performed for three light waters (*h2o*, *h2on*, *h2o95*) and one refrigerant (*r134a*) for both liquid and vapor phases
- Testing was performed for *istate* = 1, 2, 3, 5, and 6
- Testing was performed for equilibrium and metastable single-phase properties

Testing (cont'd)

- The Linux script *rundplt** (* = 1, 2, 3, 5, or 6)
  - Contains a list of input files
  - For each input file in the list, the script
    - Calls the driver program (*polated.x*) to generate thermodynamic results
    - Calls AptPlot scripts that generate graphs of the thermodynamic results
    - Makes output directories based on the name of the input file and moves the output files and figures into the designated directory
      - For example, *rundplt1* contains *T_h2o.i* in the list, calls *t.apt* to plot saturated fluid properties as a function of temperature and *td.apt* to plot derivatives as a function of temperature, and moves the results into directory *T_h2o_*

polated.x reads the input files

- The input files use normalized or relative thermodynamic data
- This approach allows similar input files for the testing of different fluids (i.e. *h2o* and *r134a*)
- The input files use normalized temperatures with *istate* = 1
  - $T_N = \frac{T - T_{trip}}{T_{crit} - T_{trip}}$
- The input files use normalized pressures for *istate* = 2, 3, 5, and 6
  - $P_N = \frac{P - P_{trip}}{P_{crit} - P_{trip}}$

polated.x reads the input files (cont'd)

- The input files use differential temperature relative to the saturation temperature for istate = 3, 5, and 6
  - $\Delta T = T^s - T$ for liquid and $\Delta T = T - T^s$ for gas
  - Negative values correspond to metastable states while positive values correspond to the equilibrium state for the phase
  - The input differential temperatures are converted to specific enthalpy for *istate* = 5 and specific internal energy for *istate* = 6
- The user can easily change the input files to test other fluids, interpolators, or thermodynamic region

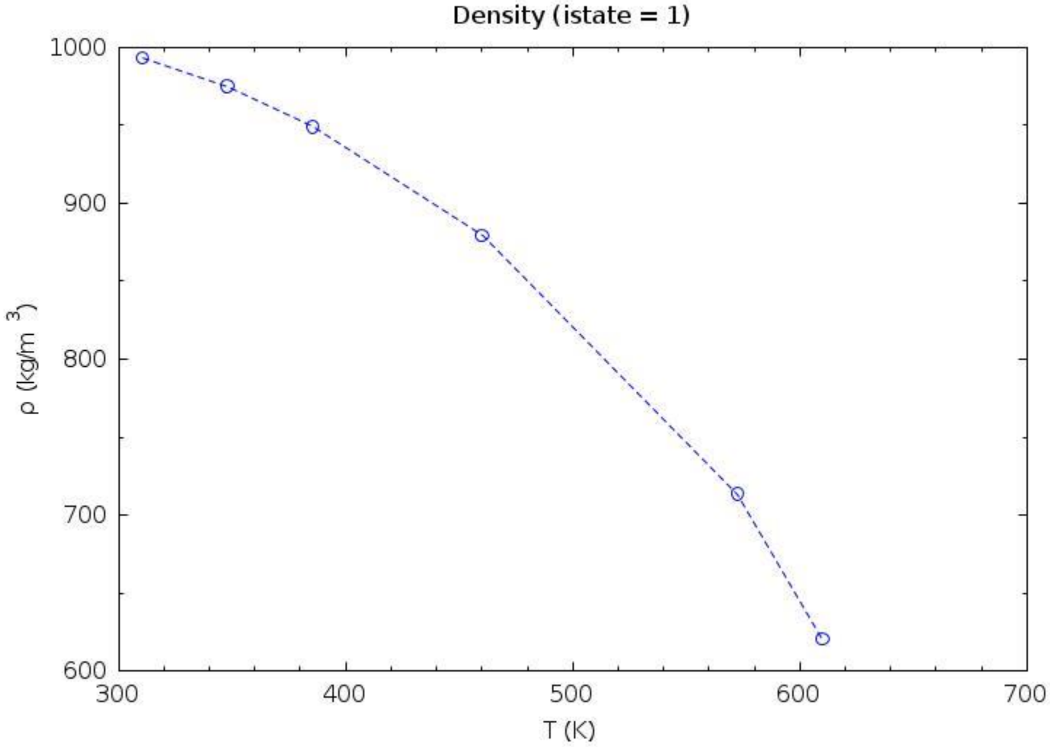
An example listing of input file PT_h2on_f.i for istate = 3 follows:

```
P-dT tables for P-h interpolator testing

 3 for istate number (istate: 1 for T, 2 for P, 3 for P-T, 5 for P-h, 6 for P-u)

 1 for phase identifier (itin: 1 is liquid, 3 is vapor)

12 for fluid type (fnum: 1 is H2O, 12 is H2O_N, 18 is H2O_95, 28 is SUVA)

 6 PN as follows :

     0.01 0.05 0.1 0.2 0.5 0.8

10 dT as follows :

   1.0  5.0 10. 20.  25.0
   30. 50.0 70. 90. 100.0
```

