Basic hydrodynamics and Introduction to FLASH

Kuo-Chuan Pan (潘國全)
Institute of Astronomy, NTHU

Outline:

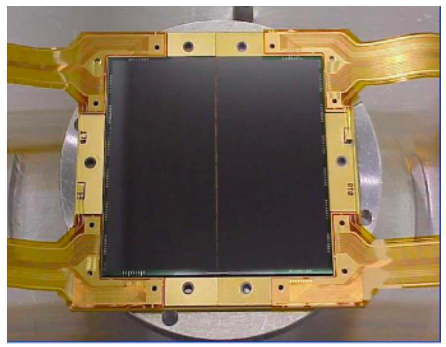
- Astrophysical fluids
- Basic hydrodynamic equations
- Introduction to the FLASH code

Why we need simulations in Astronomy?

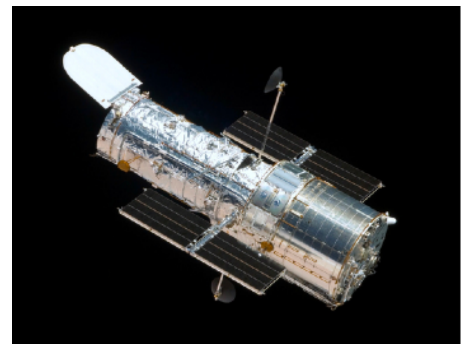
- We receive and record, but we do not experiment or perturb (too distant)
- Astrophysical systems involve extreme conditions that cannot be replicated in lab (mostly)
- Timescales are >> human lifetime (mostly)

Observation and Computation Analogies

Astrophysical object



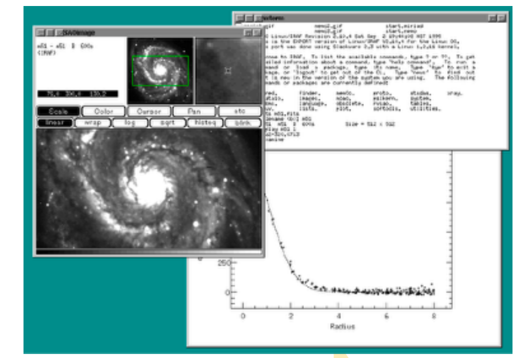
Detectors



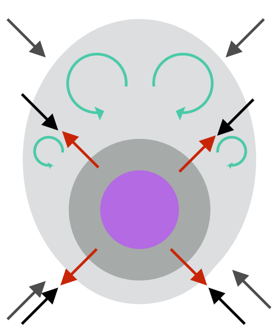
Telescope

Observation

DATA



Data reduction / Calibration



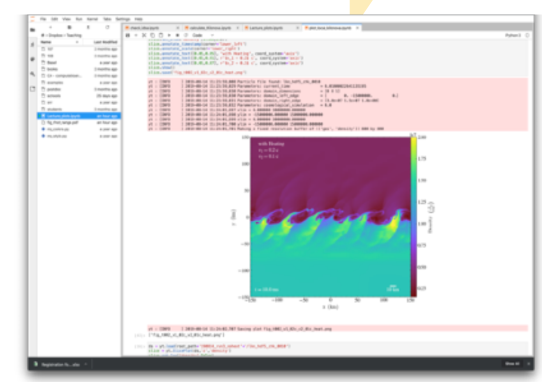
Mathematical Model

Codes

Supercomputer

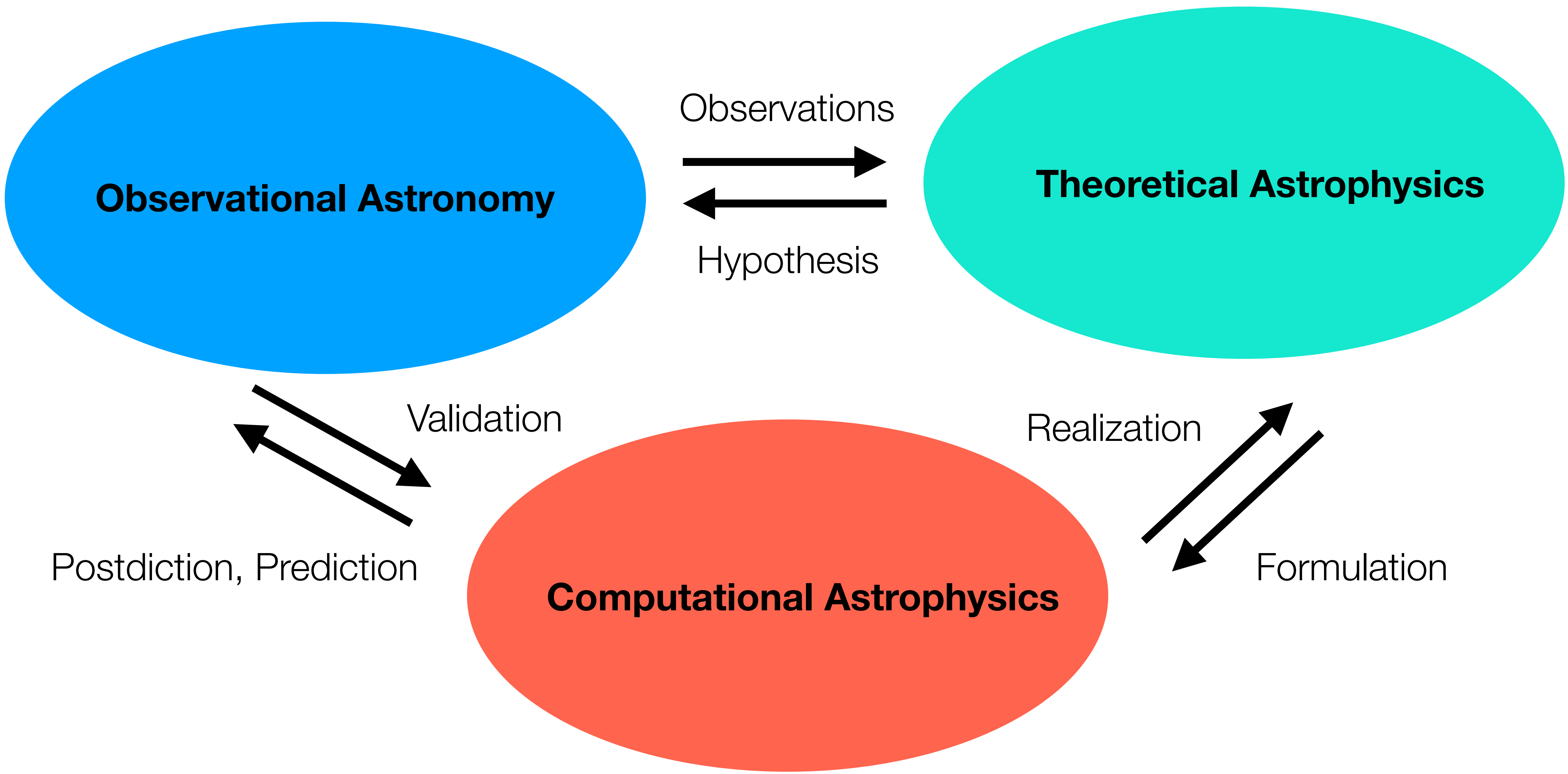
Simulation

DATA



Data analysis / Visualization

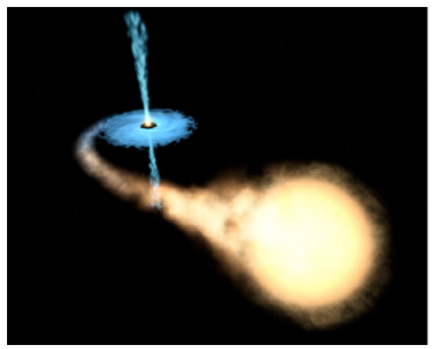
Frontier Astronomy Research

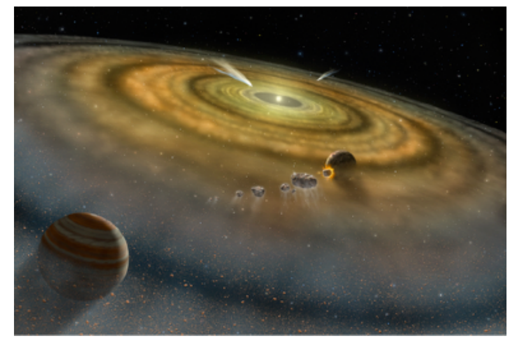


Michael Norman (1997)

Astrophysical Problems


Planets
(~ Earth radius)

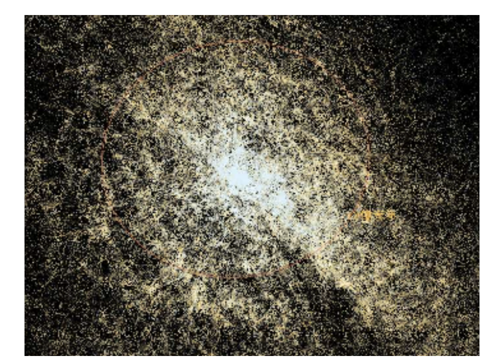

Stars/Binaries
(~0.1-1000 AU)


Planetary disk
(~100-1000 AU)

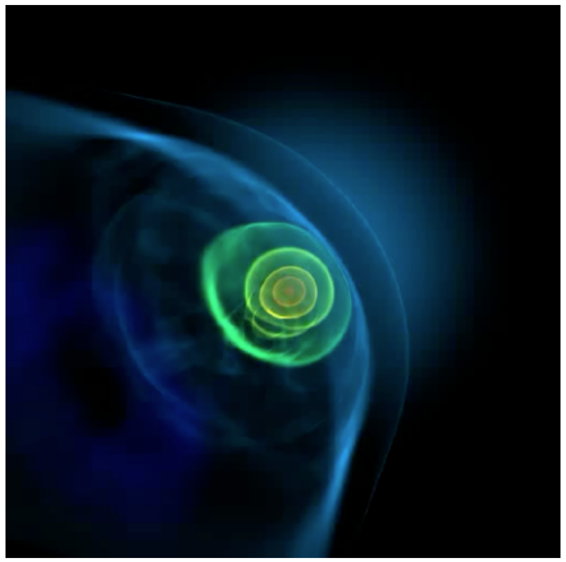

Interstellar medium
(~10-100 pc)


Galaxy
(~50 kpc)

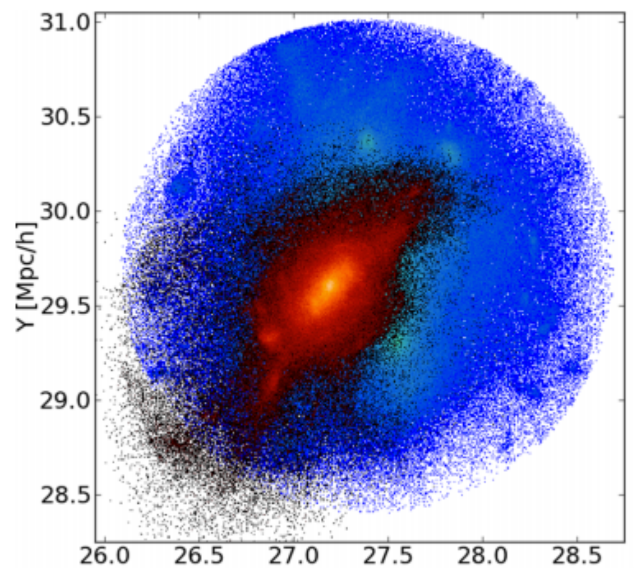

Galaxy clusters
(~1 Mpc)


Large scale
structure
(~1 Gpc)

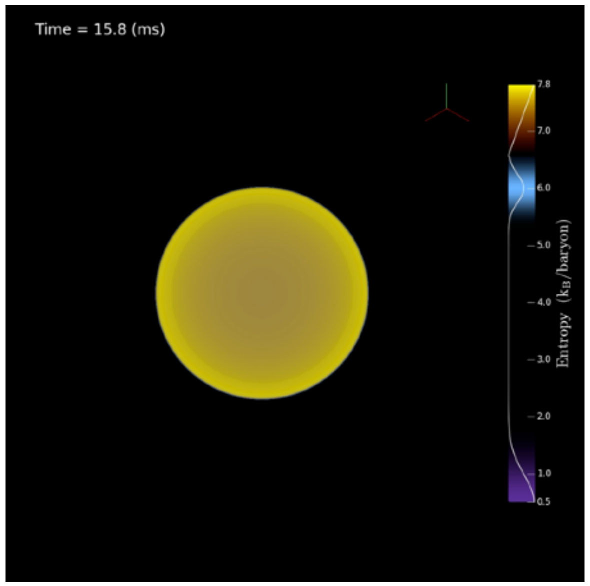
Astrophysical Simulations



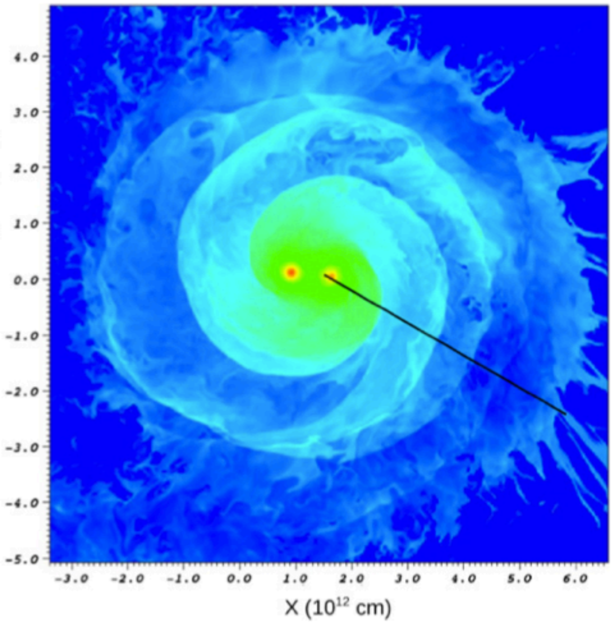
2012 ApJ, 750, 151



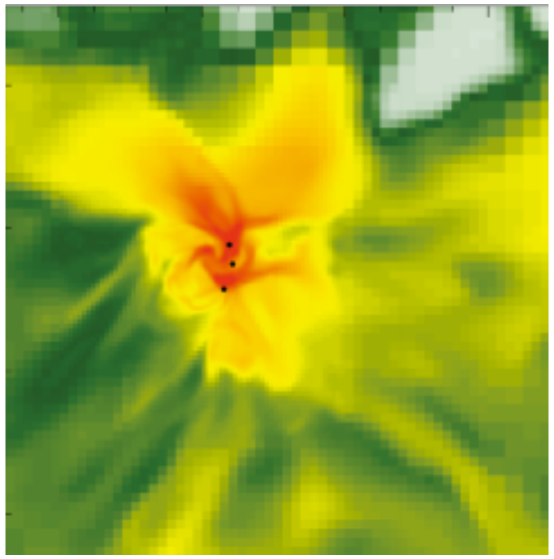
2013MNRAS



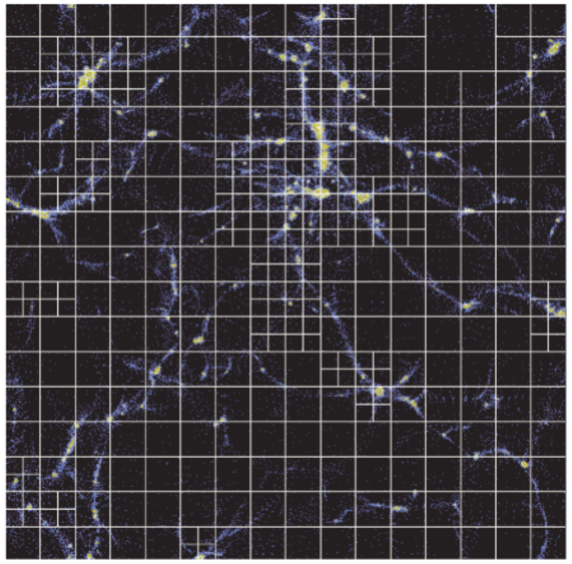
2018 ApJ



2012 ApJ, 746, 74



2009 MNRAS 398, 1082



2005 ApJS 160, 28

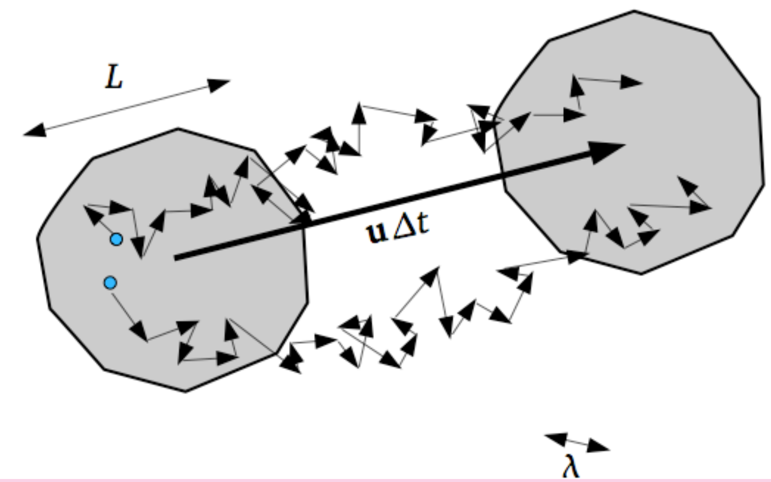
Fluid Approximation

- To solve the gas dynamics equations, one could use N-body method if we know the force between two particles.
- However, the number of particles is usually huge ($N_A=10^{23}$). It is impossible to use direct N-body method to study the gas dynamics. -> Fluid

