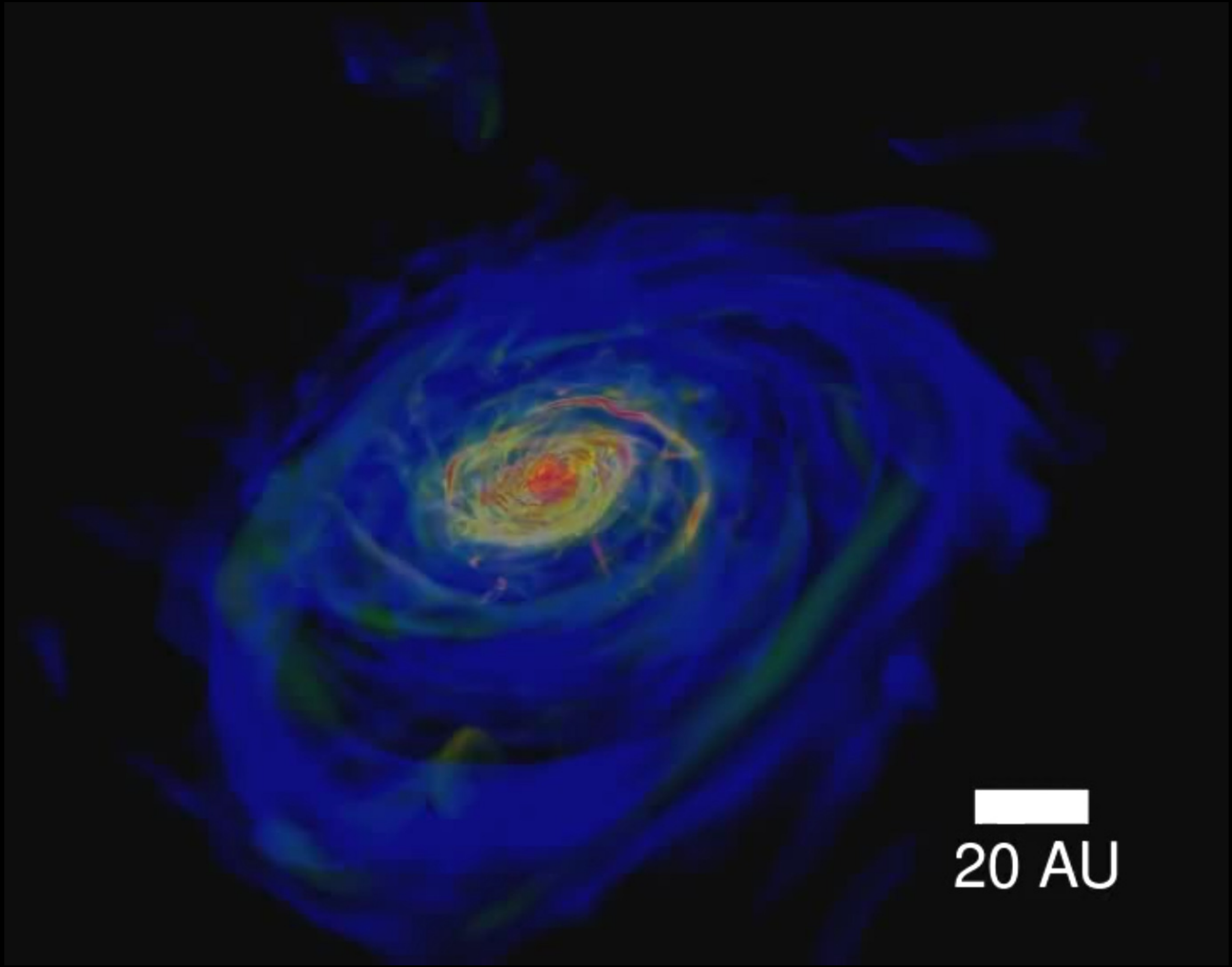
Grow your own planet...

How simulations help us understand the Universe

by caro (Carolin Kimmig, Uni Heidelberg) & miosta (Anna Penzlin, Uni Tübingen)



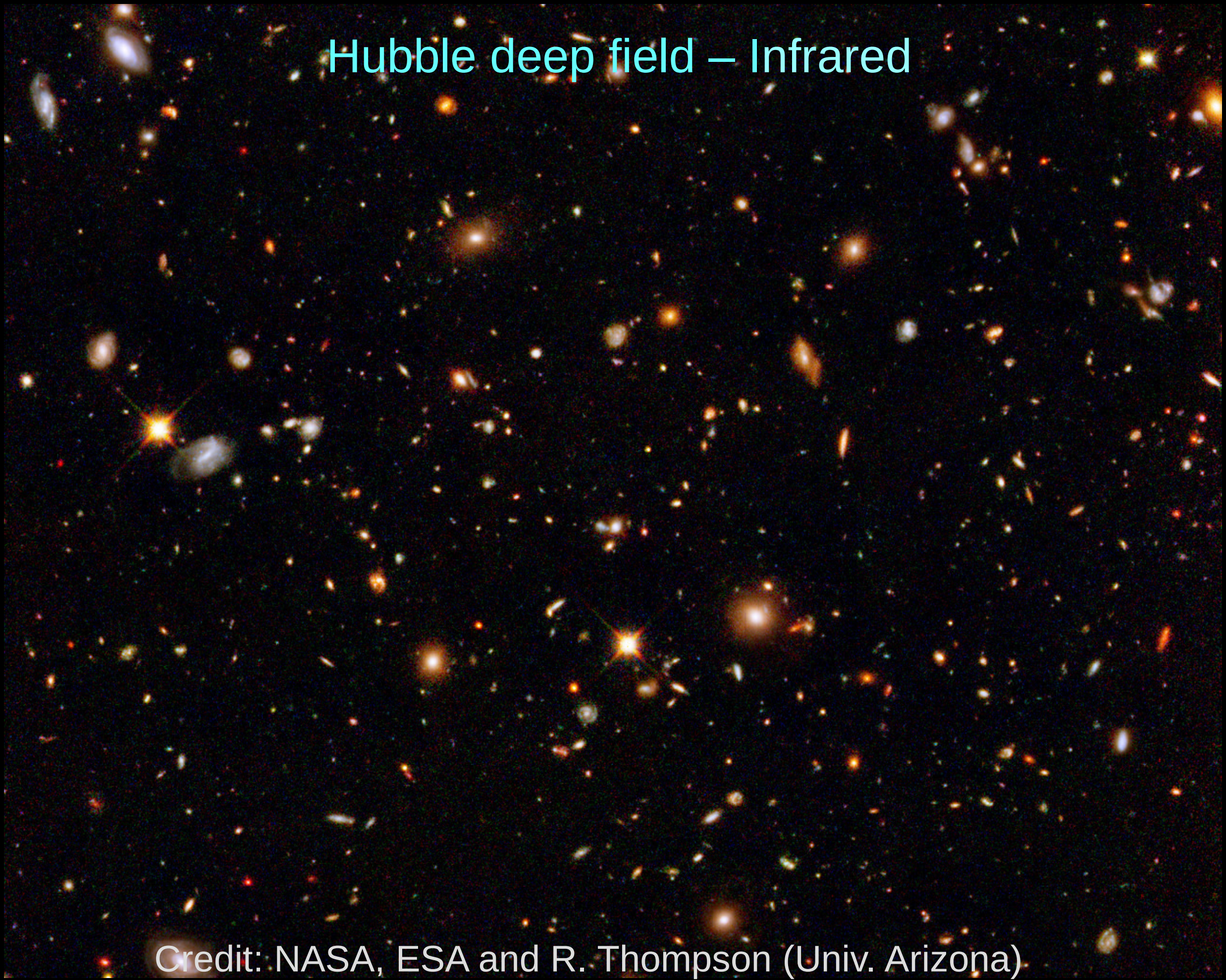
1 AU

is the distance from Earth to the Sun

150 mio km

Credit: Michael Küffmeier

Space – The final frontier



Empty except for a few specks called Galaxies

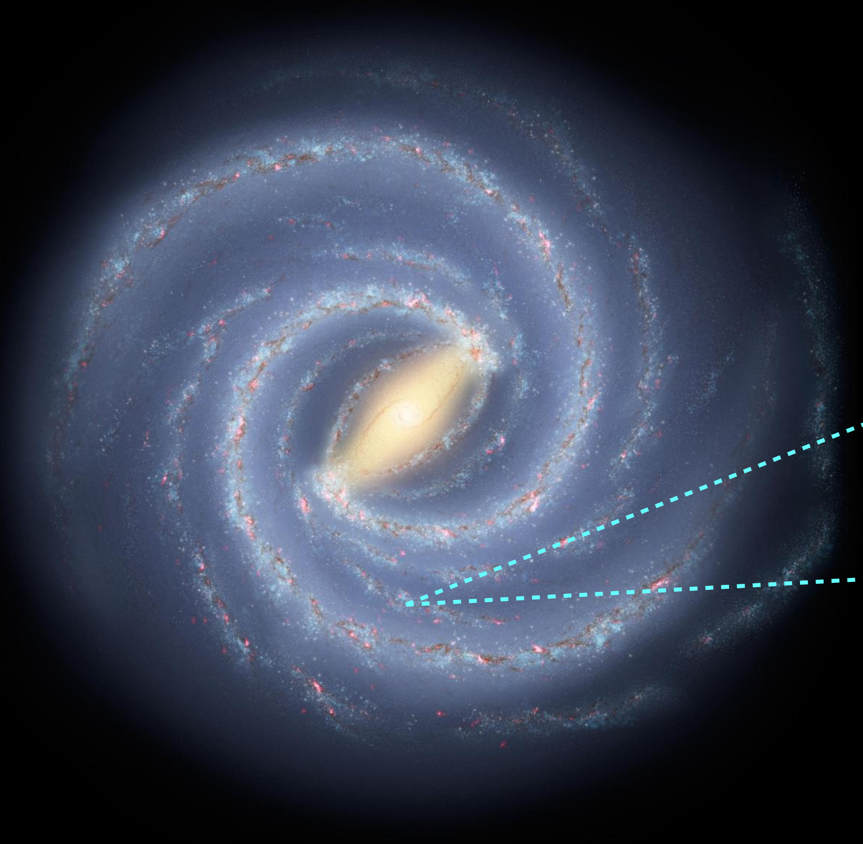
Contains:

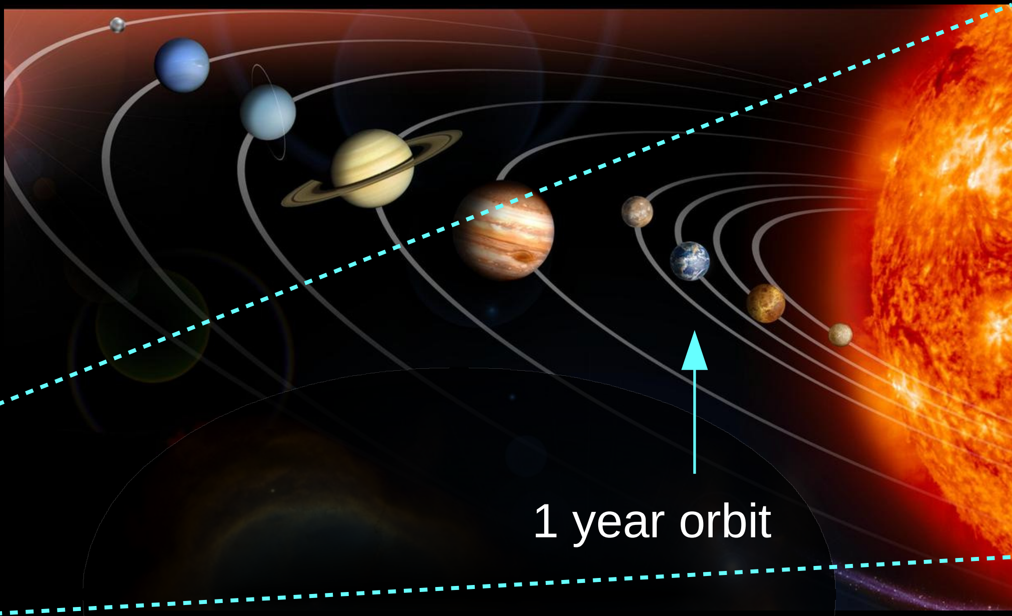
1) Mysterious expansion energy called dark energy
2) Mysterious extra gravity called dark matter
3) Hydrogen & helium
4) Everything else

3) and 4) is all we see in this Picture

Time – Back to the future?

