Refitting Thermodynamic Data

Philipp Boettcher, Jack Ziegler, Vaughan Thomas,
Remy Mevel, Jason Damazo, and Joseph Shepherd

March 5, 2012

Thermodynamic data, including specific heat, enthalpy and entropy for each species, are part of the chemical mechanism used to compute the ignition in the slowly heated vessel in Boettcher et al. (2011). In the thermodynamic data included as part of the mechanism published by Ramirez et al. (2011), many of the species have a discontinuity at the point where the low temperature fit connects to the high temperature fit as shown in Figure 1 for $C_2H_5CO_2$. These discontinuities are problematic for some numerical solvers and should be avoided.

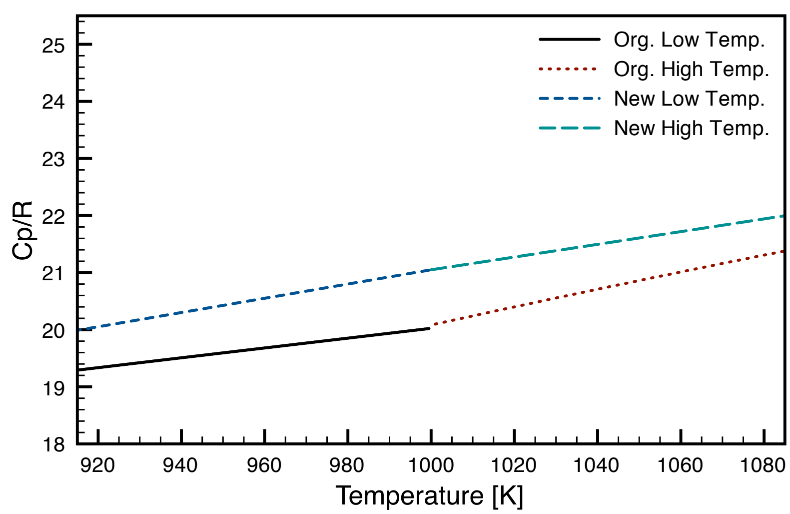

Figure 1: Original thermodynamic data - c_p/R for $C_2H_5CO_2$

The NASA polynomial representation is used for complex equilibrium calculations as discussed by Gordon and McBride (1994). Further discussion of the polynomials and fitting is given in Shepherd et al. (2006)[1]. For each species the data has two sets of seven coefficients, a_n for the low temperature regime and seven coefficients, b_n for the high temperature regime. For example the specific heat at low temperature is given by the following equation

$$\frac{c_p}{R} = \sum_{n=0}^{4} a_n T^n \,. \quad (1)$$

In this case we take the available fits and create new ones without discontinuities. The first step is to select the species whose polynomials require refitting and generating a data set based on the original fits. A choice has to be made about the step size in which to create the data set, which creates stable final polynomials. In this case data has been generated every 100 K and at the mid point the average of the

[1] http://www.galcit.caltech.edu/EDL/public/cantera/doc/tex/ShockDetonation/ShockDetonation.pdf

high and low temperature is taken. Then a constrained least squares fitting of the data is performed while keeping the enthalpy of formation and formation entropy the same.

The new fit must conserve the enthalpy of formation, $\Delta_f h^\circ$, and the formation entropy, $s_\circ(T^\circ)$. Both of these quantities can be computed from the original data. The enthalpy is computed using the first 6 coefficients using the following equation

$$\frac{h}{RT} = \sum_{n=0}^{4} \frac{a_n T^n}{n+1} + \frac{a_5}{T} \;, \tag{2}$$

where

$$a_5 = \frac{\Delta_f h^\circ}{R} - \sum_{n=0}^{4} \frac{a_n}{n+1} \left(T^\circ\right)^{n+1} \;. \tag{3}$$

Thus to conserve, $\Delta_f h^\circ$, it is computed from the initial data and we solve the following equation for the first five constants in the least squares fitting using

$$\left[\frac{h}{RT} - \frac{\Delta_f h^\circ}{RT}\right]_{\text{org}} = \sum_{n=0}^{4} \frac{a_n}{n+1} \left[T^n - \frac{\left(T^\circ\right)^{n+1}}{T}\right] \tag{4}$$

and then Equation 3 for a_5.

