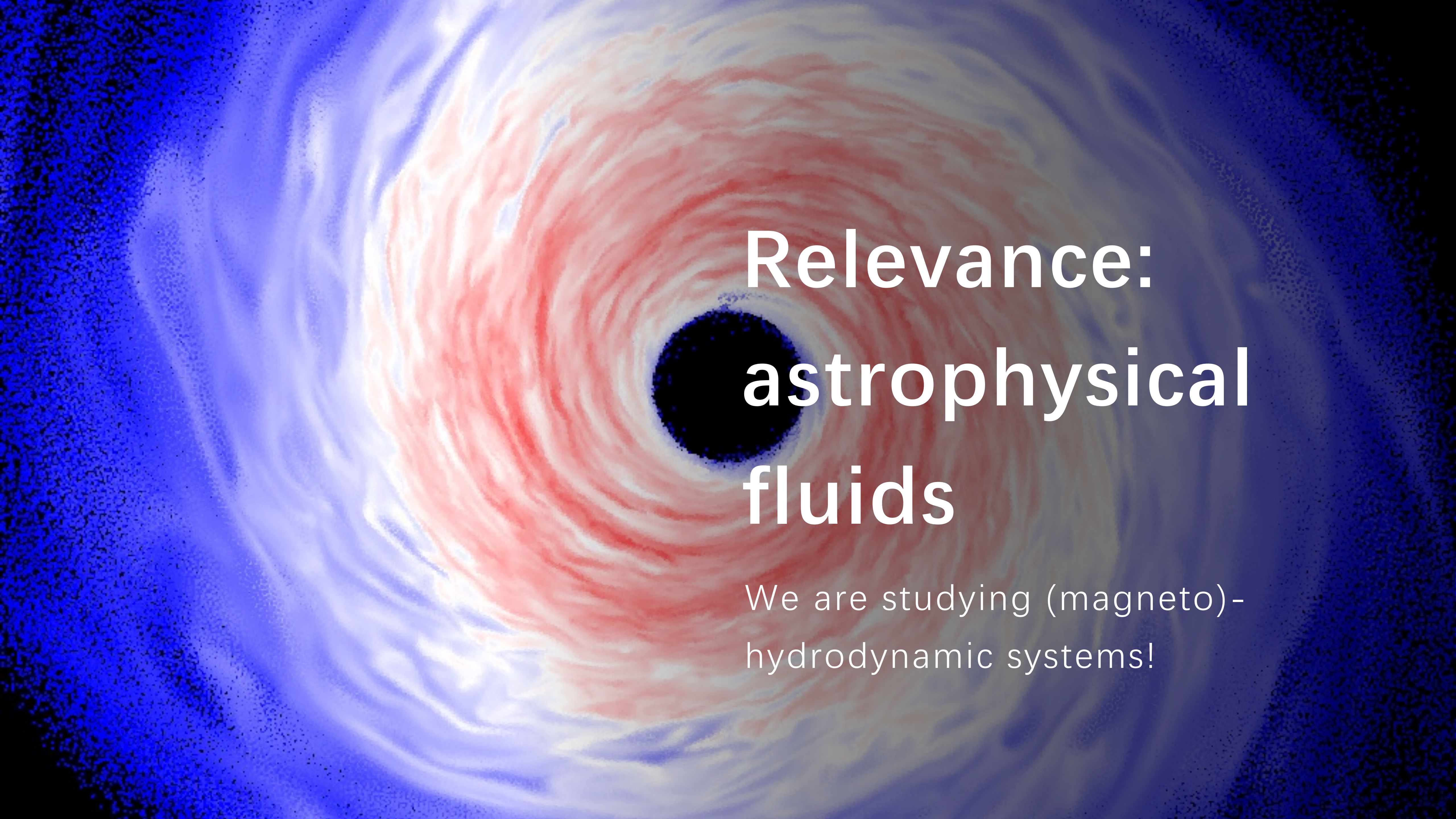
Lagrangian Computational Astrophysics

Dr. Hongping Deng, hpdeng353@shao.ac.cn

Beijing 2024 summer



Relevance: astrophysical fluids

We are studying (magneto)-hydrodynamic systems!

References

SPH

Smoothed particle hydrodynamics and magnetohydrodynamics
Daniel Price 2012

Smoothed Particle Hydrodynamics in Astrophysics
Volker Springel 2010

Moving mesh

E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh

Volker Springel 2010

Meshless Finite Mass

A new class of accurate, mesh-free hydrodynamic simulation methods
Philip Hopkins 2015