Fluid Approximation (continue)

$L >> \lambda_{\rm mfp}$, Mean free path

$T >> \tau_{\rm coll}$, Collision time scale



Exercise: Consider the air in this room

$$\rho \sim 10^{-3}\ {\rm g/cm^3},\ \mu \sim 28 m_p,\ T \sim 300\ {\rm K}$$

$$\text{Cross section } \sigma \sim 3 \times 10^{-15}\ {\rm cm^2}$$

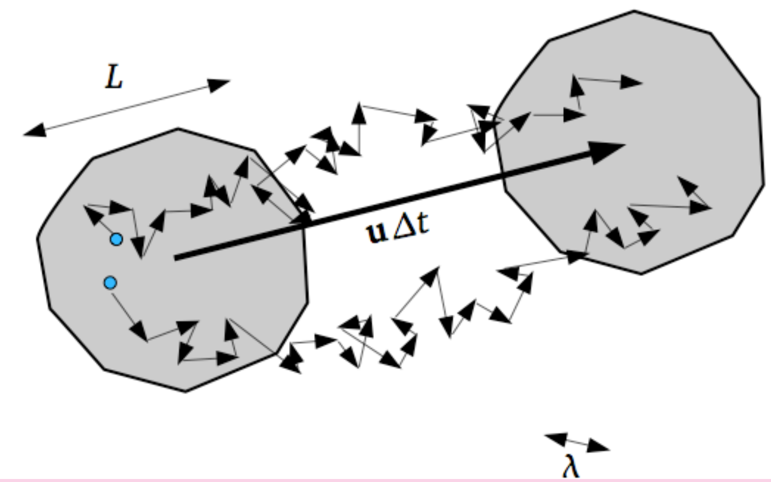
$$\text{RMS speed of nitrogen } v \sim \sqrt{\frac{8kT}{\pi\mu}} \sim 5 \times 10^4\ {\rm cm/s}$$

$$\tau_{\rm coll} \sim (n\sigma v)^{-1} \sim 3 \times 10^{-10}\ {\rm sec} \qquad \lambda_{\rm mfp} \sim (n\sigma)^{-1} \sim 3 \times 10^{-5}\ {\rm cm}$$

Fluid Approximation (continue)

$L >> \lambda_{\mathrm{mfp}}$, Mean free path

$T >> \tau_{\mathrm{coll}}$, Collision time scale



Exercise: In ISM

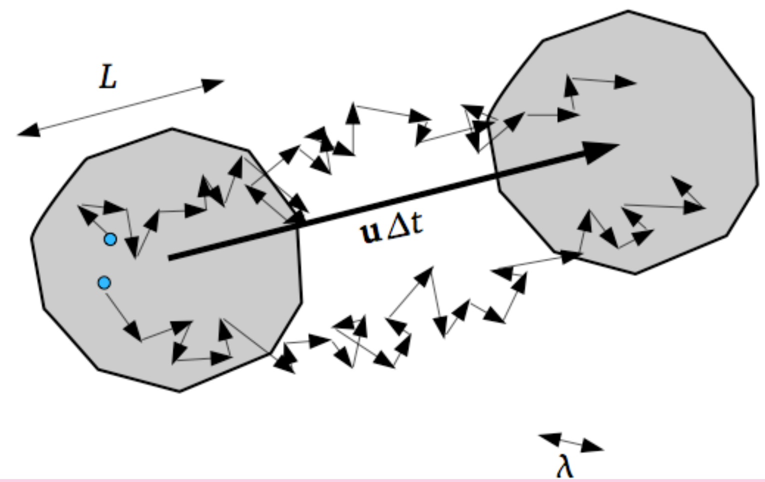
$$\text{Cross section } \sigma \sim ?\ \mathrm{cm}^2$$

$$\text{number density } n \sim ?\ \mathrm{cm}^{-3}$$

Fluid Approximation (continue)

$L >> \lambda_{\rm mfp}$, Mean free path

$T >> \tau_{\rm coll}$, Collision time scale



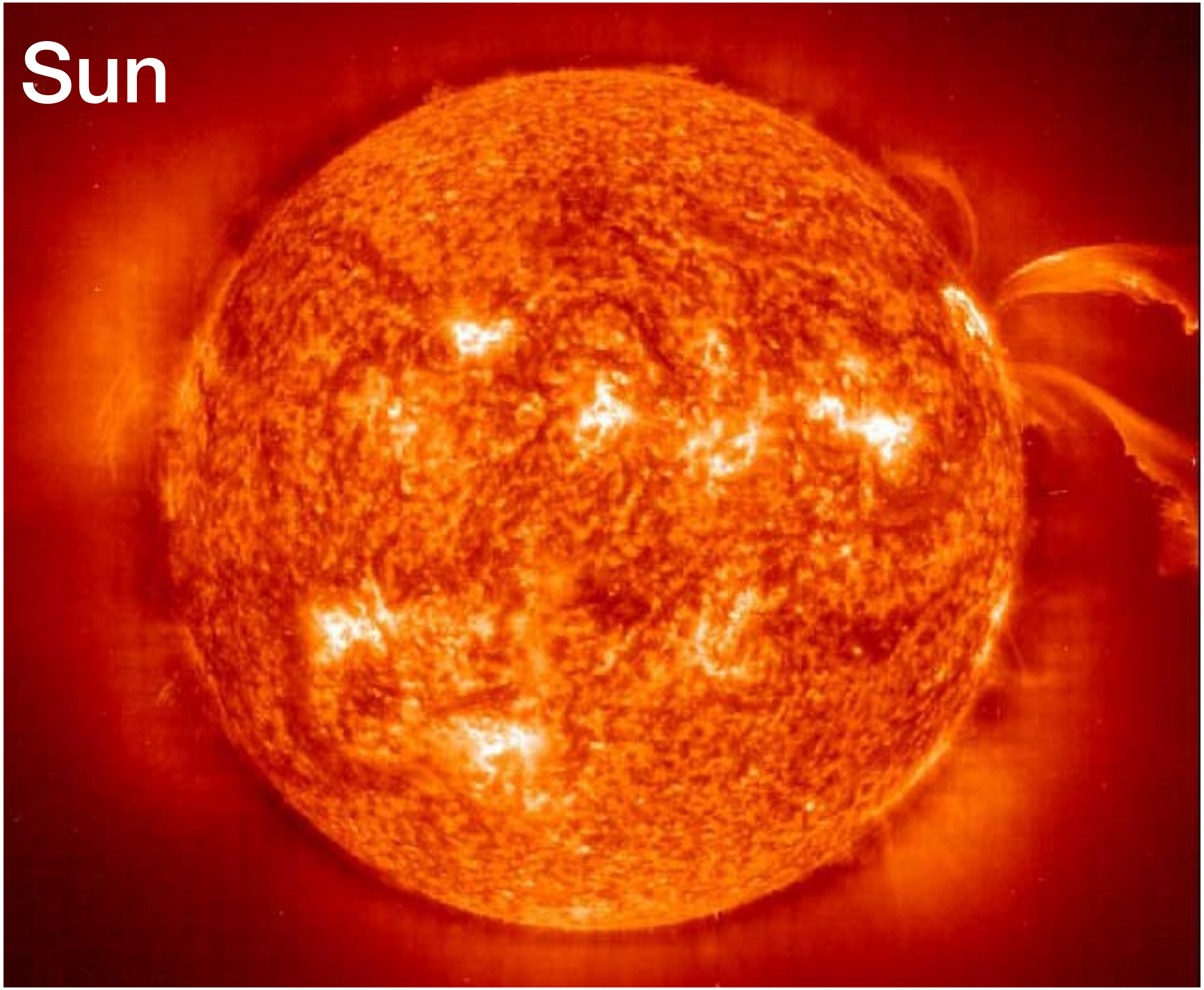
Exercise: In ISM

$$\text{Cross section } \sigma \sim 3 \times 10^{-15}\ \text{cm}^2$$

$$\text{number density } n \sim 1\ \text{cm}^{-3}$$

$$\lambda_{\rm mfp} \sim (n\sigma)^{-1} \sim 10^{15}\ \text{cm}$$

$$\tau_{\rm coll} \sim (n\sigma v)^{-1} \sim 10^{9}\ \text{sec} \sim 40\ \text{yrs}$$


Sun

Star forming region (S106)

