Zac Schrecengost MAT 499

Integration of The Saha Equation of State into AstroBEAR

An overwhelming percentage of the matter in the universe consists of plasma; a fundamental state of matter categorized as an assembly of large numbers of unbound charged particles. Due to the electric and magnetic fields that exist within plasmas, a more powerful form of hydrodynamics, known as magnetohydrodynamics (MHD), must be used to account for these fields. The AstroBEAR software package is currently one of the most advanced MHD simulation codes. There are a variety of equation of state (EOS) methodologies that can be used when solving these types of problems. An equation of state is an equation that can be used to compute all thermodynamic quantities given only two thermodynamic variables. This paper will briefly discuss AstroBEAR and MHD, and then outline the math employed by a particular method, known as the Saha equation of state, as well as the approximation techniques needed to integrate the Saha EOS into AstroBEAR.

AstroBEAR

Developed at the University of Rochester, AstroBEAR is an adaptive mesh refinement code for computational astrophysics. Its numerical methods focus on the conservative form of the ideal equations of MHD, which are given in equation 1. With the focus of this paper not centered on the MHD equations, it is still important to note the various quantities the are associated with "normal" hydrodynamics, such as the velocities and densities, as well as the quantities that account for electromagnetics, such as electric and magnetic fields.

The current EOS that AstroBEAR uses to handle the ionization of elements is to dynamically track the ratios of the number of atoms in a particular ionization state to the total number of atoms of that element, which are known as the ionization fractions. This method does not require thermal equilibrium, the condition where a single temperature can be used to describe

all particles in a region. Tracking ionization fractions is effective while only considering a small number of elements with only a few ionization states; however, to include more elements in the plasma mixture, this method becomes computationally intensive.

The objective of implementing the Saha EOS into AstroBEAR is to provide a method to incorporate more elements, without the computational cost, by considering local thermal equilibrium. The remaining content of this paper discusses the Saha EOS, and the methods that are used to integrate it into the existing structure of AstroBEAR.

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho v_x\\ \rho v_y\\ \rho v_z\\ \varepsilon\\ B_x\\ B_y\\ B_z\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho v_x\\ \rho v_x^2+P+B^2/2-B_x^2\\ \rho v_y v_x-B_x B_y\\ \rho v_z v_x-B_x B_z\\ (\varepsilon+P+\boldsymbol{B}^2/2)v_x-B_x(\boldsymbol{B}\cdot\boldsymbol{v})\\ 0\\ -E_z\\ E_y\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\rho v_y\\ \rho v_x v_y-B_y B_x\\ \rho v_y^2+P+B^2/2-B_y^2\\ \rho v_z v_y-B_y B_z\\ (\varepsilon+P+\boldsymbol{B}^2/2)v_y-B_y(\boldsymbol{B}\cdot\boldsymbol{v})\\ E_z\\ 0\\ -E_x\end{bmatrix}+\frac{\partial}{\partial z}\begin{bmatrix}\rho v_z\\ \rho v_x v_z-B_z B_x\\ \rho v_y v_z-B_z B_y\\ \rho v_z^2+P+B^2/2-B_z^2\\ (\varepsilon+P+\boldsymbol{B}^2/2)v_y-B_z(\boldsymbol{B}\cdot\boldsymbol{v})\\ -E_y\\ E_x\\ 0\end{bmatrix}=\boldsymbol{S} \tag{1}$$

The Saha Equation

The Saha equation is used in iteratively to determine a convergent value of the electron density N_e (electrons per volume). The implementation used here takes the density ρ and the temperature T to reach convergence. Saha's equation yields the ratio of the number of $r+1$ ionized particles of element i to the number of r ionized particles of element i , and is given by

$$\frac{n_i^{r+1}}{n_i^r}=\left(\frac{2\pi mk}{h^2}\right)^{3/2}T^{3/2}\frac{2B_i^{r+1}\exp\left(-\chi_i^r/kT\right)}{B_i^r N_e} \tag{2}$$

where m is the mass of an electron, k is Boltzmann's constant, h is Planck's constant, and, for

element i and ionization level r, B_i^r is the partition function and χ_i^r is the ionization energy. For a given element and ionization level, all terms on the right side of equation 2 are constant except for the temperature, which is provided, and the electron density, an initial guess value of which is used to determine the ratio n_i^{r+1}/n_i^r. The degree of the r^{th} ionization of element i is defined as