Astronomical Timescales





Star formation takes about 10,000,000 years

Our Solar System formed 4,5 billion years ago

Credit: NASA

Hubble Space Telescope

It's all just **bubbly fluid**

–

So we just need
to get the flow

Credit: NASA, ESA, Hubble Heritage Team

How a star is born – The recipe

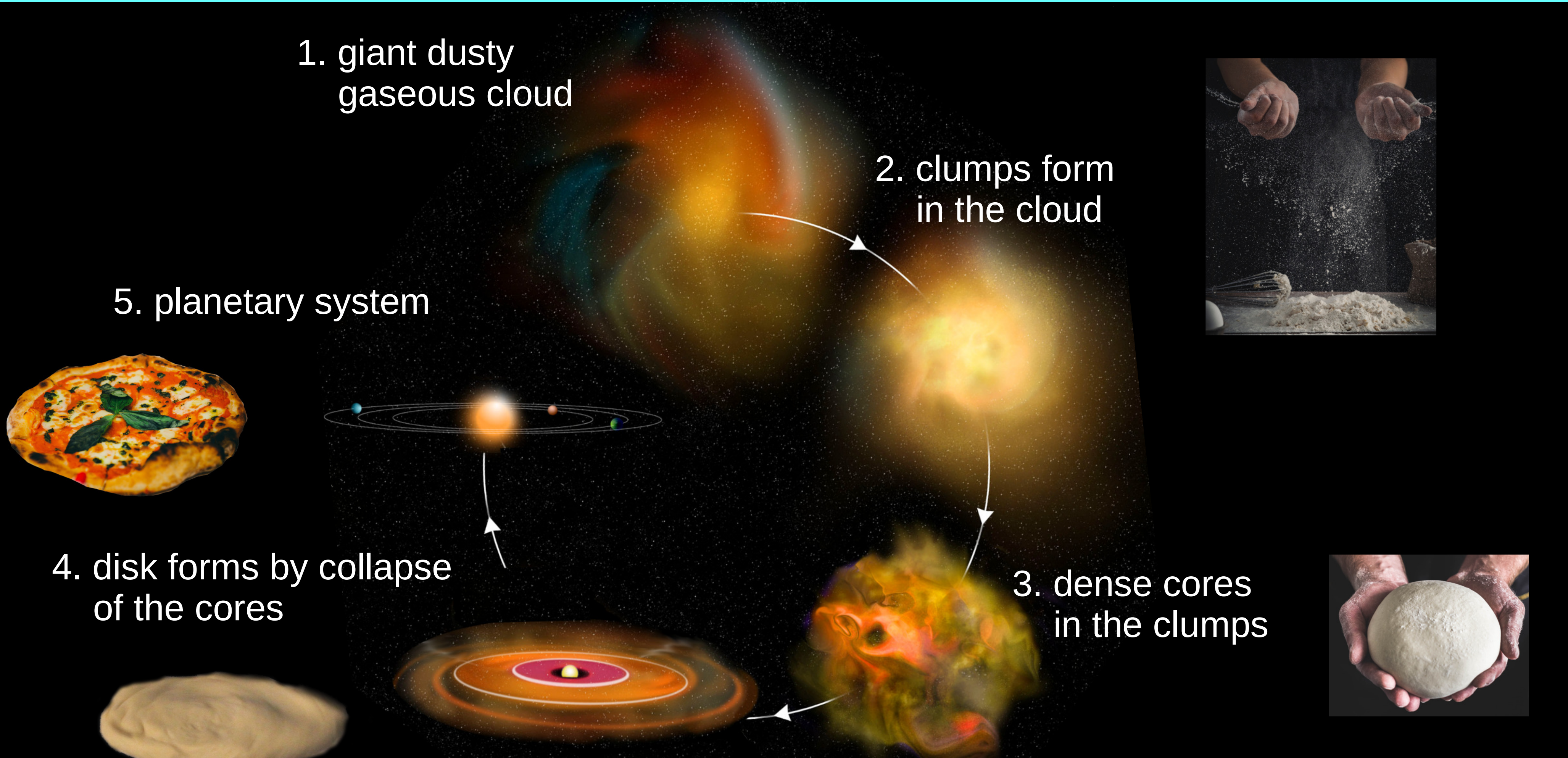
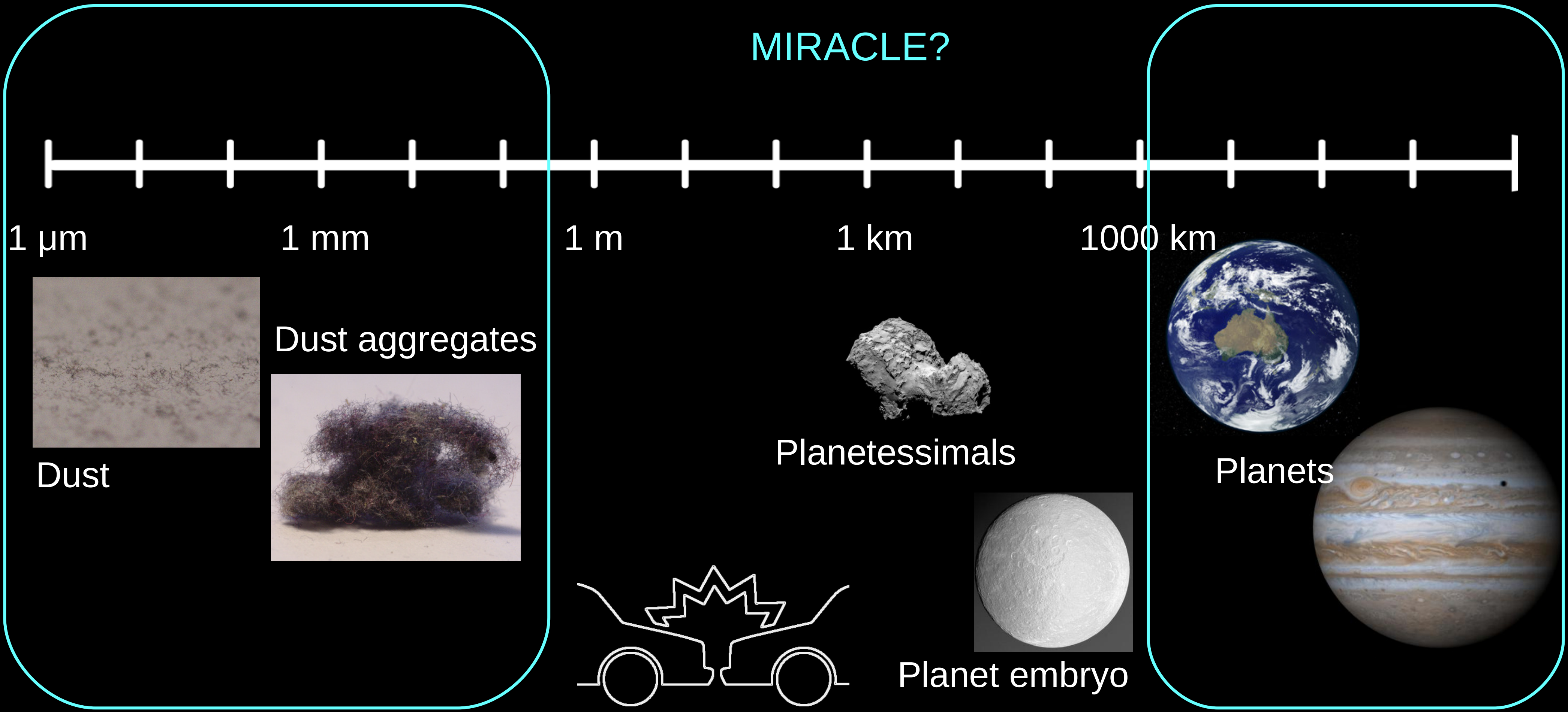


Image credit: Bill Saxton/NRAO/AUI/NSF

How a planet is born

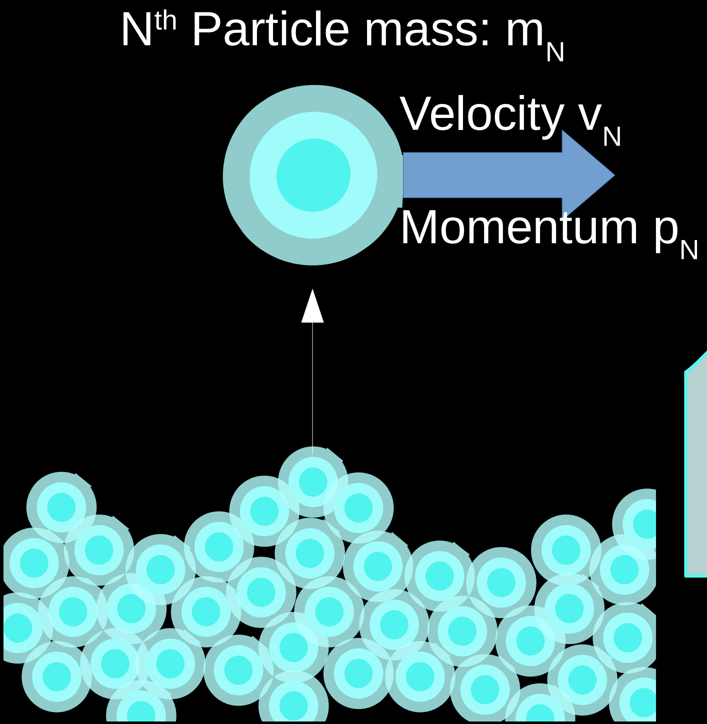


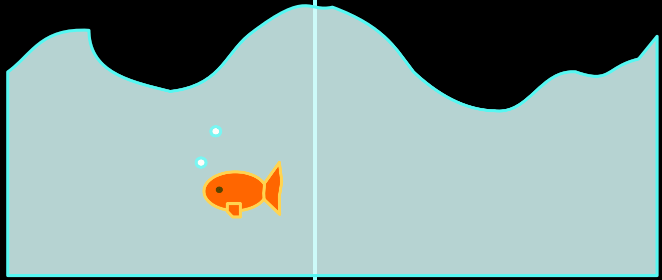
C. Dullemond

Image credit: NASA

Go with the flow or box it

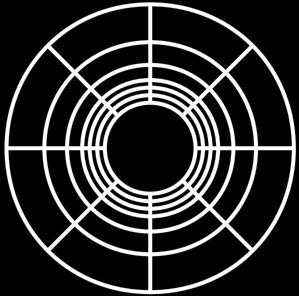
Lagrangian description

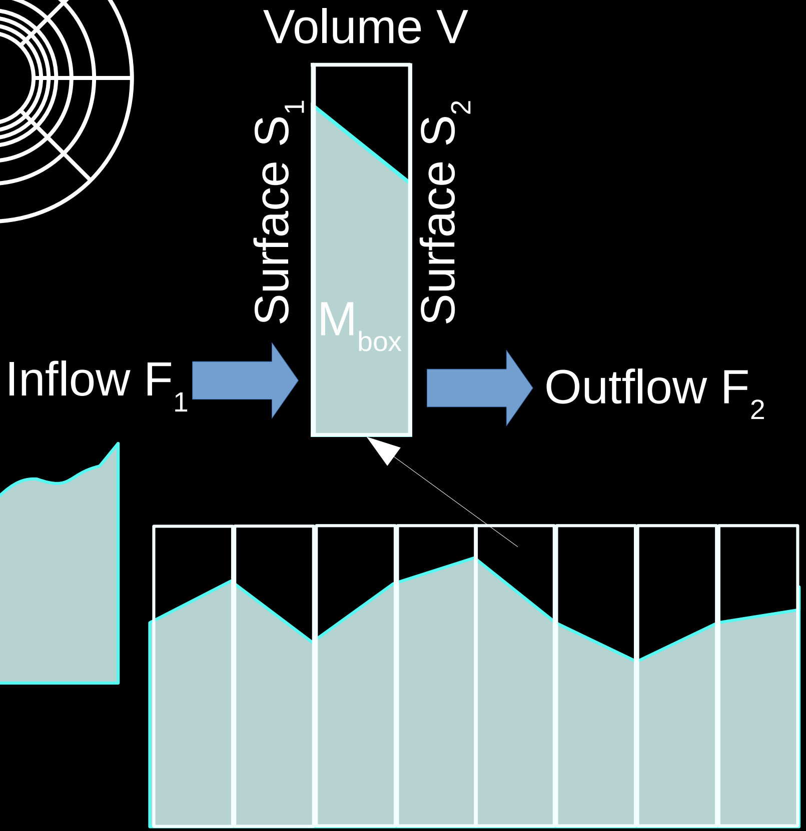




(see https://media.ccc.de/v/35c3-9451-simulating_universes)

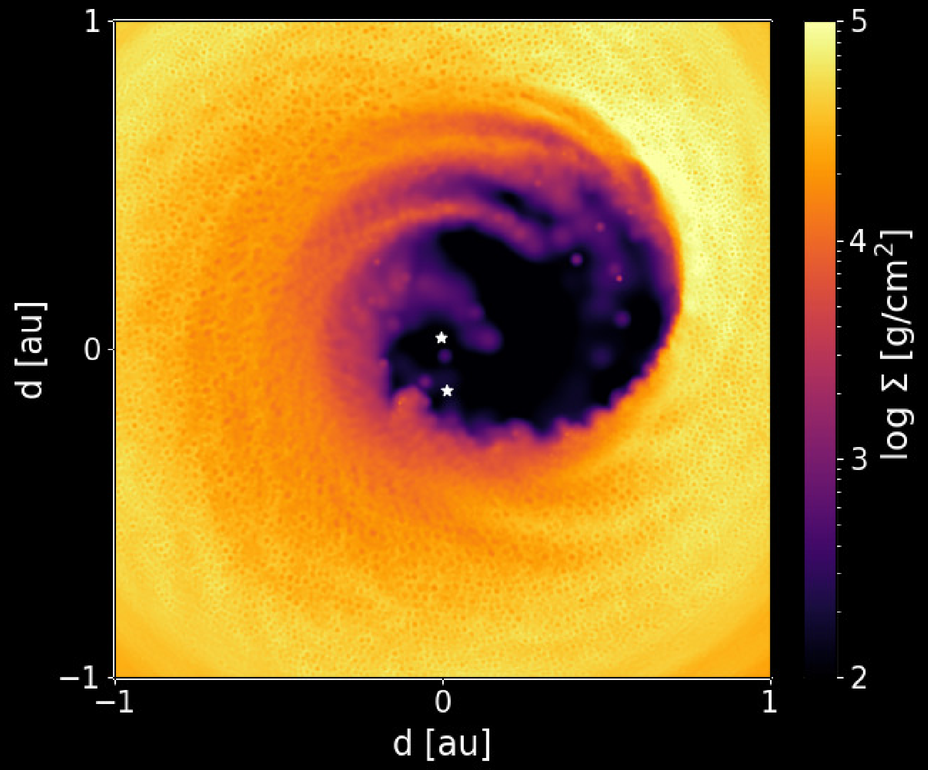
Eulerian description





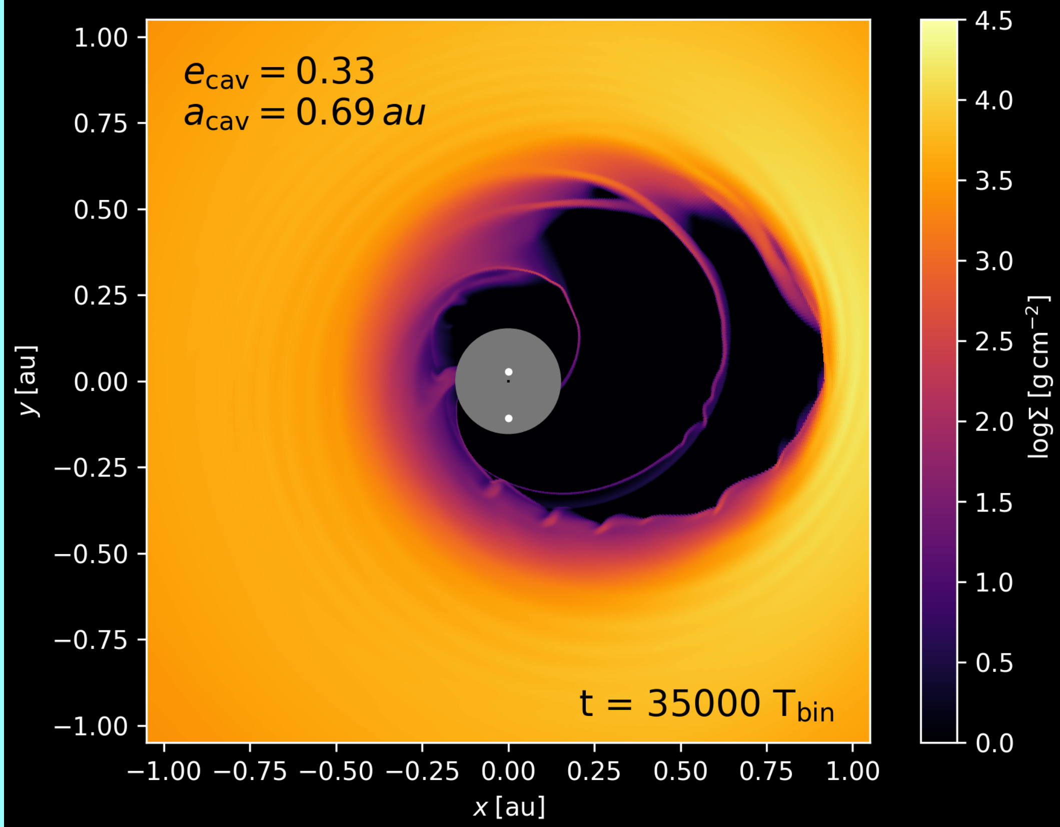
Go with the flow or box it

Lagrangian description



Hugo Audiffren 2019

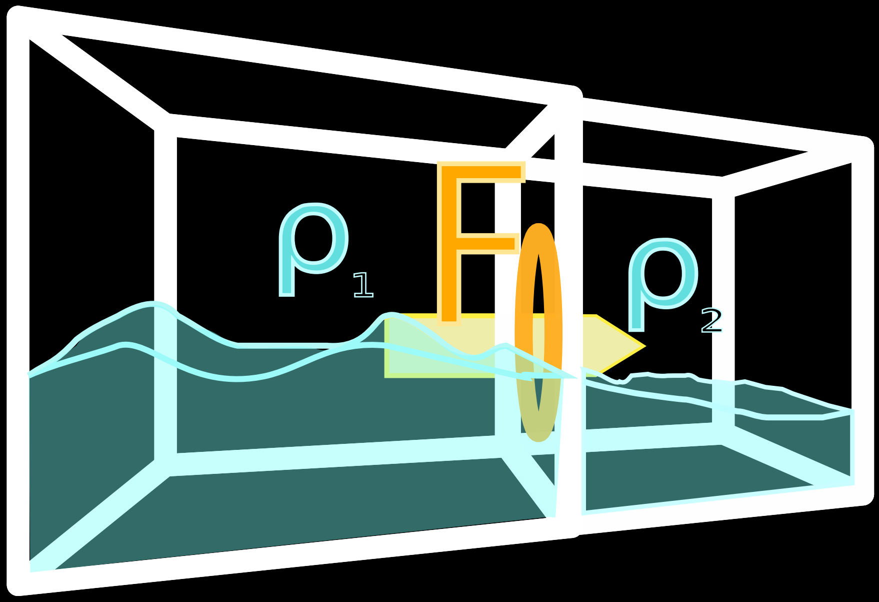
Eulerian description



Me (Anna Penzlin) 2019

Flow between boxes - Hydrodynamics

The **flow** or **flux F** between in every point **in space** is the **change in density ρ** at every point **in time** (given we don't lose any crumbs)



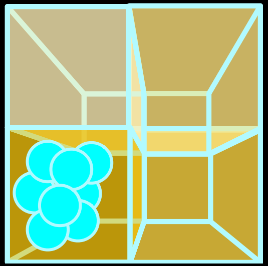
$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (v \rho) = -\nabla \cdot F$$

- Marie Kondo Principle ;) -

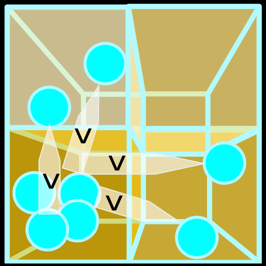
Let's start by putting everything in small boxes

Conservative physics

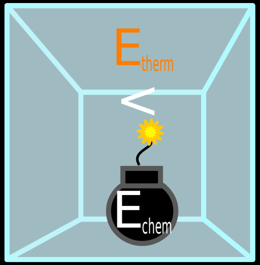
1) What's in the box stays in the box – Mass conservation:



$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (v\rho)$$

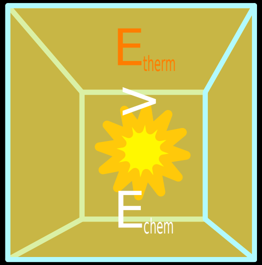


2) The energy in the box remains the same – Energy conservation:

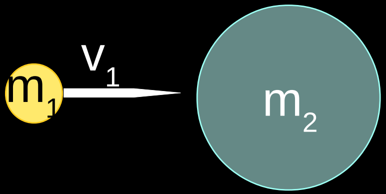


$$E_{pot} + E_{kin} + E_{chem} + E_{therm} + E_{eletric} + \ldots = U$$

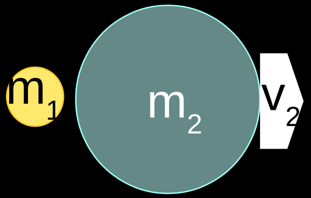
$$\frac{dU}{dt} = 0$$



3) Keep the momentum – Momentum conservation:



$$\frac{d(m_1 v_1)}{dt} = \frac{-d(m_2 v_2)}{dt}$$



Simple enough, right?