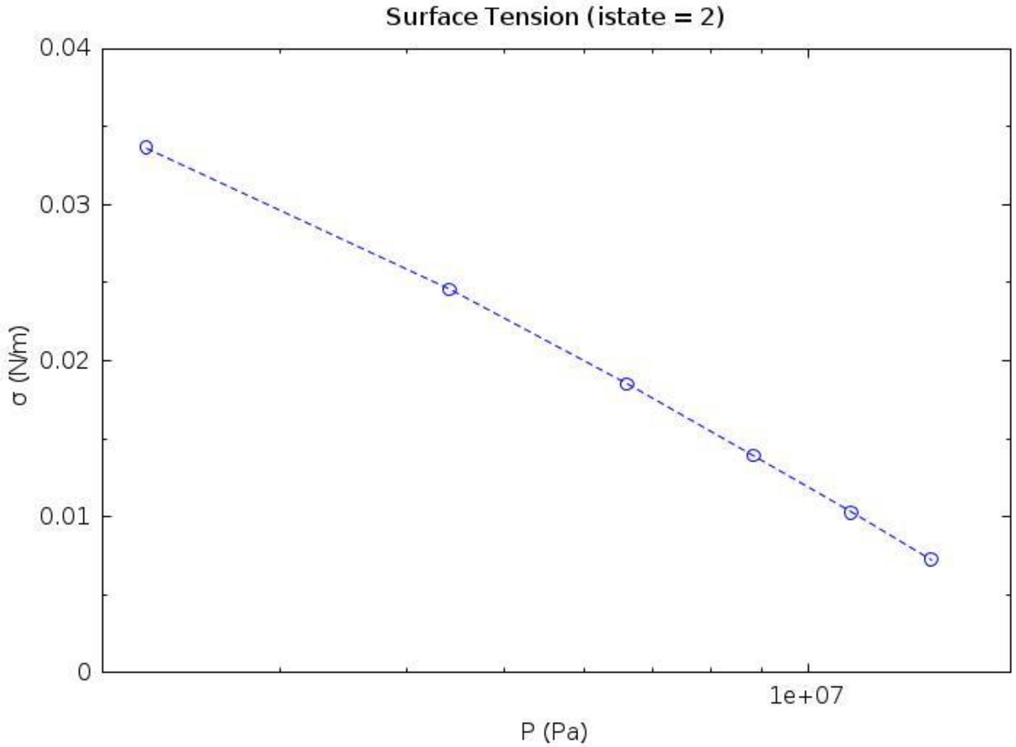
The output of the testing includes:

- Plots of single-phase state properties (T, ρ, U, h, β, κ, C_P, S, μ, and k)
- Plots of saturated properties (P^s or T^s, ρ, U, h, β, κ, C_P, S, μ, k, and σ)
- Plots comparing analytical and numerical derivatives
- Plots of the translation error, which illustrate the accuracy of the interpolators
  - For example, a (*P*,*T*) call to obtain *h* followed by a (*P*,*h*) call to obtain *T* determines the translation error in *T*

