

**T. E. A.**  
**THERMOCHEMICAL EQUILIBRIUM ABUNDANCES**  
**CODE**

---

**Theory Document**

---

*Author:*

JASMINA BLECIC

*Lead Scientist:*

JASMINA BLECIC

*Coder:*

M. OLIVER BOWMAN

*PI:*

JOSEPH HARRINGTON

25 June 2014

Copyright (C) 2014 University of Central Florida.  
ALL RIGHTS RESERVED.

This document goes along with the TEA code. The document is not peer-reviewed and the code is a test version only. They may not be redistributed to any third party. Please refer such requests to us. The document and the program are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

We welcome your feedback, but do not guarantee support. Please send feedback or inquiries to both:

Jasmina Blecic: [jasmina@physics.ucf.edu](mailto:jasmina@physics.ucf.edu)

Joseph Harrington: [jh@physics.ucf.edu](mailto:jh@physics.ucf.edu)

or alternatively,

Jasmina Blecic and Joseph Harrington  
UCF PSB 441  
4000 Central Florida Blvd  
Orlando, FL 32816-2385  
USA

Thank you for testing TEA!

## Abstract

We developed a program with 12 modules that produces equilibrium abundances of the gaseous species in planetary atmospheres for a wide range of hot-Jupiter temperatures. The code is based on the methodology formulated by White et al. (1958) and Eriksson (1971). It applies Gibbs free energy minimization using an iterative, Lagrangian method. We tested the code against the method developed by Burrows & Sharp (1999), the open source thermochemical equilibrium code CEA (Chemical Equilibrium with Applications), and the example given in White et al. (1958). We confirmed our results within the numerical precision given by the tabulated Gibbs free energies. The code is modular, well documented, and is available to the scientific community under an open-source license.

# 1 Thermochemical Equilibrium Abundances Code

This project was completed with the support of the NASA Earth and Space Science Fellowship Program, grant NNX12AL83H, held by Jasmina Bleicic, PI Joseph Harrington. Project development included graduate student Jasmina Bleicic (lead scientist and coder) and undergraduate M. Oliver Bowman (coder). In this document, we describe the theoretical basis for the method applied in the code. In the TEA Manual (in preparation) we describe the code itself.

The Thermochemical Equilibrium Abundances (TEA) code calculates the equilibrium abundances of gaseous molecular species present in hot-Jupiter atmospheres. It works with high numerical precision, without adjustments for fractional abundances (mixing fractions) up to  $10^{-14}$  and a temperature range of 1000 - 4000 K. The code is based on the Gibbs free energy minimization calculation, originally developed by White et al. (1958) and Eriksson (1971). This approach has been used by many authors in the exoplanetary field (e.g., Seager 1999, 2010, Madhusudhan & Seager 2009, 2010, Sharp & Huebner 1990). More recent astrophysical adaptations are described in Sharp & Huebner (1990), Petaev & Wood (1998), and Burrows & Sharp (1999).

## 1.1 Gibbs Free Energy Minimization Method

There are two methods to calculate chemical equilibrium: using equilibrium constants and reaction rates or minimizing the free energy of a system. Although chemical equilibrium can be calculated almost trivially for several reactions present in the system, as the number of reactions increases, the set of numerous equilibrium constant relations becomes impossible to solve simultaneously. The advantage of the free energy minimization method is that each species present in the system can be treated independently without specifying complicated sets of reactions a priori, and therefore, a limited set of equations needs to be solved (see Section 1.2).

Equilibrium abundances can be obtained by using different combinations of thermodynamical State functions (temperature and pressure –  $tp$ , Enthalpy and pressure –  $Hp$ , Entropy and pressure –  $Sp$ , temperature and volume –  $tv$ , Internal Energy and volume –  $Uv$ , etc.). Depending on how the system is described, the condition for equilibrium can be stated in terms of Gibbs free energy, Helmholtz energy, or Entropy. If a thermodynamical state is defined with temperature and pressure, Gibbs free energy ( $G$ ) is most easily minimized, since those two states are its natural, dependent variables.

Gibbs free energy represents a thermodynamical potential that measures the useful work obtainable by the system at a constant temperature and pressure. Thus, the Gibbs free energy minimization method minimizes the total chemical potential of all involved species when the system reaches equilibrium.