$$y_i^r = \frac{n_i^r}{n_i} \tag{3}$$

where n_i is the total number of particles, in all states of ionization, of element i. Using equation 3, the number of free electrons contributed by an average particle of element i is given by

$$\nu_e(i) = \sum_{s=0}^{z(i)} s\, y_i^s \tag{4}$$

Notice that first term in the summation in equation 4 is always zero. This means that calculating y_i^0 never needs to occur. The new value of the electron density is calculated by

$$N_{e1} = \sum_i \nu_e(i)\, n_i = \rho\, N_o \sum_i \frac{x_i}{A_i} \nu_e(i) \tag{5}$$

where N_o is Avogadro's number, x_i is the number of grams of element i per gram of mixture, and A_i is the molecular weight of an atom of element i. If, for a given tolerance TOL, equation 6 holds,

$$\frac{|N_e - N_{e1}|}{N_e} < TOL \tag{6}$$

then the algorithm finishes. Otherwise, the calculation is repeated, using equation 7 or equation 8, until convergence is reached.

$$N_e = N_{e1} \tag{7}$$

$$N_e = \frac{N_e + N_{e1}}{2} \tag{8}$$

Depending on the initial guess of the electron density, the new value determined from the Saha equation can oscillate around the convergent value, leading to very slow convergence, if at all. For this reason, equation 7 is typically used to adjust the value of the electron density, however, if the values returned from iteration k and iteration $k+2$ are within a set tolerance, equation 8 is used to set the new value of the electron density, which leads to convergence in a reasonable amount of time. Taking a mixture of pure hydrogen as an example, the plot of y_H^1 vs. temperature is given below.

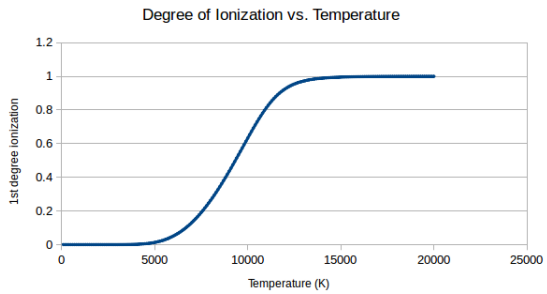

Figure 1: Hydrogen first degree ionization

The key to calculating y_i^r is in calculating the ratios n_i^{r+1}/n_i^r for all ionization levels of a particular element. Since n_i, the total number of particles in all states of ionization of element i is unknown, y_i^r must be expressed in terms of the ratios. Although there are different ways to calculate y_i^r, the method outlined below can easily be implemented in software. This makes it a feasible method when considering elements with more than two or three ionization levels. Taking hydrogen as a simple example, with ionization levels 0 and 1,

$$y_H^1 = \frac{n_H^1}{n_H^0 + n_H^1} = \frac{1}{\frac{1}{\frac{n_H^1}{n_H^0}} + 1} \tag{9}$$

where $\frac{n_H^1}{n_H^0}$ comes from the Saha equation. The algorithmic nature of these calculations does not become clear until considering an element with a more ionization levels. Carbon, for example, has 7 ionization levels, ranging from 0 to 6. The formulation of the expressions for the various y_i^r terms can be determined by looking at the first few ionization levels. The first, second, and third degrees are given in equations 10, 11, and 12.

$$y_C^1 = \frac{n_C^1}{n_C^0+n_C^1+n_C^2+n_C^3+n_C^4+n_C^5+n_C^6} = \frac{1}{\frac{n_C^0}{n_C^1}+\frac{n_C^1}{n_C^1}+\frac{n_C^2}{n_C^1}+\frac{n_C^3}{n_C^1}+\frac{n_C^4}{n_C^1}+\frac{n_C^5}{n_C^1}+\frac{n_C^6}{n_C^1}} =$$

$$= \frac{1}{\frac{1}{\frac{n_C^1}{n_C^0}}+1+\frac{n_C^2}{n_C^1}+\frac{n_C^2}{n_C^1}\frac{n_C^3}{n_C^2}+\frac{n_C^2}{n_C^1}\frac{n_C^3}{n_C^2}\frac{n_C^4}{n_C^3}+\frac{n_C^2}{n_C^1}\frac{n_C^3}{n_C^2}\frac{n_C^4}{n_C^3}\frac{n_C^5}{n_C^4}+\frac{n_C^2}{n_C^1}\frac{n_C^3}{n_C^2}\frac{n_C^4}{n_C^3}\frac{n_C^5}{n_C^4}\frac{n_C^6}{n_C^5}} \quad (10)$$