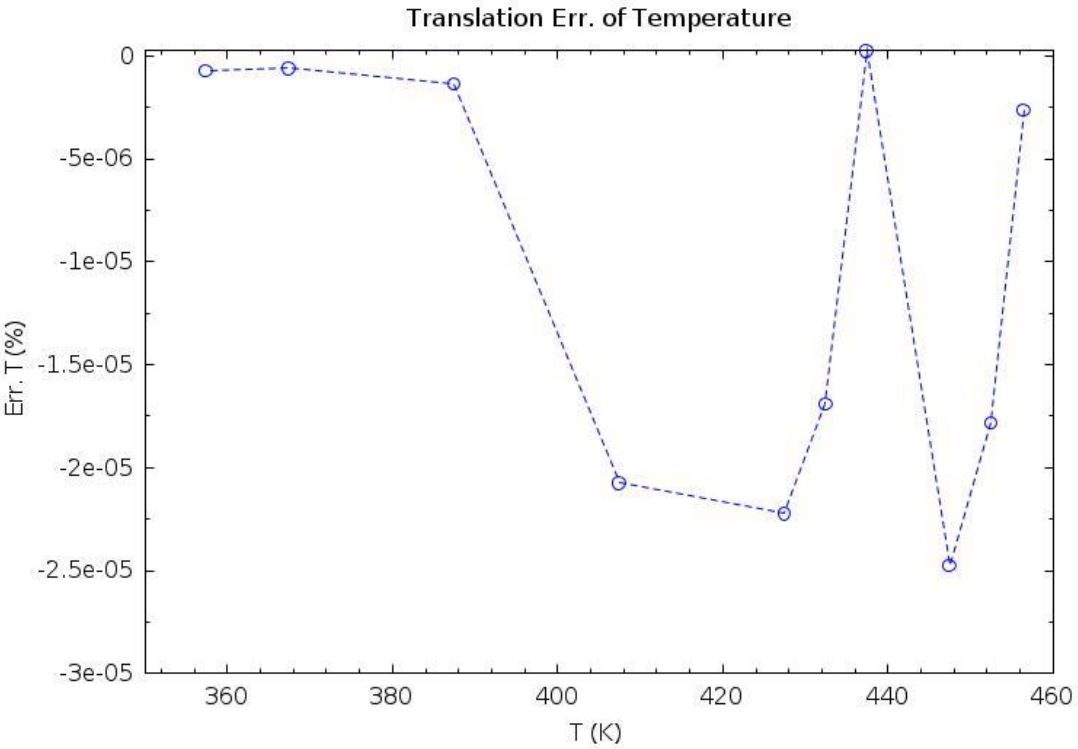
Sample results for a basic thermodynamic property (ρ_f of h2o versus temperature)



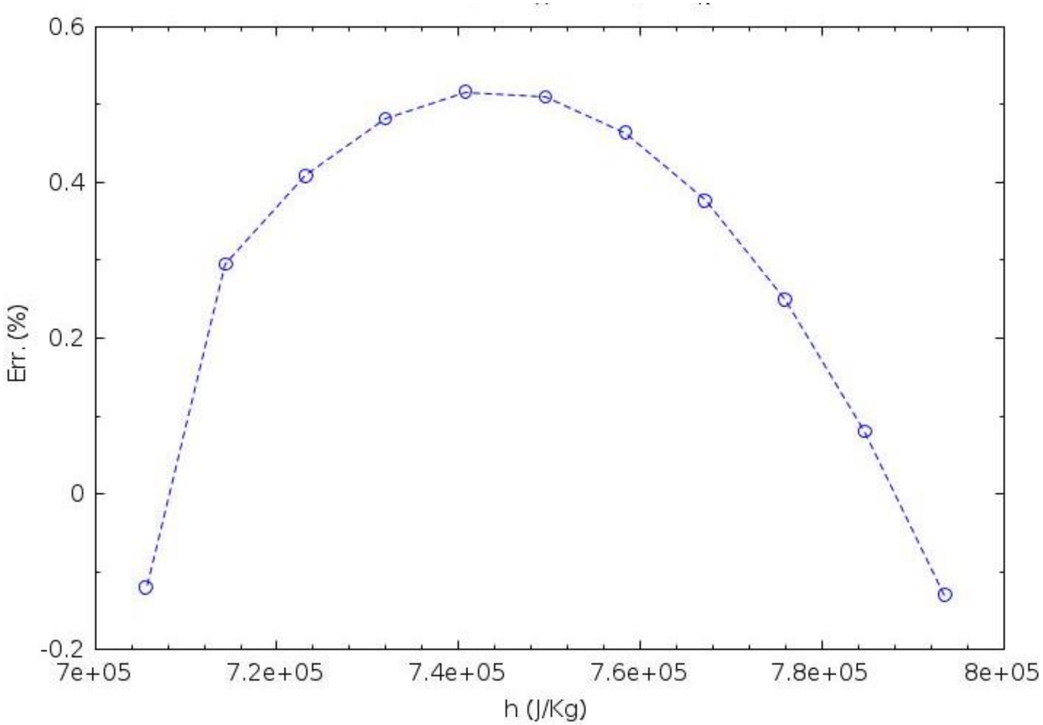
Sample results for a basic transport property (σ of h2on versus pressure)