The Gibbs free energy of the system at a certain temperature is the sum of the Gibbs free energies of its constituents:

$$\Delta G(T) = \sum_i^n G_i(T), \quad (1)$$

where  $\Delta G(T)$  is the total Gibbs free energy of the system for  $n$  chemical species and  $G_i(T)$  is the Gibbs free energy of a gas species  $i$ . The total Gibbs free energy of the system is expressed as the sum of the number of moles  $x$  of the species  $i$ ,  $x_i$ , and their chemical potentials  $g_i(T)$ :

$$\Delta G(T) = \sum_i^n x_i g_i(T). \quad (2)$$

The chemical potential  $g_i(T)$  depends on the chemical potential at the standard state  $g_i^0(T)$  and the activity  $a_i$ ,

$$g_i(T) = g_i^0(T) + RT \ln a_i, \quad (3)$$

where  $R$  is the gas constant,  $R = k_B N_A$ , and  $k_B$  and  $N_A$  are the Boltzmann constant and the Avogadro's number, respectively. Activities for gaseous species, which are treated as ideal, are equal to the partial pressures, and for condensates they equal 1:

$$a_i = P_i = P \frac{x_i}{N}, \quad \text{for gases} \quad (4)$$

$$a_i = 1, \quad \text{for condensates} \quad (5)$$

$P$  is the total pressure of the atmosphere,  $N$  is the total number of moles of all species involved in the system. Hence, Equation (3) for gaseous species becomes:

$$g_i(T) = g_i^0(T) + RT \ln P_i. \quad (6)$$

Combining Equation (6) with Equation (2), the Gibbs free energy of the system becomes:

$$\Delta G(T) = \sum_i^n x_i \left( g_i^0(T) + RT \ln P_i \right), \quad (7)$$

or,

$$\Delta G(T) = \sum_i^n x_i \left( g_i^0(T) + RT \ln P + RT \ln \frac{x_i}{N} \right), \quad (8)$$

For our purposes, it is more convenient to write Equation (8) in unitless terms:

$$\frac{\Delta G(T)}{RT} = \sum_{i=1}^n x_i \left[ \frac{g_i^0(T)}{RT} + \ln P + \ln \frac{x_i}{N} \right]. \quad (9)$$

Equation (9) requires a knowledge of the free energy of each species as a function of temperature. These can be obtained from the JANAF tables (<http://kinetics.nist.gov/janaf/>, Chase et al. 1982, Chase 1986, Burrows & Sharp 1999), or easily derived from other tabulated functions.

To extract free energies,  $g_i^0(T)/RT$ , from JANAF tables we used the expression given in Eriksson (1971), Equation (2):

$$\frac{g_i^0(T)}{RT} = 1/R \left[ \frac{G_i^0 - H_{298}^0}{T} \right] + \frac{\Delta_f H_{298}^0}{RT}, \quad (10)$$

where  $g_i^0(T)$  is given in J/mol,  $R = 8.3144621$  J/K/mol,  $H_{298}^0$  is the enthalpy (heat content) in the thermodynamical standard state at a reference temperature of  $25^\circ\text{C} = 298.15$  K,  $G_i^0$  is the Gibbs free energy in J,  $(G_i^0 - H_{298}^0)/T$  is the free energy function in J/K/mol, and  $\Delta_f H_{298}^0$  is the heat of formation at 298.15 K in kJ/mol. Thus, our conversion equation becomes:

$$\frac{g_i^0(T)}{RT} = 1/R \left[ \frac{G_i^0 - H_{298}^0}{T} \right] + \frac{\Delta_f H_{298}^0 * 1000}{RT}, \quad (11)$$

$G_i^0 - H_{298}^0/T$  is the fourth term in JANAF tables and  $\Delta_f H_{298}^0$  is the sixth.

To determine the equilibrium composition, we need to find a non-negative set of values  $x_i$ , which minimizes Equation (9) and satisfies the mass balance constraint:

$$\sum_{i=1}^n a_{ij} x_i = b_j, \quad (j = 1, 2, \dots, m), \quad (12)$$

where the stoichiometric coefficient  $a_{ij}$  indicates the number of atoms of element  $j$  in species  $i$  (e.g., for  $\text{CH}_4$  the stoichiometric coefficient of C is 1 and the stoichiometric coefficient of H is 4), and  $b_j$  is the total number of moles of element  $j$  originally present in the mixture.

We use the reference table containing elemental solar abundances given in Asplund et al. (2009) for  $b$  values. These values are transformed into elemental fractions by number, i.e., ratio of number densities. We convert each dex elemental abundance into counts (number density) and divide it by the total number of all elemental counts (number densities) in the mixture (see Section 3 in Asplund et al. 2009). The final output are fractional abundances (mole mixing fractions), i.e., the ratio of each species' mole numbers to the total sum of all mole numbers in the mixture.

## 1.2 Lagrangian Method of Steepest Descent

To find equilibrium abundances of the desired molecular species at a given temperature and pressure, we need to minimize Equation (9). To do so, we have to apply a technique that minimizes a multi-variable function under constraint. There are many optimization techniques used to find the minima of a function subject to equality constraints (e.g., line search method, Dantzig-simplex method for linear programming, Newton-Rapson method, Hessian-conjugate gradient method, Lagrangian steepest descent method). The main advantage of the Lagrangian steepest descent method is that the number of equations to solve scales with the number of different types of atoms present in the mixture, which is usually

a much smaller number than the possible number of molecular constituents. This allows the code to be executed much faster than in other methods.