No, not really...

Navier-Stokes momentum equation

Conservation equation · Pressure · Viscosity (honey factor) · Compression (sqeezing) · Gravity

$$\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla \bar{p} + \mu \nabla^2 \boldsymbol{u} + \frac{1}{3}\mu \nabla (\nabla \cdot \boldsymbol{u}) + \rho \boldsymbol{g} + \ldots$$

Derivatives

Our poor computers can’t solve that…
So we need to get clever

Numerics of flow between boxes

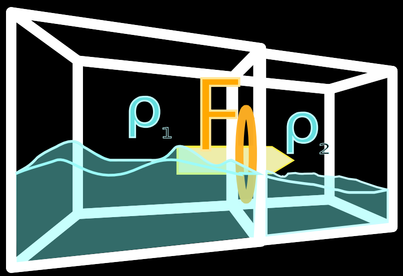
How to calculate that in a discrete way on a computer?

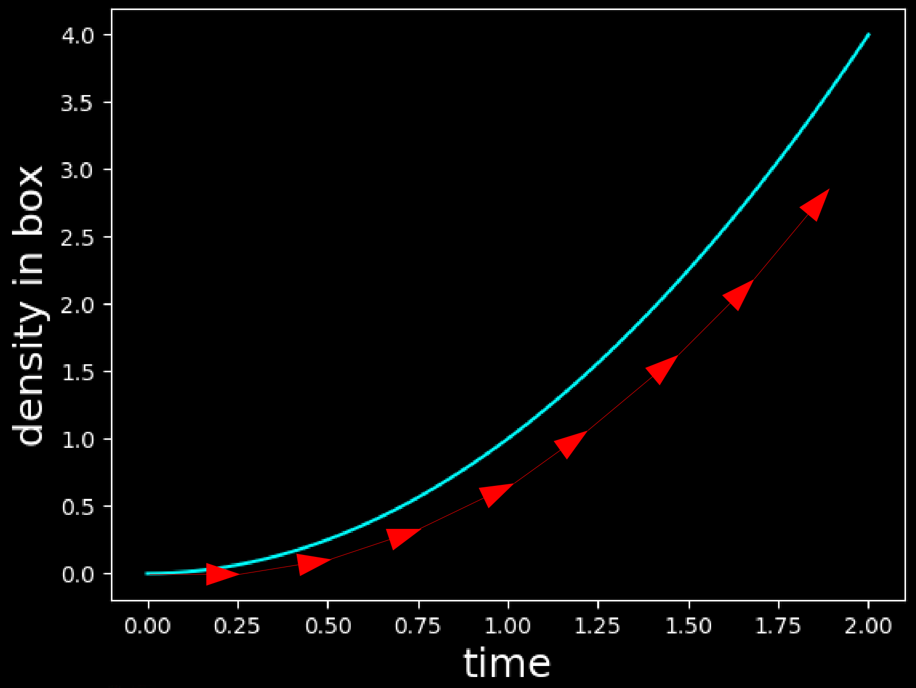
$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (v\rho) = -\nabla \cdot F$$

Simplest solution: Euler method

We add the current trend to the next **time step**.
This trend in the **flux F** coming
through the **surface S** of our box **volume V**.

$$\rho_{t+1} = \rho_t + \Delta t \cdot F_t \frac{S}{V}$$





BUT can we do better?

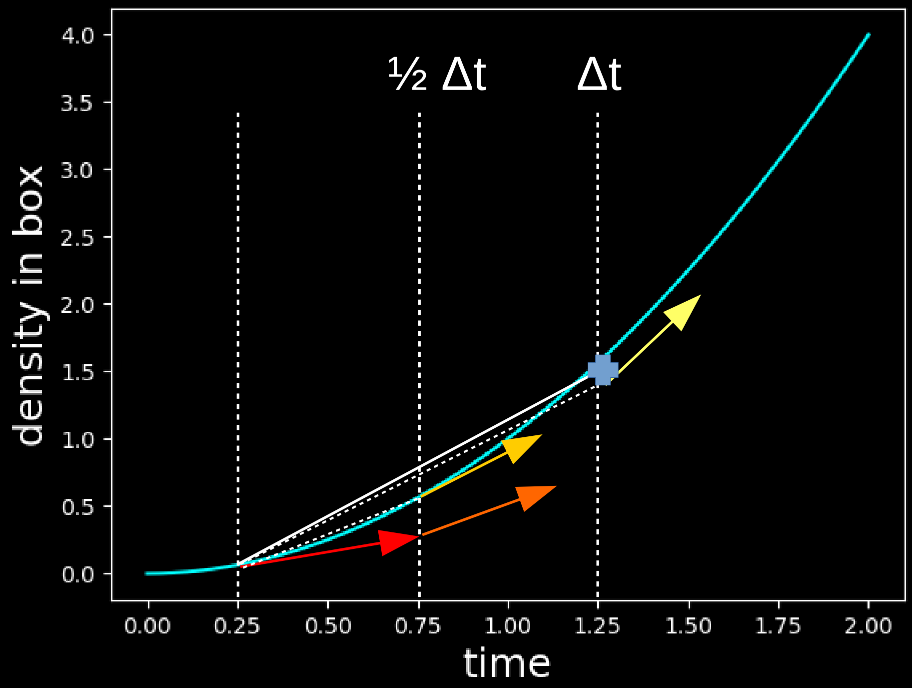
Numerics of flow between boxes

How to calculate that in a discrete way on a computer?

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (v\rho) = -\nabla \cdot F$$

Better solution: Runge Kutta

1) Find the slope to the next **half time step**.
2) Average the slope at **½Δt** and 1)
3) Average the slope at **½Δt** and 2)
4) Average the slope at **Δt** to the 3)



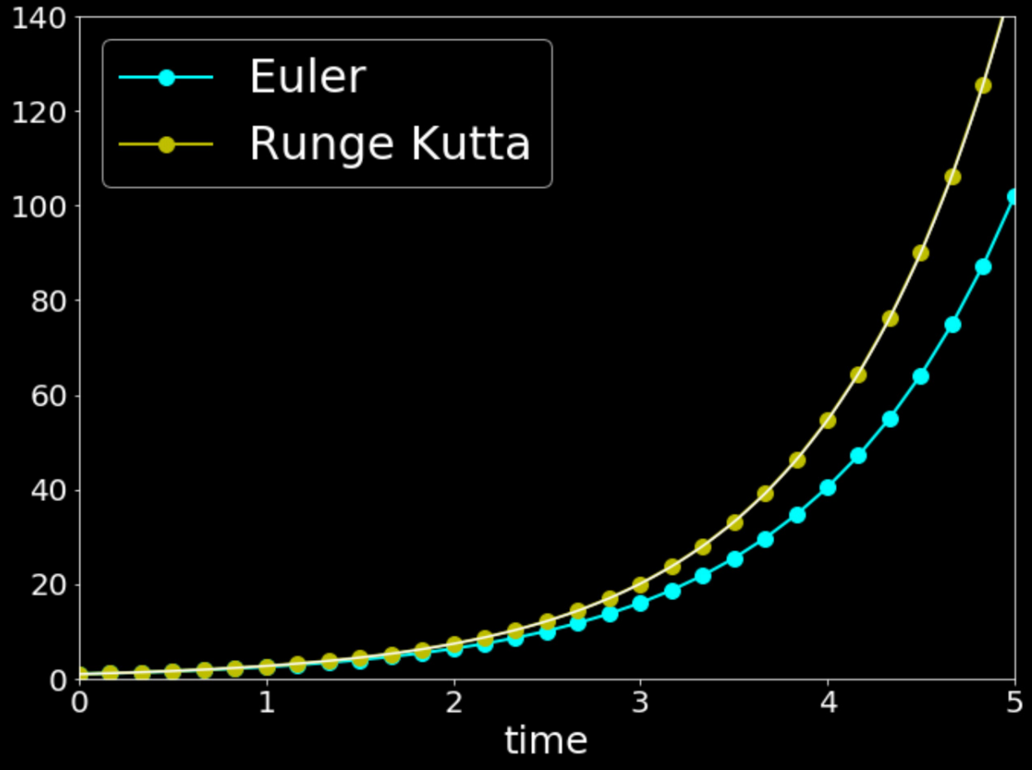
Numerics of flow between boxes

How to calculate that in a discrete way on a computer?

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (v\rho) = -\nabla \cdot F$$

Better solution: Runge Kutta

1) Find the slope to the next **half time step**.
2) Average the slope at **½Δt** and 1)
3) Average the slope at **½Δt** and 2)
4) Average the slope at **Δt** to the 3)



What’s needed for photos of dust

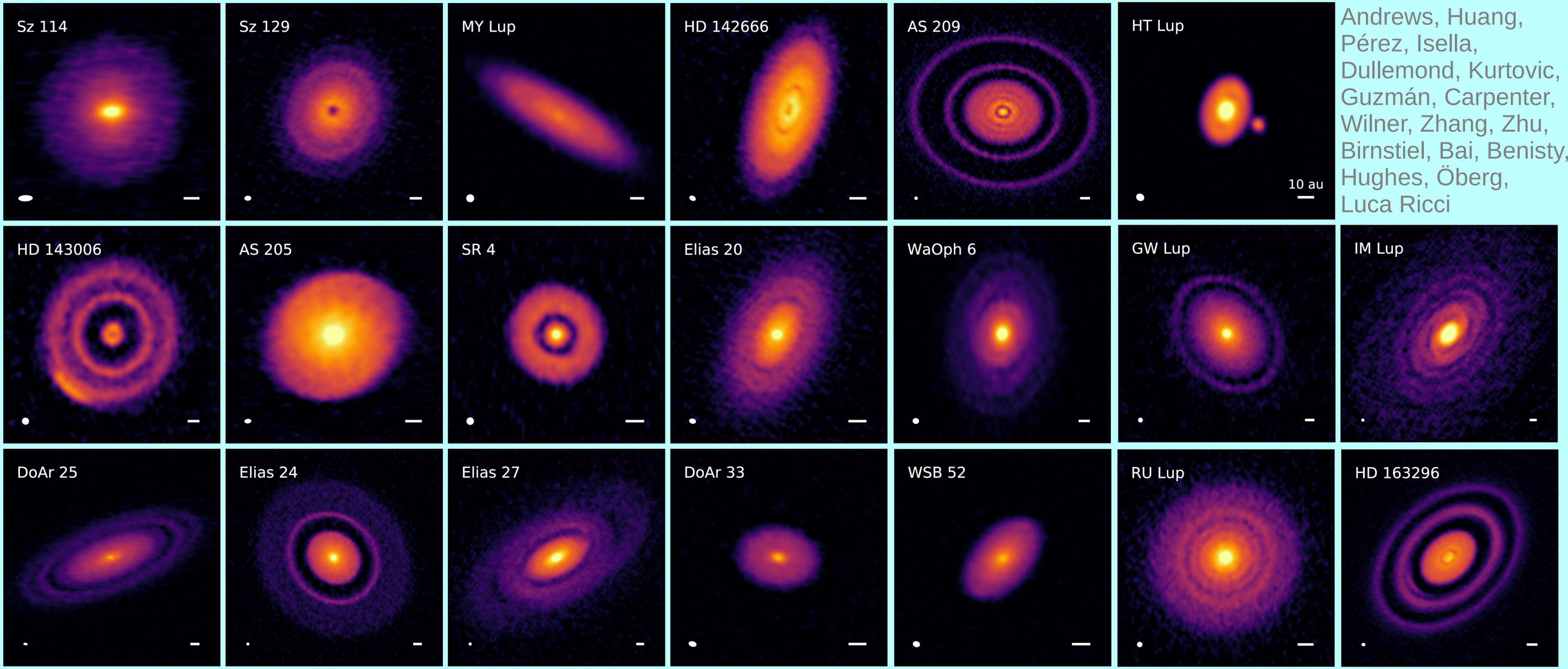
Atacama Large Millimeter Array (ALMA)

Credit: ESO/NAOJ/NRAO

Hydrodynamics in the wild

Disks around young stars – Images taken last year!

Credit:
DSHARP Project

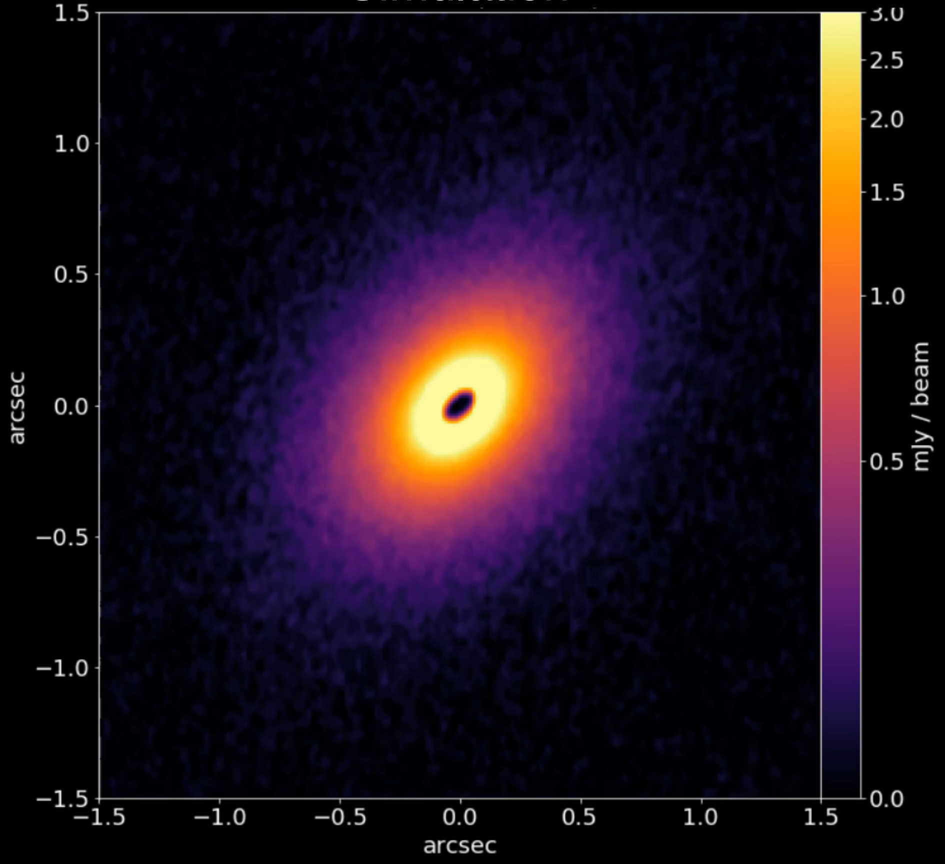


Andrews, Huang, Pérez, Isella, Dullemond, Kurtovic, Guzmán, Carpenter, Wilner, Zhang, Zhu, Birnstiel, Bai, Benisty, Hughes, Öberg, Luca Ricci