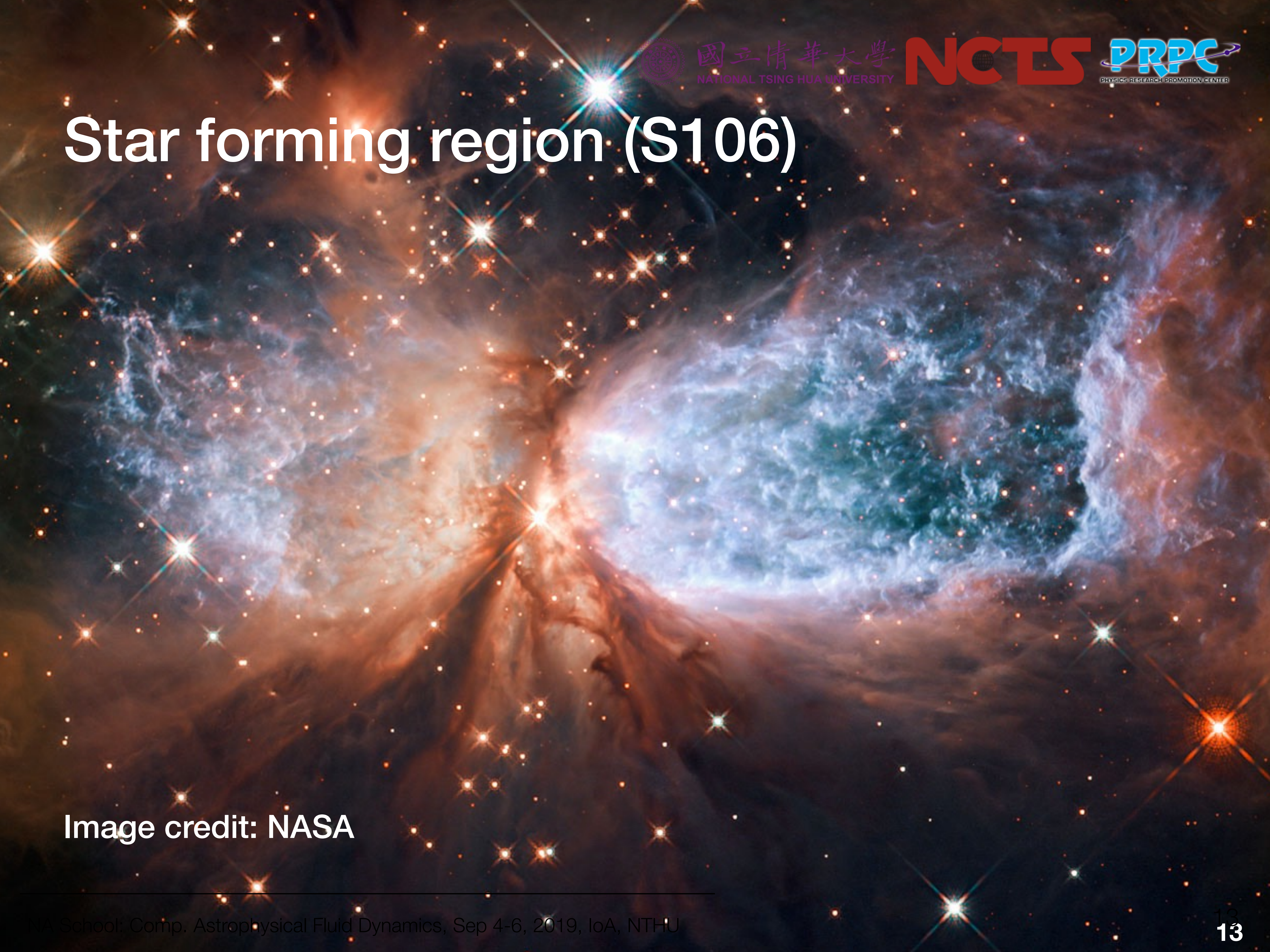


Image credit: NASA

Cas A Supernova Remnant

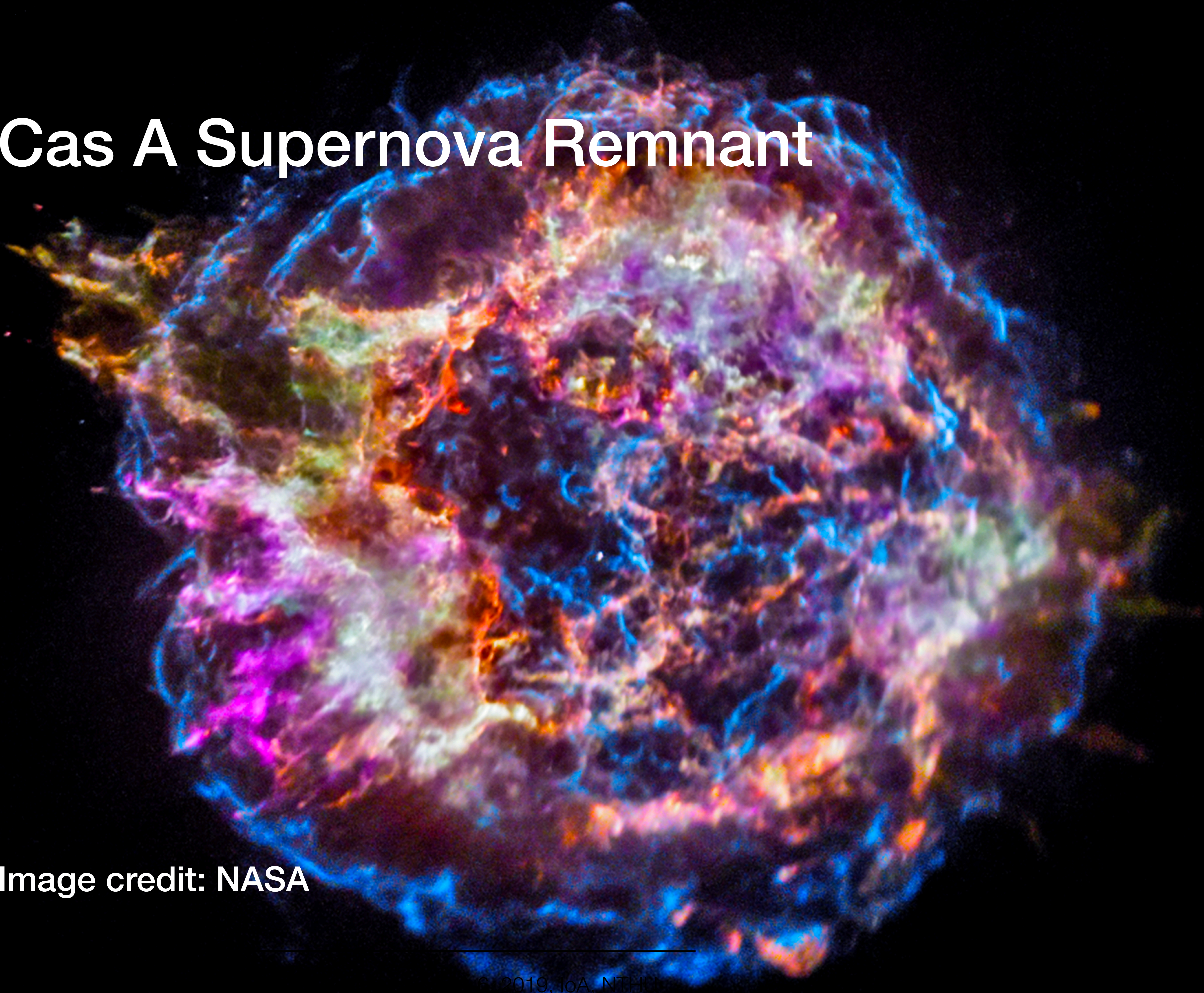


Image credit: NASA

Bullet cluster

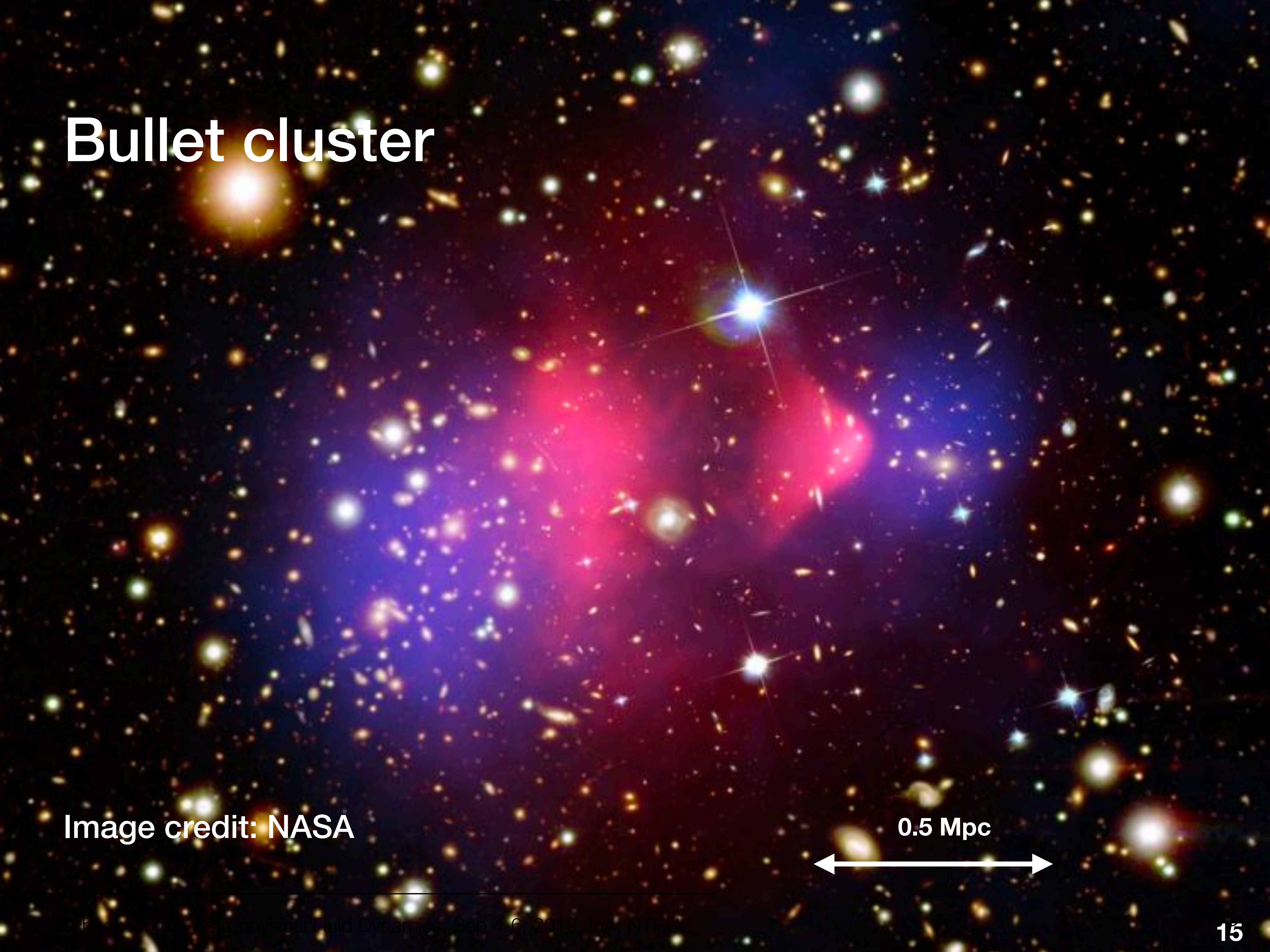
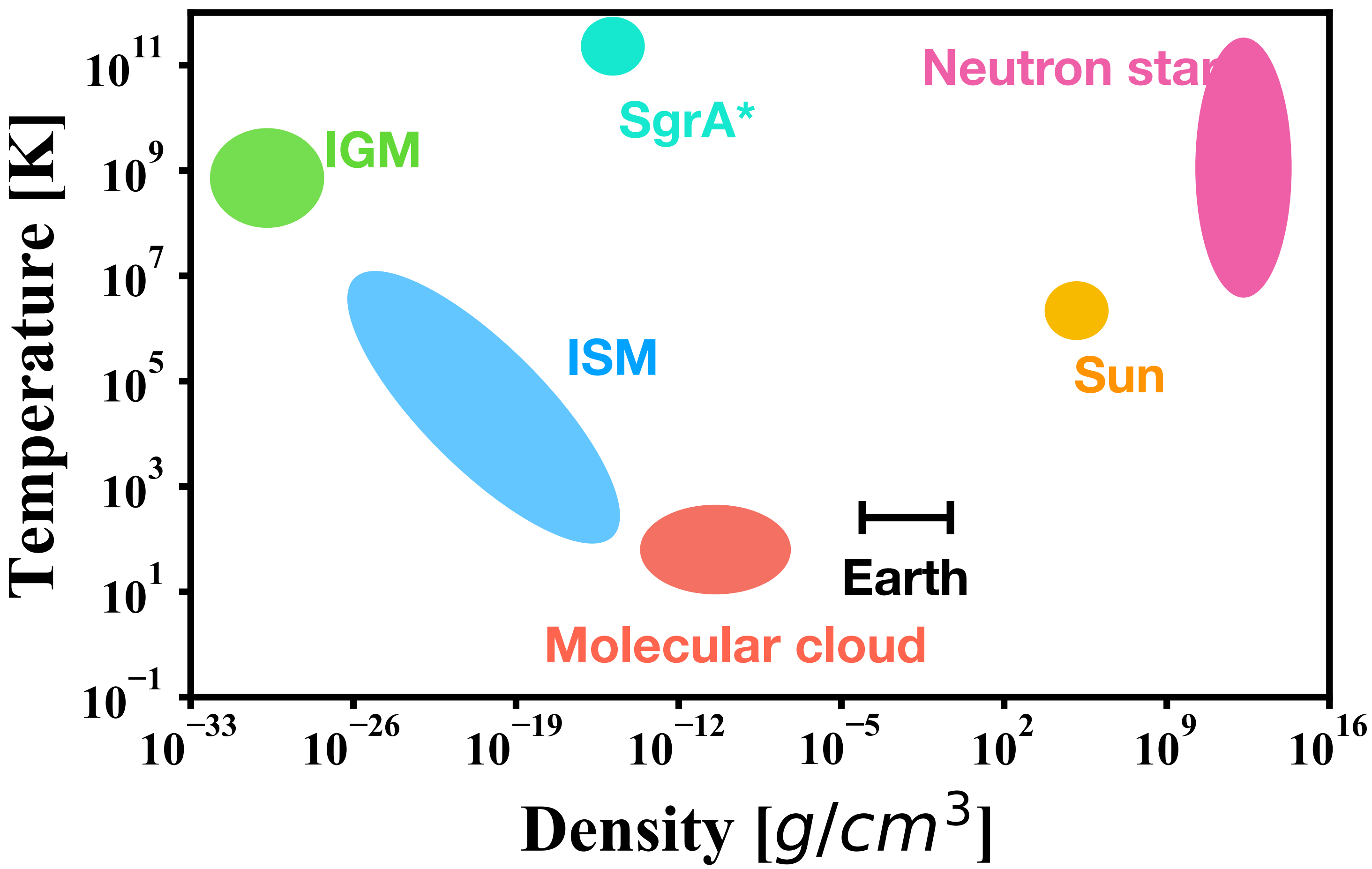


Image credit: NASA

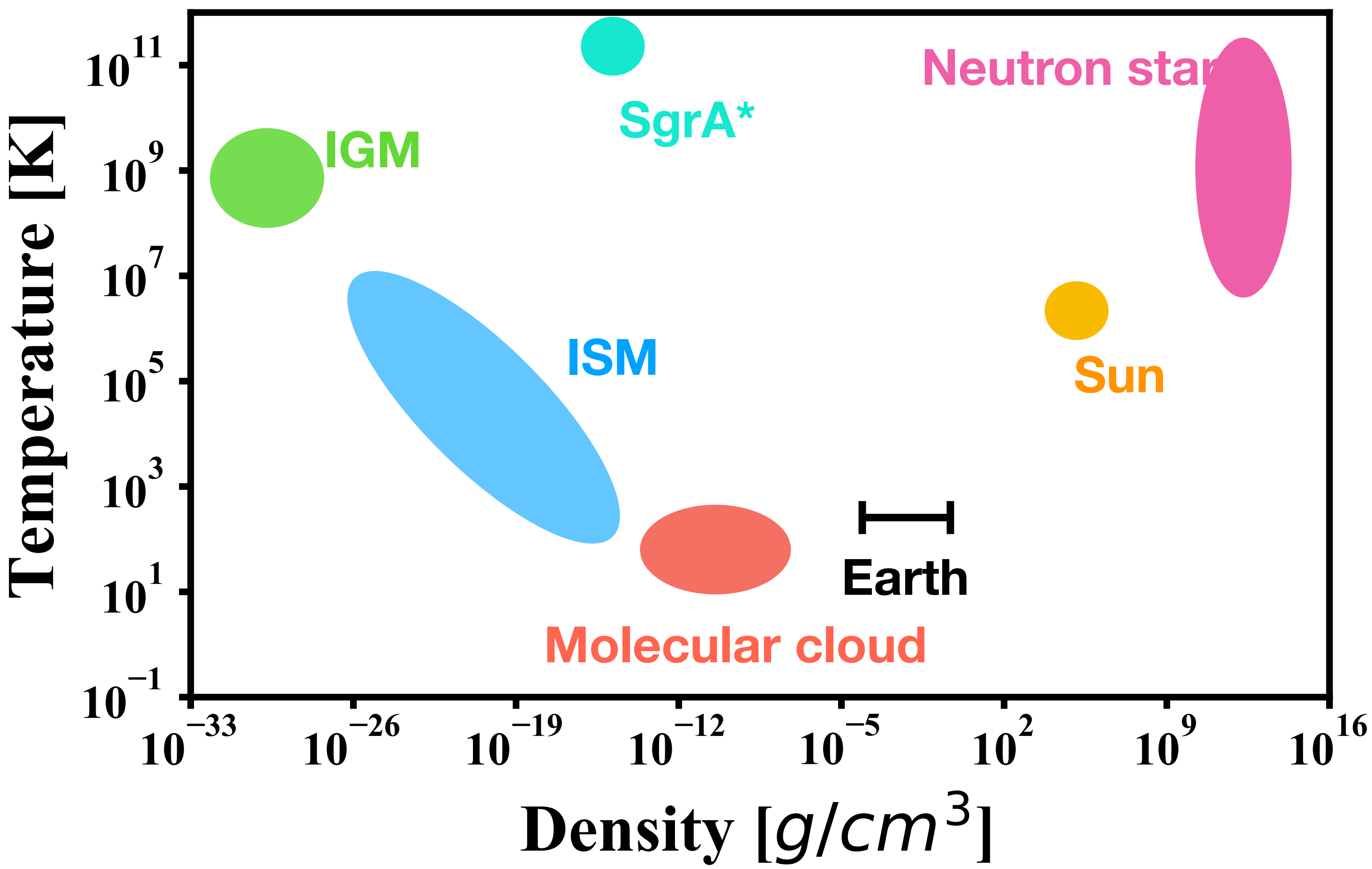
Astrophysical Fluids

Astrophysical fluids



- Different with Earth-bound fluid
- Wide range of physical Conditions

Astrophysical fluids



- Magnetic fields are often important
- Compressibility is often important
- Occasionally self-gravitating and relativistic
- Non-ideal process are often unimportant
- Sometimes involves interactions with non-thermal process and strong radiation fields

Governing equations

The governing equations for ideal (inviscid) hydrodynamics. The main physical ideas are simple:

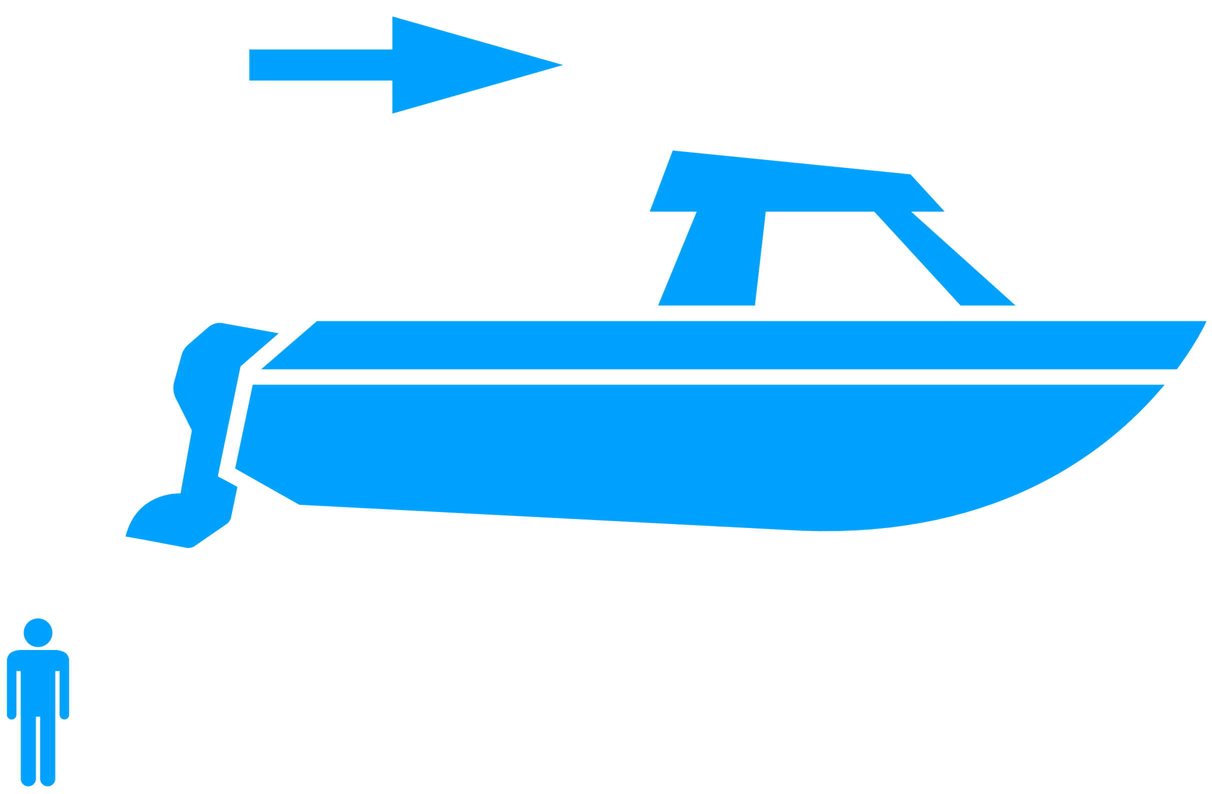
- Mass is conserved (1 constraint)
- Momentum is conserved (3 constraints)
- Energy is conserved (1 constraint)

$$\text{variables: } \rho, \boldsymbol{v}, \text{ and } u$$

ρ	Gas density
$\boldsymbol{v}$	Gas velocity
u	Gas internal energy

Coordinates

Eulerian



Lagrangian

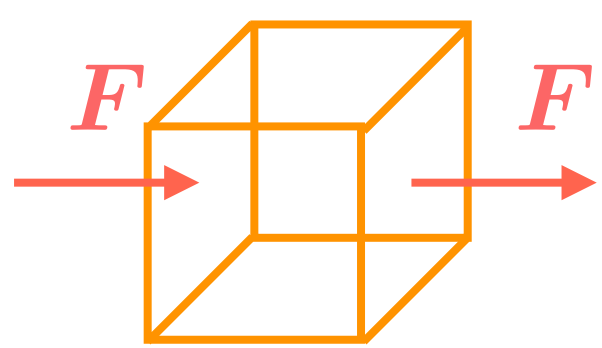
Conservational laws

The conservational laws can be written in what is called “conservation form”:

$$\partial_t(U) = -\nabla \cdot \boldsymbol{F}$$

“Gauss’s Theorem" or "divergence theorem”

$$\int_V (\nabla \cdot \boldsymbol{F})\, dV = \oint_S \boldsymbol{F} \cdot \boldsymbol{n} dS$$



U	Density of a quantity
$\boldsymbol{F}$	Flux density for that quantity

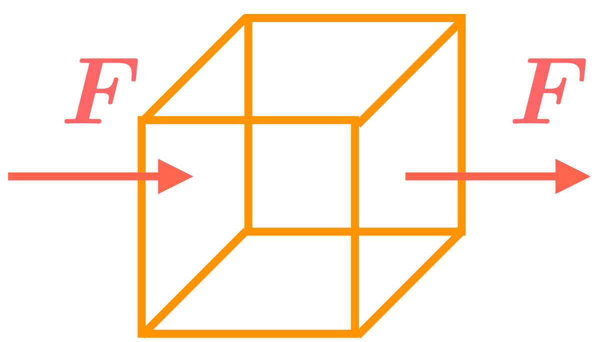
Continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0$$

ρ	Gas density
$\boldsymbol{v}$	Gas velocity

“Gauss’s Theorem" or "divergence theorem”

$$\int_V (\nabla \cdot \boldsymbol{F})\, dV = \oint_S \boldsymbol{F} \cdot \boldsymbol{n} dS$$



Convective derivative

- Consider a fluid element

$$\boldsymbol{v} = \frac{d\boldsymbol{r}}{dt}, \qquad \boldsymbol{a} = \frac{d\boldsymbol{v}}{dt},$$

$$\begin{aligned}
\boldsymbol{a} &= \frac{d\boldsymbol{v}(\boldsymbol{r},\boldsymbol{t})}{dt} \\
&= \frac{\partial v}{\partial t} + \frac{\partial v}{\partial r} \cdot \frac{\partial r}{\partial t} \\
&= \frac{\partial v}{\partial t} + (v \cdot \nabla) v \\
&\equiv \frac{Dv}{Dt}
\end{aligned}$$

Convective derivate

- Rewrite continuity equation

$$\begin{aligned}
\partial_t \rho &= -\nabla \cdot (\rho \boldsymbol{v}) \\
&= -(v \cdot \nabla)\rho - \rho(\nabla \cdot v)
\end{aligned}$$

→ $$\partial_t \rho + (v \cdot \nabla)\rho = -\rho(\nabla \cdot v)$$

→ $$\frac{D\rho}{Dt} = -\rho(\nabla \cdot v)$$

Continuity equation

Momentum equation

- What accelerations act on a fluid element?

$$\boldsymbol{a} = -\frac{\nabla p}{\rho}$$

- This gives us the Euler equations

$$\frac{D\boldsymbol{v}}{Dt} = -\frac{\nabla p}{\rho}$$

or

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}\boldsymbol{v} + P \cdot \boldsymbol{I}) = 0$$

Momentum equation

- What accelerations act on a fluid element?

$$\boldsymbol{a} = -\frac{\nabla p}{\rho}$$

- This gives us the Euler equations

$$\frac{D\boldsymbol{v}}{Dt} = -\frac{\nabla p}{\rho}$$

or

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}\boldsymbol{v} + P \cdot \boldsymbol{I}) = 0$$

Your exercise!

Internal Energy equation

- From first law of thermodynamics,

$$d\epsilon = T\cancelto{0}{ds} - pdV, \qquad dV = d\left(\frac{1}{\rho}\right) = \frac{-d\rho}{\rho^2}$$

Volume per unit mass

$$d\epsilon \equiv d\left(\frac{u}{\rho}\right) = \frac{du}{\underline{\rho}} - u\frac{d\rho}{\rho^2} = -pdV = p\frac{d\rho}{\rho^2}$$

$$\frac{Du}{Dt} = \frac{u+p}{\rho}\frac{D\rho}{Dt} = \frac{\partial u}{\partial t} + (v\cdot\nabla)u = -(u+p)\nabla\cdot v$$

- Combine internal energy equation with the momentum equation

$$\frac{\partial \rho E}{\partial t} + \nabla\cdot[(\rho E + P)\boldsymbol{v}] = 0$$

$$\boxed{E = \frac{1}{2}v^2 + \epsilon}$$

Total Energy equation

Summary: Euler equations

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0$$

Continuity equation

Stress tensor　　Unit tensor

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}\boldsymbol{v} + P \cdot \boldsymbol{I}) = 0$$

Momentum equation

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot [(\rho E + P)\boldsymbol{v}] = 0$$

Energy equation

ρ	Gas density
$\boldsymbol{v}$	Gas velocity
p	Gas pressure
ϵ	Gas specific internal energy
E	Gas specific total energy
u	Gas internal energy

$$u = \rho\epsilon$$

$$E = \frac{1}{2}v^2 + \epsilon$$

國立清華大學 NATIONAL TSING HUA UNIVERSITY NCTS PRPC PHYSICS RESEARCH PROMOTION CENTER

Summary: Euler equations + Gravity

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0$$

Continuity equation

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}\boldsymbol{v} + P \cdot \boldsymbol{I}) = -\rho \nabla \Phi$$

Momentum equation

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot [(\rho E + P)\boldsymbol{v}] = -\rho \boldsymbol{v} \cdot \nabla \Phi$$

Energy equation

$$\nabla^2 \Phi = 4\pi G \rho$$

Poisson equation

- ρ Gas density
- $\boldsymbol{v}$ Gas velocity
- p Gas pressure
- ϵ Gas specific internal energy
- E Gas specific total energy
- u Gas internal energy
- Φ Gravitational Potential

$$u = \rho \epsilon$$

$$E = \frac{1}{2} v^2 + \epsilon$$

Summary: Lagrangian form

$$\frac{D\rho}{Dt} = -\rho\nabla \cdot v$$

Continuity equation

$$\frac{Dv}{Dt} = -\frac{\nabla p}{\rho} - \nabla\Phi$$

Momentum equation

$$\frac{Du}{Dt} = -(u+p)\nabla \cdot v$$

Energy equation

ρ	Gas density
$\boldsymbol{v}$	Gas velocity
p	Gas pressure
ϵ	Gas specific internal energy
E	Gas specific total energy
u	Gas internal energy
Φ	Gravitational Potential

$$u = \rho\epsilon$$

$$E = \frac{1}{2}v^2 + \epsilon$$

Equation of State (EoS)

Need an Equation of State to close system of equations.

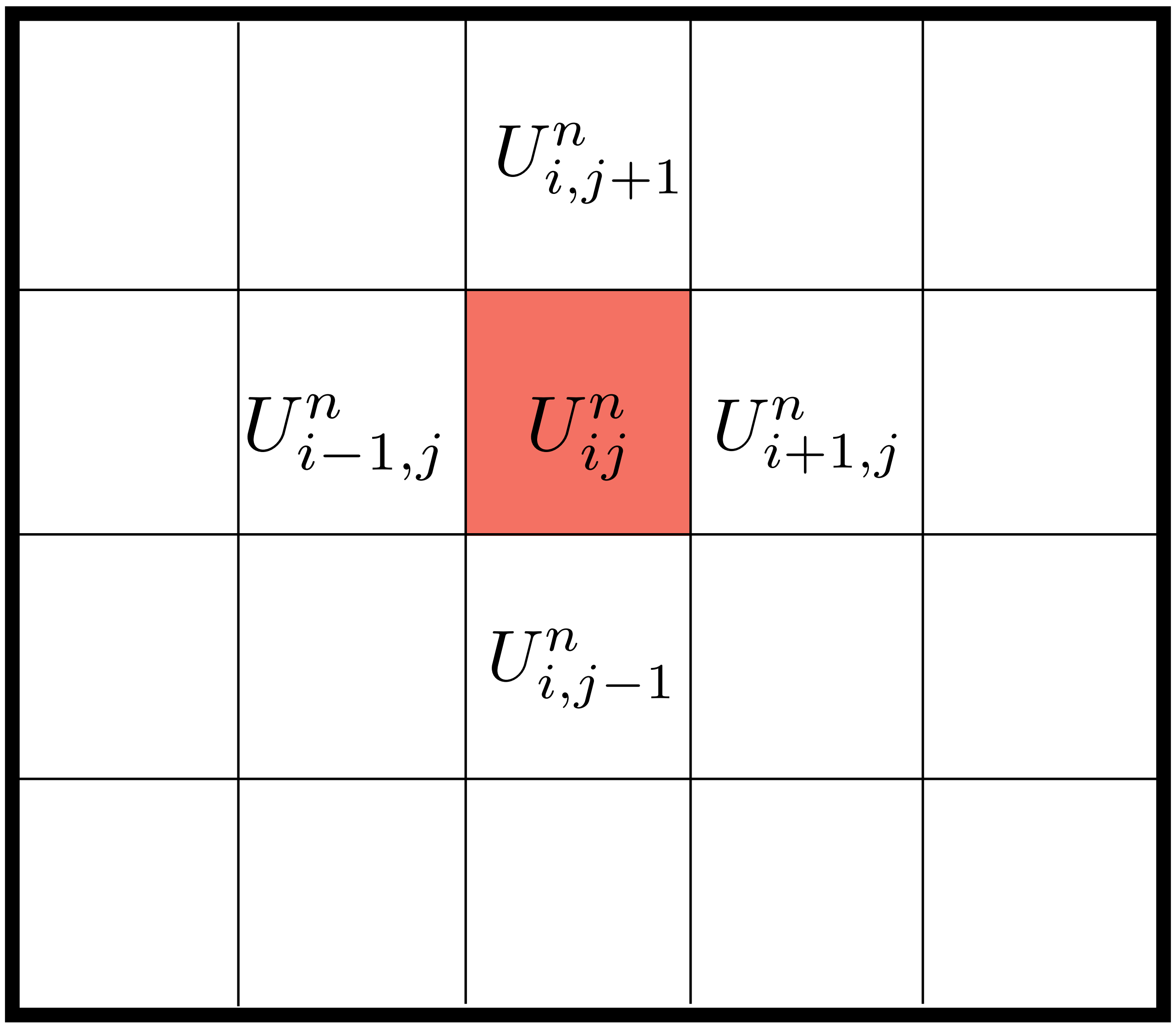
For ideal gas,

$$P = (\gamma - 1)u = nk_{\rm B}T$$

$n = \rho/m$	Number density
T	Temperature
$k_{\rm B}$	Boltzmann constant
$\gamma = C_v/C_p$	Adiabatic index

- Isothermal EoS: $\gamma = 1$ $\quad P = \rho c_s^2$
- Polytropic EoS: $\quad P = K\rho^\gamma$

Finite Volume Method



- Initial conditions of all hydrodynamics variables at time step n=0 (a given EoS)
- Boundary conditions at simulation boundaries
- Evolve hydrodynamics variables with a time step dt

$$\frac{\partial}{\partial t}\underbrace{\begin{pmatrix}\rho \\ \rho U \\ E\end{pmatrix}}_{W(x,t)} + \frac{\partial}{\partial x}\underbrace{\begin{pmatrix}\rho U \\ \rho U^2 + p \\ (E+p)U\end{pmatrix}}_{F(W)} = 0,$$

See Hsi-Yu Schive's lecture tomorrow

FLASH Code

FLASH Code: summary

- 3D AMR HD/MHD multi-physics code
- Mainly developed at FLASH center at University of Chicago (http://flash.uchicago.edu)
- Contribution from many individual groups
- Written principally in Fortran, with some C and Python (> 1 million lines; 25% are comments)
- Current release version 4.6.1
- > 700 scientists around the world
- > 1000 publications

FLASH Code: Requirements

- Fortran compiler
- MPI - Message Passing Interface
- HDF5 - Hierarchical Data Format

FLASH Code: research applications

- Thermonuclear flashes
- High energy density physics (HEDP)
- Fluid—structure interaction
- Star formation
- Star-star & Star-plantets interactions
- Core-collapse supernova
- Galaxy-galaxy cluster
- Magnetic field amplification
- Turbulence
- Cosmology
- … (more)