Gradient descent, also known as steepest descent, is an algorithm for finding a local minimum of a function. At each iteration, the method takes steps towards the minimum, where each step is proportional to the negative gradient of the function at the current point. If a function  $f(x)$  is defined and differentiable in the neighbourhood of a point  $a$ , then  $f(x)$  decreases most rapidly in the direction of the negative gradient,  $-\nabla f(a)$ . From this, it follows that if  $b = a - \lambda \nabla f(a)$ , then  $f(a) > f(b)$  if  $\lambda$  is small enough. Starting with a guess  $x_0$  for a local minima of  $f$ , and considering a sequence  $x_0, x_1, x_2, \dots$  such that  $x_{n+1} = x_n - \lambda \nabla f(x_n)$ ,  $n \geq 0$ , one gets  $f(x_0) \geq f(x_1) \geq f(x_2) \geq \dots$ . This sequence of  $x_n$  converges to a desired local minima if the correct  $\lambda$  value is assigned. The value of  $\lambda$  can vary at each iteration. If the function  $f$  is convex, the local minimum is also the global minimum.

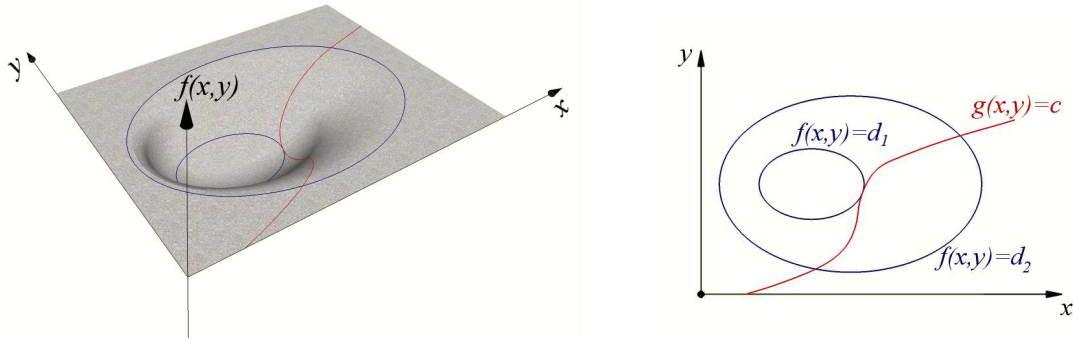


Figure 1: Example of the Lagrangian minimization approach. **Left:** The 3D illustration of the the minimization problem. Blue lines indicate starting and ending values of  $f(x, y)$  during minimization. An  $(x, y)$  pair is found that minimizes  $f(x, y)$  (bottom blue line) subject to a constraint  $g(x, y) = C$  (red line). **Right:** Contour map of the left figure. The point where the red line (constraint) tangentially touches a blue contour is the solution. Since  $d_1 < d_2$ , the solution is the minimum of  $f(x, y)$ .

Our code implements a more complex version of the method outlined above. The problem consists of some function  $f(x, y)$  subject to a constraint  $g(x, y) = C$ . In this case, we need both  $f$  and  $g$  to have continuous first partial derivatives. Thus, we introduce a new variable called the *Lagrangian multiplier*,  $\pi$ , where:

$$\Lambda(x, y, \lambda) = f(x, y) \pm \pi (g(x, y) - C), \quad (13)$$

which allows us to find where the contour of  $g(x, y) = C$  tangentially touches  $f(x, y)$  (see Figure 1). The point of contact is where their gradients are parallel:

$$\nabla_{xy} f(x, y) = -\pi \nabla_{xy} g(x, y). \quad (14)$$

The constant  $\pi$  allows these gradients to have different magnitudes. To find the minimum, we need to calculate all partial derivatives of the function  $\Lambda$ , equate them with zero,

$$\nabla_{x,y,\pi} \Lambda(x, y, \pi) = 0, \quad (15)$$

and follow the same iteration procedure as explained above.

### 1.2.1 Lagrangian Method in TEA

To implement this in our code, we followed the methodology derived in White et al. (1958). We applied an iterative solution to the energy minimization problem, where the mole numbers of the desired molecular species are recomputed at each step and the new direction of steepest descent is calculated. This produces improved mole number values, which however, could be negative. Thus, two short procedures are required in each iteration cycle: solving a set of simultaneous linear equations for an improved direction of descent (described in this Section) and approximately minimizing a convex function of one variable,  $\lambda$ , to ensure that all improved mole number values are positive (see Section 1.3).

To calculate the direction of steepest descent (following the methodology derived in Section 1.2) and initiate the first iteration cycle, we first need to solve the mass balance Equation (12). We start from any positive set of values for the initial mole numbers,  $y = (y_1, y_2, \dots, y_n)$ , as our initial guess:

$$\sum_{i=1}^n a_{ij} y_i = b_j \quad (j = 1, 2, \dots, m). \quad (16)$$

To satisfy the mass balance Equation (16), some  $y_i$  variables must remain as free parameters. In solving these equations, we leave as many free parameters as we have elements in the system, thus ensuring that the mass balance equation can be solved for any number of input elements and output species the user chooses. We set all other  $y_i$  to a known, arbitrary number. Initially, the starting values for the known species are set to 0.1 moles, and the mass balance equation is calculated. If that does not produce all positive mole numbers, the code automatically sets known parameters to 10 times smaller and tries again. The initial iteration input is set when all mole numbers are positive, and the mass balance equation is satisfied.

