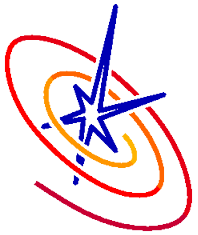

EOS and Opacity Units

Klaus Weide
University of Chicago

FLASH Tutorial at RAL
May 31, 2012

Outline

Two parts in the talk

- **EOS**
  - Multispecies
- Opacity

Three-Temperature Model

- Level of Approximation: Three-Temperature Model
- A fluid with three components that
  - May have different temperatures
  - Share same bulk motion (only one fluid velocity vector)
  - “Add up” in terms of pressure, internal energy

- Components will represent ions, electrons, and radiation.
  - Radiation is treated very summarily: moves with the matter, not resolved with respect to frequency or direction.

- As far as FLASH is concerned, “three temperatures” really means “three internal energies” - energies are concerned quantities, prime objects for Hydro.
  - 1T FLASH Hydro solvers already advances total internal energy.

Equations – Three-Temperature Model

- 1T FLASH implements Hydro solver for:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla p = 0$$

$$\frac{\partial \rho e_{\rm tot}}{\partial t} + \nabla \cdot [(\rho e_{\rm tot} + p)\mathbf{u}] = 0$$

$$\frac{\partial \rho e_{\rm i}}{\partial t} + \nabla \cdot (\rho e_{\rm i} \mathbf{u}) = -p\, \nabla \cdot \mathbf{u}$$

- *Internal energy equation is usually redundant.*
- p and e_i are coupled by an EOS equation: $p=p(\rho,e_i)$

Note: We do not want to make ideal gas assumptions!

The basic idea for turning this into 3-temperature system is described in Paul Drake's HEDP book:

- Turn the internal energy equation into three equations
- Each new equation relates partial internal energy with partial pressure:

  $e_i^{(1)}$ and $p^{(1)}$, $e_i^{(2)}$ and $p^{(2)}$, $e_i^{(3)}$ and $p^{(3)}$.

  Here $e_i = e_i^{(1)} + e_i^{(2)} + e_i^{(3)}$ and $p = p^{(1)} + p^{(2)} + p^{(3)}$.

  Some variables redundant, but we neeed to evolve at least 5 even in 1D.
- We need EOS equations to know how $p^{(c)}$ relate to the $e_i^{(c)}$.

HEDP Equations and Approach

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0, \\
\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla P &= 0, \\
\frac{\partial E_{\mathrm{tot}}}{\partial t} + \nabla \cdot [(E_{\mathrm{tot}} + P)\mathbf{u}] &= S_{\mathrm{i}} + S_{\mathrm{e}} + S_{\mathrm{R}} + \nabla \cdot (\kappa_{\mathrm{e}} \nabla T_{\mathrm{e}}) + \nabla \cdot (\kappa_{\mathrm{R}} \nabla T_{\mathrm{rad}}), \\
\frac{\partial}{\partial t} \rho e_{\mathrm{e}} + \nabla \cdot (\rho e_{\mathrm{e}} \mathbf{u}) &= -P_{\mathrm{e}} \, \nabla \cdot \mathbf{u} + \nabla \cdot (\kappa_{\mathrm{e}} \nabla T_{\mathrm{e}}) + k^{(\mathrm{i,e})}(T_{\mathrm{i}} - T_{\mathrm{e}}) + k^{(\mathrm{e,R})}(T_{\mathrm{rad}} - T_{\mathrm{e}}) + S_{\mathrm{e}} \\
\frac{\partial}{\partial t} E_{\mathrm{R}} + \nabla \cdot (E_{\mathrm{R}} \mathbf{u}) &= -P_{\mathrm{R}} \, \nabla \cdot \mathbf{u} + \nabla \cdot (\kappa_{\mathrm{R}} \nabla T_{\mathrm{rad}}) - k^{(\mathrm{e,R})}(T_{\mathrm{rad}} - T_{\mathrm{e}}) + S_{\mathrm{R}},
\end{aligned}
$$

