Using Tabular EOS at Low Temperatures

MultiMat 2013

September 2-6, 2013

Robert A. Managan

Lawrence Livermore National Laboratory

LLNL-POST-593432

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Abstract

In multi-physics hydro-codes there are many algorithms that update the internal energy of a zone. It is the job of the equation of state to take this new internal energy and provide an updated temperature. When the physics of the problem requires that the code use separate electron and ion temperatures problems can arise when determining the new temperature. The cause of this problem is that in the LEOS tables the electron energy is zero until there are free electrons while the algorithms that couple energy into the electrons generally assume a plasma and don't account for whether the electrons are bound or free.

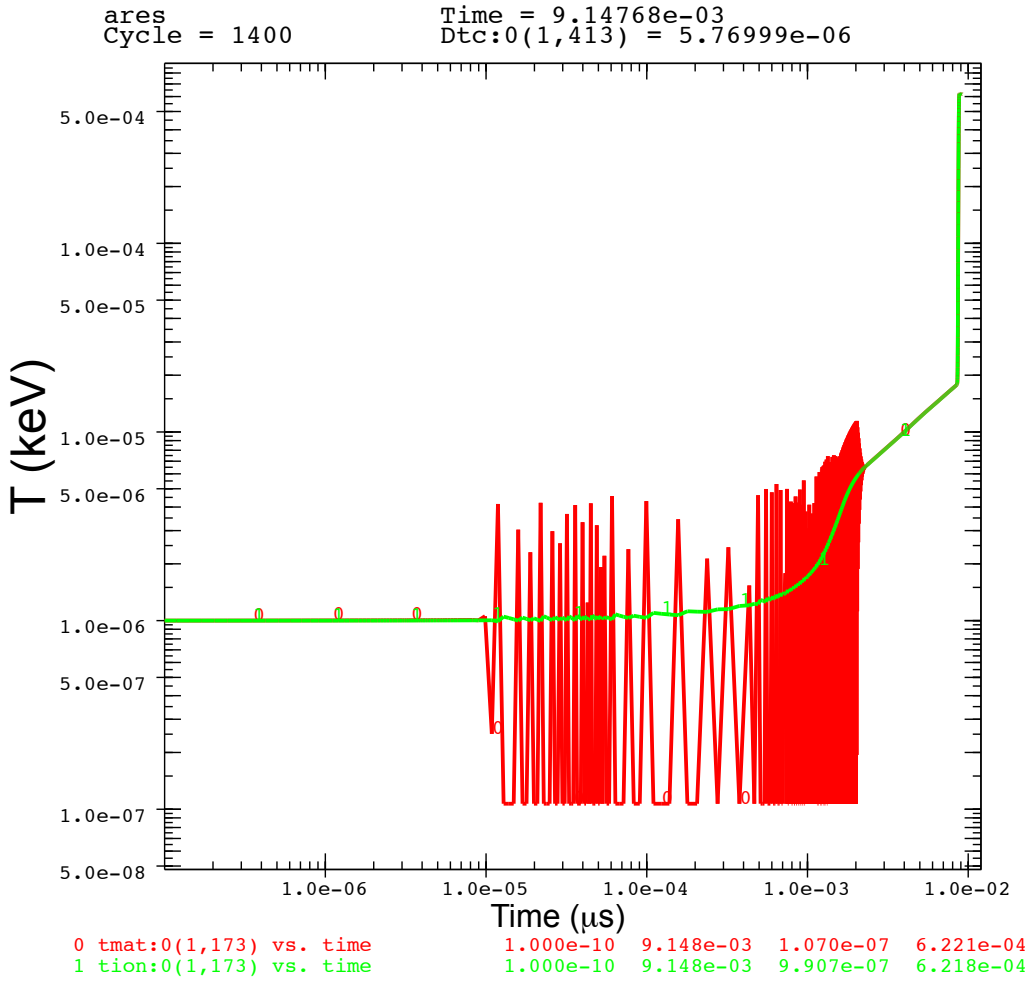
A new approach for dealing with these issues is presented that avoids discontinuous jumps in the electron temperature. It does this by smoothly switching from a total energy solution to separate temperature updates for electrons and ions.

Introduction

- The LEOS tables are built with separate tables for the cold, electron, and ion values. These three tables are also combined into a table with the total energy. When a multi-physics code is run in a mode that has separate electron and ion temperatures we need to use the separate tables for the EOS. However, most multi-physics codes are based on algorithms that are derived in the plasma physics regime and make different assumptions in the low temperature limit than the EOS tables do. In particular the table's electron energy is zero at low temperatures until there are free electrons, but the radiation-electron coupling transfers energy to the electrons without ionizing them. The result is that once a little energy is put into the electron field the electron temperature jumps up to the ionization threshold only to drop the next cycle when the energy is coupled to the ions.