To follow with the Lagrangian method, we then set basic equations. In Equation (9), we denote the portion to the right as the variable  $c$ :

$$c_i = \frac{g_i^0(T)}{RT} + \ln P, \quad (17)$$

where  $P$  is the pressure in bar. Using  $c_i$ , we denote the right side of Equation (9) as the variable  $f_i(Y)$ :

$$f_i(Y) = y_i \left[ c_i + \ln \frac{y_i}{\bar{y}} \right], \quad (18)$$

where  $Y = (y_1, y_2, \dots, y_n)$  and  $\bar{y}$  is the sum of initial number of moles. The left-hand side of Equation (9),  $\Delta G(T)/RT$ , we denote as function  $F(Y)$ :

$$F(Y) = \sum_{i=1}^n y_i \left[ c_i + \ln \frac{y_i}{\bar{y}} \right]. \quad (19)$$

Then, we do a Taylor series expansion of the function  $F$  about  $Y$ . This yields a quadratic approximation  $Q(X)$ :

$$Q(X) = F(X) \Big|_{X=Y} + \sum_i \frac{\partial F}{\partial x_i} \Big|_{X=Y} \Delta_i + \frac{1}{2} \sum_i \sum_k \frac{\partial^2 F}{\partial x_i \partial x_k} \Big|_{X=Y} \Delta_i \Delta_k. \quad (20)$$

where  $\Delta_i = x_i - y_i$ , and  $x_i$  are the improved mole numbers. This function is minimized using the Lagrangian principle. We now introduce Lagrangian multipliers as  $\pi_j$ :

$$G(X) = Q(X) + \sum_j \pi_j \left( - \sum_i a_{ij} x_i + b_j \right), \quad (21)$$

and calculate the first derivatives,  $\partial G / \partial x_i$ , of the new function. We equate them to zero to find the minima,  $\partial G / \partial x_i = 0$ .

We solve for  $x_i$  from Equation (21) by combining Equation (16) and (18) with the fact that  $\bar{x}$  is the sum of the improved mole numbers,  $\bar{x} = \sum_{i=1}^n x_i$ . The improved number of moles,  $x_i$ , are given as:

$$x_i = -f_i(Y) + \left( \frac{y_i}{\bar{y}} \right) \bar{x} + \left( \sum_{j=1}^m \pi_j a_{ij} \right) y_i, \quad (22)$$

while the Lagrangian multipliers,  $\pi_j$ , are expressed as:

$$\sum_{j=1}^m \pi_j \sum_{i=1}^n a_{ij} y_i = \sum_{i=1}^n y_i \left[ \frac{g_i^0(T)}{RT} + \ln P + \ln \frac{y_i}{\bar{y}} \right], \quad (23)$$

where  $j$  iterates over the  $m$  elements and  $i$  iterates over the  $n$  species.  $\bar{x}$  and  $\bar{y}$  are the sums of improved and initial number of moles. Using Equation (18), we can now rewrite Equation (23) as:

$$\sum_{j=1}^m \pi_j b_j = \sum_{i=1}^n f_i(Y). \quad (24)$$

If we further denote the constants with:

$$r_{jk} = r_{kj} = \sum_{i=1}^n (a_{ij} a_{ik}) y_i, \quad (25)$$

combining Equations (22), (24) and (25) we get the following system of  $m+1$  equations that can be easily solved:



$$\begin{aligned}
r_{11}\pi_1 + r_{12}\pi_2 + \dots + r_{1m}\pi_m + b_1 u &= \sum_{i=1}^n a_{i1} f_i(Y), \\
r_{21}\pi_1 + r_{22}\pi_2 + \dots + r_{2m}\pi_m + b_2 u &= \sum_{i=1}^n a_{i2} f_i(Y), \\
&\vdots \\
&\vdots \\
&\vdots \\
r_{m1}\pi_1 + r_{m2}\pi_2 + \dots + r_{mm}\pi_m + b_m u &= \sum_{i=1}^n a_{im} f_i(Y), \\
b_1\pi_1 + b_2\pi_2 + \dots + b_m\pi_m + 0 u &= \sum_{i=1}^n f_i(Y),
\end{aligned} \tag{26}$$

where:

$$u = -1 + \bar{x}/\bar{y}. \tag{27}$$

The solutions to Equations (26) and (27) will give  $\pi_j$  and  $u$ , and from them using Equation (22) we can calculate the next set of improved mole numbers, i.e., an improved direction of descent,  $\Delta_i = x_i - y_i$ .