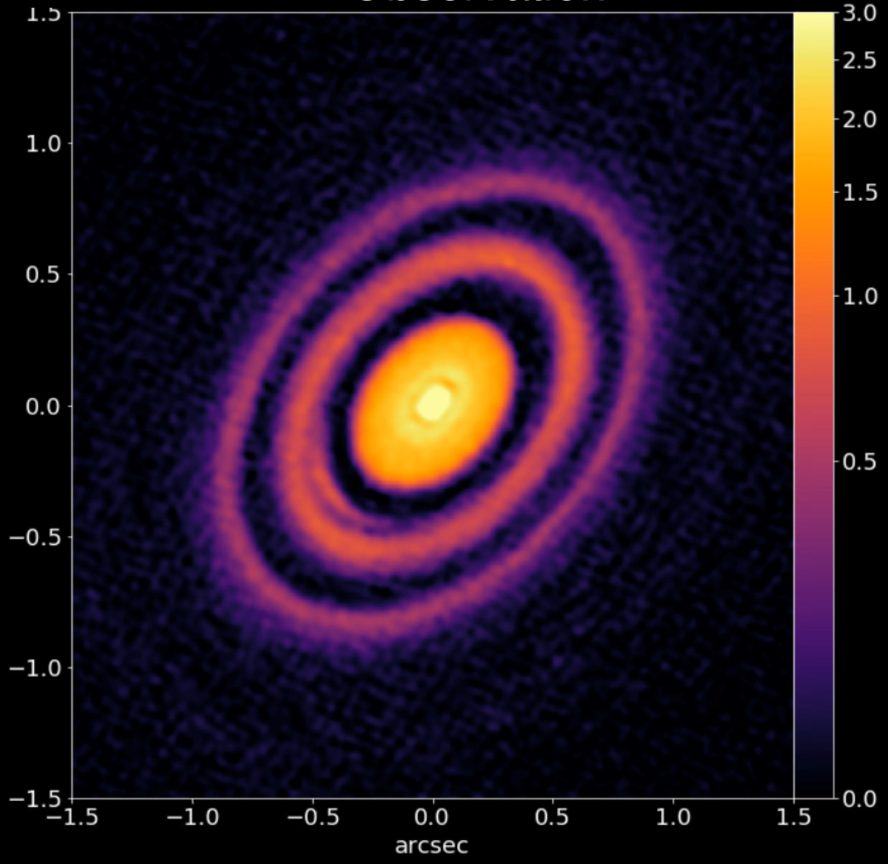
Hydrodynamics on a computer

Disks around young stars – Name: HD 163296

Simulation



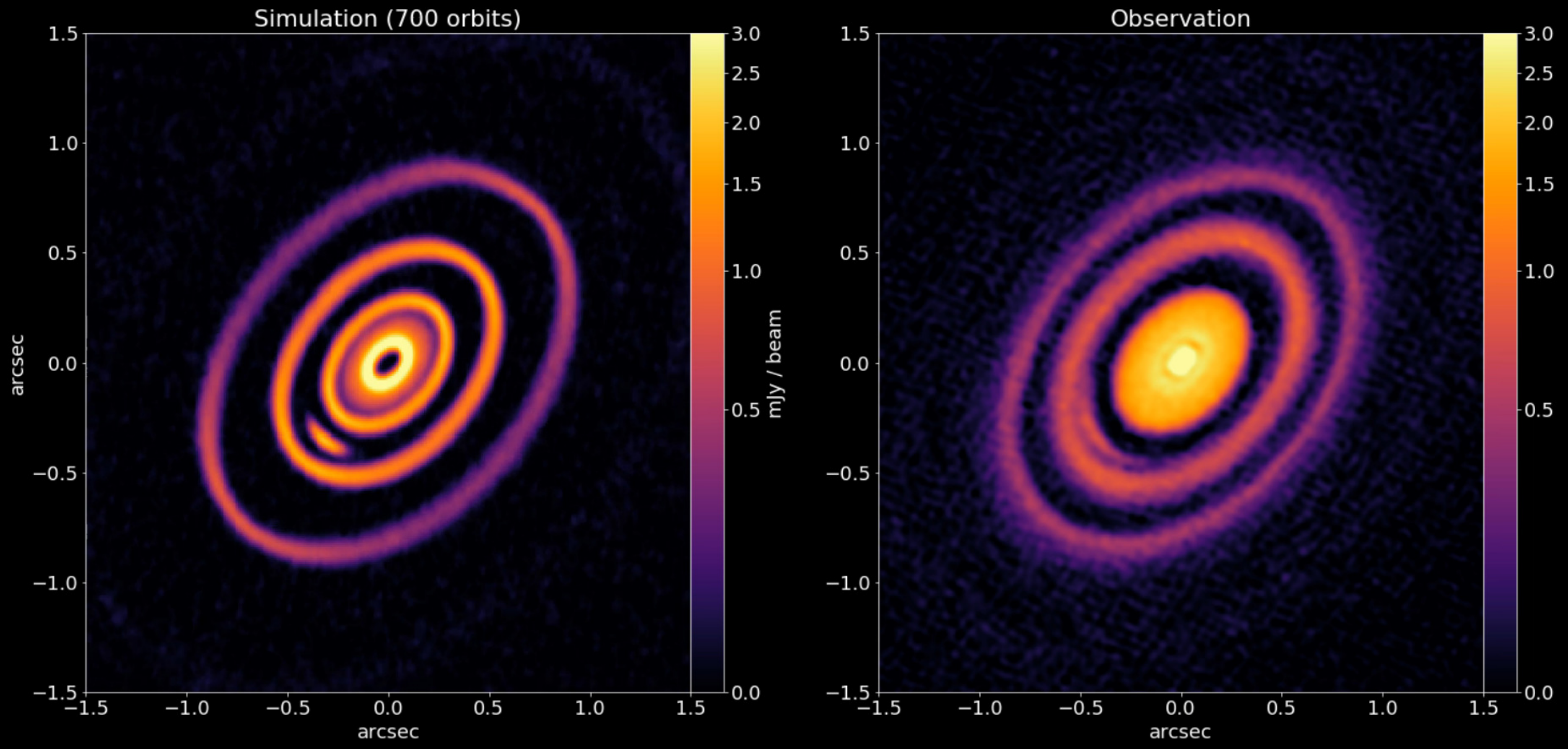
Observation



Credit: Peter Rodenkirch

Hydrodynamics on a computer

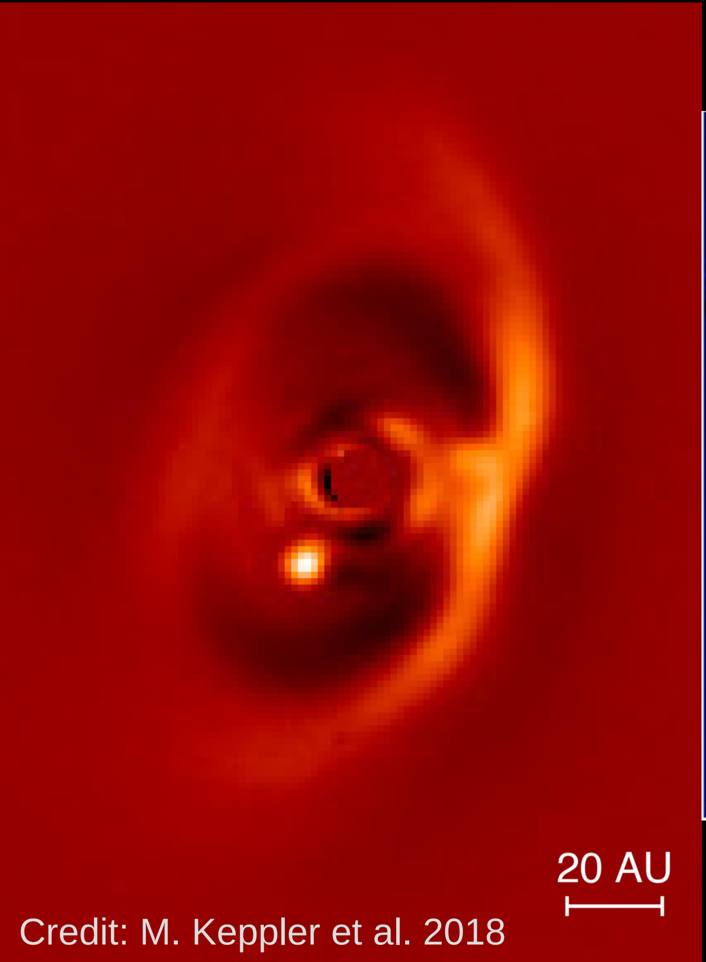
Disks around young stars



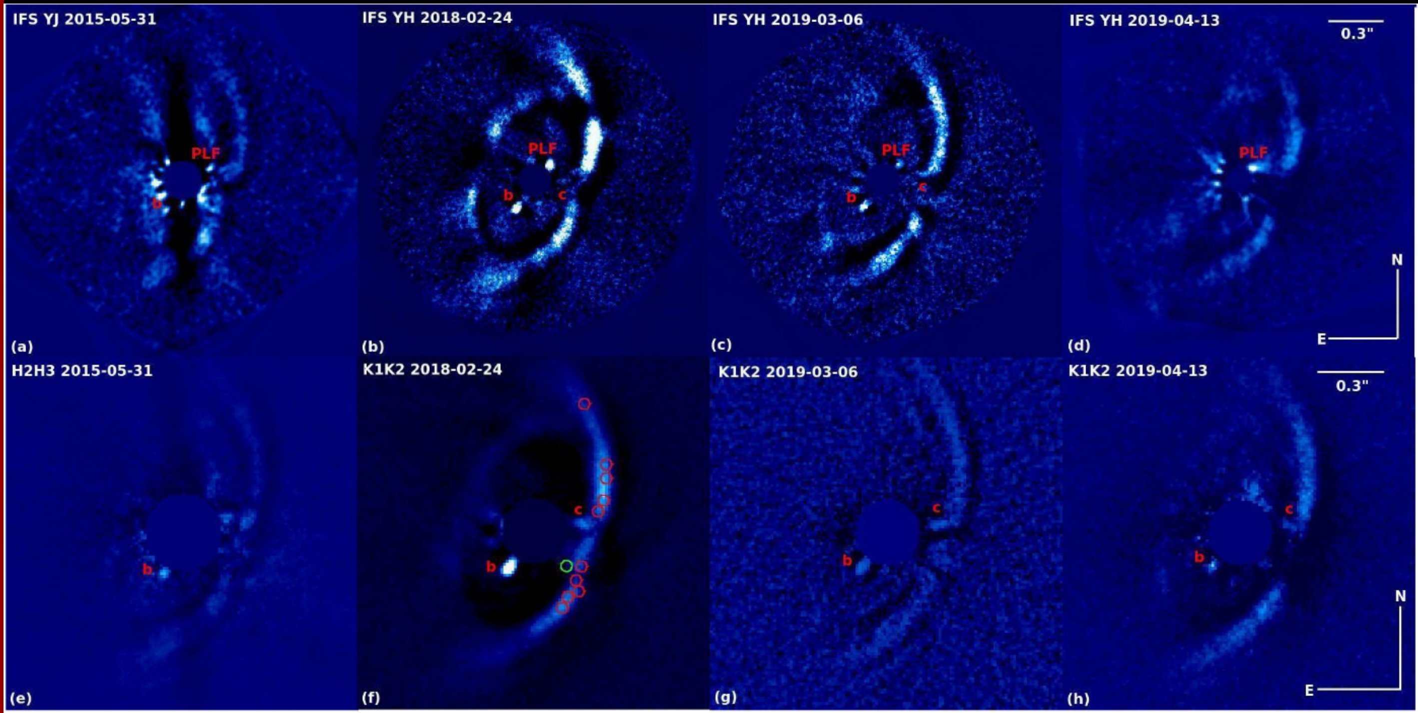
Credit: Peter Rodenkirch

Planets in the wild

Embedded planets in PDS70



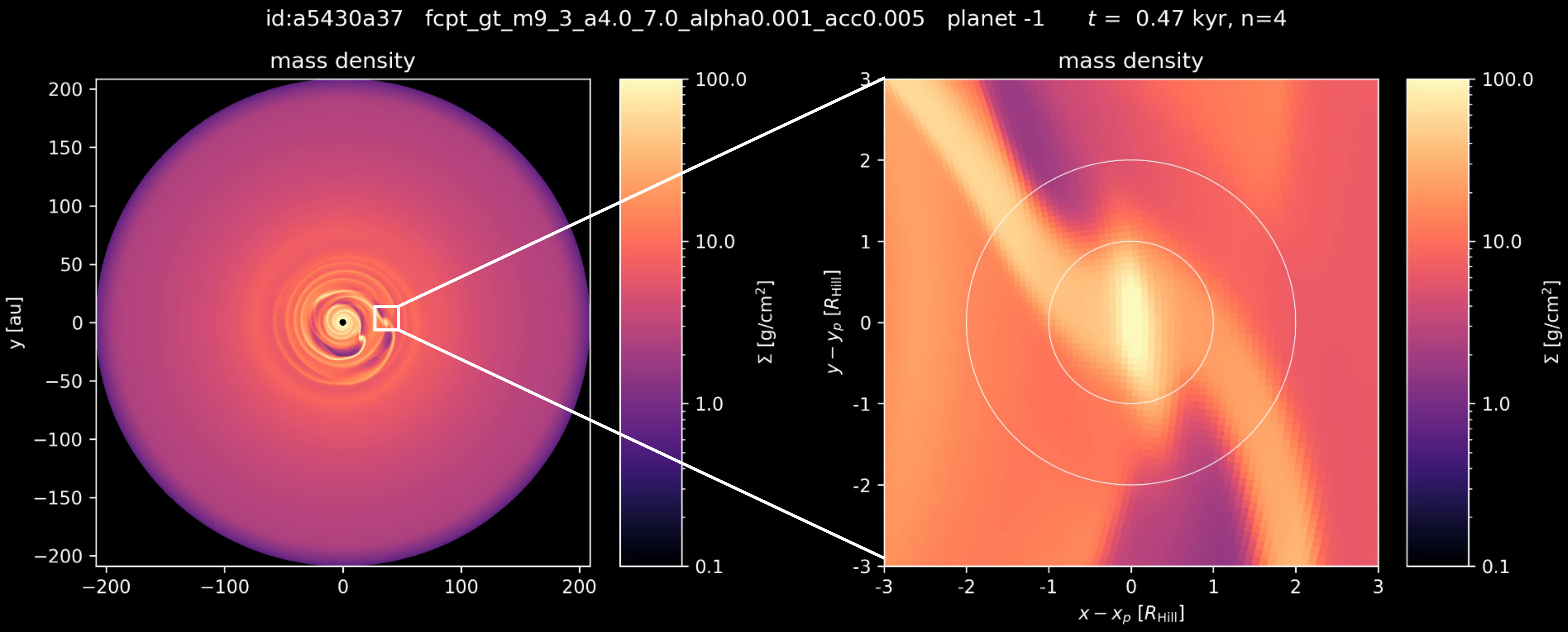
This is 370 light-years away from us!