FLASH Code: Physic solvers

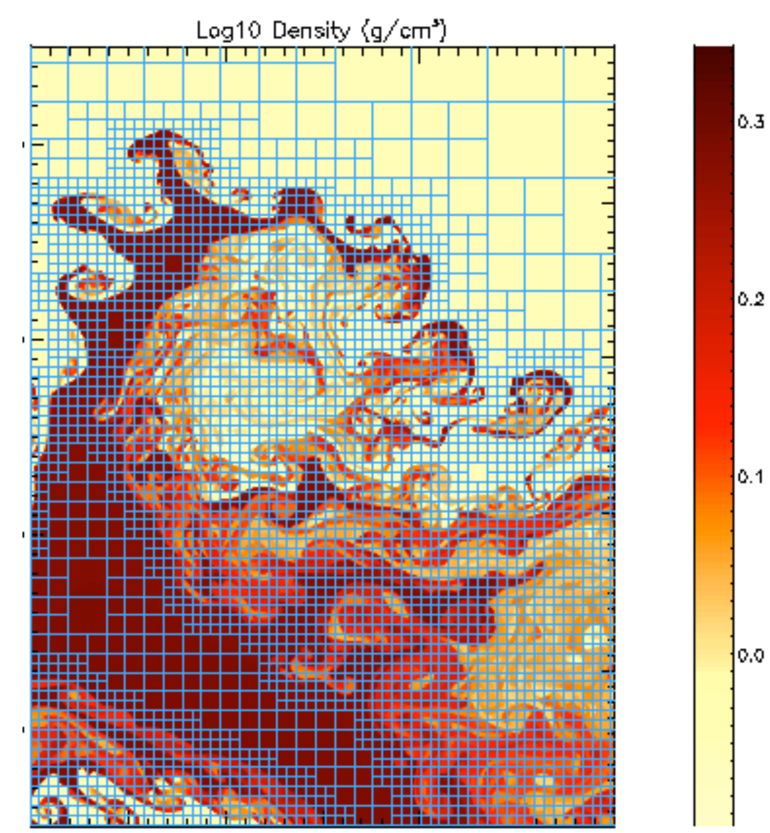
- Hydrodynamics: unspilt PPM, WENO; split PPM; 2T + Radiation
- Magneto-hydrodynamics (MHD): unspilt staggered mesh; split 8-wave
- Equation of State (EoS): idea gas; degenerate ionized plasma; nuclear,..
- Radiation Transfer: multi-group flux-limited diffusion
- Diffusion and Conduction: implicit with AMR
- Laser Energy Deposition: geometric optics with inverse Bremsstrahlung
- Opacity: constant; multi-material tabular
- Particles: tracer; massive; sink; charged
- Nuclear Burning
- Gravity: constant; point mass; planar; self-gravity
- Cosmology
- Heating and Cooling as source terms
- Magnetic Resistivity; Conductivity
- Primordial Chemistry

FLASH Code: Physic solvers

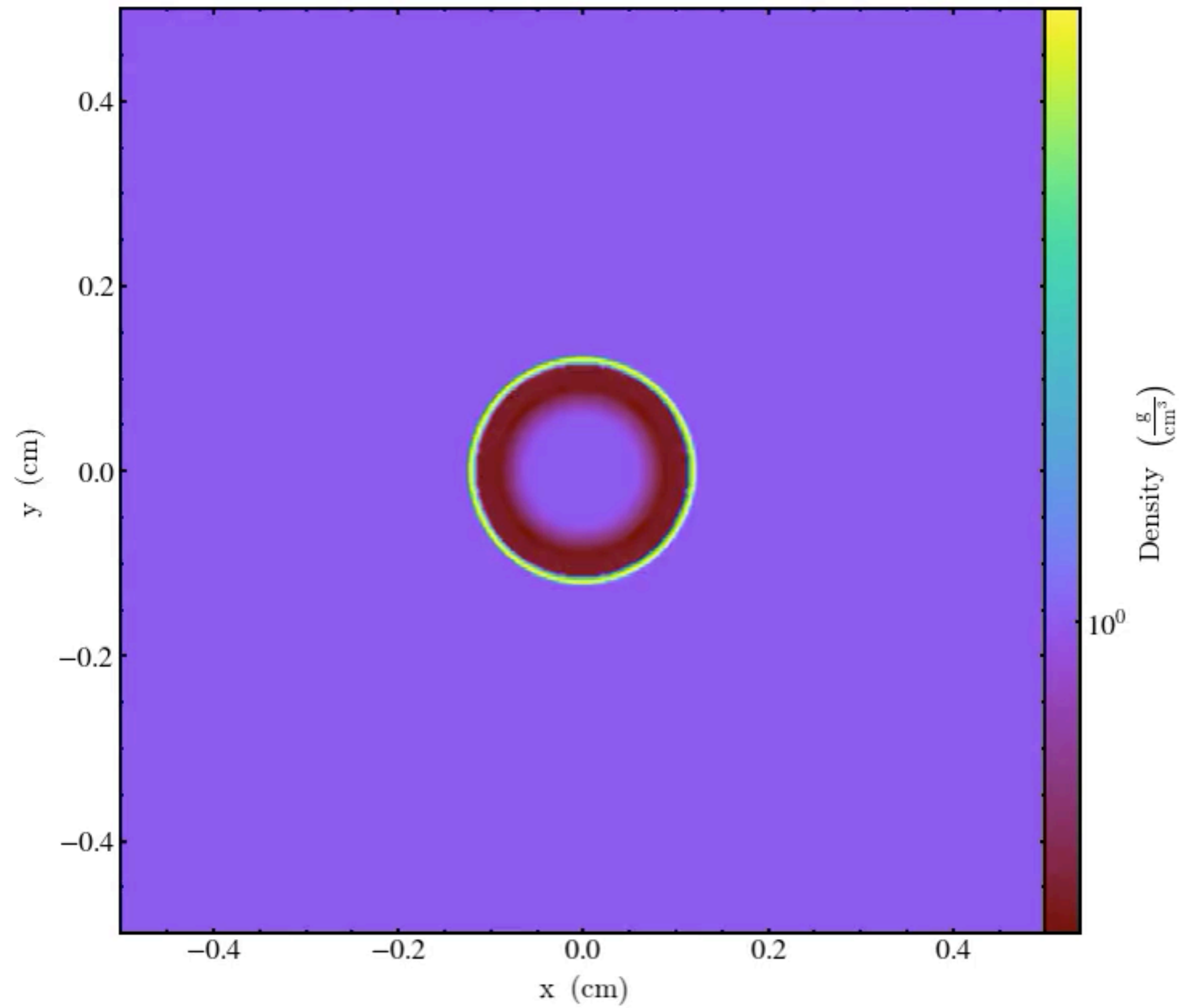
- Hydrodynamics: unspilt PPM, WENO; split PPM; 2T + Radiation
- Magneto-hydrodynamics (MHD): unspilt staggered mesh; split 8-wave
- Equation of State (EoS): idea gas; degenerate ionized plasma; nuclear,..
- Radiation Transfer: multi-group flux-limited diffusion
- Diffusion and Conduction: implicit with AMR
- Laser Energy Deposition: geometric optics with inverse Bremsstrahlung
- Opacity: constant; multi-material tabular
- Particles: tracer; massive; sink; charged
- Nuclear Burning
- Gravity: constant; point mass; planar; self-gravity
- Cosmology
- Heating and Cooling as source terms
- Magnetic Resistivity; Conductivity
- Primordial Chemistry

FLASH Code: Infrastructure

- Driver: split; unsplit
- Grid: uniform Grid; Adaptive Mesh Refinement (AMR)
- GridParticles: Lagrangian framework
- GridSolvers: multigrid; multipole; Barnes-Hut Tree; PFFT; direct solvers
- IO: Hierarchical Data Format 5 (HDF5); PnetCDF
- Multispecies
- Runtime parameters
- Monitor: MPI Timers; Hooks for TAU



Block structure PARAMESH AMR



See the afternoon lectures for data analysis and visualization

FLASH Code: research applications

- Highly modular (setup.py)

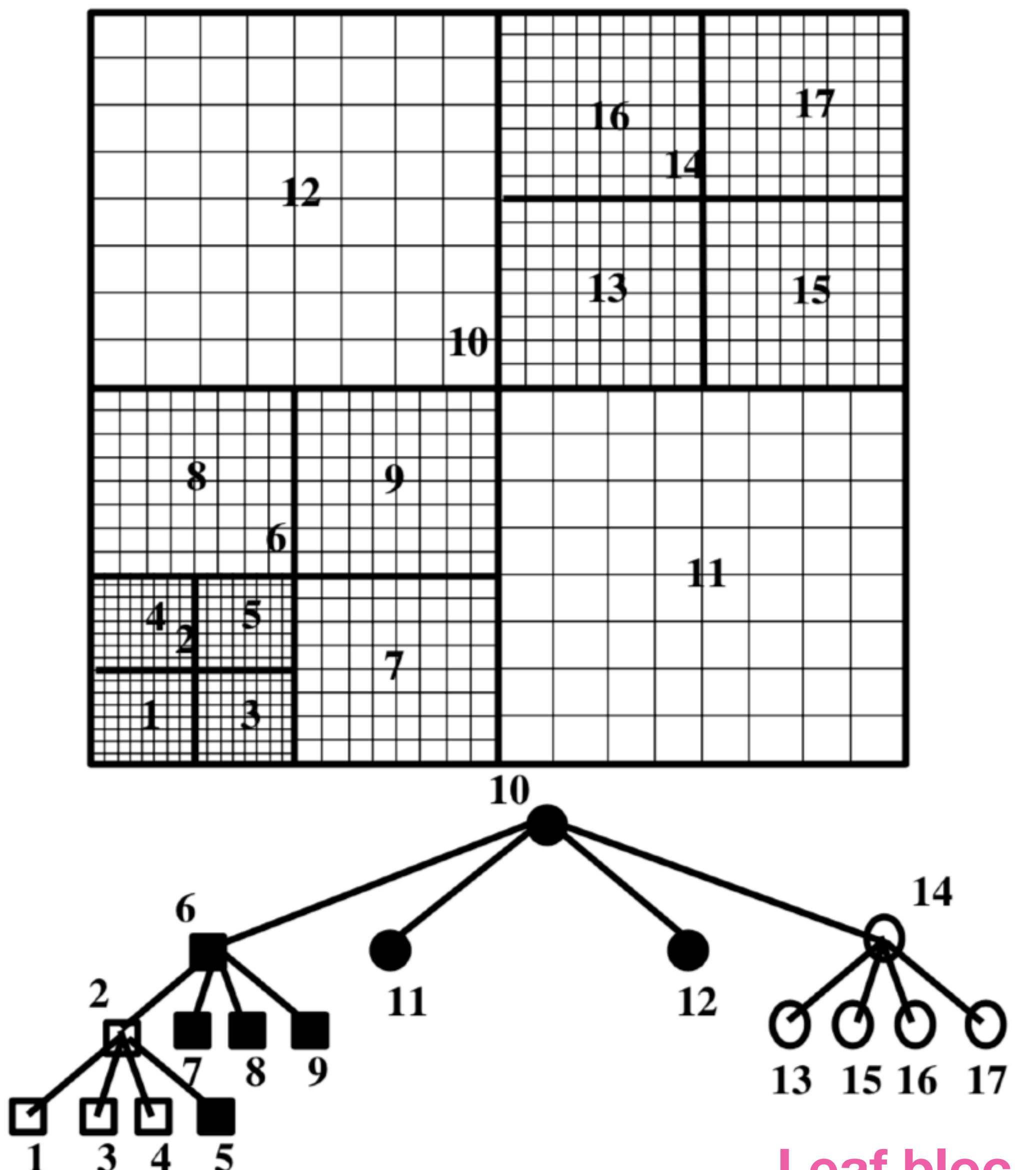
```
Driver/             Particles/            diagnostics/            numericalTools/
Grid/               PhysicalConstants/    flashUtilities/         physics/
IO/                 RuntimeParameters/    monitors/
Multispecies/       Simulation/           multiprocessorTools/
```

```
Cosmology/     Hydro/        RayTrace/             sourceTerms/
Diffuse/       ImBound/      SolidMechanics/       utilities/
Eos/           IncompNS/     TreeRay/
Gravity/       RadTrans/     materialProperties/
```

```
Burn/          EnergyDeposition/    Heatexchange/    PrimordialChemistry/
Cool/          Flame/               Ionize/          Stir/
Deleptonize/   Heat/                Polytrope/       Turb/
```