### 1.3 Lambda Correction Algorithm

Thus, the improved values of  $x_i$  fall from Equation (22) and they satisfy the mass balance equation. However, solving the system of linear equations can also lead to negative mole numbers for some species. Values such as these are not physically possible, so a short additional step is needed to eliminate this possibility and guarantee a valid result.

We consider the computed changes  $\Delta_i = x_i - y_i$  to be *directional numbers* indicating the preferred direction of descent given by the last Lagrangian calculation. To ensure that all improved values of the mole numbers are positive, we introduce a new value,  $\lambda$ , that defines the fraction of the total distance ( $\lambda\Delta_i$ ) the system moves in the direction of the steepest descent. Other than providing all positive mole numbers, we determine the value  $\lambda$  so that the Gibbs energy of the system must decrease (the minimum point is not passed).

At each Lagrangian iteration cycle we start with our initial guesses  $y_i$  and we get the next set of improved values of  $x_i$  given as:

$$x_i = y_i + \Delta_i. \tag{28}$$

Since we don't want any  $x_i$  to be negative, the variable  $\lambda$  performs a small correction:

$$x'_i = y_i + \lambda\Delta_i. \tag{29}$$

$\lambda$  takes the values between 0 and 1, where value of *zero* implies no step is taken from the iteration's original input,  $y_i$ , and *one* implies that the full Lagrangian distance is travelled,  $\Delta_i$ . We now rewrite Equation (18) using Equation (29) as:

$$f_i(X') = x'_i \left( \frac{g_i^0(T)}{RT} + \ln P + \ln \frac{x'_i}{\bar{x}} \right), \quad (30)$$

which can be written in the form:

$$f_i(\lambda) = (y_i + \lambda\Delta_i) \left( \frac{g_i^0(T)}{RT} + \ln P + \ln \frac{y_i + \lambda\Delta_i}{\bar{y} + \lambda\bar{\Delta}} \right), \quad (31)$$

where  $\bar{\Delta} = \bar{y} - \bar{x}$ . Summing over  $i$ , we get a new function,  $F(\lambda)$ :

$$F(\lambda) = \sum_i (y_i + \lambda\Delta_i) \left( \frac{g_i^0(T)}{RT} + \ln P + \ln \frac{y_i + \lambda\Delta_i}{\bar{y} + \lambda\bar{\Delta}} \right). \quad (32)$$

We choose  $\lambda$  values using sampling that will ensure effective exploration of the  $[0,1]$  range. Half of the range is sampled exponentially, and the other half linearly, totaling 150 points. The exponential sampling is chosen to prevent steps that can cross the negative threshold for  $\Delta_i$  (where mole numbers become negative). For lower temperatures, this threshold has proven to be very low, thus it requires the smallest step in  $\lambda$  space. Increasing the number of points in  $\lambda$  space in certain cases (very low temperatures) can provide better numerical precision.

Thus, to ensure that the new corrected values  $x'_i$  are all positive, the distance travelled will be limited to fractional amounts defined by  $\lambda\Delta_i$ , using the largest possible value of  $\lambda$  that satisfies the conditions:

1. The function called the *directional derivative* is defined and exists:

$$\frac{dF(\lambda)}{d\lambda} = \sum_{i=1}^n \Delta_i \left[ \frac{g_i^0(T)}{RT} + \ln P + \ln \frac{y_i + \lambda\Delta_i}{\bar{y} + \lambda\bar{\Delta}} \right]. \quad (33)$$

2. The directional derivative does not become positive (the minimum point is not passed).

Every new iteration produces a new  $\lambda$  value.  $\lambda$  will be found to approach unity after some number of iterations. Unity in  $\lambda$  indicates the solution is near.

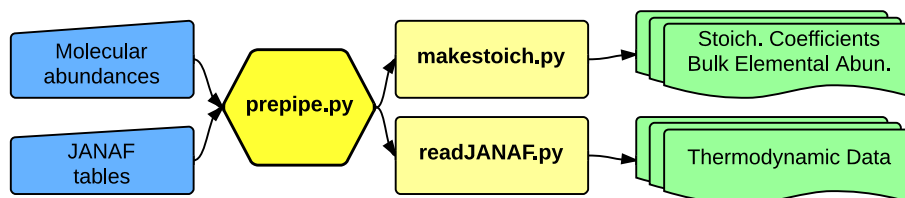
We repeat the Lagrangian method and the lambda correction until a pre-defined maximum number of iterations is met. The final abundances are given as fractional abundances (mole mixing fractions), i.e., the ratio of each species' mole numbers to the total sum of mole numbers of all species in the mixture.

## 1.4 Code Structure

Our TEA code is written entirely in Python and uses the Python packages NumPy and SciPy along with SymPy, an external linear equation solver (<http://sympy.org/>).