Credit: D. Mesa et al. 2019

Planets on a computer

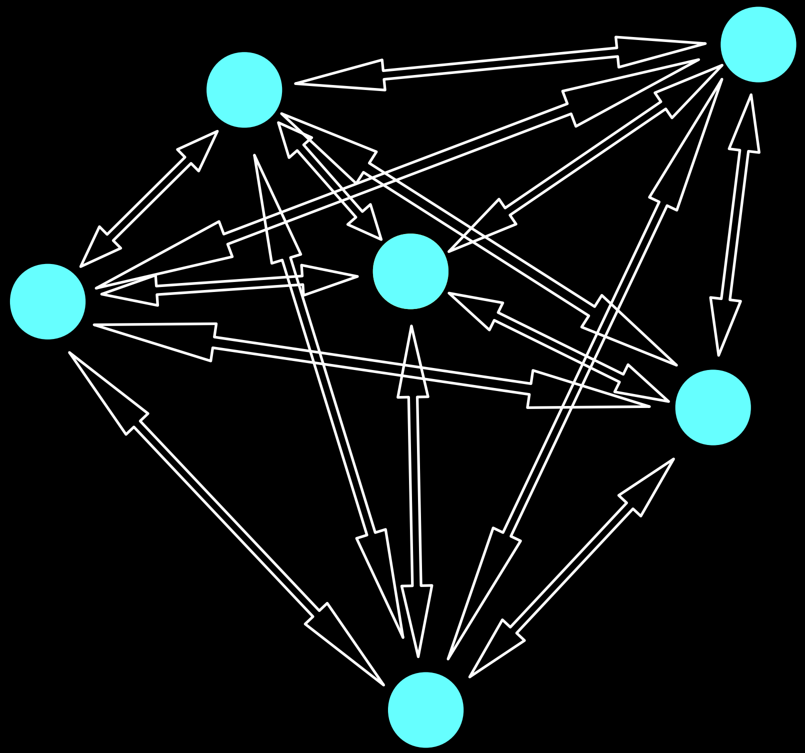
Embedded planets in PDS70 – with Fargo



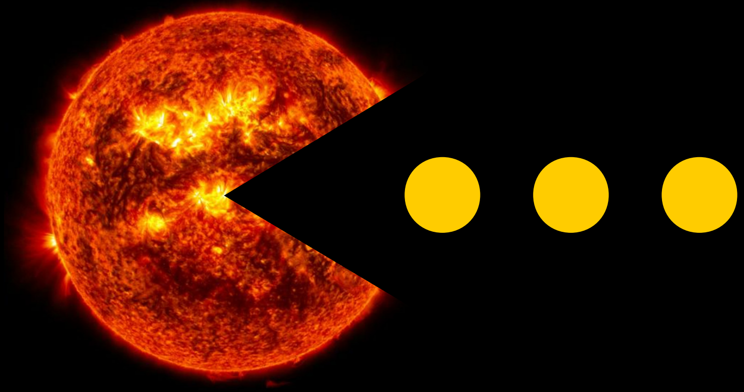
Credit: T. Rometsch

So there is more…

N-body interaction

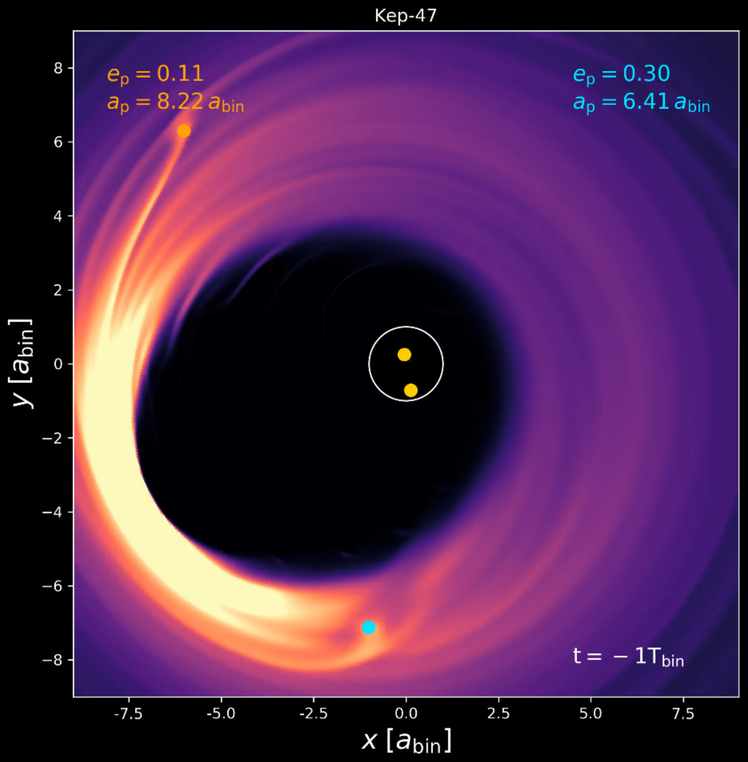


Accretion



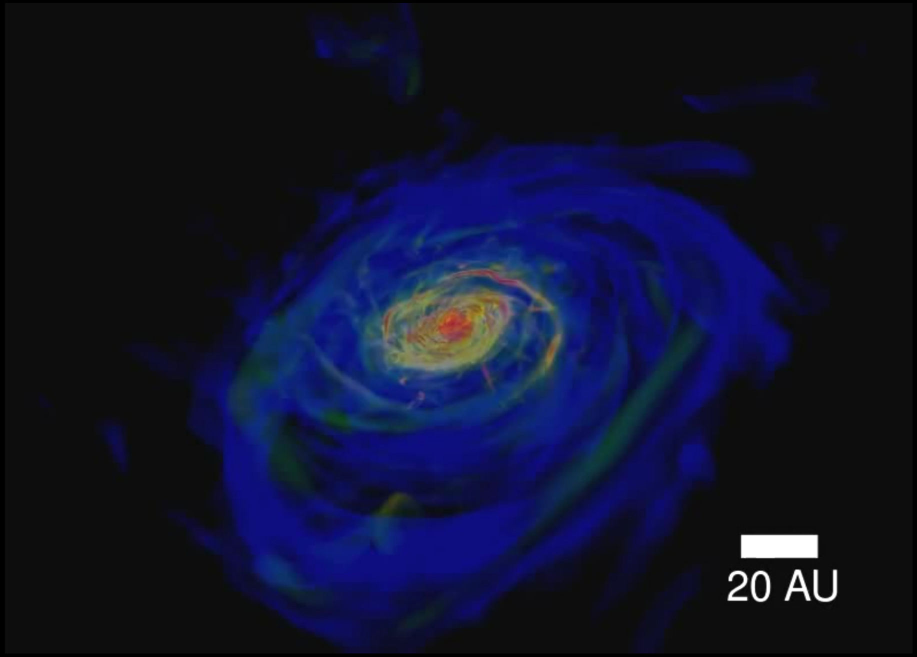
So there is more…

N-body interaction



Credit: A. Penzlin

Accretion

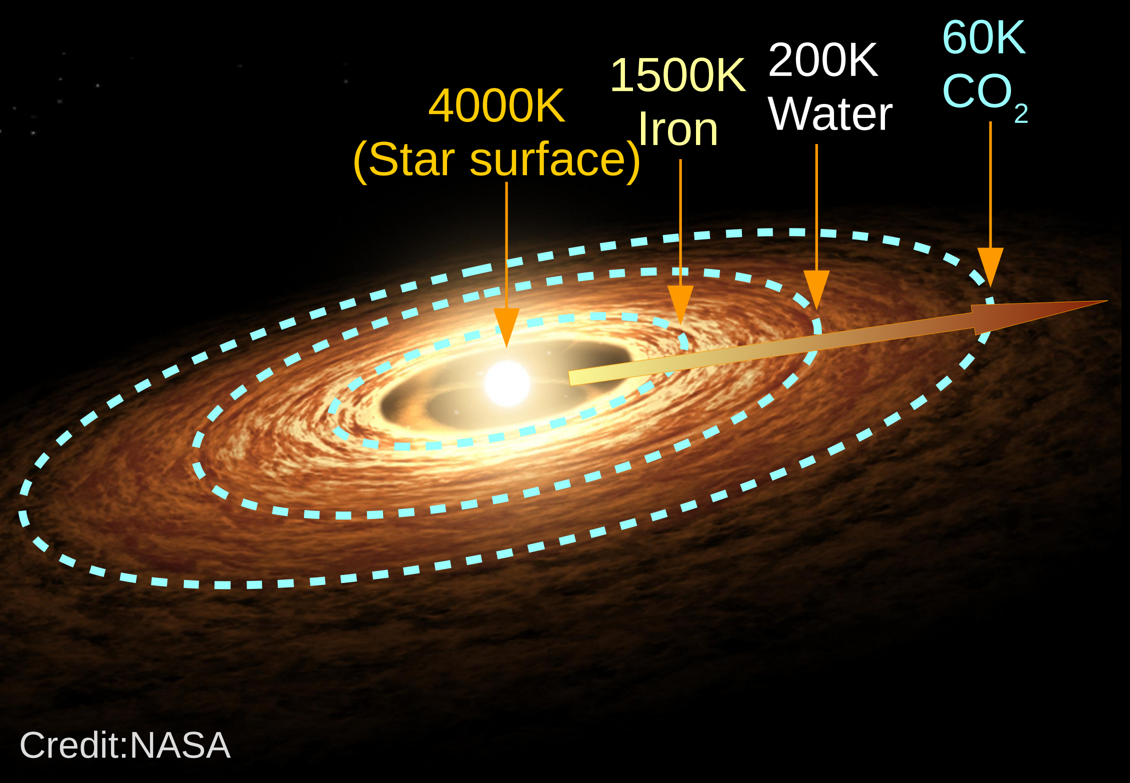


Credit: M. Küffmeier

Let's turn up the heat

Temperature

Condensation of different chem. compounds



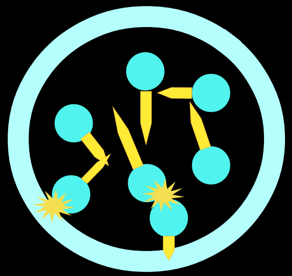
We only expect **water** on planets from beyond the **snow line.**

This is beyond 1 AU!
(1 AU = Earth-Sun distance)

Let’s turn up the heat

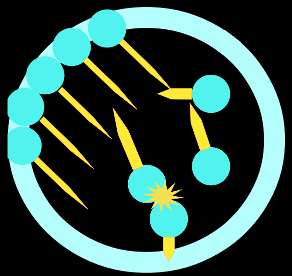
Pressure and shocks

Thermodynamics is like pogo, all about collision.
Gas **temperature** is determined the **speed of gas molecules**.

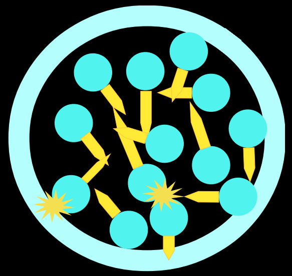


Two ways to heat up the pogo party:

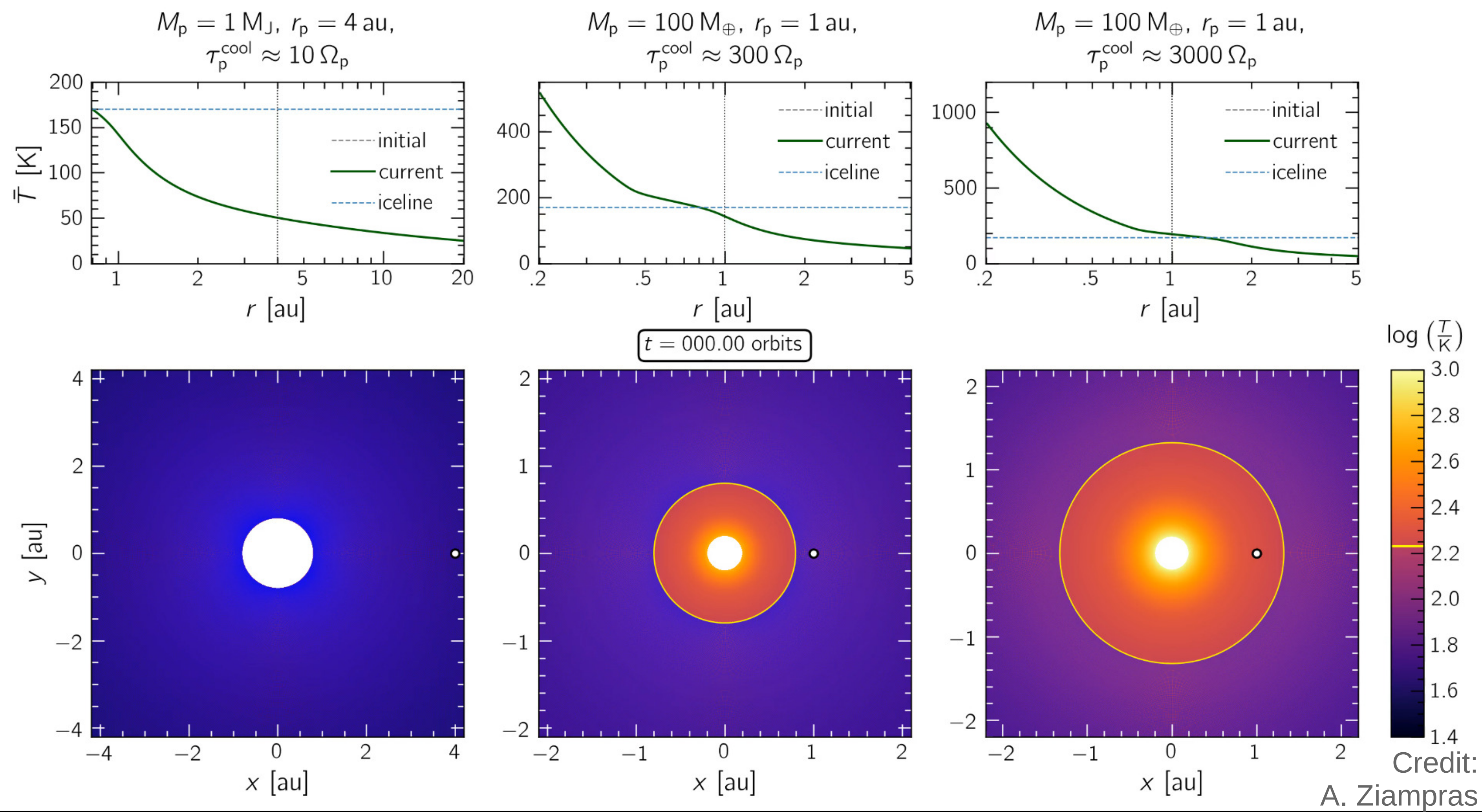
1) **Shock**: Get in a blast of velocity



2) **Pressure**: Get more people in less space!