Preface

《易经·系辞》形而上者谓之道 形而下者谓之器

子曰：君子不器

Lagrangian vesus Eulerian

A gas cloud

As a goup of fluid parcels

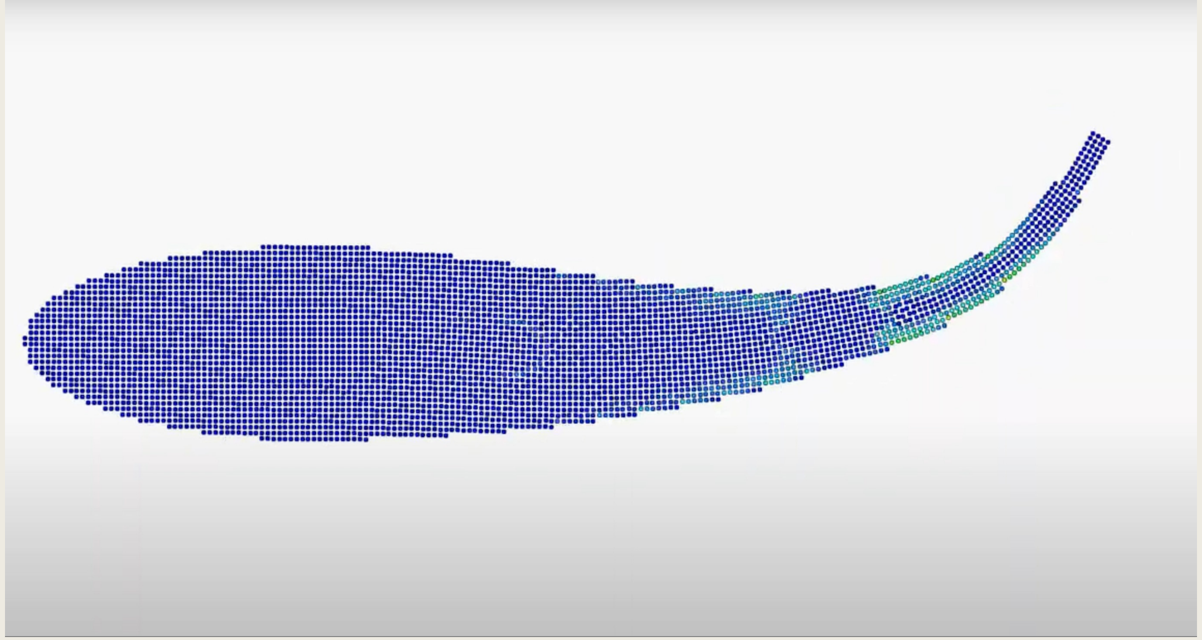
As a field on cells

Evolution of fluid parcels

$$\frac{d\rho}{dt} + \rho \nabla \cdot \mathbf{v} = 0,$$

$$\frac{d\mathbf{v}}{dt} + \frac{\nabla P}{\rho} = 0,$$

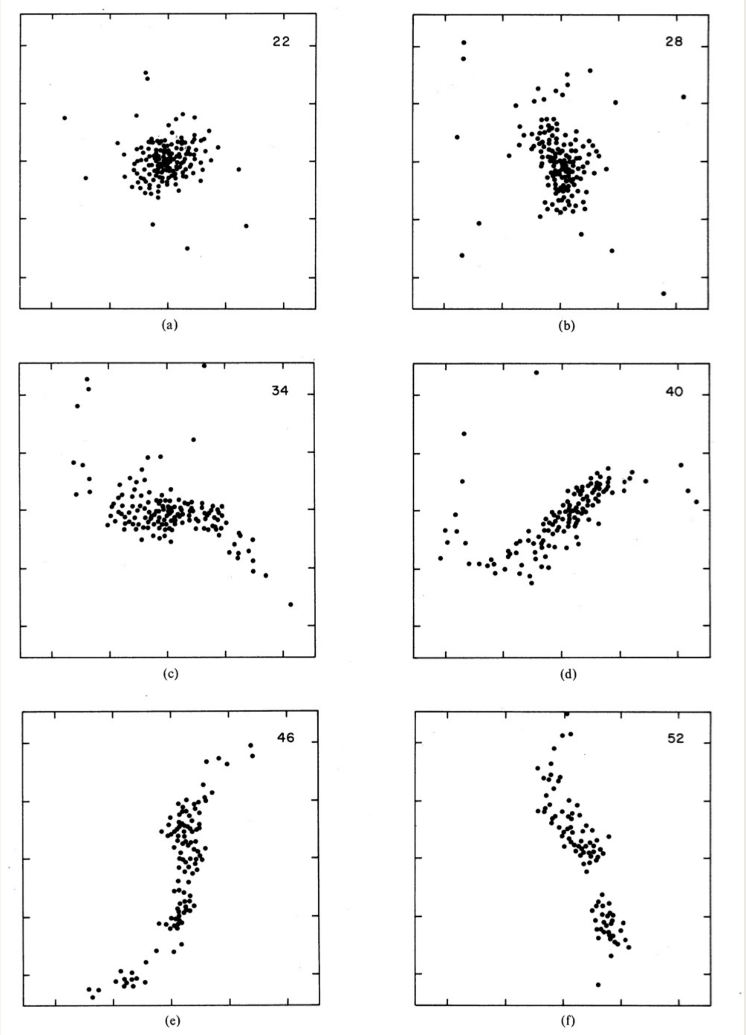
$$\frac{du}{dt} + \frac{P}{\rho} \nabla \cdot \mathbf{v} = 0,$$



Credit: Xiangyu Hu

$$d/dt = \partial/\partial t + \mathbf{v} \cdot \nabla$$

Material derivative



Smoothed particle hydrodynamics

The first ever meshless method (Lucy 1977, Gingold & Monaghan 1977) compared to finite difference method by Lewis Fry Richardson in 1920s.

Heuristic derivation of the density

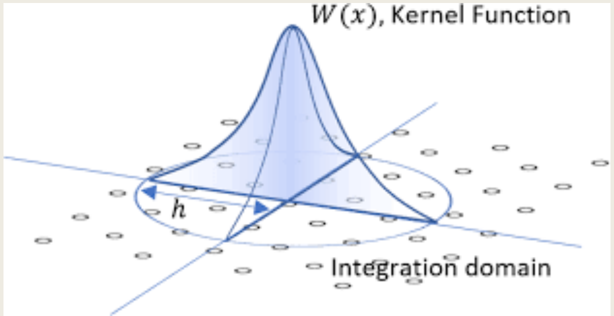
Share with your neighbours, say 60 particles

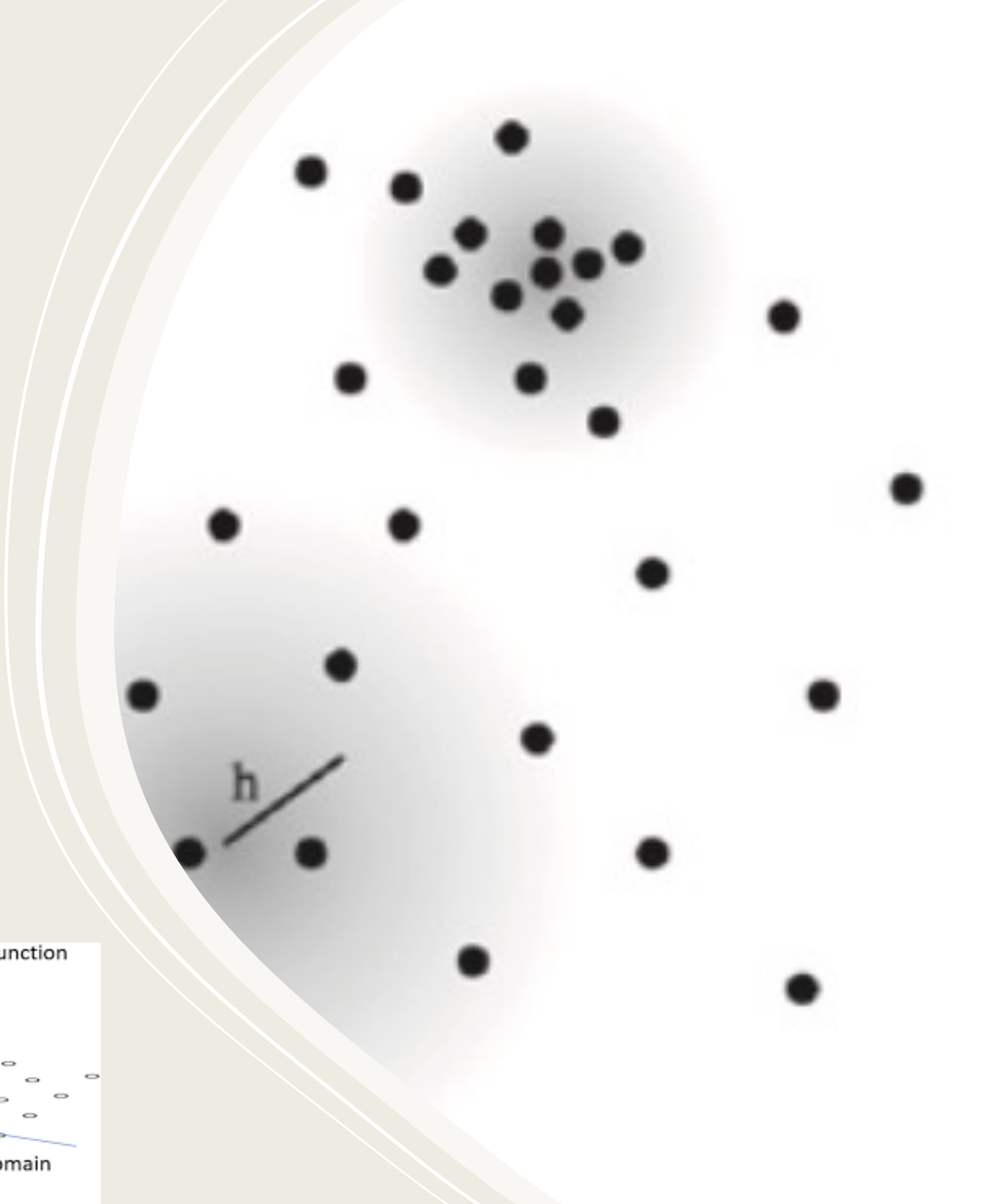
$$\rho_i = \sum_{j=1}^{N} m_j W(\mathbf{r}_i - \mathbf{r}_j, h_i).$$

Smooth of mass distribution -> density

The kernel function should be:

- Smooth
- Isotropic
- Even function





Varying smoothing length

Denser regions with smaller h
Constant mass in the kernel

$$\rho_i h_i^3 = const$$

$$\frac{\partial \rho_j}{\partial h_j}\frac{\partial h_j}{\partial \mathbf{r}_i}\left[1 + \frac{3\rho_j}{h_j}\left(\frac{\partial \rho_j}{\partial h_j}\right)^{-1}\right] = -\nabla_i \rho_j.$$

$$\frac{\partial \rho_j}{\partial \mathbf{r}_i} = \left(1 + \frac{h_j}{3\rho_j}\frac{\partial \rho_j}{\partial h_j}\right)^{-1}\nabla_i \rho_j.$$

Density a function of particle positions

$$\rho_i = \rho_i(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N, h_i)$$

$$h_i = h_i(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N)$$

$$\frac{\partial \rho_j}{\partial \mathbf{r}_i} = \nabla_i \rho_j + \frac{\partial \rho_j}{\partial h_j}\frac{\partial h_j}{\partial \mathbf{r}_i},$$

The equation of motion from the Lagrangian

Lagrangian of invicid fluids

$$L = \int \rho \left(\frac{\mathbf{v}^2}{2} - u \right) \mathrm{d}V.$$

Euler-Lagrange equation

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{\mathbf{r}}_i} - \frac{\partial L}{\partial \mathbf{r}_i} = 0.$$

$$m_i \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N} m_j \frac{P_j}{\rho_j^2} \frac{\partial \rho_j}{\partial \mathbf{r}_i},$$

Discrete version for ideal gas (not necessary, see Price 2012)

$$L_{\mathrm{SPH}} = \sum_i \left(\frac{1}{2} m_i \mathbf{v}_i^2 - m_i u_i \right),$$

$$P_i = A_i \rho_i^{\gamma} = (\gamma - 1) \rho_i u_i,$$

The equation of motion from the Lagrangian

We know previously

$$\frac{\partial \rho_j}{\partial \mathbf{r}_i} = \left(1 + \frac{h_j}{3\rho_j}\frac{\partial \rho_j}{\partial h_j}\right)^{-1} \nabla_i \rho_j.$$