The code is divided into two parts: the pre-pipeline that makes thermochemical data library and stoichiometric tables, and the pipeline that performs abundance calculations following the method of Gibbs free energy minimization with the Lagrangian steepest descent, described earlier. The user manual and the detailed explanation of the code functions are given in the TEA Manual (in preparation). In Figure 2, we show the layout of the TEA packages. Below, we list the main modules with their basic characteristics.

### Pre-pipeline modules to make thermodynamic library



### Pipeline modules

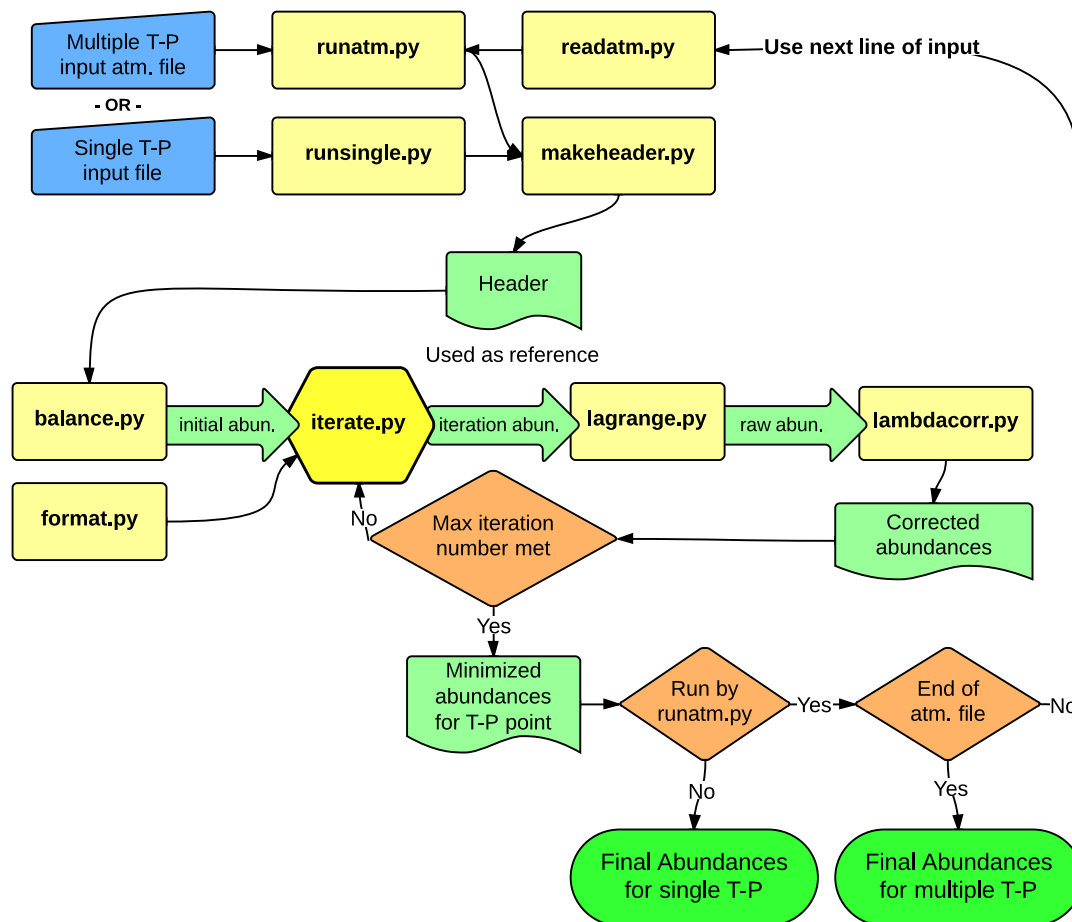


Figure 2: Layout of the pre-pipeline and pipeline packages. Files read and created by the code are in green and user inputs are in blue. The modules included are listed in the text.

1. **prepipe.py**: This module runs the `readJANAF.py` and `makestoich.py` modules and provides their common setup.
2. **readJANAF.py**: Reads data from all available NIST-JANAF Thermochemical Tables and returns ASCII files.
3. **makestoich.py**: Reads the chemical formula to obtain species names and their stoichiometric coefficients from each JANAF file, and bulk dex elemental abundances from an ASCII file based on the Asplund et al. (2009) solar photosphere abundances. The code produces an output file containing species, stoichiometric coefficients, and bulk abundances.
4. **runsingl.py**: This module will run TEA for a single  $T - P$  point.
5. **runatm.py**: This module will run TEA over a pre-atmosphere file containing a list of  $T - P$  points.
6. **readatm.py**: This module reads the pre-atmospheric file with multiple  $T - P$  points.
7. **makeheader.py**: Combines the stoichiometric information, Gibbs free energy per species at specific temperatures, and the user input to create a single file used as a header for the pipeline.
8. **balance.py**: Takes species and stoichiometric information to establish viable, mass-balanced, initial mole numbers.
9. **format.py**: Auxiliary program that manages in/out operations in each piece of the pipeline.
10. **lagrange.py**: Takes data from the most recent iteration's corrected mole numbers and implements the Lagrangian method for minimization. Produces output with raw, non-corrected mole numbers for each species (values are temporarily allowed to be negative).
11. **lambdacorr.py**: Takes non-corrected mole numbers and implements lambda correction to obtain only valid, positive number of moles. Output is the iterations corrected mole numbers for each species.
12. **iterate.py**: Driver program that repeats *lagrange.py* and *lambdacorr.py* until a pre-defined maximum number of iterations is met.