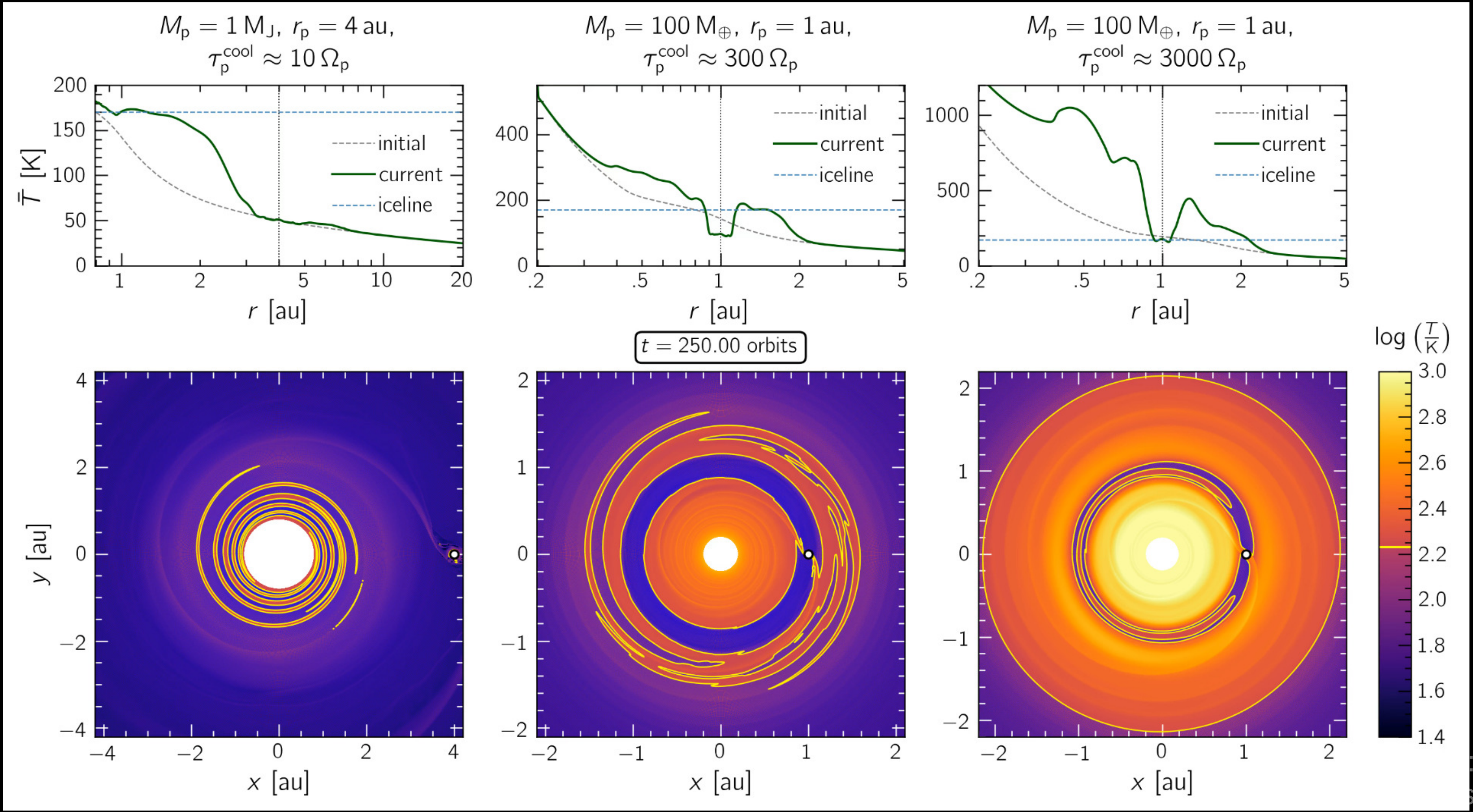


Let’s turn up the heat



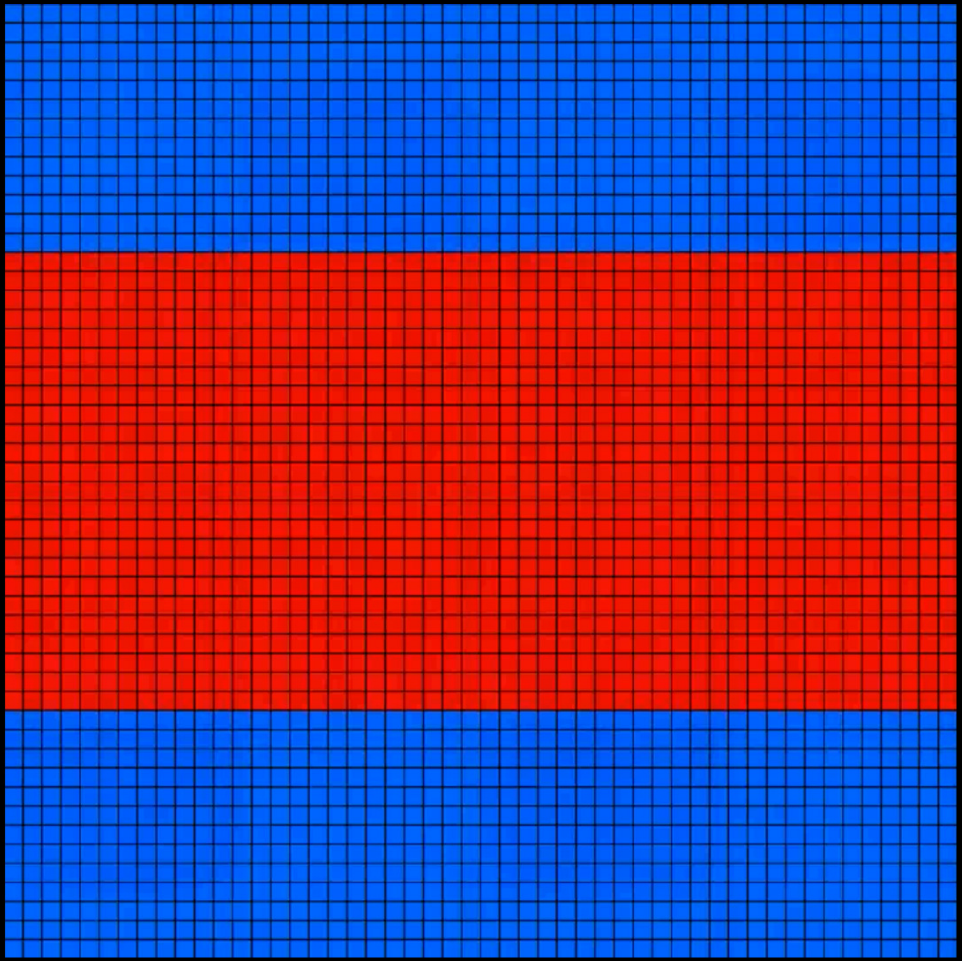
Credit:
A. Ziampras

Let’s turn up the heat



Think outside the box

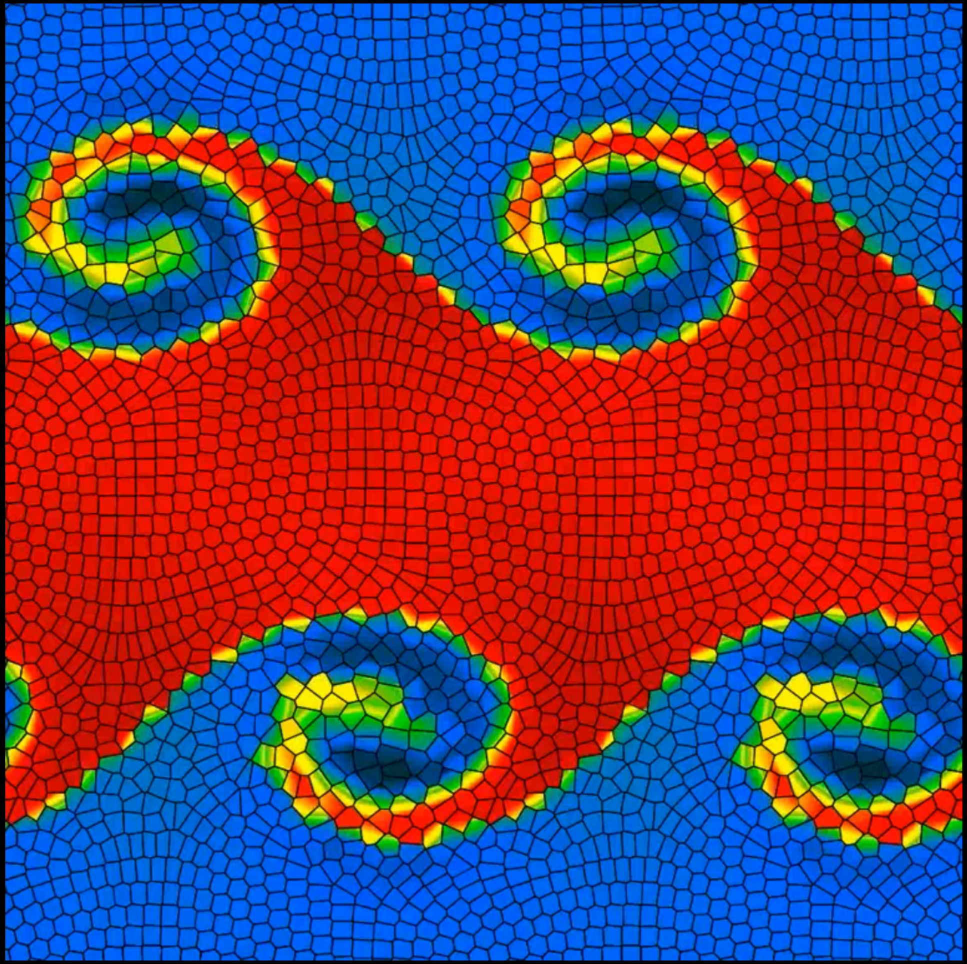
Adaptive Mesh refinement



Credit: Volker Springel, 2009

Think outside the box

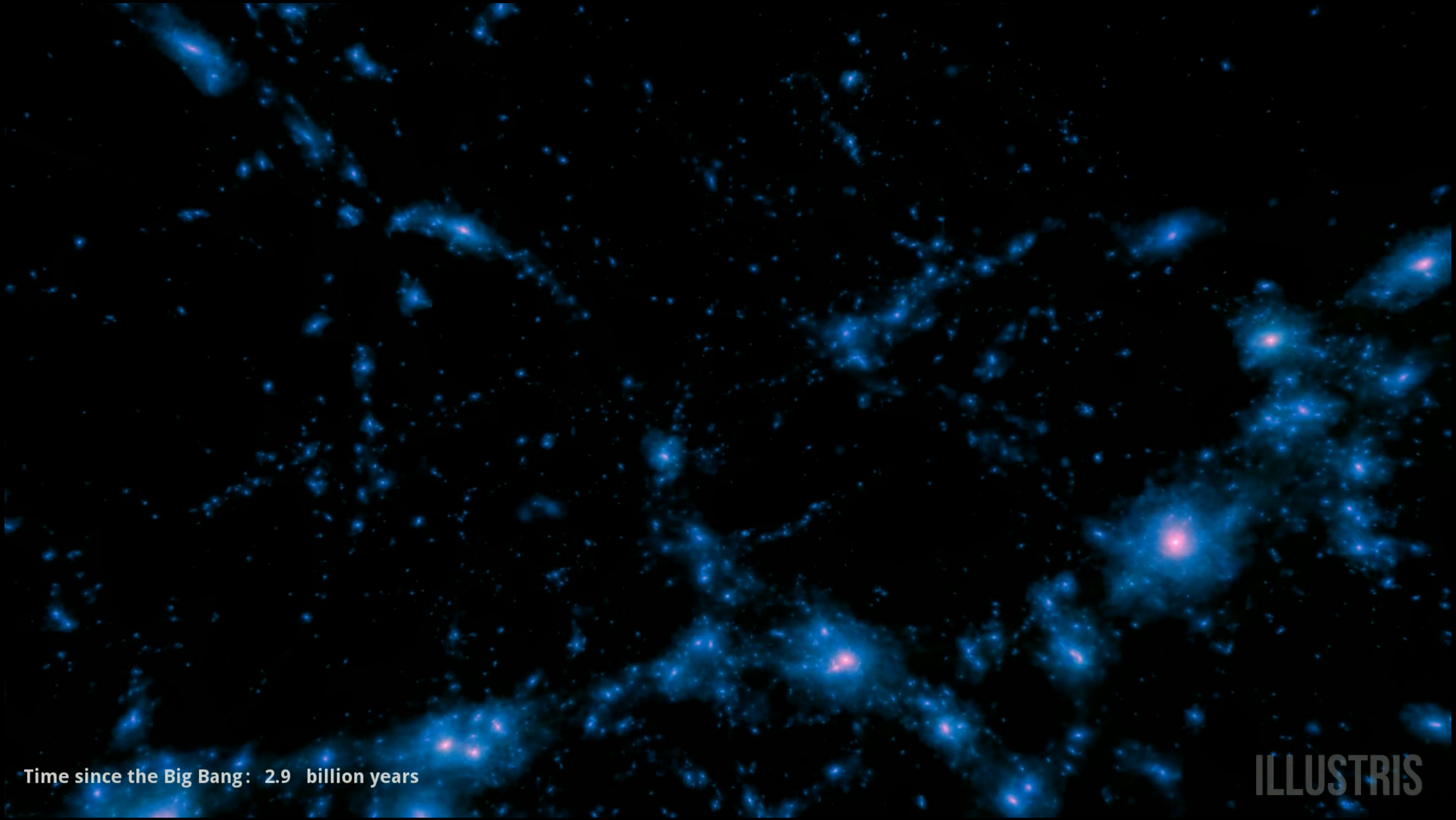
Adaptive Mesh refinement



Credit: Volker Springel, 2009

Think outside the box

Adaptive Mesh refinement



Credit:
Illustris
Simulation

Astro-hydro programms

	FARGO3D fargo.in2p3.fr	PLUTO plutocode.ph.unito.it	AREPO arepo-code.org
Computing:			
GPU	✔	✘ (exists, but not in open source version)	✘
Adaptive mesh	✘	✘	✔
Physics:			
Includes viscosity	✔	✔	✘
Optimal energy conservation	✘	✔	✔

Behind the scenes – FARGO3D

public/setups/fargo/fargo.par

```
Setup                   fargo

### Disk parameters

AspectRatio         0.05                Thickness over Radius in the disc
Sigma0              6.3661977237e-4     Surface Density at r=1
Nu                  1.0e-5              Uniform kinematic viscosity
SigmaSlope          0.0                 Slope for the surface density
FlaringIndex        0.0                 Slope for the aspect-ratio

# Radial range for damping (in period-ratios). Values smaller than one
# prevent damping.

DampingZone         1.15

# Characteristic time for damping, in units of the inverse local
# orbital frequency. Higher values means lower damping

TauDamp             0.3

### Planet parameters

PlanetConfig        planets/jupiter.cfg
ThicknessSmoothing  0.6
RocheSmoothing      0.0
Eccentricity        0.0
ExcludeHill         no
IndirectTerm        Yes

### Mesh parameters

Nx                  384                 Azimuthal number of zones
Ny                  128                 Radial number of zones
Xmin                -3.14159265358979323844
Xmax                3.14159265358979323844
Ymin                0.4                 Inner boundary radius
Ymax                2.5                 Outer boundary radius
OmegaFrame          1.0005              Angular velocity for the frame of reference (If Frame is F).
Frame               G                   cMethod for moving the frame of reference

### Output control parameters

DT                  0.314159265359      Physical time between fine-grain outputs
Ninterm             20                  Number of DTs between scalar fields outputs
Ntot                1000                Total number of DTs

OutputDir           @outputs/fargo

### Plotting parameters

PlotLog             yes
```

http://fargo.in2p3.fr/

```
$: make
$: ./fargo3d setups/fargo/fargo.par
```

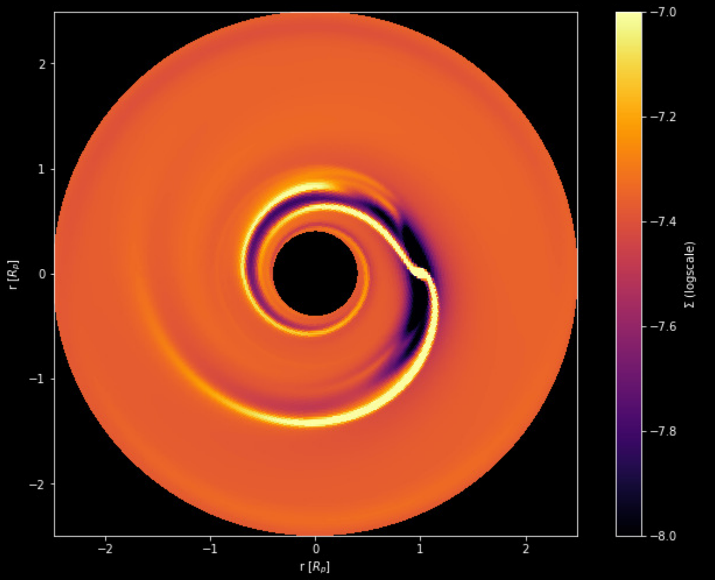
Credit: Benítez-Llambay & Masset, 2016

Behind the scenes – FARGO3D

```
        !!!! WARNING !!!!

This is a sequential built of the ./fargo3d code
If you planned to run the MPI-parallel version,
then you MUST rebuild the executable. In this case,
 issue:
make PARALLEL=1 (or make para)

Any subsequent invocation of make will build an MPI version.



===========================
FARGO3D git version 2.0-12-g92409890
SETUP: 'fargo'
===========================

The default output directory root is ./
Reading parameters file setups/fargo/fargo.par
The output directory is ./outputs/fargo/
Process 0 created the directory ./outputs/fargo/
I do not output the ghost values
Warning: The Y spacing is linear (default).
Field2D Reduction2D has been created
Field Moment_Plus_X has been created
Field Moment_Minus_X has been created
Field2D Vxhy has been created
Field2D Vxhyr has been created
Field2D Vxhz has been created
Field2D Vxhzr has been created
Field2D Eta_xi has been created
Field2D Eta_xizi has been created
Field2D Eta_zi has been created
Field2D Nshift has been created
Field2D Nxhy has been created
Field2D Nxhz has been created
Field Moment_Plus_Y has been created
Field Moment_Minus_Y has been created
Field potential has been created
Field Slope has been created
Field DivRho has been created
Field DensStar has been created
Field Qs has been created
Field Pressure has been created
Field Total_Density has been created
Grids Pressure and QLeft share their storage
Field QRight has been created
1 planet found.
Planet number 0
---------------
```

```
x = 1.0000000000        y = 0.0000000000        z = 0.0000000000
vx = -0.0000000000      vy = 1.0004998751       vz = 0.0000000000
Non-accreting.
Doesn't feel the disk potential
Doesn't feel the other planets potential

Field gasdens has been created
Field gasenergy has been created
Field2D VxMed has been created
Field gasvx has been created
Field Vx_temp has been created
Field gasvy has been created
Field Vy_temp has been created
Field2D rho0 has been created
Field2D e0 has been created
Field2D vx0 has been created
Field2D vy0 has been created
Field2D vz0 has been created
Found 0 communicators
Standard version with no ghost zones in X
OUTPUTS 0 at date t = 0.000000 OK
............
Process 0 created the directory ./outputs/fargo/monitor/gas
.............
..............
................
................
..................
................
................
...............
...............
..............
..............
.............
.............
.............
.............
.............
.............
.............
OUTPUTS 1 at date t = 6.283185 OK
.............
.............
.............
.............
.............
.............
.............
```

http://fargo.in2p3.fr/



Credit: Benítez-Llambay & Masset, 2016

Behind the scenes – PLUTO

Setup a path to PLUTO_DIR
Create your folder eg. “Test_disc”
Add pluto.ini + definitions.h + init.c

