Student Projects

Department of Earth & Space Science, Osaka University
Kengo TOMIDA
Use of Public Codes

Pros:
• You can reduce development costs and focus on science.
• You can save your computing time if the code is fast.
• You can just say “we use this code” in your paper
• You can share knowledge and techniques with other users.

Cons:
• Too easy – think carefully and understand underlying physics.
• Some codes are hard to extend or keep consistency
• Potential inconsistency between versions, or bugs.
• Risk: the core of your research relying on others
• People may not think it’s your achievement even if you write a part of the code by yourself.

Use public codes wisely!
Use of Public Codes

• Not only Athena++, but all the public codes consume a lot of resources. Please cite the papers and acknowledge the code accordingly (not only when you actually used it, but also when you read it and learned something).

• Please report any bug, problem, or suggestion of improvement as soon as possible, even if you do not know how to fix it. However, please understand our man-power is limited.

• It is YOUR responsibility to ensure the simulation results are correct. Do not trust the code too much, and validate the results are physically reasonable.

• You do not need to stick to a single code. Select a code wisely for your problem based on features, performance and reliability.

• Enjoy your happy simulation life.
**Athena++ Code Structure**

```plaintext
athena/
src/    -- source code directory
 bvals/  -- boundary conditions including communications
 coordinates/  -- coordinate definitions
  eos/    -- equation of state
 field/ -- magnetic field integrator
 hydro/ -- hydrodynamics integrators
 mesh/   -- grid generation, refinement
 outputs/ -- I/O functions
 pgen/   -- problem generators
 utils/  -- other utilities
 inputs/ -- sample input parameter files
 tst/    -- regression test scripts
 vis/    -- visualization scripts
```
Where to Start?

Want to understand the hydrodynamic solvers → hydro/
Magnetic fields → field/
Grid structure → mesh/
Simulation flow* → main.cpp, task_list/
Parallelization** → mesh/, bvals/ (**: Chaos. Careful. Cal)
*: Athena++ uses “TaskList” to control the simulation, which split
the simulation into small “tasks”, and put them in a waiting queue
with dependency flags. The code automatically change the order
of execution if possible. The “Main loop” is in TaskList::DoTaskList:

```cpp
87   while(nmb_left > 0) {
88       pmb = pmesh->pblock;
89       while (pmb != NULL) {
90           if (DoAllAvailableTasks(pmb,step) == TLCOMPLETE) nmb_left--; 
91           pmb=pmb->next;
92       }
93   }
```
Multi-Phase ISM

Interstellar Medium consists of a few different phases
• Hot Ionized Medium \( (n < 0.01 \text{ cm}^{-3}, T > 10^5 \text{K}) \)
• Warm Ionized Medium \( (n \sim 0.2 \text{ cm}^{-3}, T \sim 8000 \text{K}) \)
• Warm Neutral Medium \( (n \sim 0.5 \text{ cm}^{-3}, T \sim 6000-8000 \text{K}) \)
• Cold Neutral Medium \( (n \sim \text{a few x 10 cm}^{-3}, T \sim 100 \text{K}) \)
• Molecular Clouds \( (n > 100 \text{ cm}^{-3}, T < 50 \text{K}) \)

These phase separation occurs due to the cooling curve.
See e.g. review by Cox 2005
ISM Equilibrium Curve

\[ \text{pressure : } p/k_B \quad [ \text{K/cm}^3 ] \]

\[ \text{number density : } n \quad [ \text{cm}^{-3} ] \]
Ubiquitous Turbulence

↑Larson 1981, →Heyer & Brunt 2004
Molecular clouds (and ISM) have turbulence almost ubiquitously. The spectrum is $\sigma \propto L^{0.5}$, close to the Kolmogorov law
→ There seems no characteristic scale – the turbulence can be driven in larger scales, or every scale (?) compressibility? B-field?
Interstellar media and molecular clouds are ubiquitously turbulent. Supersonic turbulence decays quickly (Stone et al. 1998) \(\rightarrow\) turbulence must be driven by some mechanism(s) 

One of promising processes is thermal instability in shocked gas, as ISM experiences nearby SNe or galactic flow occasionally. \(\rightarrow\) cold clumps move at velocities \(c_{\text{SNe}} < v < c_{\text{WNM}}\) (Iwasaki et al. in prep.)
Magnetic fields prevent strong compression and the instability.
Project 1

Following Inoue and Inutsuka 2008 (also Koyama & Inutsuka 2002), simulate colliding flows, with and without magnetic fields in 2D, but with simpler physics, just the cooling function.

This requires:
- Implementing the initial condition
- Implementing the inflowing boundary condition
- Implementing the cooling function + time step

Goals:
- See how the thermal instability drives turbulence
- See how magnetic fields affect it

Additional Goals:
- Measure turbulence properties
- Learn some details of the ISM physics
Project 1

Step 1:
• Implement the source function as in Inoue & Inutsuka 2008
• Set up a periodic box, put thermally stable or unstable gas, add some perturbation, and see what happens

\[ \rho_n \mathcal{L} = n_n (\mathcal{L} + n_n \Lambda) \text{ erg cm}^{-3} \text{ s}^{-1}, \]
\[ \Lambda = 2 \times 10^{-26}, \]
\[ \frac{\Lambda}{\mathcal{L}} = 1.0 \times 10^7 \exp \left( \frac{-118,400}{T + 1000} \right) \]
\[ + 1.4 \times 10^{-2} \sqrt{T} \exp \left( \frac{-92}{T} \right). \]

Step 2:
• Implement the colliding flow initial and boundary conditions
• Let two WNM streams collide and see what will happen

Step 3:
• Try some simulations with higher resolution if possible
• Add vertical/horizontal magnetic fields to see their effects
Project 2

If you are already using simulation for your research project, and if you need features already available in Athena++, port it to Athena++.

Goals:
• Make this school more practical
• Compare different codes

Disclaimer:
I cannot guarantee your problem works with Athena++ (or Athena++ works with your problem), but it might be worth trying.
Normalization to Code Units

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

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

\[ \frac{\partial e}{\partial t} + \nabla \cdot [(e + p) \mathbf{u}] = 0 \]

Set a new set of three independent units as you like, for example:

\[ \rho \rightarrow \rho' \rho_0, \quad L \rightarrow L'L_0, \quad u \rightarrow u'u_0 \]

These relations imply:

\[ M \rightarrow M'(\rho_0 L_0^3), \quad t \rightarrow t' \left( \frac{L_0}{u_0} \right), \quad p \rightarrow p(\rho_0 u_0^2), \quad \text{etc...} \]

\[ \frac{\rho_0 u_0}{L_0} \left[ \frac{\partial \rho'}{\partial t'} + \nabla' \cdot (\rho' \mathbf{u}') \right] = 0 \]

\[ \frac{\rho_0 u_0^2}{L_0} \left[ \frac{\partial \rho' \mathbf{u}'}{\partial t'} + \nabla' \cdot (\rho' \mathbf{u}' \mathbf{u}' + p' \mathbf{I}) \right] = 0 \]

\[ \frac{\rho_0 u_0^3}{L_0} \left[ \frac{\partial e'}{\partial t'} + \nabla' \cdot [(e' + p') \mathbf{u}'] \right] = 0 \]