FLASH Code: Simulation Domain

- AMR



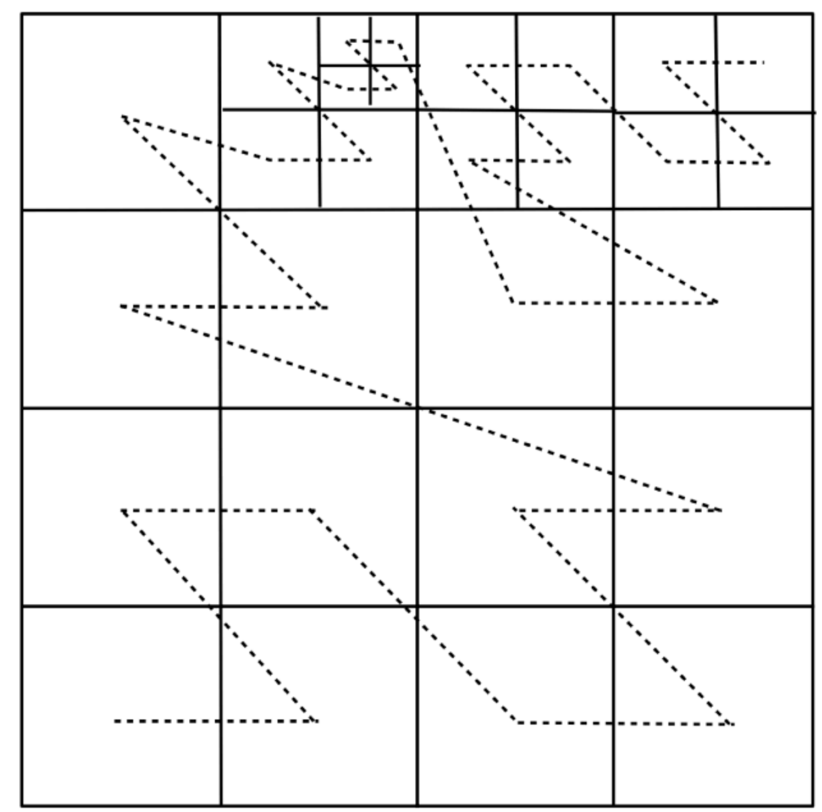
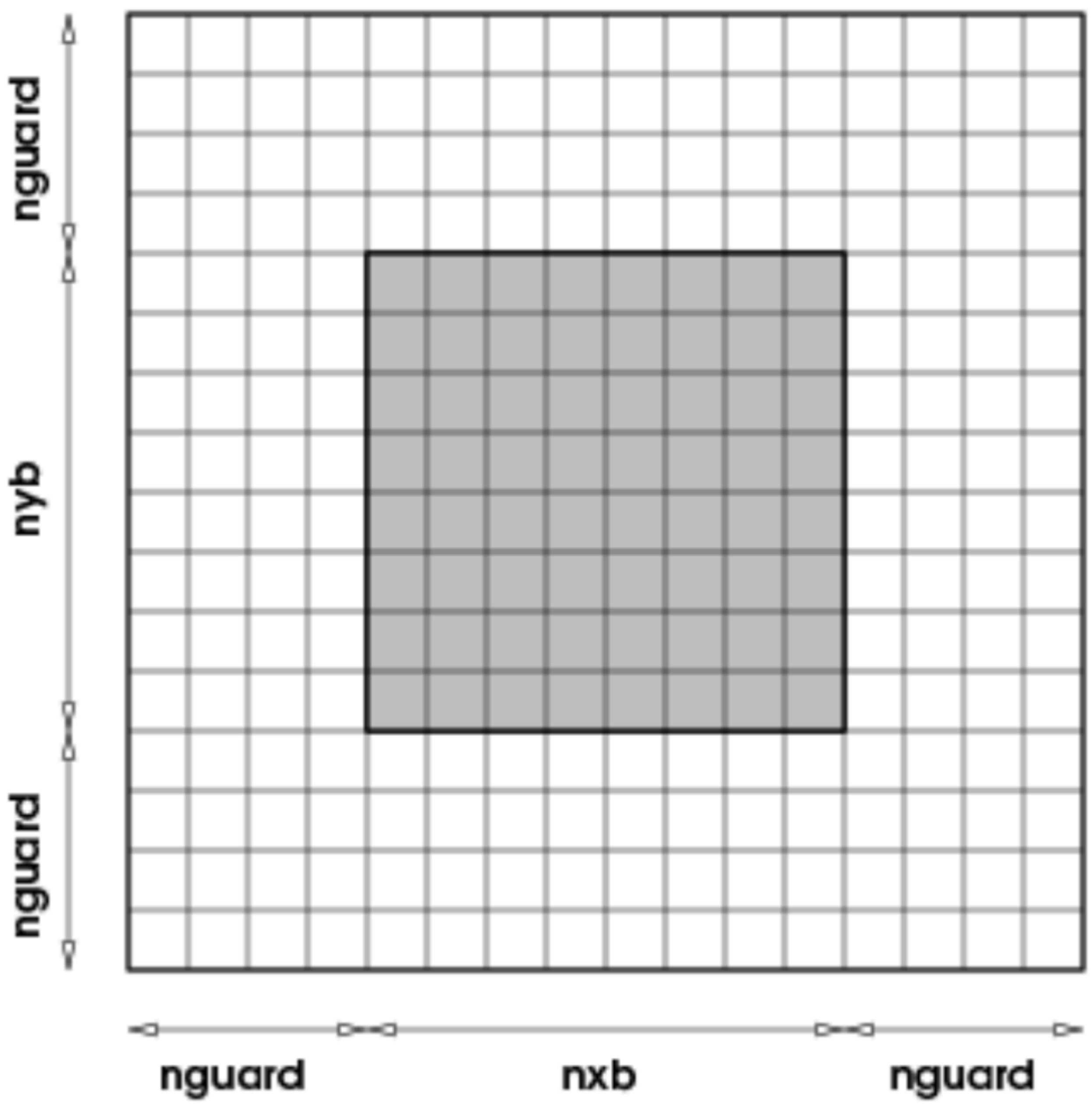


FIG. 3.—Morton space-filling curve for an arbitrary set of blocks of differing spatial resolution

Leaf blocks (blockID)

FLASH Code: Simulation Domain

- Blocks



- The grid is composed of blocks
- FLASH4: all blocks are of same size
- May cover different fractions of the physical domain, depending on a block's resolution
- Data storage area for each block reserves space for some layers of guard cells
- Size of guard cells can be changed by putting a line “GUARDCELLS 6" in the Config file. Default is 4.

FLASH Code: Variables

- Density: DENS_VAR
- Velocities: VELX_VAR, VELY_VAR, VELZ_VAR
- Temperature: TEMP_VAR
- Pressure: PRES_VAR
- Specific internal energy: EINT_VAR

Simulations

- All sample simulations can be found in

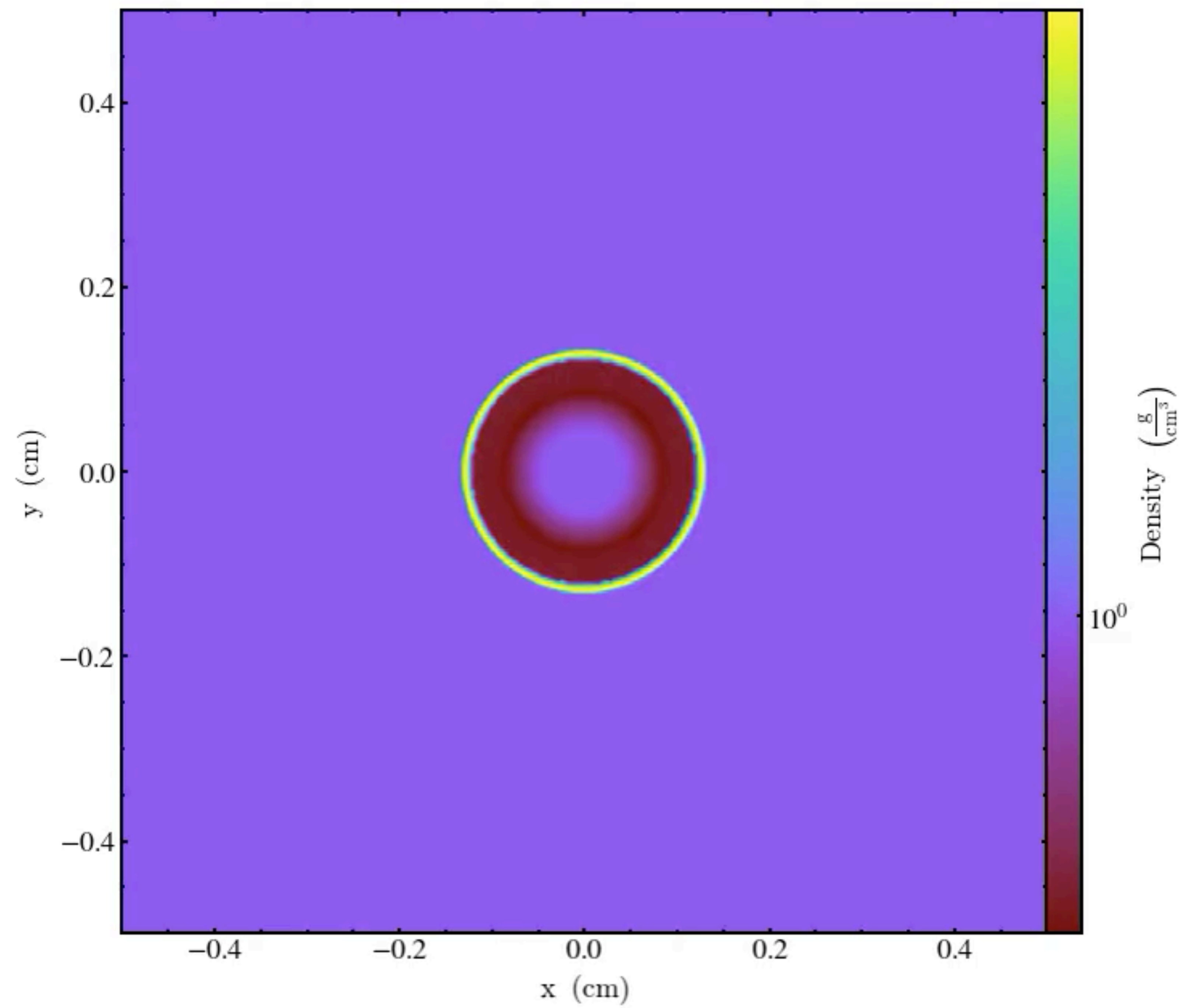
  FLASH/source/Simulation/SimulationMain/

- A simulation requires the following 5 files:
  - Config
  - Makefile:
  - Simulation_data.F90
  - Simulation_init.F90
  - Simulation_initBlock.F90

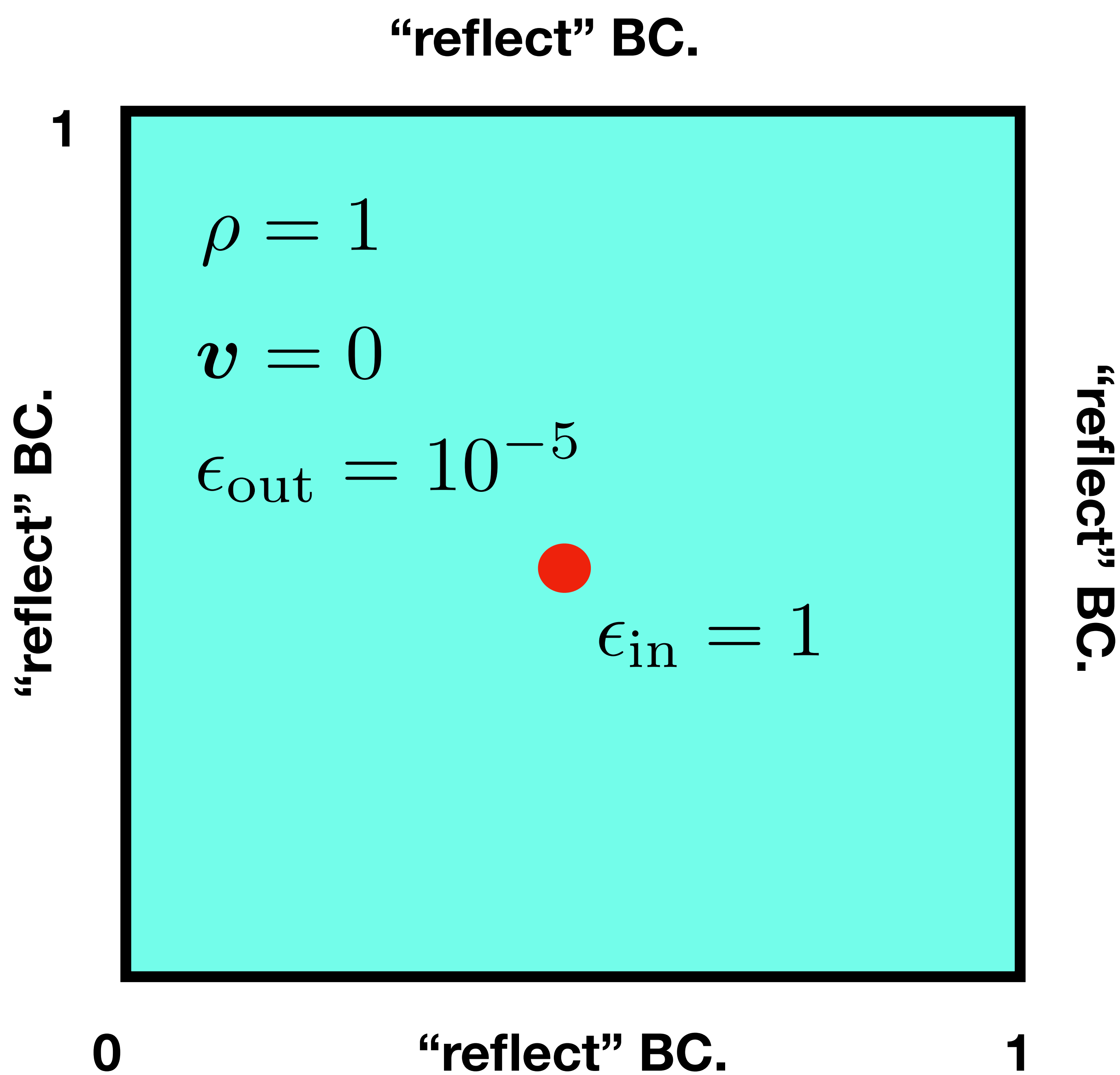
Design a FLASH simulation

- All sample simulations can be found in `FLASH/source/Simulation/SimulationMain/`
- A simulation requires the following 5 files:
    - Config
    - Makefile:
    - Simulation_data.F90
    - Simulation_init.F90
    - Simulation_initBlock.F90

A simple explosion simulation



The initial condition



Config

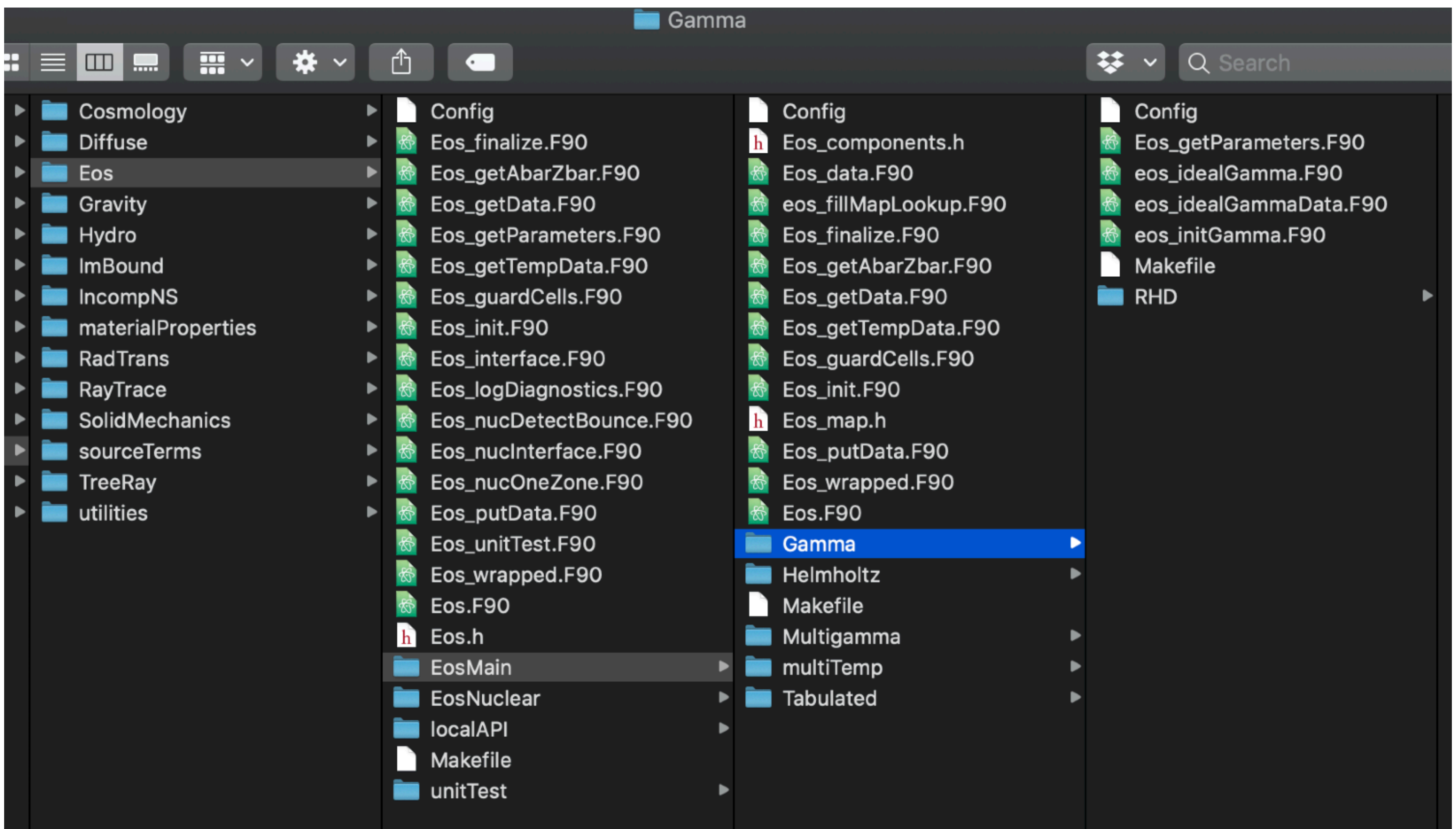
Config + (~/codes/FLASH/s...ationMain/Template) - VIM

```
# The Configuration file for a test problem

REQUIRES Driver
REQUIRES physics/Hydro
REQUIRES physics/Eos

D sim_rho0 The ambient density [g/cc]
PARAMETER sim_rho0   REAL     1.        [0 to ]


~
~
~
Config [+]                                   11,0-1          All
```

pan   vega.astr.nthu.edu.tw   ~/c/F/s/S/S/Template   vim ◂ zsh   8/28, 3:16 PM

Using a Flash Unit

- REQUIRES physics/EoS
- Default: physics/Eos/EosMain/Gamma



Makefile

Makefile (~/codes/FLASH/s...ationMain/Template) - VIM

```
Simulation += Simulation_data.o
Simulation_init.o : Simulation_data.o
Simulation_initBlock.o : Simulation_data.o