$$y_C^2 = \frac{n_C^2}{n_C^0+n_C^1+n_C^2+n_C^3+n_C^4+n_C^5+n_C^6} = \frac{1}{\frac{n_C^0}{n_C^2}+\frac{n_C^1}{n_C^2}+\frac{n_C^2}{n_C^2}+\frac{n_C^3}{n_C^2}+\frac{n_C^4}{n_C^2}+\frac{n_C^5}{n_C^2}+\frac{n_C^6}{n_C^2}} =$$

$$= \frac{1}{\frac{1}{\frac{n_C^1}{n_C^0}\frac{n_C^2}{n_C^1}}+\frac{1}{\frac{n_C^2}{n_C^1}}+1+\frac{n_C^3}{n_C^2}+\frac{n_C^3}{n_C^2}\frac{n_C^4}{n_C^3}+\frac{n_C^3}{n_C^2}\frac{n_C^4}{n_C^3}\frac{n_C^5}{n_C^4}+\frac{n_C^3}{n_C^2}\frac{n_C^4}{n_C^3}\frac{n_C^5}{n_C^4}\frac{n_C^6}{n_C^5}} \quad (11)$$

$$y_C^3 = \frac{n_C^3}{n_C^0+n_C^1+n_C^2+n_C^3+n_C^4+n_C^5+n_C^6} = \frac{1}{\frac{n_C^0}{n_C^3}+\frac{n_C^1}{n_C^3}+\frac{n_C^2}{n_C^3}+\frac{n_C^3}{n_C^3}+\frac{n_C^4}{n_C^3}+\frac{n_C^5}{n_C^3}+\frac{n_C^6}{n_C^3}} =$$

$$= \frac{1}{\frac{1}{\frac{n_C^1}{n_C^0}\frac{n_C^2}{n_C^1}\frac{n_C^3}{n_C^2}}+\frac{1}{\frac{n_C^2}{n_C^1}\frac{n_C^3}{n_C^2}}+\frac{1}{\frac{n_C^3}{n_C^2}}+1+\frac{n_C^4}{n_C^3}+\frac{n_C^4}{n_C^3}\frac{n_C^5}{n_C^4}+\frac{n_C^4}{n_C^3}\frac{n_C^5}{n_C^4}\frac{n_C^6}{n_C^5}} \quad (12)$$

Expressing the ionization degrees in this way shows how the formulation changes with the

different levels of ionization. Taking into account the number of ionization levels and the current degree of ionization being computed, this can be implemented in software in a straightforward manner. This also allows the Saha equation to consider many elements in a plasma mixture because including more ionization levels can be accomplished algorithmically.

Equation of State

Once the Saha equation converges, various thermodynamic quantities can be calculated. In particular, the pressure, internal energy, and specific heat capacity can be obtained. Here, the case of a mixture of pure hydrogen will be considered. The calculations of these quantities rely on the ionization fraction x, which is given by the same equation used to determine the degree of ionization; for hydrogen, this is equation 9. The pressure depends on the total number of particles, Boltzmann's constant, and the temperature. Since the number of hydrogen atoms n_H and the number of free electrons n_e contribute to the total number of particles, the pressure is given by

$$P_g = (n_H + n_e)kT \tag{13}$$

The sum of n_H and n_e can be related to the ionization fraction, so the gas pressure can be expressed as

$$P_g = (1+x)\frac{k}{m_H}\rho T \tag{14}$$

where m_H is the mass of a hydrogen atom. The internal energy is also related to the total number of particles, Boltzmann's constant, and the temperature, as well as the energy associated with the ionization of each excited particle. Therefore, the internal energy is

$$U_g = 1.5(n_H + n_e)kT + n_e\chi \tag{15}$$

and relating quantities to the ionization fraction yields

$$U_g = 1.5 P_g + x \frac{\chi}{m_H} \rho \tag{16}$$

The specific heat at constant volume is related to the internal energy by

$$c_{V,g} = \left[\frac{\partial (U_g / \rho)}{\partial T} \right]_\rho \tag{17}$$

Without going through the differentiation, the specific heat, using the ionization fraction, is given by

$$c_{V,g} = \frac{k}{m_H} \left[1.5(1+x) + (1.5 + \chi / kT)^2 \frac{x(1-x)}{(2-x)} \right] \tag{18}$$

These three thermodynamic variables are the particular quantities AstroBEAR deals with.