Pluto.ini

```
[Grid]

X1-grid    1    0.4    256    u    2.5
X2-grid    1    0.0    768    u    6.283185307
X3-grid    1   -1.0      1    u    1.0

[Chombo Refinement]

Levels           4
Ref_ratio        2 2 2 2 2
Regrid_interval  2 2 2 2
Refine_thresh    0.3
Tag_buffer_size  3
Block_factor     8
Max_grid_size    64
Fill_ratio       0.75

[Time]

CFL              0.4
CFL_max_var      1.1
tstop            2.0
first_dt         1.e-4

[Solver]

Solver         roe

[Boundary]

X1-beg        userdef
X1-end        userdef
X2-beg        periodic
X2-end        periodic
X3-beg        periodic
X3-end        periodic

[Static Grid Output]

uservar    0
dbl        10.0  -200   single_file
flt         1.0  -1   single_file
vtk        -1.0  -1   single_file
tab        -1.0  -1
ppm        -1.0  -1
png        -1.0  -1
log        10
analysis  -1.0  100

[Chombo HDF5 output]

Checkpoint_interval  -1.0  0
Plot_interval         1.0  0

[Parameters]

Mstar              1.0
Mdisk              0.01
Mplanet            320.0
Viscosity          1.e15
```

Definitions.h

```
#define  PHYSICS                 HD
#define  DIMENSIONS              2
#define  COMPONENTS              2
#define  GEOMETRY                POLAR
#define  BODY_FORCE              POTENTIAL
#define  COOLING                 NO
#define  RECONSTRUCTION          LINEAR
#define  TIME_STEPPING           RK2
#define  DIMENSIONAL_SPLITTING   NO
#define  NTRACER                 0
#define  USER_DEF_PARAMETERS     4

/* -- physics dependent declarations -- */

#define  EOS                     ISOTHERMAL
#define  ENTROPY_SWITCH          NO
#define  THERMAL_CONDUCTION      NO
#define  VISCOSITY               NO
#define  ROTATING_FRAME          YES

/* -- user-defined parameters (labels) -- */

#define  Mstar                   0
#define  Mdisk                   1
#define  Mplanet                 2
#define  Viscosity               3

/* [Beg] user-defined constants (do not change this line) */

#define  UNIT_LENGTH             (5.2*CONST_au)
#define  UNIT_DENSITY            (CONST_Msun/(UNIT_LENGTH*UNIT_LENGTH*UNIT_LENGTH))
#define  UNIT_VELOCITY           (sqrt(CONST_G*g_inputParam[Mstar]*CONST_Msun/UNIT_LENGTH)/(2.*CONST_PI))

/* [End] user-defined constants (do not change this line) */

/* -- supplementary constants (user editable) -- */

#define  INITIAL_SMOOTHING   NO
#define  WARNING_MESSAGES    YES
#define  PRINT_TO_FILE       YES
#define  INTERNAL_BOUNDARY   NO
#define  SHOCK_FLATTENING    NO
#define  CHAR_LIMITING       NO
#define  LIMITER             VANLEER_LIM
```

init.c

```
/* ********************************************************************* */
void Init (double *us, double x1, double x2, double x3)
/*
 *
 *
 *
 ********************************************************************* */
{
  double r, th, R, z, H, OmegaK, cs;
  double scrh;

  #if EOS == IDEAL
   g_gamma = 1.01;
  #endif

  #if ROTATING_FRAME == YES
   g_OmegaZ  = sqrt(1.0 + g_inputParam[Mplanet]/g_inputParam[Mstar]*CONST_Mearth/CONST_Msun);
   g_OmegaZ *= 2.0*CONST_PI;
  #endif

  #if GEOMETRY == POLAR
   R  = x1;
   #if DIMENSIONS == 2
    z  = 0.0;
    r  = R;
    th = 0.5*CONST_PI;
   #else...

  #endif
}
/* ********************************************************************* */
void UserDefBoundary (const Data *d, RBox *box, int side, Grid *grid)
/*
 *
 ********************************************************************* */
{
  int   i, j, k, nv;
  double *x1, *x2, *x3, R, OmegaK, v[256];
  static int do_once = 1;

  x1 = grid[IDIR].x;
  x2 = grid[JDIR].x;
  x3 = grid[KDIR].x;

  if (side == X1_BEG){
    X1_BEG_LOOP(k,j,i){
      NVAR_LOOP(nv) d->Vc[nv][k][j][i] = d->Vc[nv][k][j][2*IBEG - i - 1];
      d->Vc[VX1][k][j][i] *= -1.0;
      #if GEOMETRY == POLAR
       R = x1[i];
      OmegaK = 2.0*CONST_PI/(R*sqrt(R));
      d->Vc[iVPHI][k][j][i] = R*(OmegaK - g_OmegaZ);
    }
  }

  if (side == X1_END){…
  }
```

Behind the scenes – PLUTO

Setup a path to PLUTO_DIR
Create your folder eg. “Test_disc”
Add pluto.ini + definitions.h + init.c

Pluto.ini

Defines:

- Grid
- Initial values
- Simulation time
- Parameter values
- Output

```
[Grid]

X1-grid    1    0.4    256    u    2.5

[Chombo Refinement]

Levels          4
Refine_thresh   0.3
Max_grid_size   64

[Time]

CFL
CFL_max_var     1.1

[Solver]

Solver      roe

[Boundary]

X1-beg      userdef
X1-end      userdef
X2-beg      periodic
X2-end      periodic
X3-beg      periodic
X3-end      periodic

[Static Grid Output]

dbl        10.0  -200   single_file
tab        -1.0  -1
ppm        -1.0  -1
png        -1.0  -1
log        10
analysis  -1.0  100

[Chombo HDF5 output]

Checkpoint_interval  -1.0  0
Plot_interval         1.0  0

[Parameters]

Mstar          1.0
Mdisk          0.01
Mplanet        320.0
Viscosity      1.e15
```

Definitions.h

Defines:

- Physics
- Coordinate system
- Time step integrator
- Simulation units
- User-defined parameters

```
#define  PHYSICS                 HD
#define  DIMENSIONS              2
#define  COMPONENTS              2
#define  COOLING                 NO
#define  RECONSTRUCTION          LINEAR
#define  TIME_STEPPING           RK2
#define  DIMENSIONAL_SPLITTING   NO

#define  EOS                     ISOTHERMAL
#define  VISCOSITY               NO

/* -- user-defined parameters (labels) -- */

#define  Mdisk                   1
#define  Mplanet                 2
#define  Viscosity               3

/* [Beg] user-defined constants (do not change this line) */

#define  UNIT_LENGTH     (5.2*CONST_au)
#define  UNIT_DENSITY    (CONST_Msun/(UNIT_LENGTH*UNIT_LENGTH*UNIT_LENGTH))
#define  UNIT_VELOCITY   (sqrt(CONST_G*g_inputParam[Mstar]*CONST_Msun/UNIT_LENGTH)/(2.*CONST_PI))

/* [End] user-defined constants (do not change this line) */

/* -- supplementary constants (user editable) -- */

#define  INITIAL_SMOOTHING   NO
#define  WARNING_MESSAGES    YES
#define  PRINT_TO_FILE       YES
#define  INTERNAL_BOUNDARY   NO
#define  SHOCK_FLATTENING    NO
#define  CHAR_LIMITING       NO
#define  LIMITER             VANLEER_LIM
```

init.c

Defines:

- How is the disc initalized
- What happens at the edge
- How does gravity work
- What will be analysed
- ...

```
/* ********************************************************************* */
void Init (double *us, double x1, double x2, double x3)
{
  double r, th, R, z, H, OmegaK, cs;
  double scrh;

  #if EOS == IDEAL
   g_gamma = 1.01;
  #endif

   g_OmegaZ  = sqrt(1.0 + g_inputParam[Mplanet]/g_inputParam[Mstar]*CONST_Mearth/CONST_Msun);
   g_OmegaZ *= 2.0*CONST_PI;
  #endif

   R  = x1;
   #if DIMENSIONS == 2
    th = 0.5*CONST_PI;
   #else...
  #endif
}
/* ********************************************************************* */
void UserDefBoundary (const Data *d, RBox *box, int side, Grid *grid)
 ********************************************************************* */
{
  int   i, j, k, nv;
  double *x1, *x2, *x3, R, OmegaK, v[256];
  static int do_once = 1;

  x1 = grid[IDIR].x;
  x2 = grid[JDIR].x;
  x3 = grid[KDIR].x;

  if (side == X1_BEG){
    X1_BEG_LOOP(k,j,i){
      NVAR_LOOP(nv) d->Vc[nv][k][j][i] = d->Vc[nv][k][j][2*IBEG - i - 1];
      d->Vc[VX1][k][j][i] *= -1.0;
      #if GEOMETRY == POLAR
       R = x1[i];
      OmegaK = 2.0*CONST_PI/(R*sqrt(R));
      d->Vc[iVPHI][k][j][i] = R*(OmegaK - g_OmegaZ);
    }
  }

  if (side == X1_END){…
  }
```

Get the right makefile for your setup, make it and run it!

Behind the scenes – PLUTO

$ ./pluto

```
   ____  __    __  ____________
  / _ \/ /   / / / /_  __/ __ \
 / ___/ /__/ /_/ / / / / /_/ /
/_/  /_____/\____/ /_/  \____/
=============================    v. 4.2

> System:

! sysconf.out file not found

> Local time:         Mon Dec 23 10:48:10 2019

> Cmd line args:

> Header configuration:

  PHYSICS:            HD
  DIMENSIONS:         2
  COMPONENTS:         2
  GEOMETRY:           Polar
  BODY_FORCE:         NO
  RECONSTRUCTION:     Linear TVD (Primitive lim)
  TRACERS:            0
```

Computers

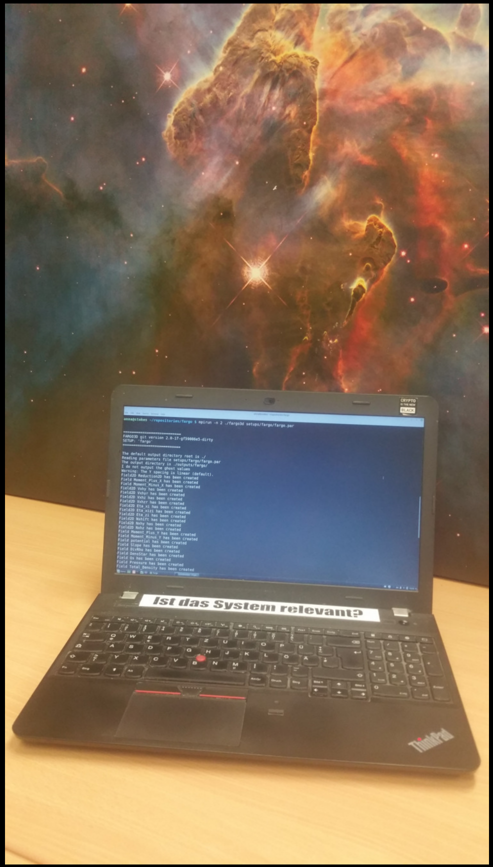
Please calculate, my dear laptop!

So let's run ./fargo3d!

....

....

...just need to wait
another 10 years now



Credit: stokes

Computers

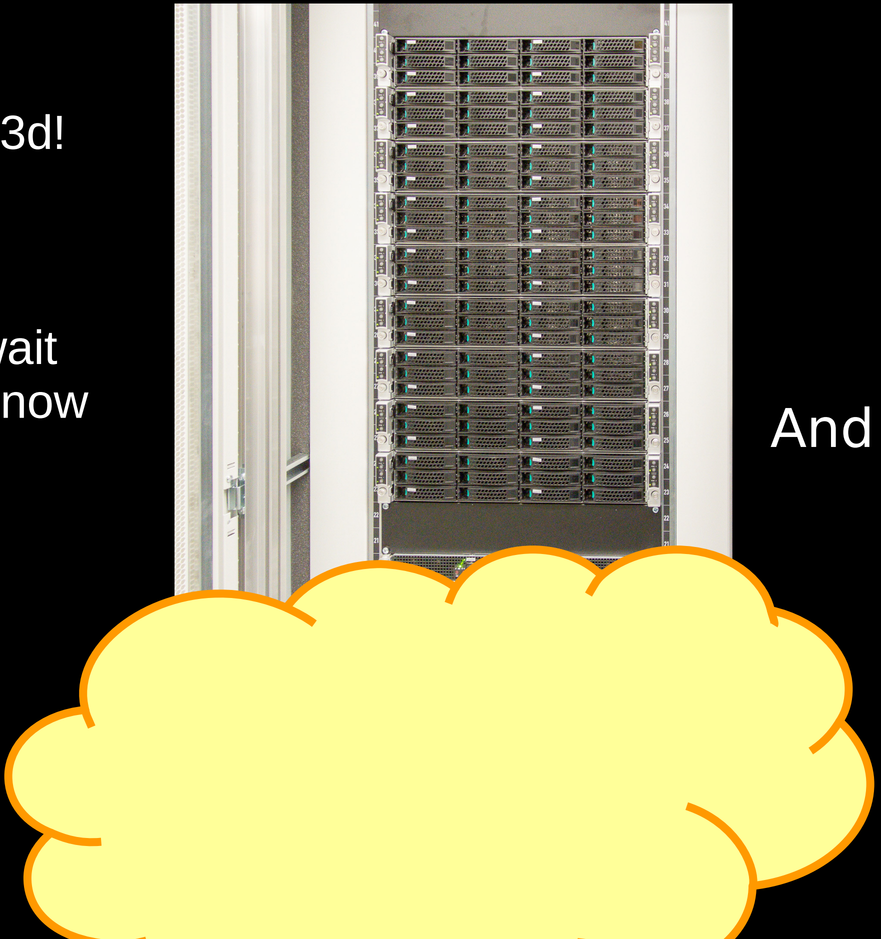
BinAC – Bioinformatics Astrophysics Cluster

So let's run ./fargo3d!

……..

……..

...just need to wait
another 10 years now



BEHOLD!
MORE POWER!