## 1.5 Code validity and comparison with other methods

To test the validity of our code, we first used the example from White et al. (1958). We determined the composition of the gaseous species arising from the combustion of a mixture of hydrazine and oxygen at  $T = 3500$  K and the pressure of  $750 \text{ psi} = 51.034 \text{ atm}$ . We used the free energy functions and  $b$  values from their Table 1, and produced the same final abundances as they did with slightly higher numerical precision (see our Table 1).

We also compared the TEA code with our TEBS (Thermochemical Equilibrium by Burrows & Sharp) code that implements the Burrows & Sharp (1999) analytical method for calculating the abundances of the five major molecular species present in hot-Jupiter atmospheres ( $\text{N}_2$ ,  $\text{H}_2\text{O}$ ,  $\text{CH}_4$ ,  $\text{CO}$ ,  $\text{NH}_3$ ). The TEBS code uses Asplund et al. (2009) dex elemental abundances, but does not need Gibbs free energies of formation from JANAF tables.

Table 1: Comparison White et al. *vs.* TEA

Species	$\frac{g_i^0(T)}{RT}$	White et al. final abundances	TEA final abundances
H	-10.021	0.040668	$4.0654772 \times 10^{-2}$
H <sub>2</sub>	-21.096	0.147730	$1.4771009 \times 10^{-1}$
H <sub>2</sub> O	-37.986	0.783153	$7.8318741 \times 10^{-1}$
N	-9.846	0.001414	$1.4138556 \times 10^{-3}$
N <sub>2</sub>	-28.653	0.485247	$4.8524791 \times 10^{-1}$
NH	-18.918	0.000693	$6.9312544 \times 10^{-4}$
NO	-28.032	0.027399	$2.7397203 \times 10^{-2}$
O	-14.640	0.017947	$1.7941230 \times 10^{-2}$
O <sub>2</sub>	-30.594	0.037314	$3.7308528 \times 10^{-2}$
OH	-26.111	0.096872	$9.6857103 \times 10^{-2}$

As another comparison, we used the open-source thermochemical equilibrium code CEA (Chemical Equilibrium with Applications, <http://www.grc.nasa.gov/WWW/CEAWeb>). This code uses the Newton-Raphson descent method within the Lagrange optimization scheme to solve for chemical abundances. Their approach is described in Gordon & McBride (1994) and Zeleznik & Gordon (1960, 1968). The thermodynamic data included in the CEA code are partially from JANAF tables (Chase 1986) that we used in our TEA code, but also from numerous other sources (i.e., Cox et al. 1982, Gurvich et al. 1989, McBride et al. 1993). The comparison between TEA, TEBS and CEA is shown on Figure 3.

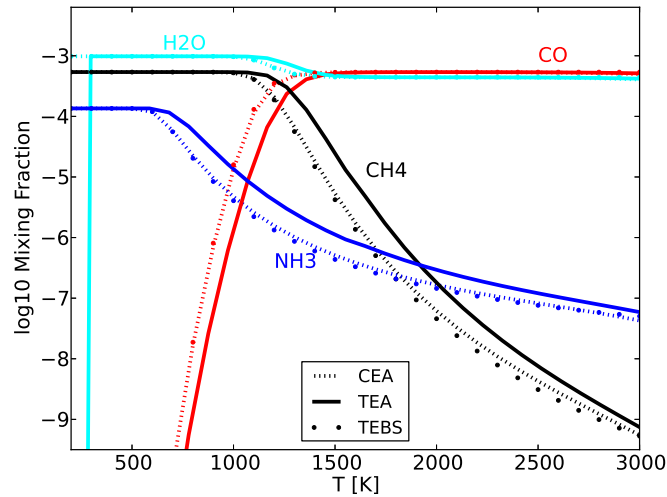


Figure 3: Comparison of the TEA results with the results from CEA and TEBS. TEBS is an analytic method, while CEA and TEA are numerical methods. We show the most abundant spectroscopically-active species in the infrared that are expected to be present in hot-Jupiter atmospheres, that all codes can cover. We run the codes for the same range of temperatures and the same atmospheric pressure of  $P = 1$  bar. The differences between the results are due to different data sources used for the tabulated Gibbs free energies of formation. TEA, in addition, has incorporated water condensation below  $T = 273$  K.

In Figure 4, right panel, we reproduced Figure 17 from Sharp & Burrows (2007), left panel.