AstroBEAR Integration

Despite the Saha equation requiring density and temperature, AstroBEAR does not track temperature during its simulations. The density is always available along with one of the following: pressure, internal energy, total energy, or specific heat. Currently, AstroBEAR is able to calculate the quantities not available by taking the mixture to be an ideal gas.

To calculate these quantities with the Saha equation, the temperature must be known. The approach taken here is to create tables, explained below, at the beginning of the simulation, and then use the tables to determine the temperature of the mixture. For example, a table is created where density varies across the columns and temperature varies down the rows, and the contents of the table are internal energies. A temperature is then determined from a given density and internal energy, as described below. This temperature and density is then used in the Saha equation to determine the other thermodynamic variables.

As an initial strategy to determine the temperature, an approximation using two planes was used. Given a density ρ and an internal energy U_g, the first density larger than the given density and the first internal energy larger than the given internal energy are found; denoted ρ_2 and

$U_{g,22}$, respectively. The last density smaller than the given value is ρ_1 with a corresponding internal energy $U_{g,11}$. Due to the structure of the table, given in the figure below, the desired T is between T_1, which is associated with ρ_1 and $U_{g,11}$, and T_2, which is associated with ρ_2 and $U_{g,22}$. The four points of interest are thus denoted $A=(\rho_2, T_2, U_{g,22})$, $B=(\rho_2, T_1, U_{g,12})$, $C=(\rho_1, T_2, U_{g,21})$, and $D=(\rho_1, T_1, U_{g,11})$.

	ρ_1	ρ_2
T_1	$U_{g,11}$	$U_{g,12}$
T_2	$U_{g,21}$	$U_{g,22}$

Figure 2: Points of Interest

The equations of two planes are generated considering the equation below.

$$a\rho + bT + cU_g + offset = 0 \tag{19}$$

The first plane uses points A, B, and C to calculate the coefficients and point D to calculate the offset. The second plane uses points D, B, and C to calculate the coefficients and point A to calculate the offset. The temperature corresponding to ρ and U_g is calculated using each plane equation and the average temperature is reported as the approximated value.

To test the accuracy of this method, another table was created for a range of densities and internal energies such that the values would be in the range of those in the initial table. For each internal energy in the second table, a temperature was determined using the above method. This temperature was then used in Saha's equation to calculate the corresponding internal energy. The percent error of the input internal energy and the calculated internal energy was calculated and stored. For temperatures and densities of interest, the largest percent error was approximately 5.5% with the average error being in the range of 1.5 to 3%. These errors are roughly half that of the errors that result from only approximating temperature with one plane.

Future Work

The method of approximating a temperature from a density and internal energy has been tested and the magnitude of the errors is acceptable. Calculating a temperature from a density and pressure, and from a density and specific heat, must still be implemented and tested. Alternative approximation methods, using splines for example, also must be examined. Once these methods are employed and tested, the Saha EOS can be integrated with the AstroBEAR software package. After integration, a test problem must be developed such that the effects of ionization can clearly be seen versus the results from an ideal gas EOS.

Conclusion

With the variety of methods to compute equation of state variables, the Saha equation provides a means to incorporate the effects of ionization. While the assumption of thermal equilibrium is required, the ionization of many elements can be handled in an algorithmic manner. Tables generated at the beginning of a simulation provide a means for AstroBEAR to compute equation of state variables using the Saha equation. Although there are several steps between what was outlined in this paper and what AstroBEAR needs for full implementation, the foundation has been formed for this future work.

References

Cunningham, Andrew J., et al., Simulating Magnetohydrodynamical Flow with Constrained Transport and Adaptive Mesh Refinement; Algorithms & Tests of the AstroBEAR Code, *APJS: March 2009.*

Kanbur, Shashi. An investigation into classical Cepheid Pulsation. *Thesis.*

Knapp, G. R., Ionization, Saha Equation. *Course Notes,* http://www.astro.princeton.edu /~gk/A403/ioniz.pdf