The problem we want to solve

In an ICF calculation that starts at cryogenic temperatures a zone in the ablator shows this temperature behavior as the radiation heats it and the main ablation shock arrives.



The old algorithm

- The basic algorithm for updating the temperature is to iteratively update the temperature until the energy we look up matches the energy of the zone.
- Here the primed variables are the current guess and unprimed are the value we want to match.

$$E_e = E'_e(T_e) + \frac{\partial E_e}{\partial T}\Delta T_e$$

$$E_i = E'_i(T_i) + \frac{\partial E_i}{\partial T}\Delta T_i$$

The old algorithm (2)

- With this algorithm once there is any energy in the electrons the electron temperature will jump up to the temperature at the current density where the electron energy becomes non-zero.
- For DT gas this is several thousand degrees Kelvin.
- Since NIF capsules start at cryogenic temperatures this large difference between the electron and ion temperatures causes problems.

New algorithm

- The new method uses an interpolating function, $f(T) = 2^{-(T/T_0)^2}$, to switch from total energy to separate energies, T_0 is the "half-way temperature". Currently $T_0 = 5 \times 10^{-3}$ keV at which temperature all materials have begun to ionize. Different forms for the interpolating function or different values of T_0 could be made but this choice seems to work.

New algorithm (2)

- This algorithm works with two effective energies. At low temperatures they are half of the total energy. At high temperatures they are the electron energy and the sum of the cold and the ion energy.

$$E_e^* = \frac{E_e + f(T_e)\,(E_i + E_c)}{1 + f(T_e)}$$

$$E_i^* = \frac{f(T_e)E_e + (E_i + E_c)}{1 + f(T_e)}$$

New algorithm temperature update

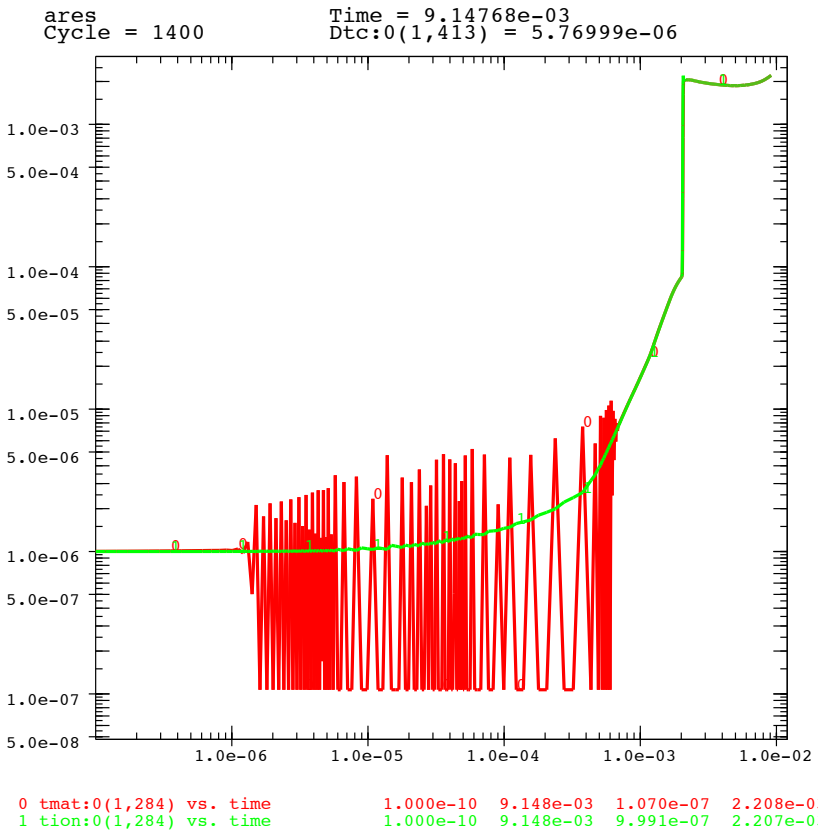
- The temperature update comes from treating E_e^* and E_i^* as functions of the electron temperature and ion temperature respectively.