From the density calculation

$$\nabla_i \rho_j = m_i \nabla_i W_{ij}(h_j) + \delta_{ij} \sum_{k=1}^{N} m_k \nabla_i W_{ki}(h_i),$$

$$W_{ij}(h) = W(|\mathbf{r}_i - \mathbf{r}_j|, h)$$

The equation of motion

$$\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N} m_j \left[f_i \frac{P_i}{\rho_i^2} \nabla_i W_{ij}(h_i) + f_j \frac{P_j}{\rho_j^2} \nabla_i W_{ij}(h_j) \right]$$

$$f_i = \left[1 + \frac{h_i}{3\rho_i}\frac{\partial \rho_i}{\partial h_i}\right]^{-1}$$

Internal energy equation

The interanl energy and density are related by

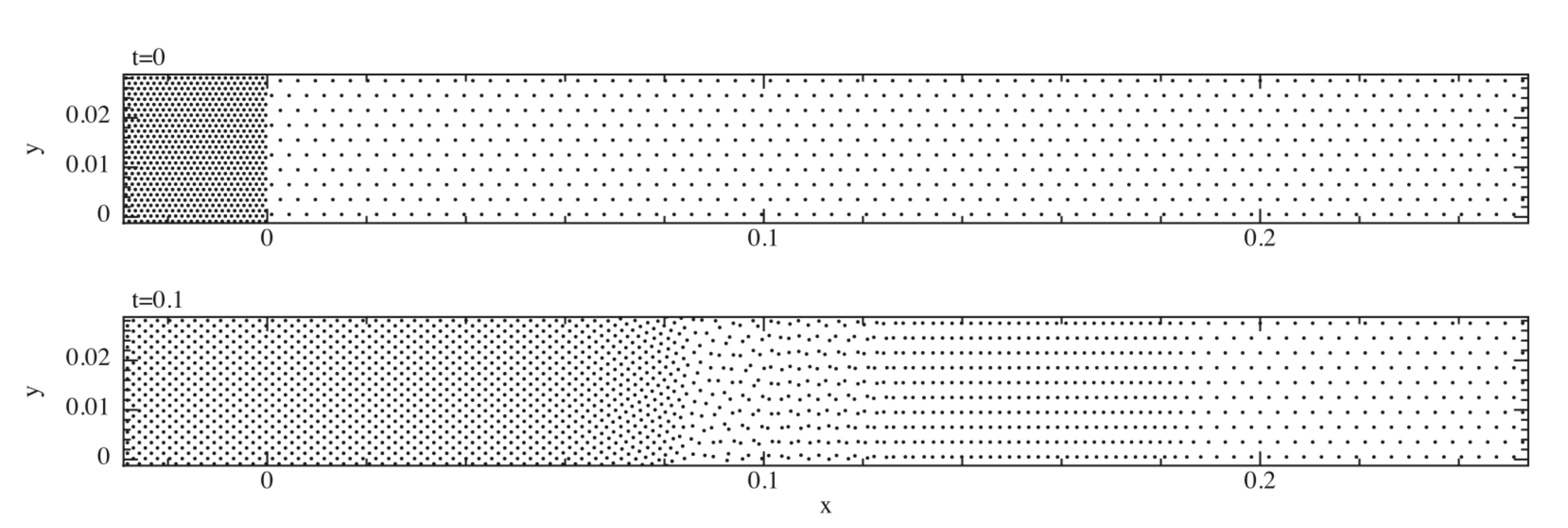
$$P_i = A_i \rho_i^{\gamma} = (\gamma - 1)\rho_i u_i,$$

$$\frac{du_i}{dt} = \frac{P_i}{\rho_i} \sum_j \mathbf{v}_j \cdot \frac{\partial \rho_i}{\partial \mathbf{r}_j}.$$

Internal energy evolution

$$\frac{du_i}{dt} = f_i \frac{P_i}{\rho_i} \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{ij}(h_i),$$

Shock capturing



Price 2012

Prevent particle interpenetration

Artificial viscosity

$$\left.\frac{d\mathbf{v}_i}{dt}\right|_{\rm visc} = -\sum_{j=1}^{N} m_j \Pi_{ij} \nabla_i \overline{W}_{ij},$$

$$\overline{W}_{ij} = \frac{1}{2}\left[W_{ij}(h_i) + W_{ij}(h_j)\right]$$

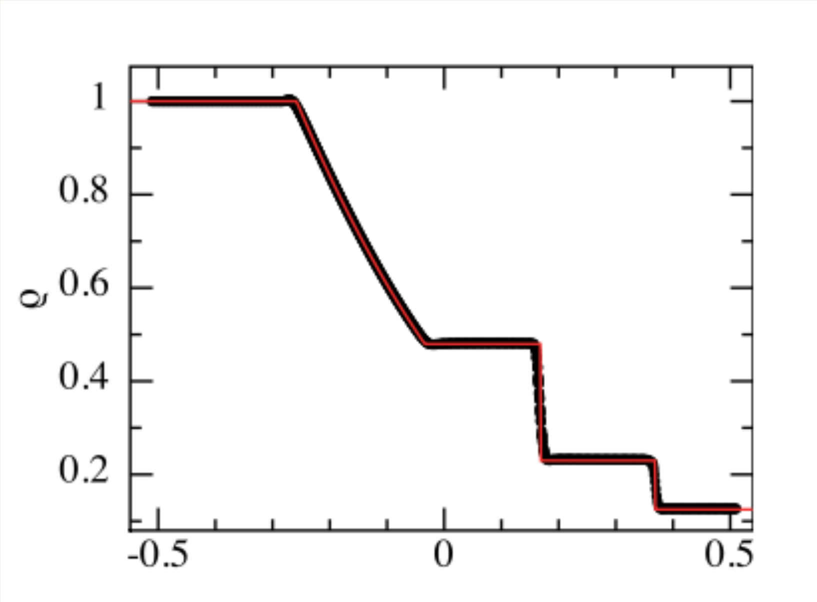
$$\left.\frac{du_i}{dt}\right|_{\rm visc} = \frac{1}{2}\sum_{j=1}^{N} m_j \Pi_{ij} \mathbf{v}_{ij} \cdot \nabla_i \overline{W}_{ij},$$

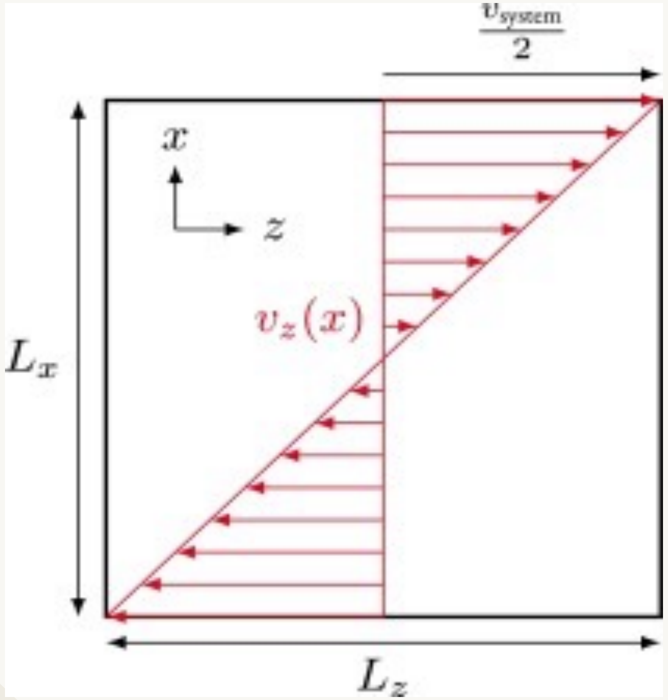
$$\Pi_{ij} = \begin{cases} \left[-\alpha c_{ij}\mu_{ij} + \beta\mu_{ij}^2\right]/\rho_{ij} & \text{if } \mathbf{v}_{ij}\cdot\mathbf{r}_{ij} < 0 \\ 0 & \text{otherwise}, \end{cases}$$

$$\mu_{ij} = \frac{h_{ij}\,\mathbf{v}_{ij}\cdot\mathbf{r}_{ij}}{\left|\mathbf{r}_{ij}\right|^2 + \varepsilon h_{ij}^2}.$$