The entropy is computed from Equations 5 and 6.

$$\frac{s_\circ}{R} = a_0 \ln\left(T\right) + \sum_{n=1}^{4} \frac{a_n T^n}{n} + a_6 \tag{5}$$

$$a_6 = \frac{s_\circ\left(T^\circ\right)}{R} - \left(a_0 \ln\left(T^\circ\right) + \sum_{n=1}^{4} \frac{a_n \left(T^\circ\right)^n}{n}\right) \tag{6}$$

Similarly, to conserve $s_\circ(T^\circ)$, it is computed from the original data and we solve Equation 7 in the least squares fitting

$$\left[\frac{s_\circ}{R} - \frac{s_\circ\left(T^\circ\right)}{R}\right]_{\text{org}} = a_0 \ln\left(\frac{T}{T^\circ}\right) + \sum_{n=1}^{4} \frac{a_n \left(T^n - \left(T^\circ\right)^n\right)}{n} \;, \tag{7}$$

and then solve for a_6 using Equation 6.

The constrained least squares fitting was successfully used with the following constraints applied for the two polynomials:

1. Match lowest and highest value of c_p/R
2. C^0 (continuous) c_p/R at mid point
3. C^1 (1st derivative continuous) c_p/R at mid point
4. Match lowest and highest value of $h/(RT)$
5. C^0 (continuous) $h/(RT)$ at mid point
6. C^0 (continuous $s_\circ/R$ at mid point.

The final result of the fitting in Figure 2 shows the successful refit of the specific heat.

In our current version the refitting is performed using MATLAB using the constrained linear least-squares solver `lsqlin` (MATLAB, 2010). The function solves the matrix equation $Ax = b$ using a minimization subject to the constraint equation $A_{eq}x = b_{eq}$. The `lsqlin` function is called in the following way:

```
[x] = lsqlin(A, b, [ ], [ ], Aeq, beq, lb, ub, x0)
```

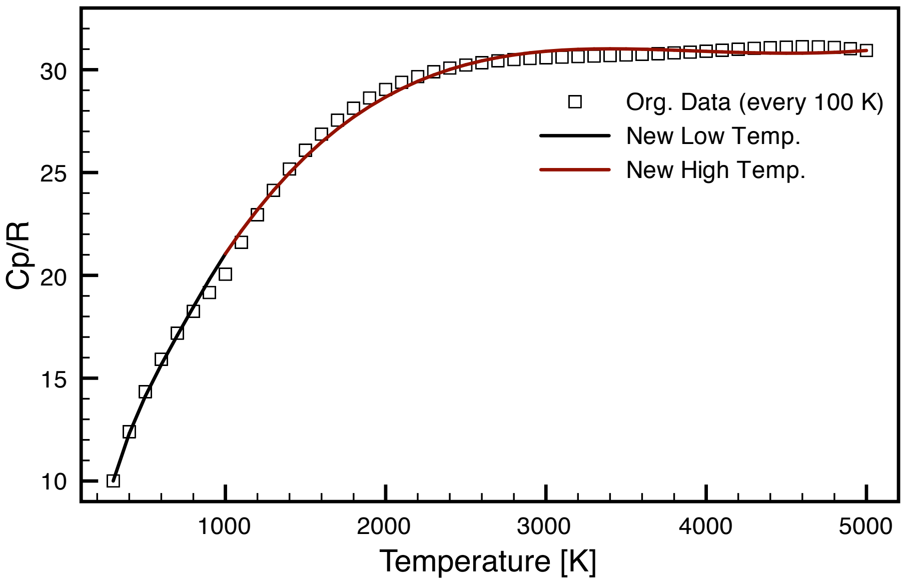

Figure 2: New thermodynamic data - c_p/R for $C_2H_5CO_2$

During the first iteration the starting point is empty, `x0 = [ ]`. The least squares fitting is then called an additional 50 times in a loop using the previous result as the initial condition for the current iteration. The lower and upper bounds, `lb` and `ub`, are simply set at -Inf and +Inf.