$$\frac{E_e + f(T_e)\,(E_i + E_c)}{1+f} = \frac{E'_e + f(T_e)\,(E'_i + E_c)}{1+f} + \frac{1}{1+f}\left[\frac{\partial E'_e}{\partial T_e} - \frac{df}{dT_e}\frac{E'_e - (E'_i + E_c)}{1+f} + f\frac{\partial E'_i}{\partial T_i}\right]\Delta T_e$$

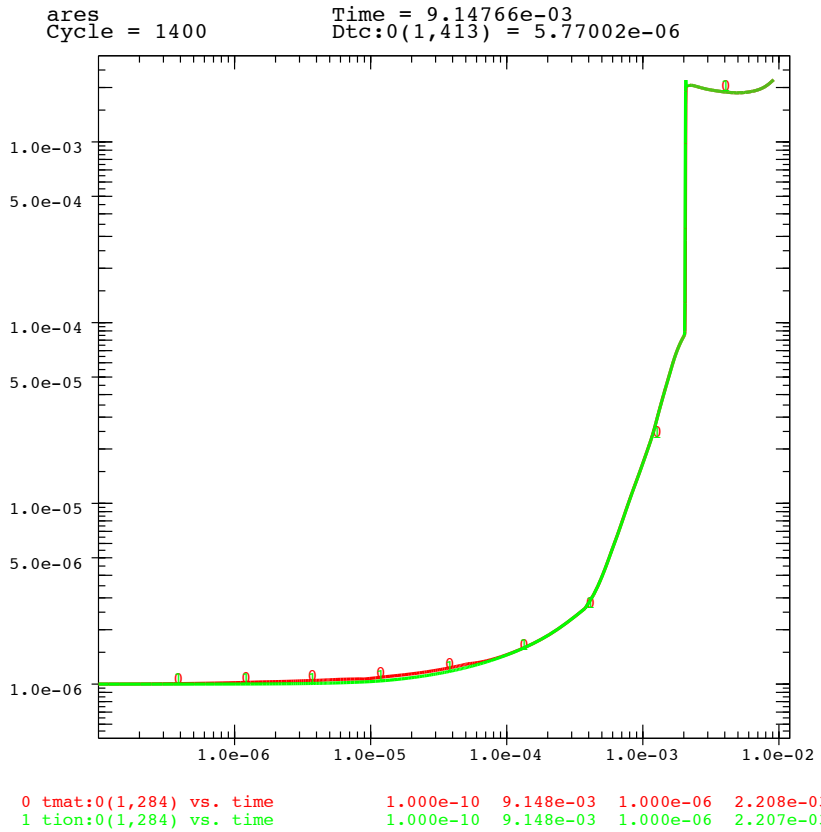
$$\frac{f(T_e)E_e + (E_i + E_c)}{1+f} = \frac{f(T_e)E'_e + E'_i + E_c}{1+f} + \frac{1}{1+f}\left[f\frac{\partial E'_e}{\partial T_e} + \frac{df}{dT_e}\frac{E'_e - (E'_i + E_c)}{1+f} + \frac{\partial E'_i}{\partial T_i}\right]\Delta T_i$$