Should be applied only at the shock fronts

Shock capturing and unwanted artificial viscosity





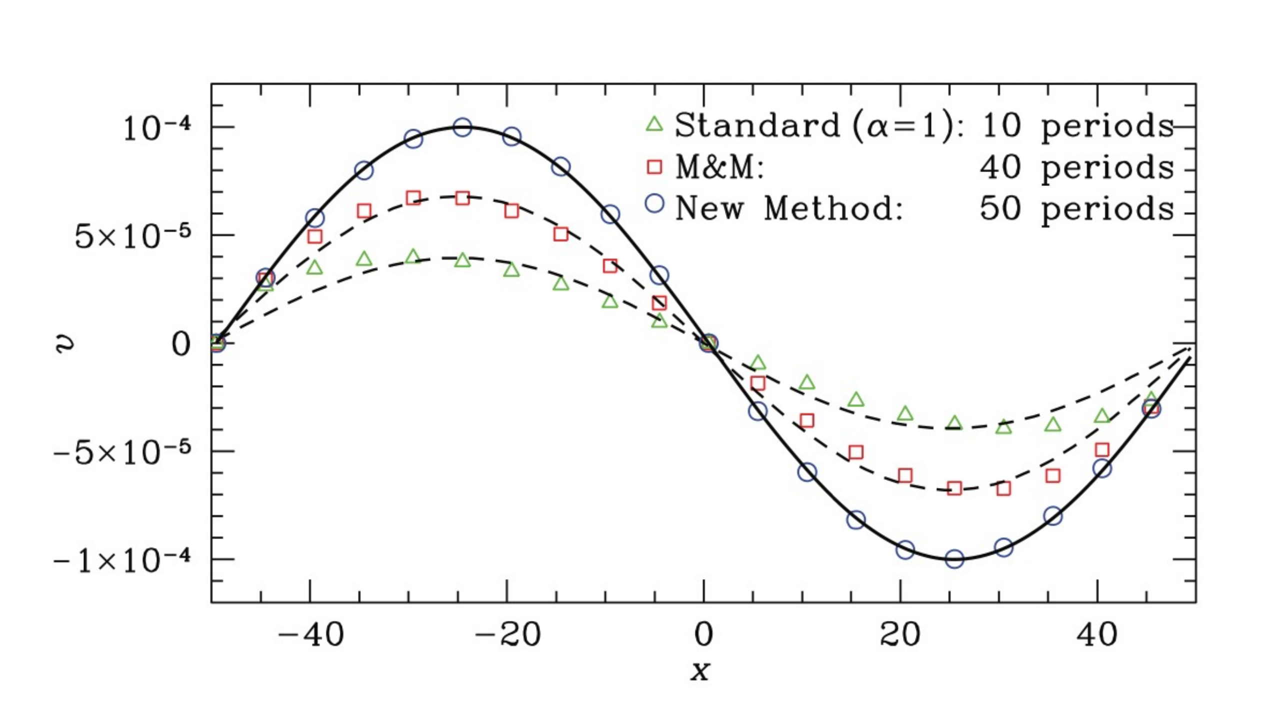
Shear flow can be mis-inteperated as shocks with the previous shock indicator

Prevent excessive artificial viscosity

New shock inidcator

$$\dot{\nabla}\cdot\boldsymbol{v} \equiv \mathrm{d}(\nabla\cdot\boldsymbol{v})/\mathrm{d}t$$

$$\dot{\alpha}_i = (\alpha_{\mathrm{loc},i} - \alpha_i)/\tau_i$$



Cullen & Dehnen 2010

Remove any unwanted viscosity away from shocks

Implementation of SPH

To name a few:

Gadget 1,2,3,4 by Volker Springel et al. (GIZMO)

Gasoline 1,2 by Wadlsley et al.

Phantom by Price et al.

Swift by Shcaller et al.

Magma 1,2 by Rosswog et al.

Further improvements in SPH

An integral approach to calculating gradients. García-Senz et al 2012

More accurate interpolation with reproducing Kernel Frontiere et al 2016

Access Code

Article Type Access Code (4)

Year All

Access Code
21 Feb 2024

GADGET

Volker Springel created the original GADGET code more than 25 years ago. Now it supports some of the largest simulations in astrophysics, and is being developed to do vastly more.

Paul Woods

Access Code
15 Nov 2023

HARM

Charles Gammie and colleagues wrote the HARM code to tackle the extreme physics close to a spinning black hole. Twenty years later, it is performing a similar task in three dimensions in 1/10,000th of the time.

Paul Woods

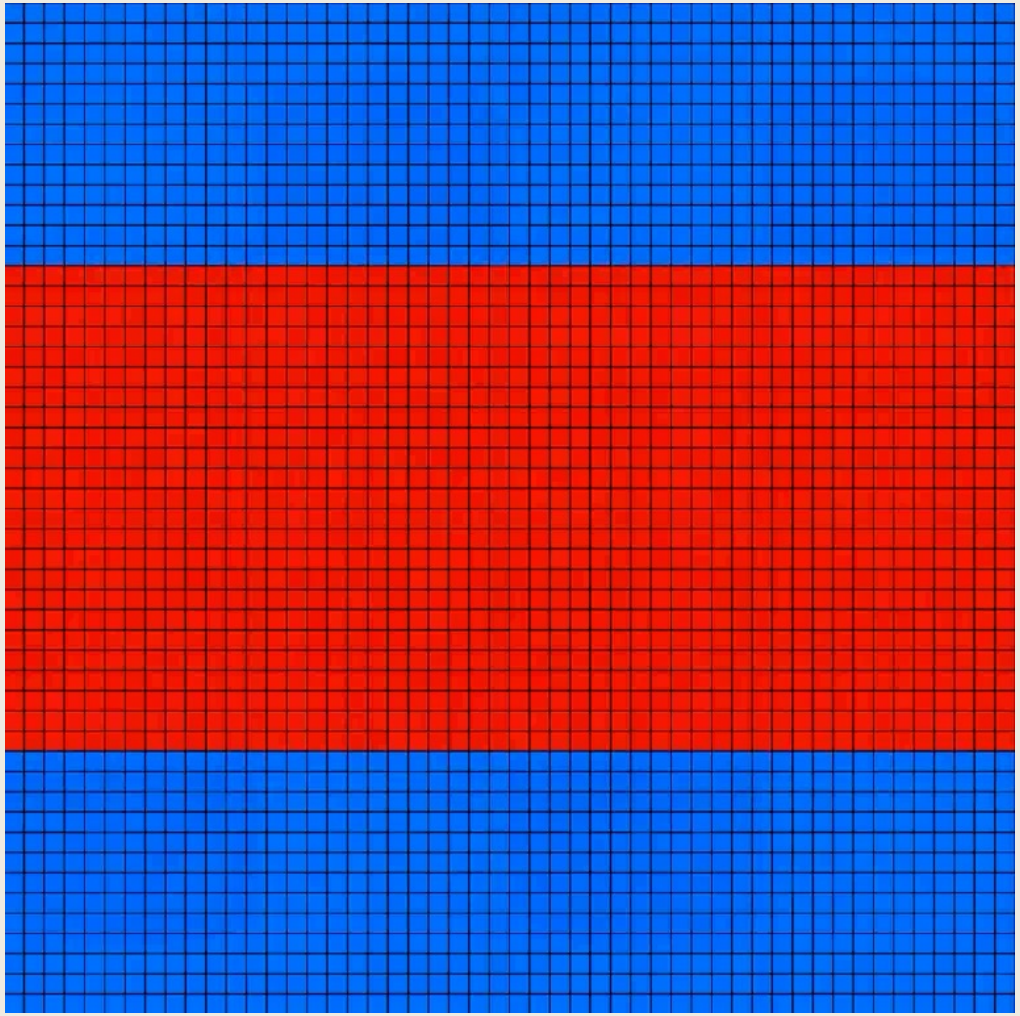
Access Code
15 Sept 2023

Athena

James Stone started developing Athena in the mid-2000s, building on several years' work on numerical methods for compressible magnetohydrodynamics in shocks. A couple of incarnations later, AthenaK is ready to face the exascale computing future.

