Assignment 9 of Computational Astrophysics in NTHU

Wei-Hsiang Yu 游惟翔

Department of Physics, National Tsing Hua University, Hsinchu, Taiwan

June 3, 2021

1 Written Assignments

Q1 : Estimate the mean free path λ_{mfp} and collision time scale τ_{col}

Consider the neutral, atomic interstellar medium at $T = 10^3$K.

On the Wikipedia[1] it provide the information of **cold neutral medium** & **high neutral medium**

$$\text{cold neutral medium } (50 - 100K) \text{ have } \textbf{higher} \text{ density : } 20 - 50 \ particle/cm^3$$

$$\text{high neutral medium } (6000 - 10^6 K) \text{ have } \textbf{lower} \text{ density : } 0.2 - 0.5 \ particle/cm^3$$

The interstellar medium we consider is in the interval between these two kinds of medium, so I suppose the density($particle/cm^3$) of neutral, atomic interstellar medium at $T = 10^3$K is $n = 1/cm^3$.

And the component of this interstellar medium is mainly hydrogen, so $\mu \sim 1m_{amu}$. The RMS speed $v \sim \sqrt{\frac{8kT}{\pi\mu}} = 2.5 \times 10^5 (cm/s)$ and the cross section σ I use the value $\sigma \sim 3 \times 10^{-15} cm^2$ provided in lecture.

$$\lambda_{mfp} \sim (n\sigma)^{-1} \sim 3.33 \times 10^{14} cm \tag{1}$$

$$\tau_{col} \sim (n\sigma v)^{-1} \sim 1.33 \times 10^9 sec \tag{2}$$

Q2 : fluid approximation

Consider a molecular cloud with:

$$L = 100pc = 100 \times (3.08 \times 10^{18}) = 3.08e20 \text{ cm}$$

$$\tau = 10^8 yr = 10^8 \times (365 \times 86400) = 3.1536e15 \text{ sec}$$

On the Wikipedia[1] it provide the information of molecular cloud :

$$\text{molecular cloud } (10 - 20K) \text{ have density : } 10^2 - 10^6 \ particle/cm^3$$

if let $n = 10^2$, $\lambda_{mfp} \sim 3.33 \times 10^{12} cm$; $\tau_{col} \sim 7.2 \times 10^7 sec.$

$n = 10^6$, $\lambda_{mfp} \sim 3.33 \times 10^8 cm$; $\tau_{col} \sim 7.2 \times 10^4 sec.$

Both length L and time τ scale is >> than the mean free path λ_{mfp} and collision time scale τ_{col}, so we can viewed this molecular cloud as fluid.

Q3 : Conservation of momentum in hydrodynamic

Derive[2]:

Given Euler equation:

$$\frac{Dv}{Dt} = -\frac{\bigtriangledown P}{\rho} \tag{3}$$

By the convective derivate in the lecture and combine with Eq.3 we have:

$$\frac{Dv}{Dt} \equiv \frac{\partial v}{\partial t} + (v \cdot \bigtriangledown)v = -\frac{\bigtriangledown P}{\rho} \tag{4}$$

By the continuity equation in the lecture we have:

$$\frac{\partial \rho}{\partial t} = -(v \cdot \bigtriangledown)\rho - \rho(\bigtriangledown \cdot v) \tag{5}$$

And we can make Eq.4 times ρ and Eq.5 times v, and add up to one equation.

$$\rho\frac{\partial v}{\partial t} + v\frac{\partial \rho}{\partial t} = -\bigtriangledown P - \rho(v \cdot \bigtriangledown)v - v(v \cdot \bigtriangledown)\rho - \rho(\bigtriangledown \cdot v)v \tag{6}$$

I recall the formula of gradient:

$$\bigtriangledown \cdot fA = f(\bigtriangledown \cdot A) + \bigtriangledown f \cdot A$$

$$\bigtriangledown \cdot (\rho vv) = \rho \bigtriangledown \cdot (vv) + \bigtriangledown \rho \cdot (vv) = \rho(v \cdot \bigtriangledown)v + \rho(v \cdot \bigtriangledown)v + \bigtriangledown \rho \cdot (vv)$$