The least square equation is set up such that x vector contains the new coefficients for the low temperature, a_n, and high temperature, b_n,

$$x = [a_0, a_1, \cdots, a_5, a_6, b_0, b_1, \cdots, b_5, b_6] \ . \tag{8}$$

The A matrix is arranged in the following way:

$$A = \begin{bmatrix} c_p/R \text{ in the low temperature range (M rows)} \\ c_p/R \text{ in the high temperature range (N rows)} \\ h/(RT) \text{ in the low temperature range (M rows)} \\ h/(RT) \text{ in the high temperature range (N rows)} \\ s_\circ/R \text{ in the low temperature range (M rows)} \\ s_\circ/R \text{ in the high temperature range (N rows)} \end{bmatrix} , \tag{9}$$

where M is the number of elements in a vector spanning from the lowest temperature to the mid temperature in increments of 100 K, and N s the number of elements in a vector spanning from the mid temperature to the highest temperature in increments of 100 K.

For example, for the specific heat $Ax = b$ is

$$\begin{bmatrix} T_1^0 & T_1^1 & \cdots & T_1^4 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots \\ T_{\text{mid}}^0 & T_{\text{mid}}^1 & \cdots & T_{\text{mid}}^4 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & T_{\text{mid}}^0 & T_{\text{mid}}^1 & \cdots & T_{\text{mid}}^4 & 0 & 0 \\ \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & T_{\text{max}}^0 & T_{\text{max}}^1 & \cdots & T_{\text{max}}^4 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_4 \\ a_5 \\ a_6 \\ b_1 \\ b_2 \\ \vdots \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} \frac{c_p}{R}\big|_{\text{org @ } T_1} \\ \vdots \\ \frac{c_p}{R}\big|_{\text{org @ } T_{\text{mid}}} \\ \frac{c_p}{R}\big|_{\text{org @ } T_{\text{mid}}} \\ \vdots \\ \frac{c_p}{R}\big|_{\text{org @ } T_{\text{max}}} \end{bmatrix} . \quad (10)$$

For the enthalpy and entropy equation the entries of b are the left-hand sides of Equations 4 and 7, respectively, computed from the original data.

The constrain equations are implemented in a similar way. For example matching the specific heat at the mid point is constraint by the following equation:

$$\begin{bmatrix} T_{\text{mid}}^1 & T_{\text{mid}}^2 & \cdots & T_{\text{mid}}^4 & 0 & 0 & -T_{\text{mid}}^1 & -T_{\text{mid}}^2 & \cdots & -T_{\text{mid}}^4 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ b_6 \end{bmatrix} = 0 \, . \quad (11)$$

The final step is to compute the remaining error in the fit at the mid point, which in our example is 1×10^{-14} and thus sufficient for the solver. If the error is too large more iterations of the least square fitting should be performed.

References

P. A. Boettcher, R. Mével, V. Thomas, and J. E. Shepherd. The effect of heating rates on low temperature hexane air combustion. *Fuel*, 2011. 1

H. P. Ramirez, K. Hadj-Ali, P. Dievart, G. Dayma, C. Togbe, G. Moreac, and P. Dagaut. Oxidation of commercial and surrogate bio-diesel fuels (B30) in a jet-stirred reactor at elevated pressure: Experimental and modeling kinetic study. *Proceedings of the Combustion Institute*, 33: 375–382, 2011. 1

S. Gordon and B. J. McBride. Computer program for calculation of complex chemical equilibrium compositions and applications. Technical report, National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Program, 1994. 1

J. E. Shepherd, S. Browne, and J. Ziegler. Numerical Solution Methods for Shock and Detonation Jump Conditions. Technical report, Aeronautical and Mechanical Engineering California Institute of Technology, Pasadena, CA, 2006. 1

MATLAB. *Version 7.9.0.529 (R2009b)*. The MathWorks Inc., Natick, Massachusetts, 2010. 3