```

```
Makefile                                              5,0-1          All
"Makefile" 5L, 116C written
```

pan   vega.astr.nthu.edu.tw   ~/c/F/s/S/S/Template   vim ◂ zsh   8/28, 3:17 PM

Simulation_data.F90

```
module Simulation_data
#include "Flash.h"
  implicit none

  !! *** Runtime Parameters *** !!
  real, save   :: sim_rho0

  !! *** Variables pertaining to Simulation
  logical, save :: sim_gCell

  integer, save :: sim_meshMe
end module Simulation_data

```

Simulation_init.F90

```fortran
subroutine Simulation_init()

  use Simulation_data
  use Driver_interface, ONLY : Driver_getMype, Driver_abortFlash
  use RuntimeParameters_interface, ONLY : RuntimeParameters_get
  use Logfile_interface, ONLY : Logfile_stamp
  implicit none
#include "constants.h"
#include "Flash.h"
#include "Multispecies.h"

  call Driver_getMype(MESH_COMM, sim_meshMe)

  call RuntimeParameters_get('sim_rho0', sim_rho0)

  call Logfile_stamp( "initializing the Template problem",  &
       "[Simulation_init]")

end subroutine Simulation_init
```

Simulation_init.F90

```fortran
subroutine Simulation_init()

  use Simulation_data
  use Driver_interface, ONLY : Driver_getMype, Driver_abortFlash
  use RuntimeParameters_interface, ONLY : RuntimeParameters_get
  use Logfile_interface, ONLY : Logfile_stamp
  implicit none
#include "constants.h"
#include "Flash.h"
#include "Multispecies.h"

  call Driver_getMype(MESH_COMM, sim_meshMe)

  call RuntimeParameters_get('sim_rho0', sim_rho0)

  call Logfile_stamp( "initializing the Template problem",  &
        "[Simulation_init]")

end subroutine Simulation_init
```

Simulation_initBlock.F90

```fortran
!!***

subroutine Simulation_initBlock(blockID)

#include "constants.h"
#include "Flash.h"
#include "Eos.h"

  use Simulation_data, ONLY : sim_rho0

  use Grid_interface, ONLY : Grid_getBlkIndexLimits, &
    Grid_getCellCoords, Grid_getBlkPtr, Grid_releaseBlkPtr
  use Eos_interface, ONLY : Eos, Eos_wrapped


  implicit none

  integer, intent(in) :: blockID


  !! Handling AMR blocks and pointer

  integer :: i, j, k, n
  real, dimension(:,:,:,:),pointer :: solnData
  integer, dimension(2,MDIM) :: blkLimits, blkLimitsGC
  integer :: sizeX,sizeY,sizeZ
  logical :: gcell = .true.

  !! Physical cooridnates
  real :: xx, yy, zz
  real :: radius
  real,allocatable, dimension(:) ::yCenter, xCoord, zCoord

  !! physical parameters
  real :: rho, pres, velx, vely

```

```fortran
 ! get the integer index information for the current block
  call Grid_getBlkIndexLimits(blockId,blkLimits,blkLimitsGC)
  call Grid_getBlkPtr(blockId,solnData)


  sizeX = blkLimitsGC(HIGH,IAXIS)
  sizeY = blkLimitsGC(HIGH,JAXIS)
  sizeZ = blkLimitsGC(HIGH,KAXIS)
  allocate(yCenter(sizeY))
  allocate(xCoord(sizeX))
  allocate(zCoord(sizeZ))
  yCenter = 0.0
  xCoord = 0.0
  zCoord = 0.0

  if (NDIM == 3) call Grid_getCellCoords&
                     (KAXIS, blockId, CENTER,gcell, zCoord, sizeZ)
  if (NDIM >= 2) call Grid_getCellCoords&
                     (IAXIS, blockId, CENTER,gcell, xCoord, sizeX)

  call Grid_getCellCoords(JAXIS, blockId, CENTER, gcell, yCenter, sizeY)

```

Simulation_initBlock.F90

```
 do k = blkLimits(LOW,KAXIS),blkLimits(HIGH,KAXIS)

    ! get the coordinates of the cell center in the z-direction
    zz = zCoord(k)
    do j = blkLimits(LOW,JAXIS),blkLimits(HIGH,JAXIS)
       ! get the coordinates of the cell center in the y-direction
       yy = yCenter(j)
       ! The position in the current yz-row.
       do i = blkLimits(LOW,IAXIS),blkLimits(HIGH,IAXIS)

          ! get the cell center, left, and right positions in x
          xx  = xCoord(i)

          radius = sqrt(xx**2 + yy**2 + zz**2)

          if (radius .le. 0.1) then
             pres = (4./3. - 1.)*1.0
          else
             pres = 1.e-5
          endif
          rho   = sim_rho0
          velx = 0.0
          vely = 0.0

          solnData(DENS_VAR, i,j,k) = rho
          solnData(PRES_VAR, i,j,k) = pres
          solnData(VELX_VAR, i,j,k) = velx
          solnData(VELY_VAR, i,j,k) = vely
          solnData(VELZ_VAR, i,j,k) = 0.0

          enddo
       enddo
 enddo

 call Grid_releaseBlkPtr(blockID, solnData)
 call Eos_wrapped(MODE_DENS_PRES,blkLimits,blockId)

 deallocate(yCenter)
 deallocate(xCoord)
 deallocate(zCoord)

 return
end subroutine Simulation_initBlock
```

Setup the Simulation

FLASH/source/Simulation/SimulationMain/Template

- A machine depend `Makefile.h`

```
~/codes/FLASH/sites(sim_school*) » ls
Aliases                     buckbeak                    ellipse09.uchicago.edu      hyades.ucsc.edu              mhd2.asci.uchicago.edu       scooter.asci.uchicago.edu
FomalhautIMP                cetus.asci.uchicago.edu     ellipse10.uchicago.edu      hydra.si.edu                 midway2.rcc.uchicago.edu     seaborg.nersc.gov
FomalhautMBP4               clark.asci.uchicago.edu     ellipse11.uchicago.edu      icc-9.0_fornax.uchicago.edu  mira.alcf.anl.gov            skeeter.asci.uchicago.edu
P640278.nist.gov            clogin1                     ellipse_pgf.uchicago.edu    ignition                     mongchi.soe.ucsc.edu         sphere.uchicago.edu
Prototypes                  code.uchicago.edu           eugenia.asci.uchicago.edu   intel-hedp                   mongchi.uchicago.edu         splash.seas.gwu.edu
SEAS10926.gwu.edu           coyote.lanl.gov             eureka.alcf.anl.gov         intel-mpi2                   myristica.asci.uchicago.edu  splash.seas.gwu.edu_KPD
SEAS10927.gwu.edu           crash2.umd.edu              fen.bluegene.bnl.gov        intel-ompi                   nag-mpi1                     sunspot.uchicago.edu
SEAS10927UB.gwu.edu         cthinkpad                   fenp.bluegene.bnl.gov       intrepid.alcf.anl.gov        nag-mpi2                     surveyor.alcf.anl.gov
SEAS12982.gwu.edu           ctsv.astro.sunysb.edu       flash.uchicago.edu          jacquard.nersc.gov           nagini.uchicago.edu          tp-login2
SEAS13033.gwu.edu           cube.uchicago.edu           flashviz.uchicago.edu       jaguar.ccs.ornl.gov          oakley.osc.edu               tsoodzil.astro.uiuc.edu
absoft-mpi2                 datastar.sdsc.edu           fleetwood.astro.sunysb.edu  jaguar.nccs.gov              optix.cs.uoregon.edu         tuxedo.uchicago.edu
alc.llnl.gov                deepthought.umd.edu_tfitz   fomalhaut                   jubl.zam.kfa-juelich.de      osel.uchicago.edu            uffda.asci.uchicago.edu
amazon.nswccd               duce.gsfc.nasa.gov          fornax.uchicago.edu         karloff.lbl.gov              p655-4.nic.uoregon.edu       variable.as.arizona.edu
animal5                     edison.nersc.gov            franklin.nersc.gov          khorba.uchicago.edu          pg-mpi2-32                   variable.ph.ua.edu
archimedes.uchicago.edu     elan.uchicago.edu           fusion.lcrc.anl.gov         klaus-laptop                 purple.llnl.gov              vega.astr.nthu.edu.tw
asterix.asci.uchicago.edu   eldorado.astro.sunysb.edu   fusion.lcrc.uchicago.edu    kraken.nics.tennessee.edu    pyramid.uchicago.edu         vestalac1
babbage.nersc.gov           eldorado.uchicago.edu       gin.asci.uchicago.edu       lahey-mpi2                   qsc.lanl.gov                 watanlsn.watson.ibm.com
bassi.nersc.gov             ellipse.uchicago.edu        gnu-ins                     lenovolaptop                 r1.oit.ua.edu                zeus.llnl.gov
beagle.ci.uchicago.edu      ellipse02.uchicago.edu      gnu-mpi2                    liturchi.uchicago.edu        ramsusii.mps.ohio-state.edu  zingiber.asci.uchicago.edu
bgl.llnl.gov                ellipse03.uchicago.edu      gnu-ompi                    login1.pads.ci.uchicago.edu  ranger.tacc.utexas.edu       zingiber.uchicago.edu
bgl.mcs.anl.gov             ellipse04.uchicago.edu      hawkmoon.uchicago.edu       login2.pads.ci.uchicago.edu  rc2.ua.edu
bgl.sdsc.edu                ellipse05.uchicago.edu      hera.llnl.gov               lupin.uchicago.edu           redstorm.sandia.gov
bonsai.cfa.harvard.edu      ellipse06.uchicago.edu      hopper.nersc.gov            macbro.uchicago.edu          saguaro.fulton.asu.edu
brassica.asci.uchicago.edu  ellipse08.uchicago.edu      hpc.msu.edu                 maitreyi                     scarf.rl.ac.uk
```

Setup the Simulation

- Run a setup script

A simple example

```
./setup Template -2d -auto
./setup Template -2d -auto -maxblocks=2000 -nxb=8 -nyb=8
```

A more complicated example

```
~/codes/FLASH(sim_school*) » ./setup CoreCollapse/IDSA -auto -2d +cylindrical -objdir ccsn2dIdsa -nxb=16 -nyb=16 -max
blocks=1400 threadBlockList=False +pm4dev threadWithinBlock=False +newMpole --with-unit=source/physics/RadTrans/RadTr
ansMain/IDSA/rk2
```

Setup the Simulation

```
~/codes/FLASH(sim_school*) » ./setup
Processing Shortcut file: /Users/pan/codes/FLASH/bin/setup_shortcuts.txt
usage:  setup <problem-name> [options] [VAR=VALUE]...

   problem-name: see source/Simulation/SimulationMain directory
   options:

   (Science Options)
            -auto -[123]d
            -maxblocks=<#> -nxb=<#> -nyb=<#> -nzb=<#>
            -with-unit=<unit> -with-library=<libname>[,args]
            -without-unit=<unit> -without-library=<libname>

   (Setup and Make Options)
            -verbose=[DEBUG|INFO|WARN|IMPINFO|ERROR]
            [-site=<site> | -ostype=<ostype>]
            -makefile=<extension>
            [-opt| -debug | -test ]
            -objdir=<relative obj directory>
            -defines=<defines> -unitsfile=<filename>
            -datafiles=<wildcard> -parfile=<filename>
            -fbs -nofbs -tau=<makefile>

   (Misc Options)
            -makehide -noclobber -portable -help

   * For GNU compatibility, options may be prefixed by -- instead of - as well
   * -unit and -library are considered equivalent to
     -with-unit and -with-library respectively.
   * For information regarding the [VAR=VALUE] options and using 'setup variables'
     refer to User's Guide.
   * To read how shortcuts work see README.shortcuts in your bin directory
```

Setup shortcuts

```
To use a shortcut add '+shortcut' to your setup line.
For example ./setup Sod -auto +ug

    3t                           ThreeT=1 --defines=FLASH_3T
    3tr                          +3t RadFlahThreeT=True
    8wave                        --with-unit=physics/Hydro/HydroMain/split/MHD_8Wave +grid --gridinterpolation=native
    asynclaser                   +laser useAsyncLaser=True --without-unit=physics/sourceTerms/EnergyDeposition/EnergyDeposition
Main/Laser/LaserIO
    cartesian                    --geometry=cartesian
    chombo_amr                   --unit=Grid/GridMain/Chombo/AMR --index-reorder Grid=Chombo --nofbs --makefile=chombo chomboCo
mpatibleHydro=True
    chombo_ug                    --unit=Grid/GridMain/Chombo/UG --index-reorder Grid=Chombo --maxblocks=1 --nofbs --makefile=ch
ombo chomboCompatibleHydro=True
    cube16                       --nxb=16 --nyb=16 --nzb=16
    cube32                       --nxb=32 --nyb=32 --nzb=32
    cube64                       --nxb=64 --nyb=64 --nzb=64
    curv-pm2                     +pm2 --unit=Grid/GridMain/paramesh/Paramesh2 --with-unit=Grid/GridMain/paramesh/Paramesh2/mono
tonic
    curvilinear                  --curvilinear
    cylindrical                  --geometry=cylindrical
    ddt                          use_ddt=True --defines=DDT --with-unit=source/monitors/Debugger/DebuggerMain/dmalloc
    default                      --with-library=mpi +io +grid --gridinterpolation=monotonic FlashAvoidOrrery=True
    dmalloc                      use_dmalloc=True --defines=DMALLOC --with-unit=source/monitors/Debugger/DebuggerMain/dmalloc
    dynamicmem-pm3               +dynamicmem-pm40
    dynamicmem-pm40              +pm40 ParameshLibraryMode=True
    gravmgrid                    +pm4dev --with-unit=physics/Gravity/GravityMain/Poisson/Multigrid
    gravmpole                    --with-unit=physics/Gravity/GravityMain/Poisson/Multipole
    gravpfftnofbs                +ug +nofbs --with-unit=physics/Gravity/GravityMain/Poisson/Pfft
    grid                         --unit=Grid
    hdf5                         IO=hdf5
    hdf5typeio                   +io +parallelIO +hdf5 typeIO=True
    io                           --with-unit=IO
    laser                        --unit=physics/sourceTerms/EnergyDeposition/EnergyDepositionMain/Laser --without-unit=Particle
s
```