And we are down to a day

Credit:
bwForCluster;
Tübingen University

Computers

BinAC – Bioinformatics Astrophysics Cluster

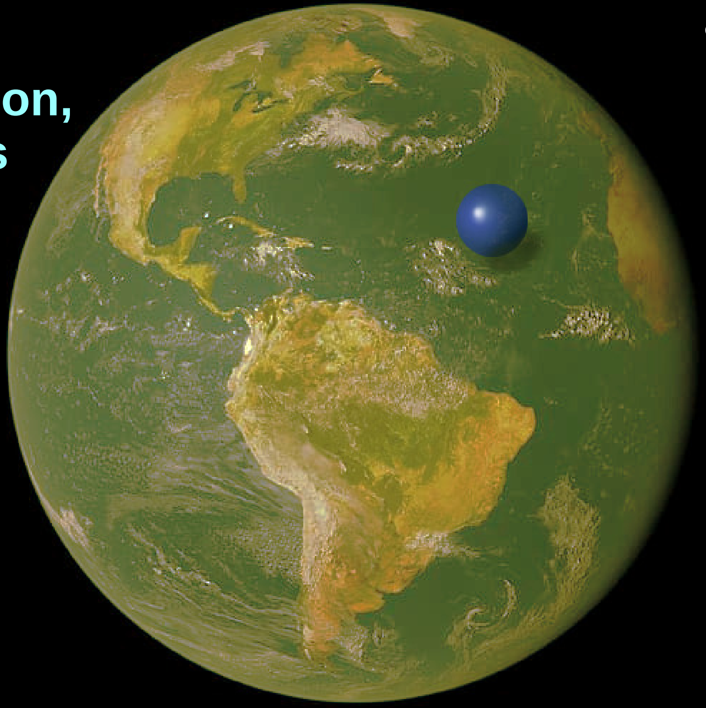
compute nodes:
236 CPU
62 GPU

Credit:
bwForCluster;
Tübingen University

Why our Earth is special

- Nice temperature (allowing fuilds on the surface)
- Enough gravity to keep an atmosphere but not to crush us
- Magnetic field to shield of solar winds
- Just enough water to have continents

Even with a fine tuned simulation, the change of getting a earth is tiny!



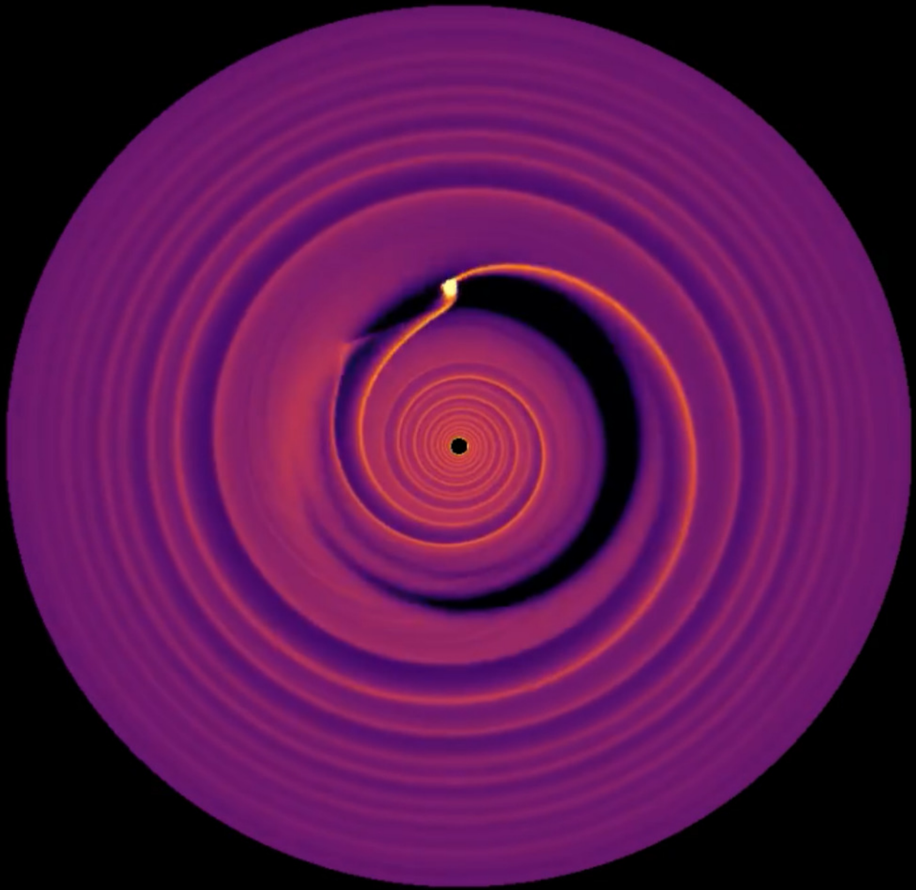
Open questions:

- Why do we have water ?
- Why not a water world ?
- Why did Earth not migrate closer to the sun?
- How could earth accrete enough mass?
- With all 8 planets how did Earth find a stable orbit?

Even in our simulation there is no Planet B!

So please, take care!

It takes a village to grow a planet



Simulation software

FARGO3d: Pablo Benítez Llambay, Frédéric Masset

http://fargo.in2p3.fr/

PLUTO: Andrea Mignone+

http://plutocode.ph.unito.it/

AREPO: Volker Springel+

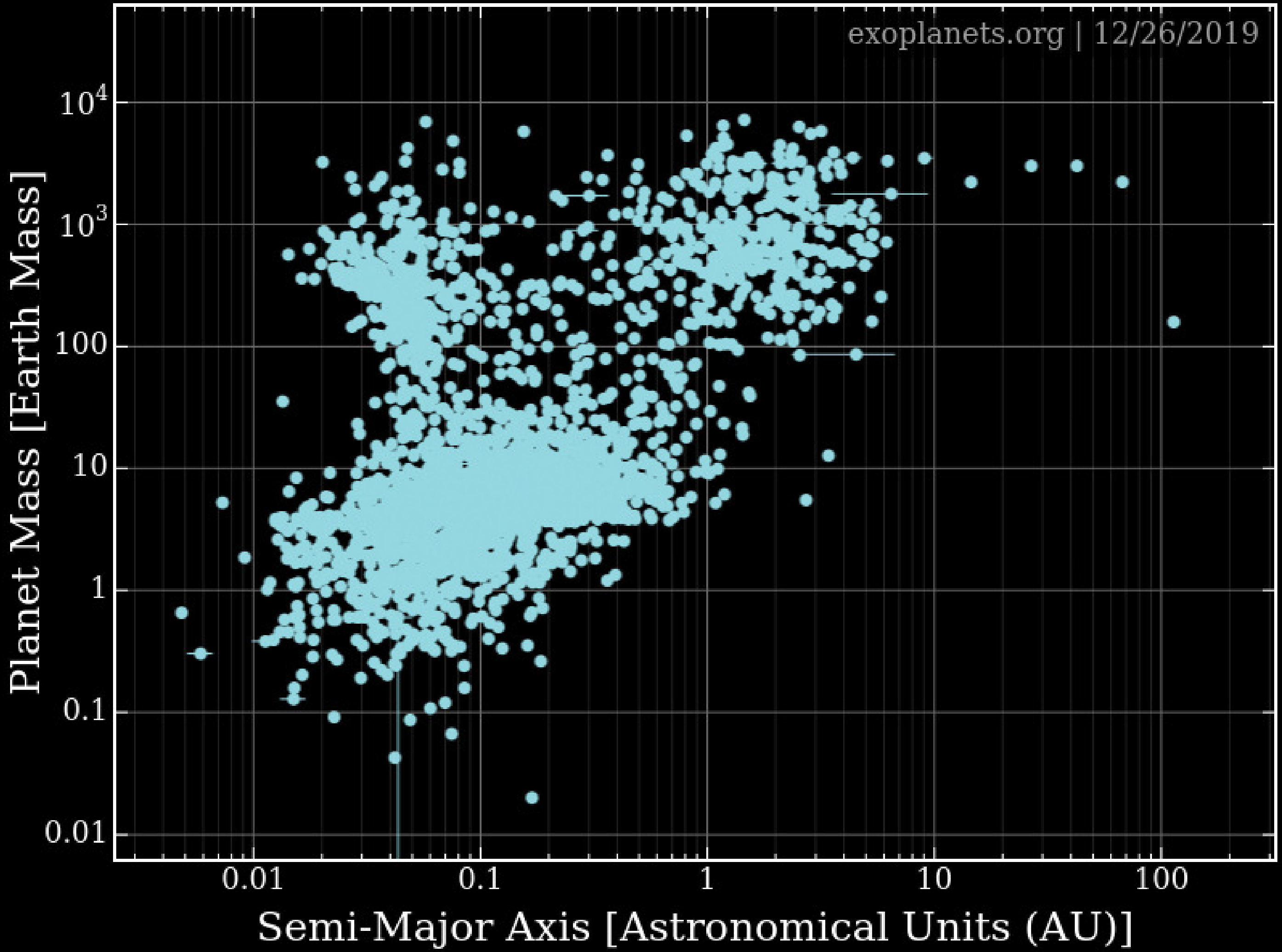
https://arepo-code.org/

Credit: C. N. Kimmig

Extra – All the known exoplanets

About 6000 exoplanets are known.

Statistically, ever star has a planet.



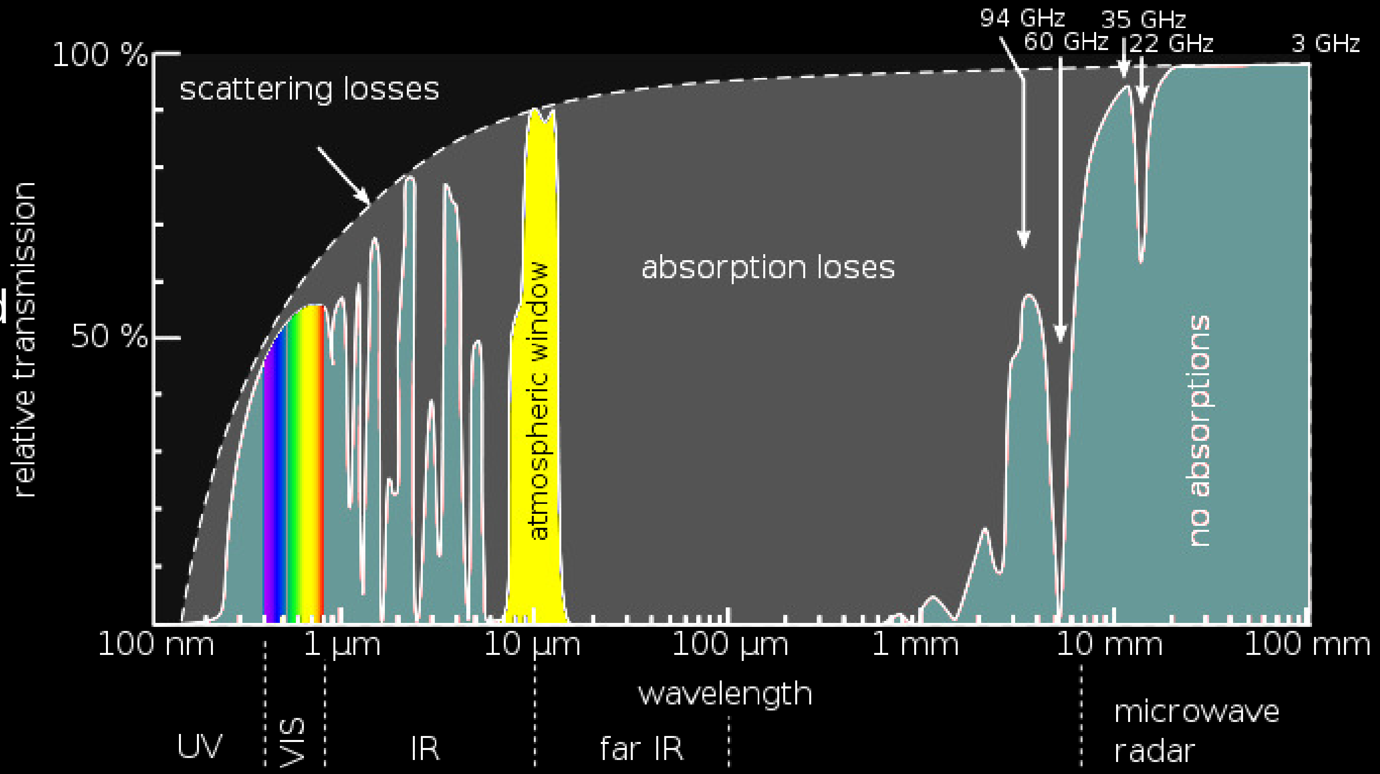
Extra – our window to star formation

Like a colored glas ceiling,
our **atmosphere** only allows
certain colors passing through

Most of **infra-red radiation** is
blocked.

But this is the faint **light** created
by **heated dust** around forming
planets.

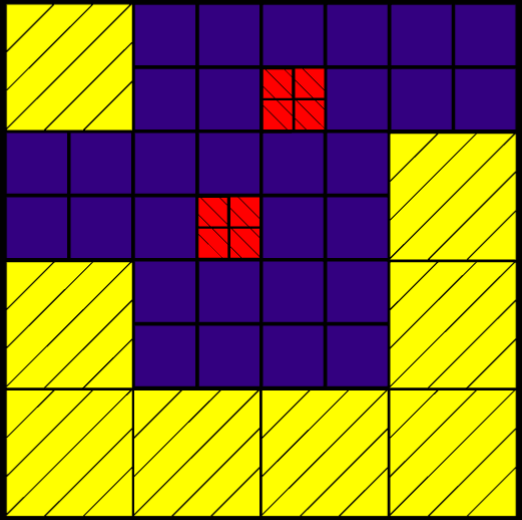
(Also it means we trap most
heat from the ground within
our atmosphere…)

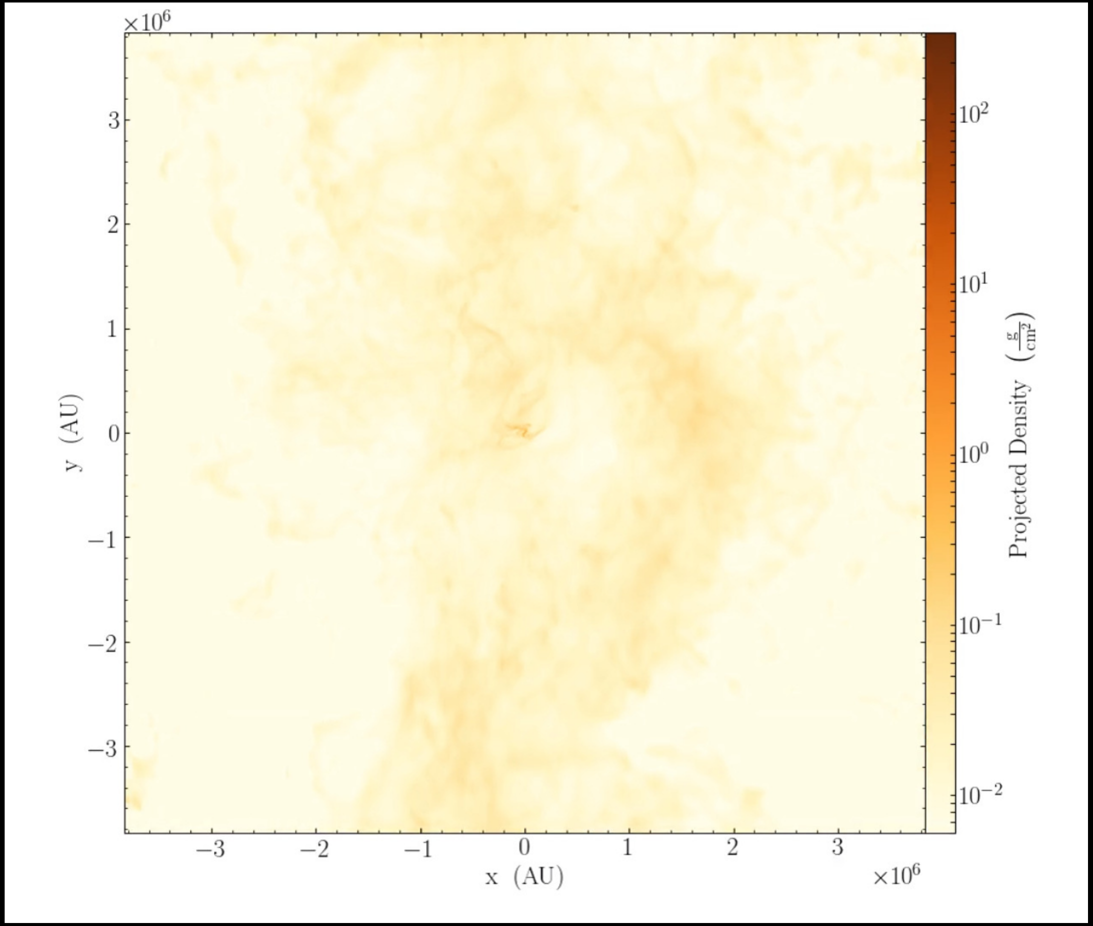


Credit: wikipedia.org

Extra – From galaxies to planets – Smaller boxes

Very different size scales





Credit:
M. Küffmeier