Paul Woods

Moving mesh with Voronoi tessellation



Conservation laws in a moving frame

Euler equation in the rest frame

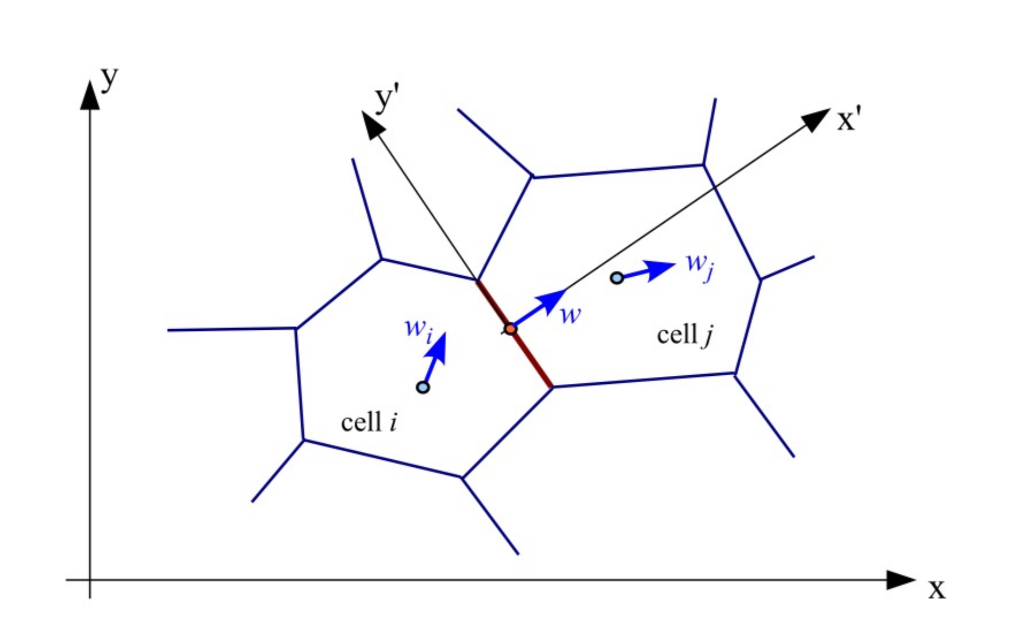
$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho \boldsymbol{v} \\ \rho e \end{pmatrix}$$

$$\boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} \rho \boldsymbol{v} \\ \rho \boldsymbol{v}\boldsymbol{v}^T + P \\ (\rho e + P)\boldsymbol{v} \end{pmatrix}$$

Conservation law in the moving frame

$$\boldsymbol{Q}_i = \begin{pmatrix} m_i \\ \boldsymbol{p}_i \\ E_i \end{pmatrix} = \int_{V_i} \boldsymbol{U}\, \mathrm{d}V.$$

$$\frac{\mathrm{d}\boldsymbol{Q}_i}{\mathrm{d}t} = -\int_{\partial V_i} \left[\boldsymbol{F}(\boldsymbol{U}) - \boldsymbol{U}\boldsymbol{w}^T\right] \mathrm{d}\boldsymbol{n}.$$



Illustrition for AREPO, *Springel 2010*

Gradients for reconstruction

Gradient at the ith particle

$$\phi(\mathbf{r}) = \phi(\mathbf{s}_i) + \langle \nabla \phi \rangle_i (\mathbf{r} - \mathbf{s}_i)$$

Prediction at it neighbouring points

$$\phi_j = \phi_i + \langle \nabla \phi \rangle_i (\mathbf{s}_j - \mathbf{s}_i)$$

To minimize the discrepency

$$S_{\mathrm{tot}} = \sum_j g_j \left(\phi_j - \phi_i - \langle \nabla \phi \rangle_i (\mathbf{s}_j - \mathbf{s}_i)\right)^2$$

Pakmor et al. 2015

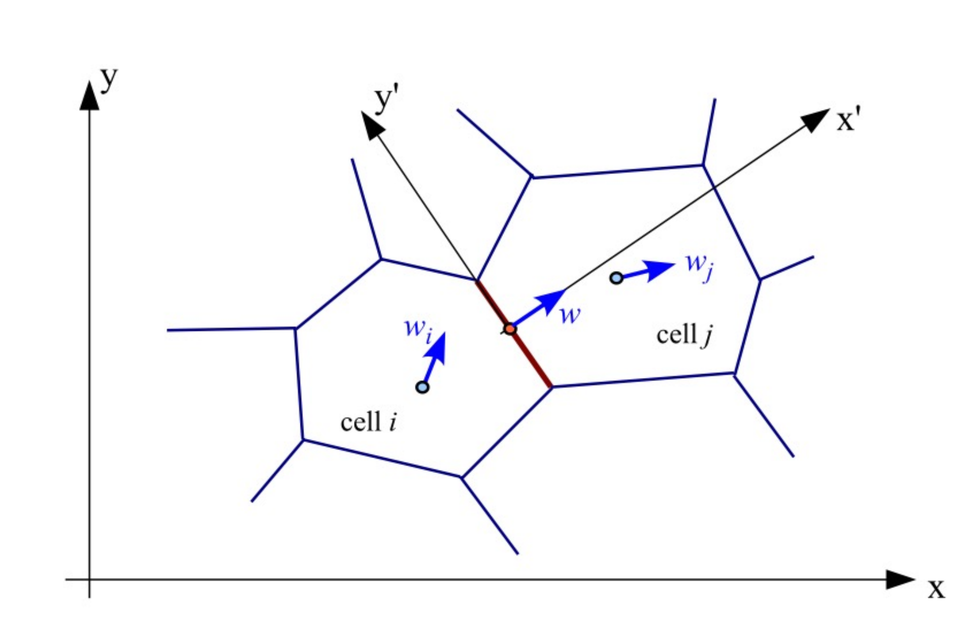
Solve the Rieman problem

To the comoving frame $\boldsymbol{W}_i = (\rho_i, \boldsymbol{v}_i, P_i)$

$$\boldsymbol{W}'_{\mathrm{L,R}} = \boldsymbol{W}_{\mathrm{L,R}} - \begin{pmatrix} 0 \\ \boldsymbol{w} \\ 0 \end{pmatrix}$$

Reconstruction to face center

$$\boldsymbol{W}''_{\mathrm{L,R}} = \boldsymbol{W}'_{\mathrm{L,R}} + \left.\frac{\partial \boldsymbol{W}'}{\partial \boldsymbol{r}}\right|_{\mathrm{L,R}} (\boldsymbol{f} - \boldsymbol{s}_{\mathrm{L,R}}) + \left.\frac{\partial \boldsymbol{W}'}{\partial t}\right|_{\mathrm{L,R}} \frac{\Delta t}{2}$$



Left cell i; right cell j

Solve the Riemann problem

Rotate the frame, the new face normal being x axis

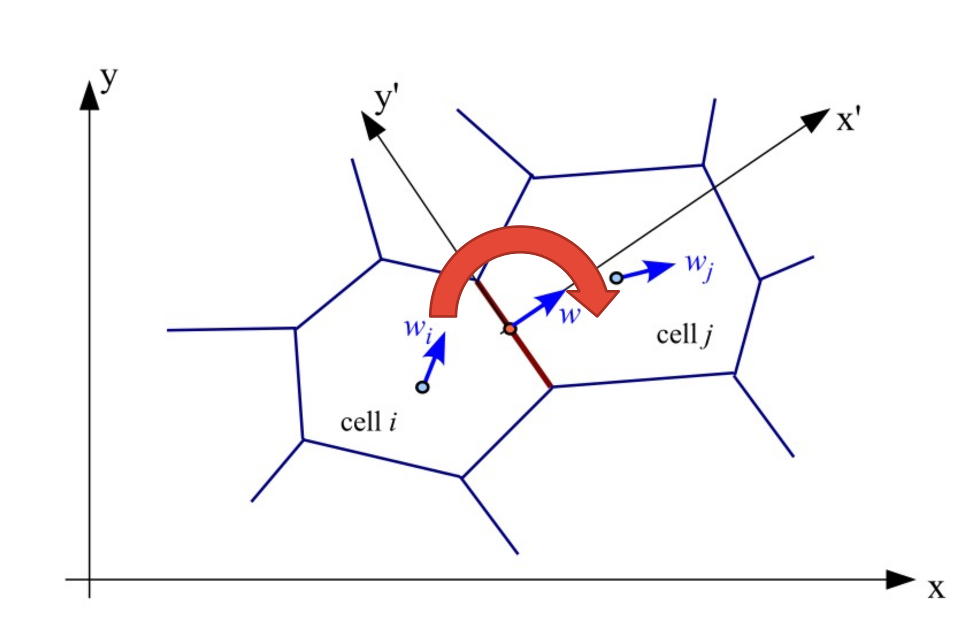
$$\boldsymbol{W}'''_{\mathrm{L,R}} = \boldsymbol{\Lambda}\,\boldsymbol{W}''_{\mathrm{L,R}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \boldsymbol{\Lambda}_{3\mathrm{D}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{W}''_{\mathrm{L,R}},$$

Riemann solver

$$\boldsymbol{W}_{\mathrm{F}} = R_{\mathrm{iemann}}(\boldsymbol{W}'''_{\mathrm{L}}, \boldsymbol{W}'''_{\mathrm{R}}),$$

Rotate and boost back to lab frame

$$\boldsymbol{W}_{\mathrm{lab}} = \begin{pmatrix} \rho \\ \boldsymbol{v}_{\mathrm{lab}} \\ P \end{pmatrix} = \Lambda^{-1}\boldsymbol{W}_{\mathrm{F}} + \begin{pmatrix} 0 \\ \boldsymbol{w} \\ 0 \end{pmatrix}$$



Left cell i; right cell j

The flux is done!