(Simplified) 3T Equations featuring

- Separate internal energies for 3 components (ion, electrons, radiation)
  - Thus 3 “temperatures” are implied – this is not necessarily physical.
- Heat conduction and radiation energy transport (multigroup not shown)
- Temperature coupling between components by emission, absorption, and ion-electron coupling
- External energy source terms (Laser energy deposition)

Use of EOS Calls

- Purpose of Eos calls:
- Make the solution state themodynamically consistent

- Where:
  - After code has changed some variables that are now inconsistent with others.
    - Example: laser energy deposition increases electron energy; it is necessary to recompute combined energy, pressures, temperatures
  - Also when Grid interpolation of cell values has taken place at fine/coarse boundaries

FLASH EOS Interface

- Most important:
- Subroutine Eos(mode, eosData, …)
  - Operates on cell date packed into eosData structure
- Subroutine Eos_wrapped(mode, blockID, …)
  - Operates on a block of the grid
- More on “mode” argument later.

FLASH Approach to HEDP Requirements

- Modified ***Hydro*** implementations as alternative versions to (fundamentally unchanged) 1T implementations
  - 3T code in "`multiTemp`" directories, overrides some routines
  - Does not interfere with traditional (1T) FLASH applications
- 3T ***Eos*** implementations as alternatives to unchanged older implementations
  - 3T code in "`multiTemp`" directories, overrides some routines
  - Additional features (for now) only in 3T variants:
    - Table-based (different from Helmholtz Energy tables for electrons)
    - Multiple materials that can have different Eos types
- New variables in UNK (and eosData) **only for 3T simulations**:
  - "tion","tele","trad" **in addition to** "temp"; similar "eion", "eele', "erad" for "eint"; "pion","pele","prad" for "pres"; similar eosData
- 3T-aware code units use theese new variables;
- Older code / 1T code can continue to work

Code Units for HEDP and Other Units

- RadTrans
  - Using Diffuse unit
- EnergyDeposition (Laser)
- Heatexchange (ion-electron coupling)
- Opacity (material properties unit)

These code units for HEDP can assume existence of 3T UNK variables (tion, tele, eion, eele, etc.).

Other code units need not be aware of 3T features.
Existing simulations can use FLASH as before, as long as they require only 1T physics!

EOS routines for Three-Temperature Model

- In a one-component fluid, pressures are related to internal energies by an Equation Of State:

  $p = p(\rho, e_i)$

- We also need other EOS forms for various purposes: $p = p(\rho,T)$ etc.
- Currently, FLASH code unit Eos provides three modes:

  MODE_DENS_TEMP: computes $e_i(\rho,T)$, $p(\rho,T)$

  MODE_DENS_EINT: computes $T(\rho, e_i)$, $p(\rho, e_i)$

  MODE_DENS_PRES: computes $T(\rho,p)$, $e_i(\rho,p)$

- For three-component operation, added many more, with component and/or total temperature, energy, pressure as inputs or outputs. Sample:

MODE_DENS_TEMP_EQUI - takes total temperature as input and equilibrates so that the components have this temperature.

MODE_DENS_EI_SCATTER - takes total internal energy as input, computes a total temperature for components in equilibrium, and distributes the same temperature, and partial internal energies, pressures, etc. to the components.

MODE_DENS_EI_GATHER - takes given component-wise internal energies as input, computes component-wise and total temperatures, internal energies, pressures, etc.

Enhanced Hydro for 3T

- Added additional (partial) internal energies, also pressures, temperatures, to state vector UNK.
- Added code to core Hydro routines to advance these variables
- Modified state vector in 3-Temperature code includes:

(ρ)
(u_x, u_y, u_z)
(e_{tot})
(e_i)
($e_i^{(1)}$, $e_i^{(2)}$, $e_i^{(3)}$)
(p)
($p^{(1)}$, $p^{(2)}$, $p^{(3)}$)
(T)
($T^{(1)}$, $T^{(2)}$, $T^{(3)}$)
(* others for convenience: gamc game)

Many of these variables are redundant!