Sample results for translation error (T of h2on liquid at P_N = 0.05)

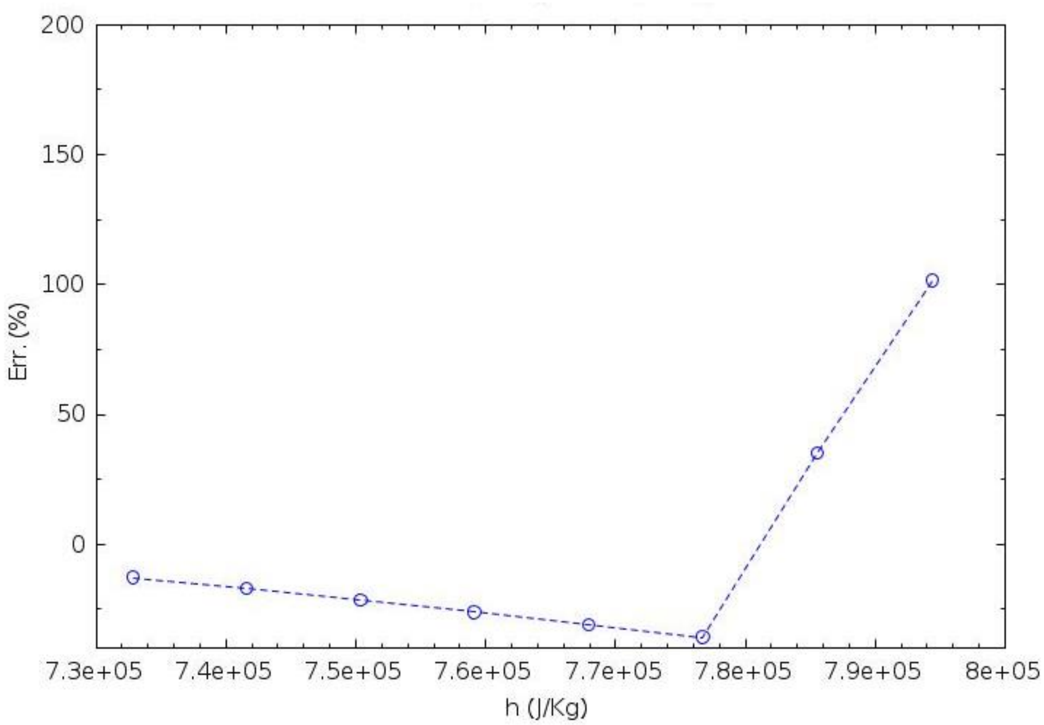


Relative error between analytical and numerical derivatives ($\left(\frac{\partial\rho}{\partial P}\right)_h$ of h2o liquid at P = 1.2E6 Pa)