Dump the updated ρ, v_{lab}, P into the flux function

$$\hat{\boldsymbol{F}} = \boldsymbol{F}(\boldsymbol{U}) - \boldsymbol{U}\boldsymbol{w}^{\mathrm{T}} = \begin{pmatrix} \rho(\boldsymbol{v}_{\mathrm{lab}} - \boldsymbol{w}) \\ \rho\boldsymbol{v}_{\mathrm{lab}}(\boldsymbol{v}_{\mathrm{lab}} - \boldsymbol{w})^{\mathrm{T}} + P \\ \rho e_{\mathrm{lab}}(\boldsymbol{v}_{\mathrm{lab}} - \boldsymbol{w}) + P\boldsymbol{v}_{\mathrm{lab}} \end{pmatrix}$$

$$\boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} \rho\boldsymbol{v} \\ \rho\boldsymbol{v}\boldsymbol{v}^T + P \\ (\rho e + P)\boldsymbol{v} \end{pmatrix}$$

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho\boldsymbol{v} \\ \rho e \end{pmatrix}$$

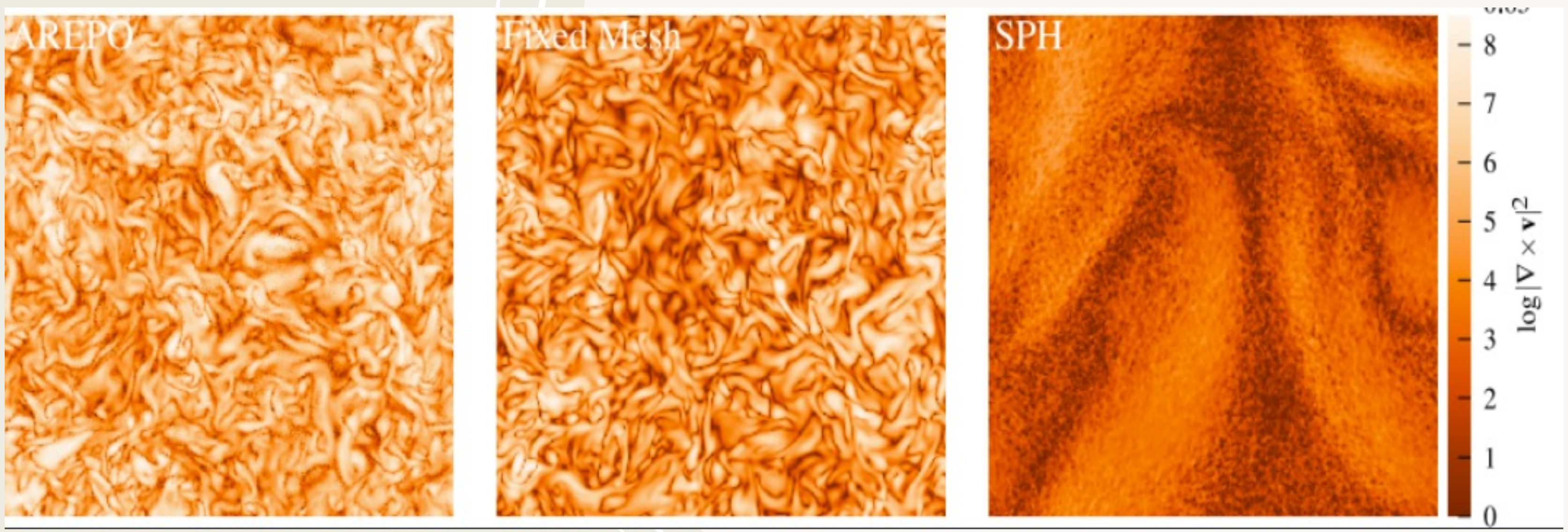
Update the euqations with the flux

$$\frac{\mathrm{d}\boldsymbol{Q}_i}{\mathrm{d}t} = -\int_{\partial V_i} \left[\boldsymbol{F}(\boldsymbol{U}) - \boldsymbol{U}\boldsymbol{w}^T\right] \mathrm{d}\boldsymbol{n}.$$

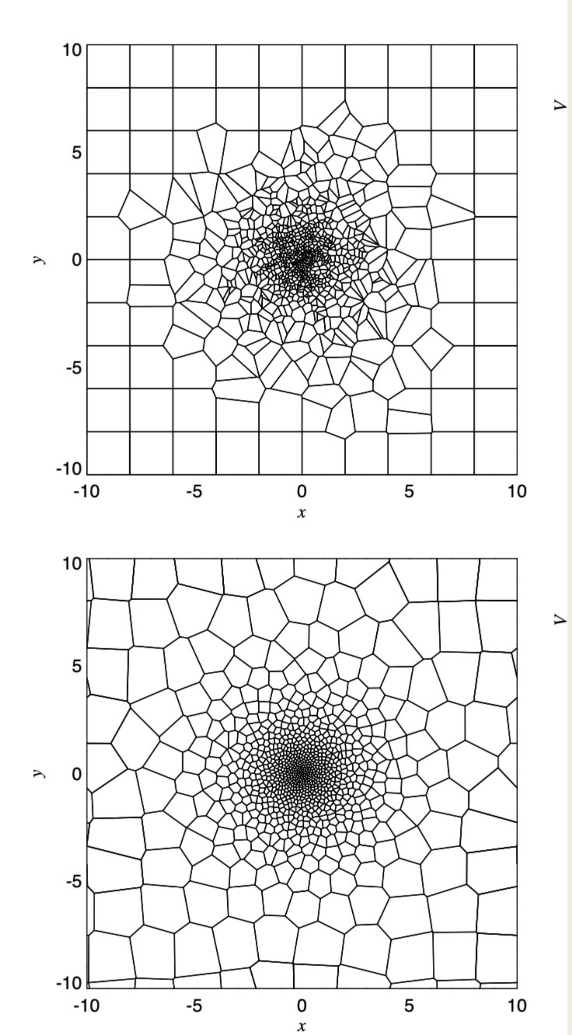
Advantages over SPH

No artificial viscosity!

No smoothing



Subsonic turbulence, *Bauer & Springel 2012*



Tricky things: cell regularization

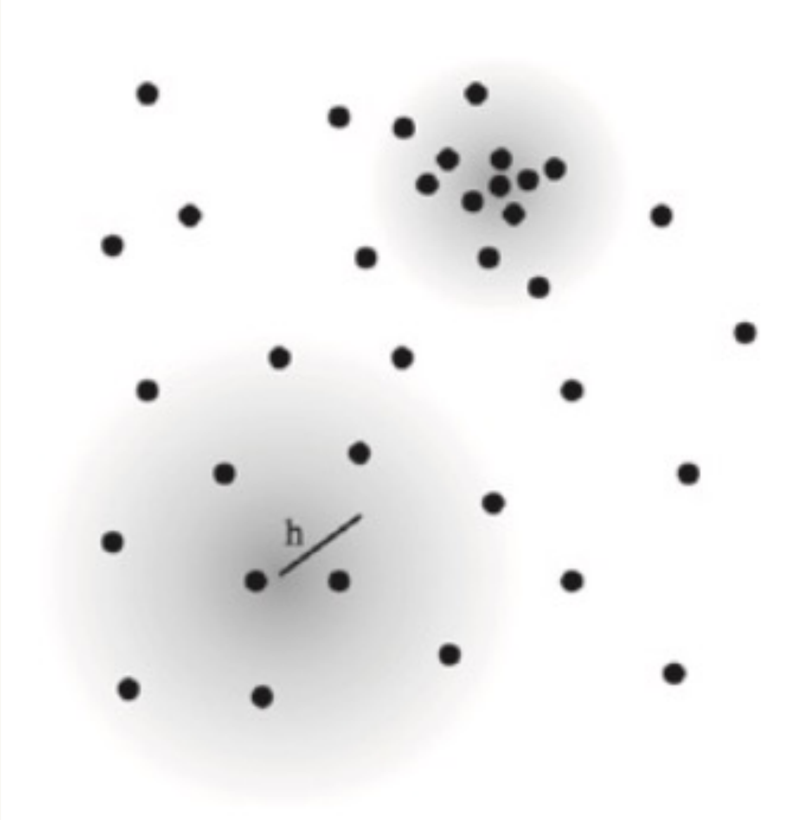
We want more regular cells!