Setup shortcuts (continues)

```
   newmpole                     +noMgrid +noDefaultMpole +gravMpole --with-unit=Grid/GridSolvers/Multipole_new
   nodefaultmpole               --without-unit=Grid/GridSolvers/Multipole
   nofbs                        --nofbs +ug parallelIO=True
   noio                         --without-unit=physics/sourceTerms/EnergyDeposition/EnergyDepositionMain/Laser/LaserIO --witho
ut-unit=IO
   nolog                        --without-unit=monitors/Logfile
   nomgrid                      --without-unit=physics/Gravity/GravityMain/Poisson/Multigrid
   npg                          npg=True
   parallelio                   parallelIO=True
   pic                          +ug --unit=Grid/GridParticles/GridParticlesMove --without-unit=Grid/GridParticles/GridParticle
sMove/UG --without-unit=Grid/GridParticles/GridParticlesMove/UG/Directional
   pipeline                     --unit=multiprocessorTools/Pipeline
   pm2                          +grid Grid=PM2
   pm3                          +pm40
   pm40                         +grid Grid=PM40
   pm4dev                       +pm4dev_basic FlashAvoidOrrery=True
   pm4dev_basic                 +grid Grid=PM4DEV
   pm4dev_clean                 +pm4dev_basic FlashAvoidOrrery=False
   pm4dev_fixed                 +pm4dev ParameshLibraryMode=False
   pnetcdf                      IO=pnetcdf
   pnettypeio                   +io +parallelIO +pnetcdf typeIO=True
   polar                        --geometry=polar
   protonemission               --unit=diagnostics/ProtonEmission --without-unit=Particles
   protonimaging                --unit=diagnostics/ProtonImaging --without-unit=Particles
   ptdens                       --without-unit=Particles/ParticlesInitialization/Lattice --without-unit=Particles/ParticlesIni
tialization/WithDensity/CellMassBins --unit=Particles/ParticlesMain --unit=Particles/ParticlesInitialization/WithDensity --pa
rticlemethods=TYPE=passive,INIT=With_Density
   ptio                         +ug --with-unit=Particles
   purehydro                    physicsMode=hydro
   rnf                          --3d --nxb=8 --nyb=16 --nzb=32 --nofbs +ug
   serialio                     parallelIO=False
   spherical                    --geometry=spherical
   spherical-pm2                +pm2 +spherical
   splithydro                   --with-unit=physics/Hydro/HydroMain/split --without-unit=physics/Hydro/HydroMain/unsplit Split
Driver=True
   supportppmupwind             SupportPpmUpwind=True
```

Compile source codes

- An object/ folder will be generated

```
cd object

make
```

```
 mpi_unpack_edges.o mpi_unpack_fluxes.o mpi_unpack_tree_info.o mpi_wrapper_dble.o mpi_wrapper_int
 mpi_wrapper_real.o nameSyntaxError.o nameValueLL_bcast.o nameValueLL_data.o nameValueLL_get.o na
alueLL_getNum.o nameValueLL_logRules.o nameValueLL_rules.o nameValueLL_set.o paramesh_comm_data.o
aramesh_interfaces.o paramesh_mpi_interfaces.o pc_interface.o pc_utilities.o physicaldata.o poiss
ist.o prolong_arrays.o rationalize_fetch_list.o removeNullChar.o rp_getArgument.o rp_getOpt.o rp_
s.o rp_storeIgnoredParams.o send_block_data.o setup_buildstamp.o setup_buildstats.o setup_flashRe
its.o timings.o tmr_buildSummary.o tmr_create.o tmr_etime.o tmr_findTimerIndex.o tmr_getMaxCallSt
erParents.o tmr_init.o tmr_lookupIndex.o tmr_stackLib.o tree.o tree_search_for_surrblks.o umap.o
constRealZeroFn1.o ut_conversionInterface.o ut_convertToArrayIndicies.o ut_convertToMemoryOffset.
Interface.o ut_getFreeFileUnit.o ut_hunt.o ut_interpolationInterface.o ut_parabolicInterpol.o ut_
o ut_qsort.o ut_qsortC.o ut_qsortInterface.o ut_quadraticInterpol.o ut_sortInterface.o ut_sortOnP
t_sysMemInterface.o ut_sysMemStats.o ut_sysMemSummaryStats.o ut_sys_mem_usage.o workspace.o -L/Us
1-gnu/lib -lhdf5 -lhdf5_fortran -lz -DH5_USE_16_API
SUCCESS
```

Run Flash

- Requires

  the executable file `flash4` and

  the runtime parameter file `flash.par`

Runtime Parameters (flash.par)

```
#   Runtime parameters for the Kelvin-Helmh

#   Parameters for initial model

sim_rho0 = 1.0


#       Grid dimensionality and geometry

geometry = cartesian

#       Size of computational volume

xmin        =  -0.5
xmax        =   0.5
ymin        =  -0.5
ymax        =   0.5

```

Runtime Parameters

```
#         Boundary conditions

xl_boundary_type = "reflect"
xr_boundary_type = "reflect"

yl_boundary_type = "reflect"
yr_boundary_type = "reflect"


#    Simulation (grid, time, I/O) parameters

cfl                  = 0.8
basenm               = "my_sim_"
restart              = .false.

```

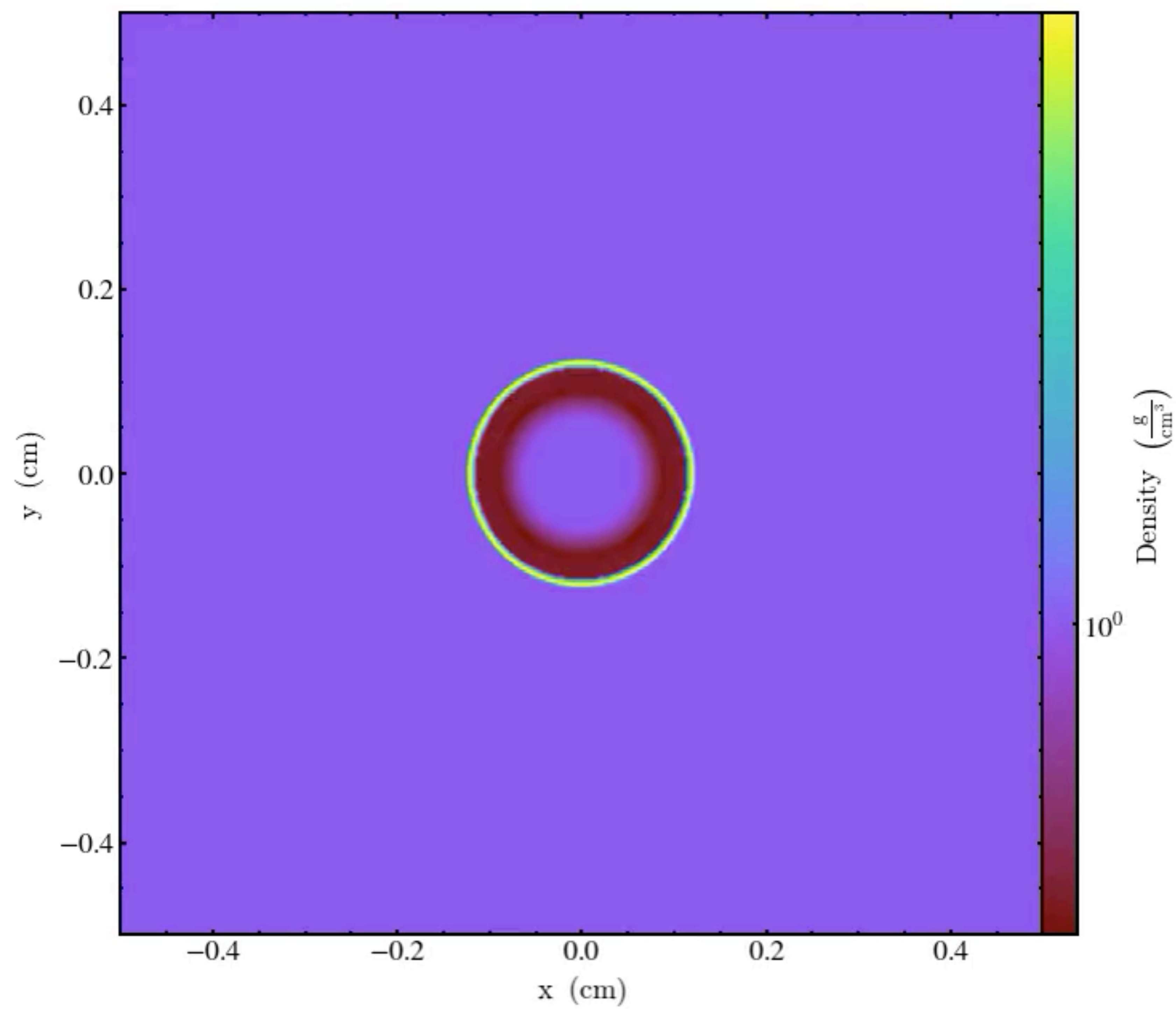
Runtime Parameters

```
#   checkpoint file output parameters
checkpointFileIntervalTime = 0.5
checkpointFileIntervalStep = 0
checkpointFileNumber       = 0

#   plotfile output parameters
plotfileIntervalTime = 0.1
plotfileIntervalStep = 0
plotfileNumber       = 0

nend                 = 1000000
tmax                 = 10.0

run_comment          = "my first flash simulation"
log_file             = "my_sim.log"
eintSwitch          = 1.e-4


plot_var_1 = "dens"
plot_var_2 = "pres"
plot_var_3 = "temp"
plot_var_4 = "velx"
plot_var_5 = "vely"
```

```
#     AMR refinement parameters
lrefine_min       = 1
lrefine_max       = 6
refine_var_1      = "dens"
refine_var_2      = "pres"

```

Outputs

- my_sim.log
- my_sim.dat
- my_sim_hdf5_chk_*
- my_sim_hdf5_plt_cnt_*



See the afternoon lectures for data analysis and visualization

Runtime parameters documentation

- http://flash.uchicago.edu/site/flashcode/user_support/rpDoc_4p61.py

RUNTIME PARAMETERS DOCUMENTATION FOR FLASH RELEASE 4.6.1

All Chapters

FLASH API

- http://flash.uchicago.edu/site/flashcode/user_support/robodoc-FLASH4_4p61/

FLASH4.6.1 API

ROBODoc 4.99 manual

Generated from /asc/asci2/site/flashcode/secure/release_4p6/ with ROBODoc v4.99.8 on Mon Sep 02 01:17:10 2019
[Sourcefiles] [Functions] [Modules]

FUNCTIONS

diagnostics/ProtonEmission/ProtonEmission
diagnostics/ProtonEmission/ProtonEmission_finalize
diagnostics/ProtonEmission/ProtonEmission_init
diagnostics/ProtonImaging/ProtonImaging
diagnostics/ProtonImaging/ProtonImaging_finalize
diagnostics/ProtonImaging/ProtonImaging_init
diagnostics/ThomsonScattering/ThomsonScattering
diagnostics/ThomsonScattering/ThomsonScattering_finalize
diagnostics/ThomsonScattering/ThomsonScattering_init
diagnostics/XrayImaging/XrayImaging
diagnostics/XrayImaging/XrayImaging_finalize
diagnostics/XrayImaging/XrayImaging_init
Driver/Driver_abortFlash
Driver/Driver_abortFlashC
Driver/Driver_checkMPIErrorCode
Driver/Driver_computeCellLocations
Driver/Driver_computeDt
Driver/Driver_diagnostics
Driver/Driver_driftBlock
Driver/Driver_driftSetSrcLoc
Driver/Driver_driftUnk
Driver/Driver_evolveFlash
Driver/Driver_finalizeFlash
Driver/Driver_finalizeSourceTerms
Driver/Driver_getComm
Driver/Driver_getDt
Driver/Driver_getElapsedWCTime
Driver/Driver_getMype
Driver/Driver_getNStep
Driver/Driver_getNumProcs
Driver/Driver_getSimTime
Driver/Driver_getTimeStamp
Driver/Driver_init
Driver/Driver_initFlash
Driver/Driver_initMaterialProperties
Driver/Driver_initNumericalTools
Driver/Driver_initParallel
Driver/Driver_initSourceTerms
Driver/Driver_logMemoryUsage
Driver/Driver_mpiThreadSupport
Driver/Driver_putTimeStamp
Driver/Driver_sendOutputData
Driver/Driver_setupParallelEnv
Driver/Driver_sourceTerms
Driver/Driver_superTimeStep
Driver/Driver_verifyInitDt
flashUtilities/Pipeline/Pipeline_finalize
flashUtilities/Pipeline/Pipeline_globalCheckStatus
flashUtilities/Pipeline/Pipeline_globalCheckStructure
flashUtilities/Pipeline/Pipeline_init
flashUtilities/Pipeline/Pipeline_localActivate
flashUtilities/Pipeline/Pipeline_localCreate
flashUtilities/Pipeline/Pipeline_localDeactivate
Particles/Particles_finalize
Particles/Particles_getCountPerBlk
Particles/Particles_getGlobalNum
Particles/Particles_getLocalNum
Particles/Particles_init
Particles/Particles_initData
Particles/Particles_initForces
Particles/Particles_initPositions
Particles/Particles_longRangeForce
Particles/Particles_manageLost
Particles/Particles_mapFromMesh
Particles/Particles_putLocalNum
Particles/Particles_sendOutputData
Particles/Particles_shortRangeForce
Particles/Particles_sinkAccelGasOnSinksAndSinksOnGas
Particles/Particles_sinkAdvanceParticles
Particles/Particles_sinkComputeDt
Particles/Particles_sinkCreateAccrete
Particles/Particles_sinkInit
Particles/Particles_sinkMarkRefineDerefine
Particles/Particles_sinkMoveParticles
Particles/Particles_sinkSortParticles
Particles/Particles_sinkSumAttributes
Particles/Particles_sinkSyncWithParticles
Particles/Particles_specifyMethods
Particles/Particles_unitTest
Particles/Particles_updateAttributes
Particles/Particles_updateGridVar
Particles/Particles_updateRefinement
PhysicalConstants/PhysicalConstants_get
PhysicalConstants/PhysicalConstants_init
PhysicalConstants/PhysicalConstants_list
PhysicalConstants/PhysicalConstants_listUnits
PhysicalConstants/PhysicalConstants_unitTest
physics/Cosmology/Cosmology_cdmPowerSpectrum
physics/Cosmology/Cosmology_computeDeltaCrit
physics/Cosmology/Cosmology_computeDt
physics/Cosmology/Cosmology_computeVariance
physics/Cosmology/Cosmology_finalize
physics/Cosmology/Cosmology_getOldRedshift
physics/Cosmology/Cosmology_getParams
physics/Cosmology/Cosmology_getRedshift
physics/Cosmology/Cosmology_init
physics/Cosmology/Cosmology_massToLength
physics/Cosmology/Cosmology_redshiftHydro
physics/Cosmology/Cosmology_redshiftToTime
physics/Cosmology/Cosmology_sendOutputData
physics/Cosmology/Cosmology_solveFriedmannEqn
physics/Cosmology/Cosmology_unitTest
physics/Diffuse/Diffuse
physics/Diffuse/Diffuse_advance1D
physics/Diffuse/Diffuse_computeDt
physics/Diffuse/Diffuse_computeFluxLimiter

Happy coding!