Approach to 3T in Hydro

- So does Hydro treat these additional variables?

Approach:

- Take existing time evolution methods for *total* entities, leave as they are.
- Add methods to also evolve in time the additional information about how total internal energy is made up of 3 components.
- Compute component-wise temperatures, and allow temperatures to diverge, via the additional EOS methods.
- Make sure that for additive entities, totals continue to equal sums of components.

  (Think "internal energy" or "pressure" for "entity")

- Note that temperatures do not enter directly into the Hydro computational cycle, they are a byproduct as far as our PPM scheme is concerned.

Gamma 3T EOS

- The “Gamma” 3T Eos implementation:
- Total internal energy is sum of 3 component energies, total pressure is sum of 3 component pressures (ions, electrons, radiation), …
  - ions treated as ideal monatomic gas (gamma = 5/3)
  - electrons treated as ideal monatomic gas (gamma = 5/3)
  - radiation is black body radiation

  - Not possible to model: ionizing/recombining plasma, solid states…
  - partial ionization (at fixed, given ionization) can be handled

Tabulated 3T EOS

- The “Tabulated” Eos implementation:
- Total internal energy is sum of 3 component energies, total pressure is sum of 3 component pressures (ions, electrons, radiation), …
  - Ion internal energy and pressure from IONMIX4 table file
  - Electron internal energy and pressure from IONMIX4 table file
  - radiation is black body radiation

  - possible to model: temperature- and density-dependent ionization
  - In principle, tables can include low temperatures.
  - Ionization level Z is an output from Eos

- IONMIX4 is a table format documented in the FLASH Users Guide.
- Tables for eion,eele,pion,pele, Z as functions of (temperature, density).
- We can convert, e.g., SESAME Eos data to this format.

Multispecies Unit

- Simulations that model multiple materials use the *Multispecies* unit.
- Multispecies provides a simple database of some material properties
  - See section “Multifluid database contents:” in a FLASH log file
  - Not all properties are used in a given simulation
- Important properties for HEDP:
  - A (avg. atomic mass); maybe Zmin, …
  - Names of mateiral-specific tables
- Initialization of Multispecies properties
  - Based on setup command line: e.g., `species=cham,targ`
  - Values assigned in flash.par file

Multiple Materials EOS

- The “Multitype” Eos implementation:
- Combines Eos output data from several materials that may be present in a cell
  - Materials are tracked by mass fractions in solution vector
    - In the **current** approach, materials are assumed to be completely mixed
  - Each material can have a different “Eos type”, currently:
    - Gamma (ideal gas)
    - Tabulated
- Hence, **MTMMMT** for multi-temp, multimaterial, multitype Eos
- This is the Eos implementation to use for realistic HEDP simulations. See LaserSlab setup!
- Easiest way to configure FLASH for this: use setup shortcut

```
./setup –auto … +mtmmmt
```

Outline

Two parts in the talk

- EOS
  - Multispecies
- **Opacity**

Opacities for 3T

- The Opacity unit in physics/materialProperties/Opacity provides opacities for multigroup radiation diffusion to the RadTrans unit.
- Opacities are group averages for an energy group (frequency range) as required by multigroup diffusion.
- Constant opacity implementations are available for testing, see

  Opacity/OpacityMain/

Opacities for 3T

- The Opacity unit in physics/materialProperties/Opacity provides opacities for multigroup radiation diffusion to the RadTrans unit.
- Opacities are group averages for an energy group (frequency range) as required by multigroup diffusion.
- The usual implementation for HEDP is table based, under

  Opacity/OpacityMain/Multispecies

  - Opacities for each material are read from an IONMIX4 table file
  - Opacities for a cell and energy group are computed by combining the material-specific opacities.
  - Mass fractions in solution vector are used for composition.

- IONMIX4 is a table format documented in the FLASH Users Guide.
- Tables for emission, absorption, transport, as functions of (temperature, density, energy group).
- We can convert other table formats to IONMIX4.

Questions ?