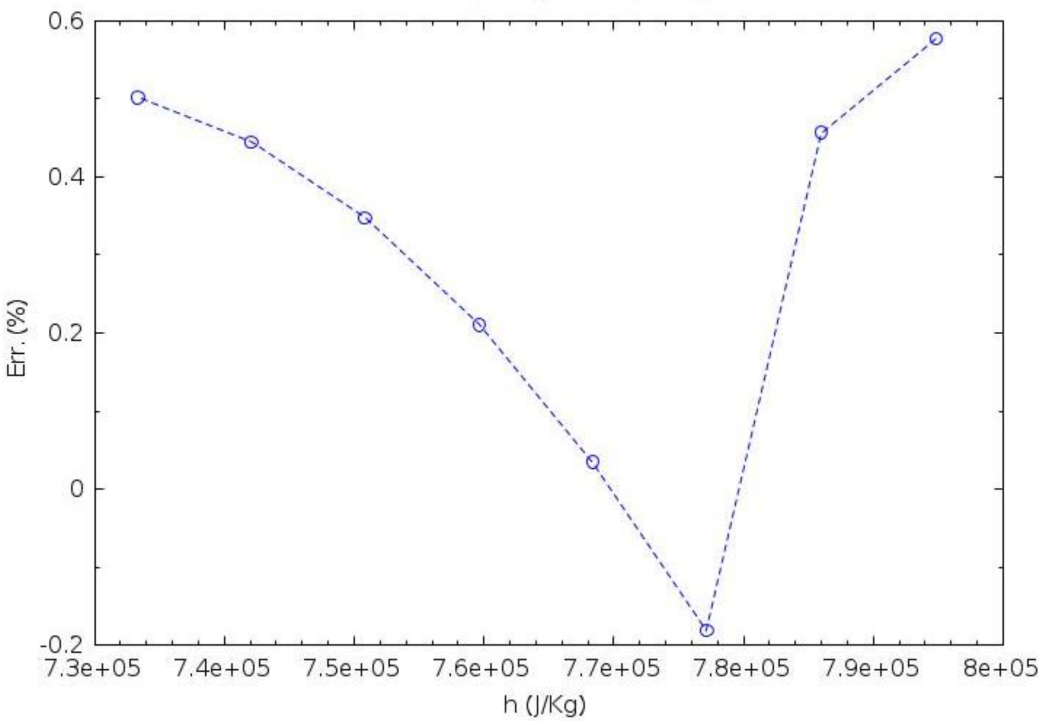
- *istate* = 5
- Liquid subcooling between 1 and 21 K
- 440 ≤ T ≤ 460 K
- Corresponds to interpolation along the edge of a single box in the *tpfh2o* file
- Error is relatively small
- Representative of the minimum error expected in this box

Relative error between analytical and numerical derivatives $\left(\frac{\partial\rho}{\partial P}\right)_h$ of h2o liquid at P = 1.1E6 Pa)



- The same input as the previous slide but the pressure has been reduced from 1.2E6 Pa to 1.1E6 Pa
- Corresponds to interpolation at a pressure point midway between adjacent grid points in the *tpfh2o* file
- State varies from 11 K subcooled to 3 K superheated (446 ≤ T ≤ 460 K)
- Representative of the maximum error expected in the box
- Error gets very large in the metastable region

Relative error between analytical and numerical derivatives ($\left(\frac{\partial\rho}{\partial P}\right)_h$ of h2on liquid at P = 1.1E6 Pa)



- The same input as the previous slide but with fluid *h2on* instead of fluid *h2o*
- The six points on the LHS are subcooled, the two points on the RHS are superheated
- The relative error is much less than for fluid *h2o*, particularly in the metastable region
- The relative error in the derivatives is expected to correlate with mass error

Conclusions and recommendations

- The *polate* programs allow easy access to the fluid properties used by RELAP5-3D for a large database of working fluids
- The *polate* programs have a variety of potential uses
  - Provide engineers with a convenient way to obtain fluid properties that are consistent with those used by RELAP5-3D
  - Code developers can easily generate graphs of fluid properties for use in debugging
  - The programs can be easily used by other computer codes, providing them with access to the RELAP5-3D database
  - The automatic testing feature can be used to provide assurance that the fluid properties have or have not changed between code versions

Conclusions and recommendations (cont'd)

- The relative error between the analytical and numerical derivatives is expected to correlate with mass error
  - Therefore, the mass error with *h2on* is expected to be less than with *h2o*, particularly in the metastable regions
  - RELAP5-3D calculations should be performed to verify if this conclusion is correct
- Source code customers will have access to the *polate* routines in the next code version
- Executable code customers interested in the *polate* subroutines should inform INL