Using this $\bigtriangledown \cdot (\rho vv)$ term to substitute Eq.6, we can finally get the form of Eq.7

$$\frac{\partial \rho v}{\partial t} + \bigtriangledown \cdot (\rho vv + P \cdot I) = 0 \tag{7}$$

2 Programming Assignments

Q2 : Antares code for Kelvin-Helmholtz instability

Q2.a

$$\text{initial condition}: \begin{cases} \rho = 2, v_x = 1, v_y = 0, P = 2.5, & \text{if } y > 0.5 \\ \rho = 1, v_x = -1, v_y = 0, P = 2.5, & \text{if } y < 0.5 \end{cases} \tag{8}$$

I use momentum transform make $p = \rho \times v$ to get the momentum of each direction.

The energy can be gotten by two components: one is provided by inner energy u, can get by the equation of states(Eos) of pressure $P = (\gamma - 1)u$; the other is by kinetic energy E_k, $E_k = \frac{p^2}{2m}$. (Show in Fig.3(a).)

(a) t=0 (b) t=0.25 (c) t=0.5 (d) t=0.75 (e) t=0.1

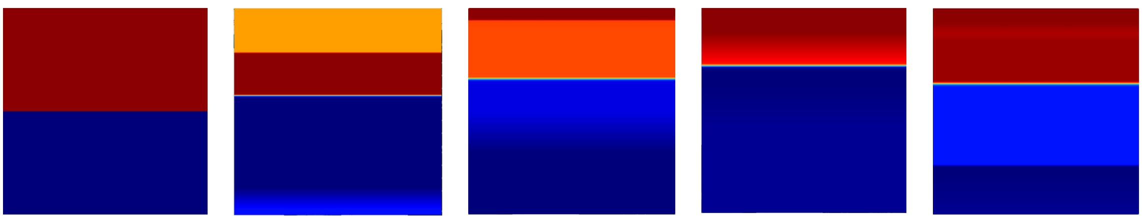

Figure 1: result of 2.a.

Q2.b

In this part, although I don't get the phenomenon of Kelvin-Helmholtz instability cloud like the question show, but change the boundary condition of this problem will get cool result. In Fig.2, I make x direction has reflect boundary and y direction has periodic boundary, the contrary condition compare to origin(Fig.1)

(a) t=0 (b) t=0.25 (c) t=0.5 (d) t=0.75 (e) t=0.1

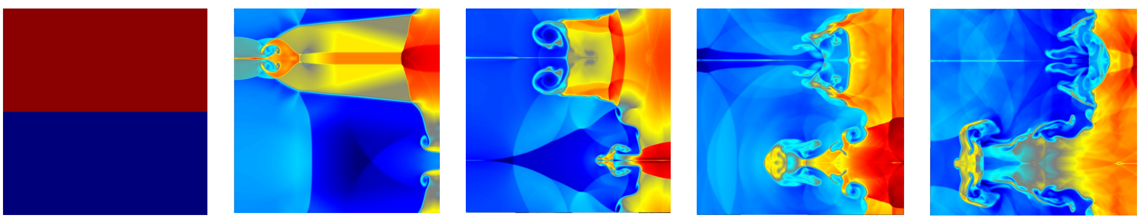

Figure 2: result of change the boundary condition

References

[1] Interstellar medium on Wikipedia
https://en.wikipedia.org/wiki/Interstellar_medium

[2] Conservation of momentum in hydrodynamic
https://slidetodoc.com/lecture-planet-formation-topic-introduction-to-hydrodynamics-and_/

(a) initial condition

```
  if ( y(j) .le. 0.5d0*ymax) then
    den(i,j,k)= 1.d0
     px(i,j,k)= -1.0d0
     py(i,j,k)= 0.0d0
     pz(i,j,k)= 0.d0

     pg       = 2.5d0
    rhoe      = pg/(den(i,j,k)*(gam-1.d0))
    ene(i,j,k)= rhoe + 0.5d0
       ! 0.5d0*(px(i,j,k)**2+py(i,j,k)**2+pz(i,j,k)**2)/den(i,j,k)

  else
    den(i,j,k)= 2.d0
     px(i,j,k)= 2.0d0
     py(i,j,k)= 0.0d0
     pz(i,j,k)= 0.d0

     pg       = 2.5d0
    rhoe      = pg/(den(i,j,k)*(gam-1.d0))
    ene(i,j,k)= rhoe + 1.d0
        !0.5d0*(px(i,j,k)**2+py(i,j,k)**2+pz(i,j,k)**2)/den(i,j,k)
```