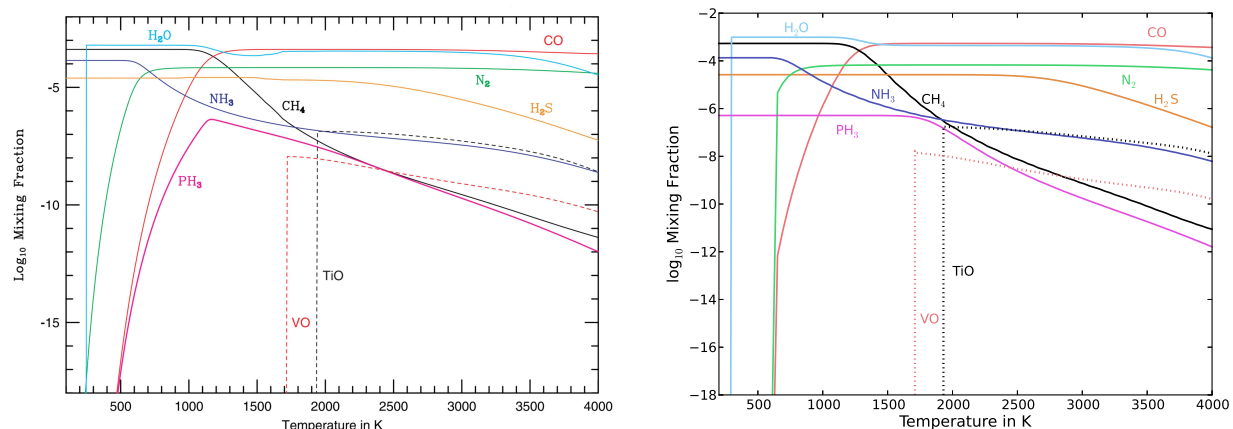


Figure 4: Comparison of the mixing fractions as a function of temperature at a total gas pressure of 1 bar between Sharp & Burrows (2007), Figure 17, left panel, and TEA, right panel. The solid curves depict the mixing fraction for  $\text{CH}_4$  (black),  $\text{CO}$  (red),  $\text{N}_2$  (green),  $\text{NH}_3$  (blue),  $\text{H}_2\text{O}$  (cyan),  $\text{H}_2\text{S}$  (orange), and  $\text{PH}_3$  (magenta), and the two dashed curves depict the mixing ratio for  $\text{TiO}$  (black) and  $\text{VO}$  (red). In the right panel, at 273 K,  $\text{H}_2\text{O}$  drops effectively to zero due to the condensation of ice. With decreasing temperature, both  $\text{TiO}$  and  $\text{VO}$  rise as they associate, then sharply drop to effectively zero when condensates involving Ti and V form. The TEA code is specialized for the gaseous species solely, thus, good comparison is observed for the temperature range above 1000 K.

## References

- Asplund, M., Grevesse, N., Sauval, A. J., & Scott, P. 2009, *ARA&A*, 47, 481
- Burrows, A. & Sharp, C. M. 1999, *ApJ*, 512, 843
- Chase, M. W. 1986, *JANAF thermochemical tables*
- Chase, M. W., Curnutt, J. L., Downey, J. R., et al. 1982, *Journal of Physics Conference Series*, 11, 695
- Cox, J. D., Wagman, D. D., & Medvedev, V. A. 1982, Hemisphere Publishing Corp. New York
- Eriksson, G. 1971, *Acta Chem.Scand.*
- Gordon, S. & McBride, B. J. 1994, *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications. I. Analysis*, Reference Publication RP-1311, NASA, describes theory and numerical algorithms behind CEA computer program
- Gurvich, L. V., Veyts, I. V., & Alcock, C. B. 1989, Hemisphere Publishing Corp. New York, 1, 1
- Madhusudhan, N. & Seager, S. 2009, *ApJ*, 707, 24
- Madhusudhan, N. & Seager, S. 2010, *ApJ*, 725, 261
- McBride, B. J., Gordon, S., & Reno, M. A. 1993, *Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species*, Technical Memorandum TM-4513, NASA, this describes the pre-1994 7-coefficient fit, which is used in Cantera.
- Petaev, M. I. & Wood, J. A. 1998, *Meteoritics and Planetary Science*, 33, 1123
- Seager, S. 1999, PhD thesis, Harvard University
- Seager, S. 2010, in *Exoplanet Atmospheres: Physical Processes* (Princeton University Press)
- Sharp, C. M. & Burrows, A. 2007, *ApJS*, 168, 140
- Sharp, C. M. & Huebner, W. F. 1990, *ApJS*, 72, 417
- White, W. B., Johnson, S. M., & Dantzig, G. B. 1958, *J. Chem. Phys.*, 28, 751
- Zelevnik, F. J. & Gordon, S. 1960, *An analytical investigation of three general methods for calculating chemical equilibrium compositions*, Technical Note TN-473, NASA
- Zelevnik, F. J. & Gordon, S. 1968, *Calculation of complex chemical equilibria*, Technical Memorandum TM X-52303, NASA