Meshless moving mesh…

Partition of unit $\sum_i \psi_i(\boldsymbol{x}) = 1$

$$\psi_i(\mathbf{x}) \equiv \frac{1}{\omega(\mathbf{x})} W(\mathbf{x} - \mathbf{x}_i, h(\mathbf{x}))$$

$$\omega(\mathbf{x}) \equiv \sum_j W(\mathbf{x} - \mathbf{x}_j, h(\mathbf{x}))$$



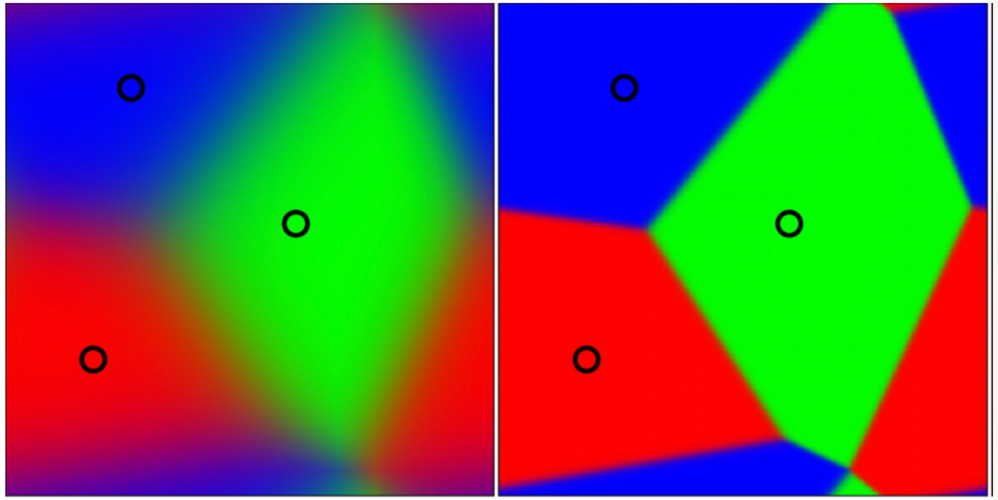
The effective volume

For arbitrary function, and a kernel function with compact support

$$\int f(\mathbf{x})\, d^\nu \mathbf{x} = \sum_i \int f(\mathbf{x})\, \psi_i(\mathbf{x})\, d^\nu \mathbf{x}$$
$$= \sum_i f_i(\mathbf{x}_i) \int \psi_i\, d^\nu \mathbf{x} + \mathcal{O}(h_i(\mathbf{x}_i)^2)$$
$$\equiv \sum_i f_i V_i + \mathcal{O}(h_i^2)$$

$$V_i = \int \psi_i(\mathbf{x})\, d^\nu\, \mathbf{x}$$

$$W(\boldsymbol{x} - \boldsymbol{x}_i) = \delta(\boldsymbol{x} - \boldsymbol{x}_i)$$



New Meshless Methods Here (MFV, MFM) Unstructured / Moving–Mesh Methods

$$\psi_i(\mathbf{x}) \equiv \frac{1}{\omega(\mathbf{x})} W(\mathbf{x} - \mathbf{x}_i, h(\mathbf{x}))$$
$$\omega(\mathbf{x}) \equiv \sum_j W(\mathbf{x} - \mathbf{x}_j, h(\mathbf{x}))$$

The conservation law

$$\frac{d}{dt}(V_i \mathbf{U}_i) + \sum_j \tilde{\mathbf{F}}_{ij} \cdot \mathbf{A}_{ij} = 0$$

With effective faces A_{ij}^{α}

$$V_i \tilde{\psi}_j^{\alpha}(\mathbf{x}_i) - V_j \tilde{\psi}_i^{\alpha}(\mathbf{x}_j).$$

$$\tilde{\psi}_j^{\alpha}(\mathbf{x}_i) \equiv \sum_{\beta=1}^{\beta=\nu} \mathbf{B}_i^{\alpha\beta} (\mathbf{x}_j - \mathbf{x}_i)^{\beta} \psi_j(\mathbf{x}_i) \equiv \mathbf{B}_i^{\alpha\beta} (\mathbf{x}_j - \mathbf{x}_i)^{\beta} \psi_j(\mathbf{x}_i)$$

$$\mathbf{B}_i \equiv \mathbf{E}_i^{-1}$$

$$\mathbf{E}_i^{\alpha\beta} \equiv \sum_j (\mathbf{x}_j - \mathbf{x}_i)^{\alpha} (\mathbf{x}_j - \mathbf{x}_i)^{\beta} \psi_j(\mathbf{x}_i)$$