(b) boundary condition

```
!=================================================================
!                       Boundary Condition
!=================================================================

!       inner :
! X
        call bnd_inner_periodic(den ,px ,py ,pz ,ene ,bx ,by ,bz ,
     &                         den2,px2,py2,pz2,ene2,bx2,by2,bz2,1)
c        call bnd_inner_reflect(den ,px ,py ,pz ,ene ,bx ,by ,bz ,
c     &                     den2,px2,py2,pz2,ene2,bx2,by2,bz2,1,uini)
! Y
        if (D2D .or. D3D)then
        call bnd_inner_reflect(den ,px ,py ,pz ,ene ,bx ,by ,bz ,
     &                         den2,px2,py2,pz2,ene2,bx2,by2,bz2,2)
c        call bnd_inner_ini(den ,px ,py ,pz ,ene ,bx ,by ,bz ,
c     &                     den2,px2,py2,pz2,ene2,bx2,by2,bz2,2,uini)
       endif
! Z
       if (D3D)then
        call bnd_inner_outflow(den ,px ,py ,pz ,ene ,bx ,by ,bz ,
     &                         den2,px2,py2,pz2,ene2,bx2,by2,bz2,3)
c        call bnd_inner_ini(den ,px ,py ,pz ,ene ,bx ,by ,bz ,
c     &                     den2,px2,py2,pz2,ene2,bx2,by2,bz2,3,uini)
       endif

!      outer :

! X
        call bnd_outter_periodic(den ,px ,py ,pz ,ene ,bx ,by ,bz ,
     &                          den2,px2,py2,pz2,ene2,bx2,by2,bz2,1)
c        call bnd_outter_outflow(den ,px ,py ,pz ,ene ,bx ,by ,bz ,
c     &                      den2,px2,py2,pz2,ene2,bx2,by2,bz2,1,uini)
! Y
        if (D2D .or. D3D)then
        call bnd_outter_reflect(den ,px ,py ,pz ,ene ,bx ,by ,bz ,
     &                          den2,px2,py2,pz2,ene2,bx2,by2,bz2,2)
c        call bnd_outter_ini(den ,px ,py ,pz ,ene ,bx ,by ,bz ,
c     &                      den2,px2,py2,pz2,ene2,bx2,by2,bz2,2,uini)
       endif
```

(c) parameter

```
!---  Choose Dimensional
      parameter ( D1D       = .false.) ! True for 1D
      parameter ( D2D       = .true. ) ! True for 2D
      parameter ( D3D       = .false.) ! True for 3D
      parameter ( MHD       = .false. ) ! True for MHD
      parameter ( ISOTHERMAL= .false.) ! True for Isothermal
!     note: we don't have isothermal MHD solver, currently
!--- Geometry
      parameter (cartesian   = .true.)
      parameter (spherical   = .false.)
      parameter (cylindrical = .false.)
!-----------------------------------------------
      parameter ( ibeg  =       1  )
      parameter ( iend  =     512  ) ! N x
      parameter ( ibuf  =       2  )
      parameter ( jbeg  =       1  )
      parameter ( jend  =     512  ) ! N y
      parameter ( jbuf  =       2  )
      parameter ( kbeg  =       1  )
      parameter ( kend  =       1  ) ! N z
      parameter ( kbuf  =       2  )
!---  Physical Dormain
      parameter ( xmin  =     0.d0 )
      parameter ( xmax  =     1.d0 )
      parameter ( ymin  =     0.d0 )
      parameter ( ymax  =     1.d0 )
      parameter ( zmin  =     0.d0 )
      parameter ( zmax  =     1.d0 )
      parameter ( gam   =     1.4d0)
      parameter ( tf    =     1.d0 )
      parameter ( smalld  =     1.d-14 )
      parameter ( smalle  =     1.d-14 )
      parameter ( smallt  =     1.d-4 )
!-----------------------------------------------
      parameter ( iso_snd =    1.d0 )  ! Isothermal sound speed
      parameter ( cfl   =    0.4d0 )  ! Courant number
```

Figure 3: problem set of folder **khi**