Time dependence of temperatures – outer ablator

Old algorithm

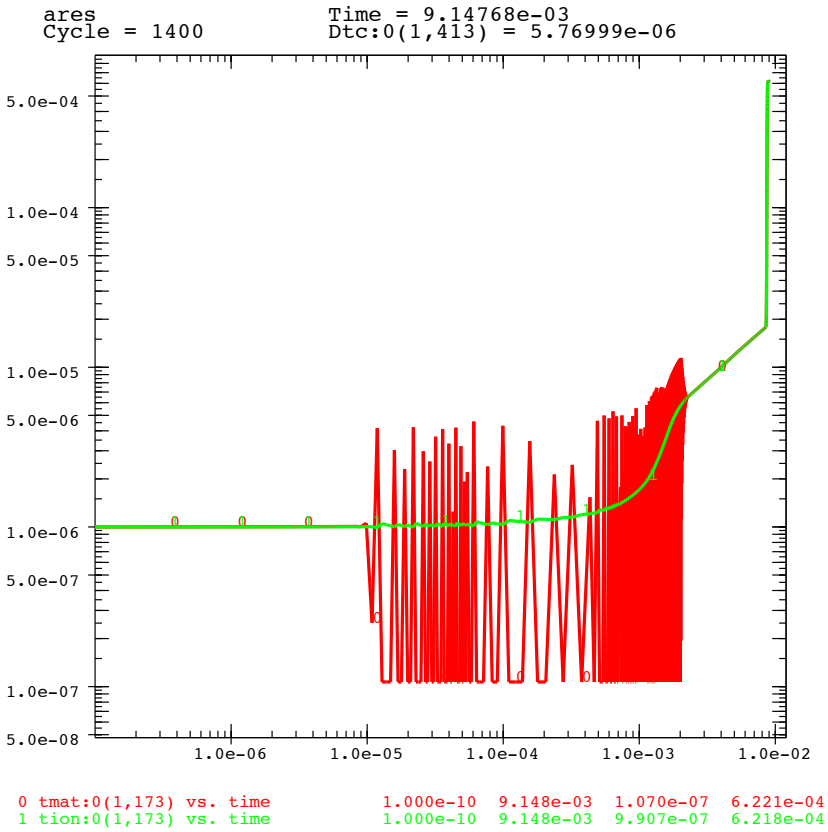


New algorithm

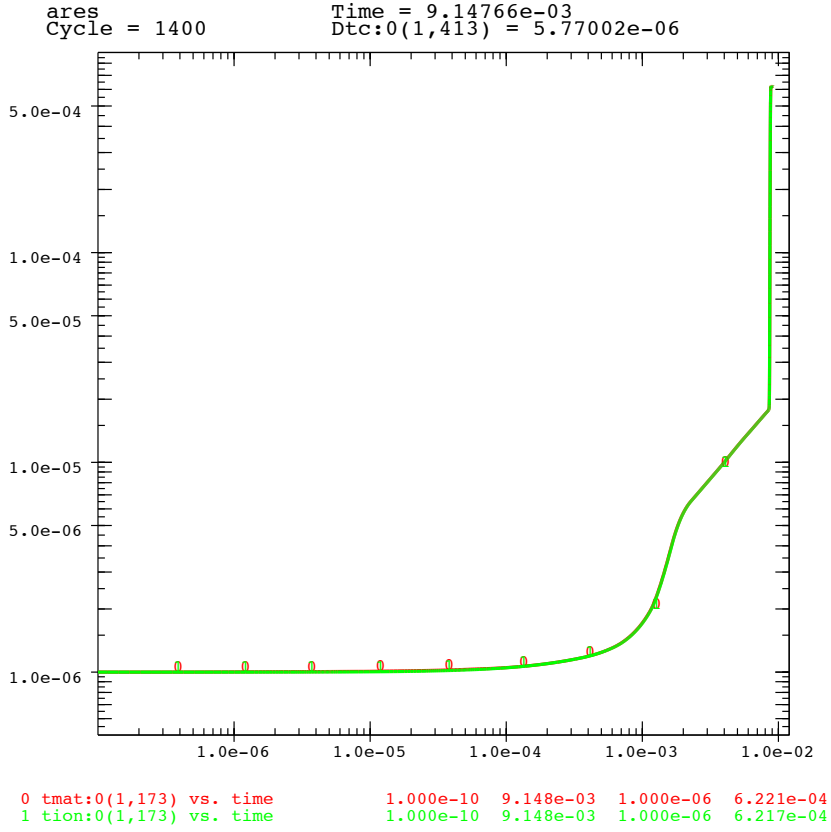


Time dependence of temperatures – inner ablator

Old algorithm



New algorithm



Issues

- Technically E_e^* and E_i^* should depend on the updates to both the electron and ion temperatures.

$$\frac{E_e + f(T_e)\,(E_i + E_c)}{1+f} = \frac{E'_e + f(T_e)\,(E'_i + E_c)}{1+f} + \frac{f}{1+f}\frac{\partial E'_i}{\partial T_i}\Delta T_i + \frac{1}{1+f}\left\{\frac{\partial E'_e}{\partial T_e} - \frac{df}{dT_e}\frac{E'_e - (E'_i + E_c)}{1+f}\right\}\Delta T_e$$

$$\frac{f(T_e)E_e + (E_i + E_c)}{1+f} = \frac{f(T_e)E'_e + E'_i + E_c}{1+f} + \frac{1}{1+f}\frac{\partial E'_i}{\partial T_i}\Delta T_i + \frac{1}{1+f}\left[f(T_e)\frac{\partial E'_e}{\partial T_e} + \frac{df}{dT_e}\frac{E'_e - (E'_i + E_c)}{1+f}\right]\Delta T_e$$

Issues

- However, this scheme is unstable since $\frac{\partial E'_e}{\partial T_e}$ is zero at low temperatures (when E_e is zero) and replaced by a small limiting value in the code.
- The adopted scheme is ad hoc but appears to work well.
- Similarly, the modified heat capacity is only used in the temperature update and not in the rest of the code.

Lawrence Livermore
National Laboratory