See e.g. *Hopkins 2015*

Compared to the above two methods

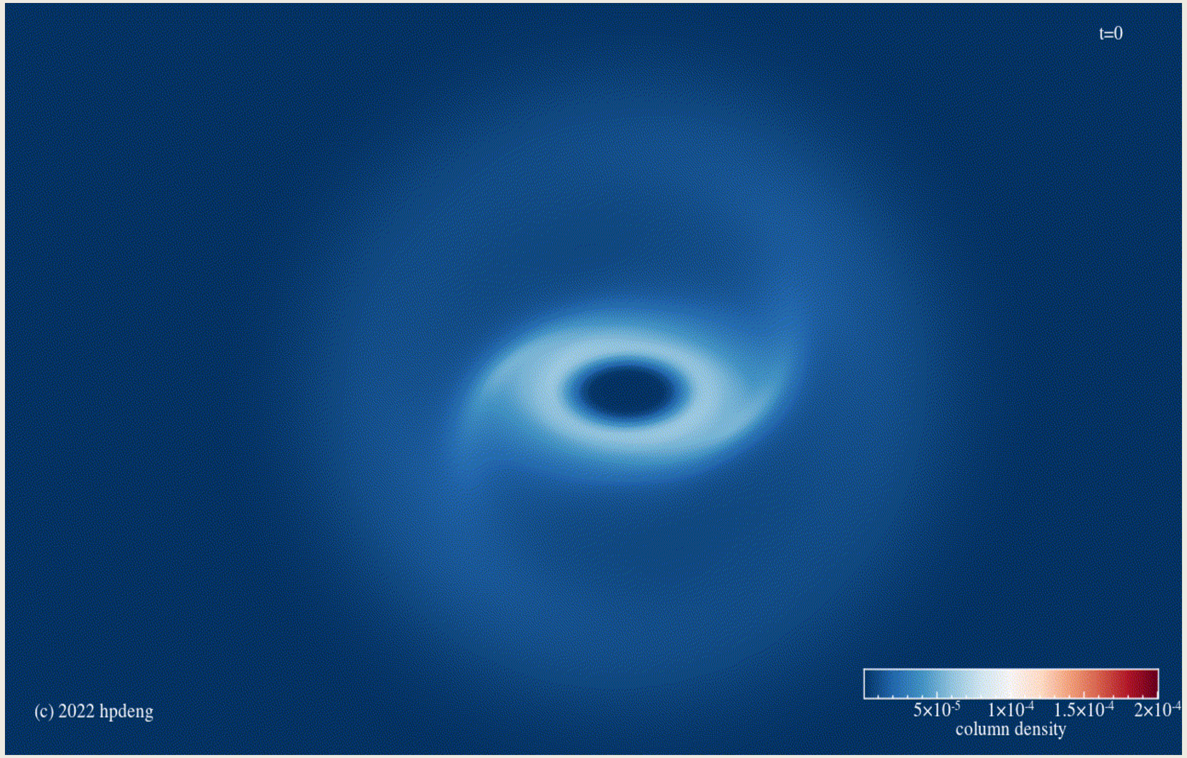
No artificial viscosity

No cell regularization

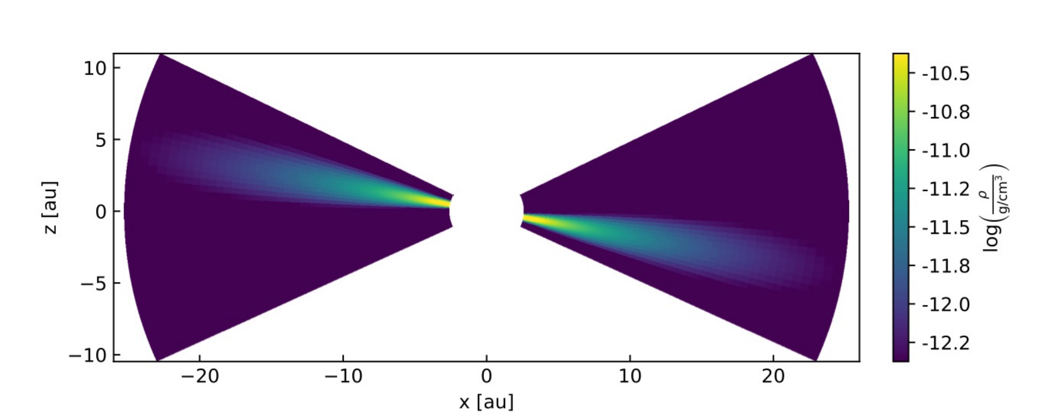
With smoothing…

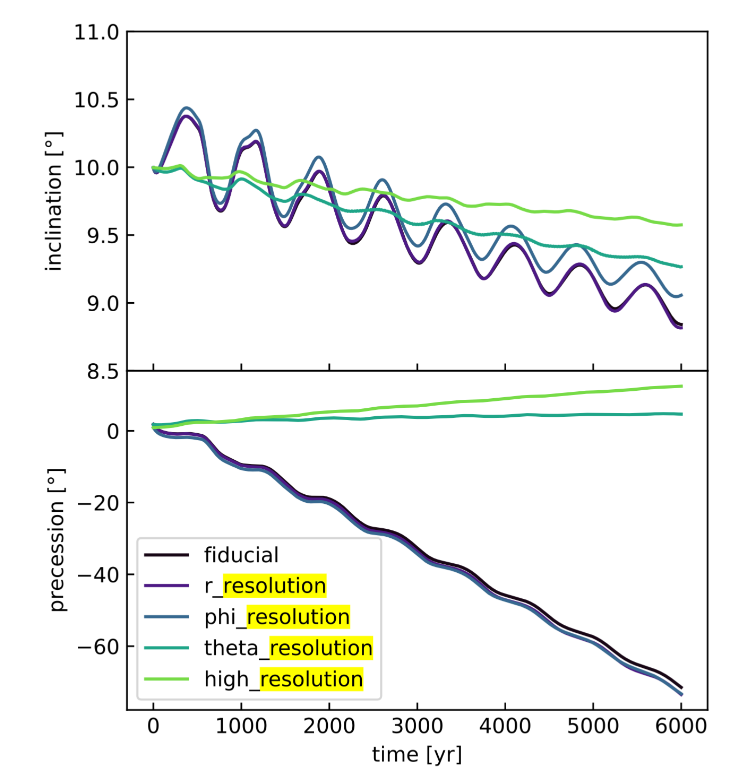
Features

- Applicable to arbitrary flow geometry, tracer
- Resolution adaptivity
- Easy self-gravity
- Galilean invariance/ low advection error



Advection errors

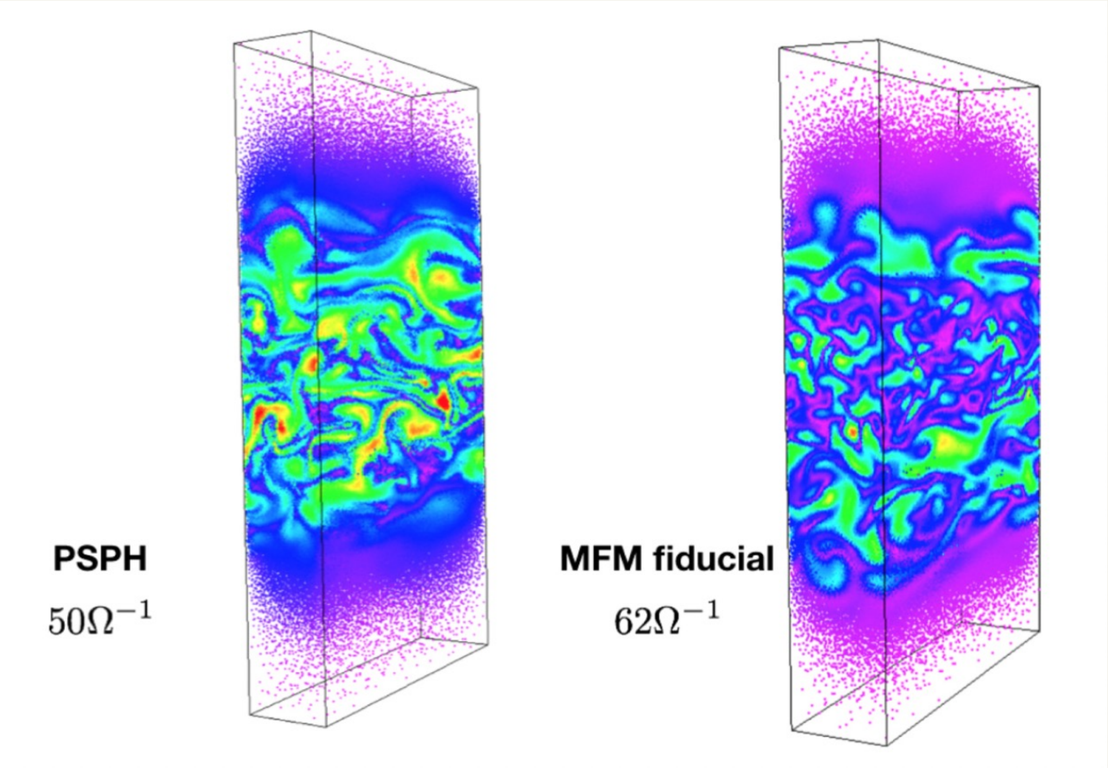




Disk misaligned to grids can be numerically aligned in FARGO3D, even in simulations with 18 cells per disk scale height (Kimming & Dullemond 2024)

Drawback

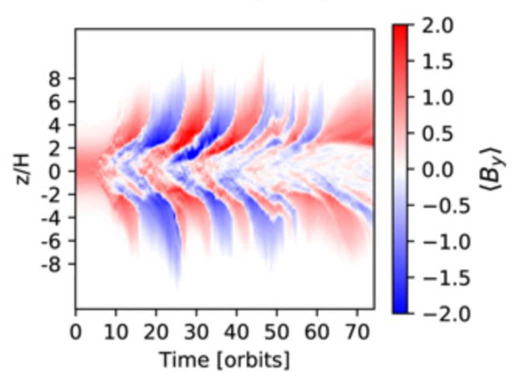
- Numerical noise (esp. artificial viscosity in SPH)
- Divergence of B fileds
- Smoothing, though it is adaptive

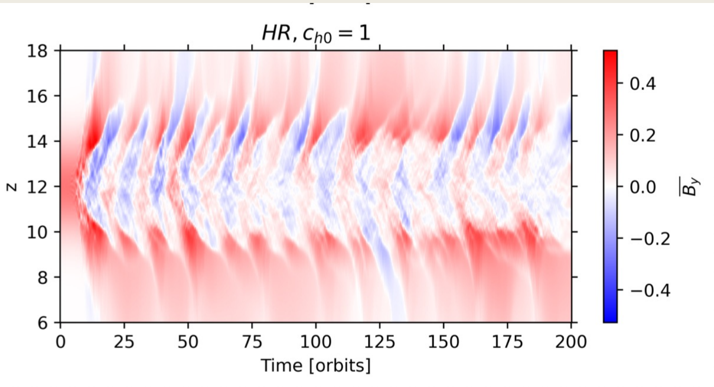


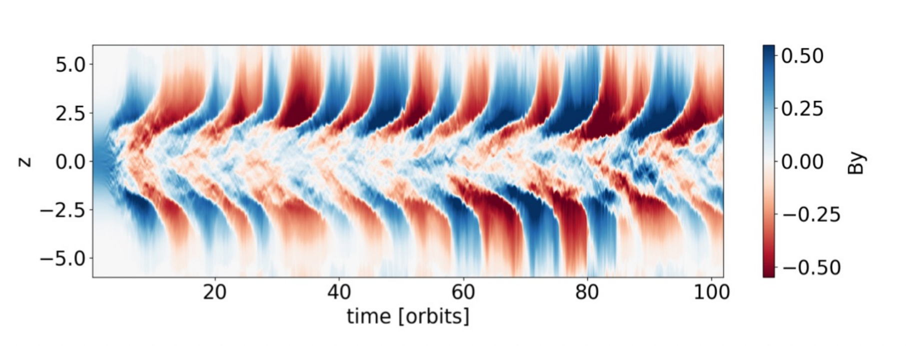
MRI simulation with Lagrangian codes (Deng et al. 2019)

Things are improving

Well maintained MRI turbulence
with SPH & Moving mesh







Applications in planet formation modeling

Distorted disks: fly by, companion, etc.

Circumbinary disk, triple

Planet disk interaction, be careful

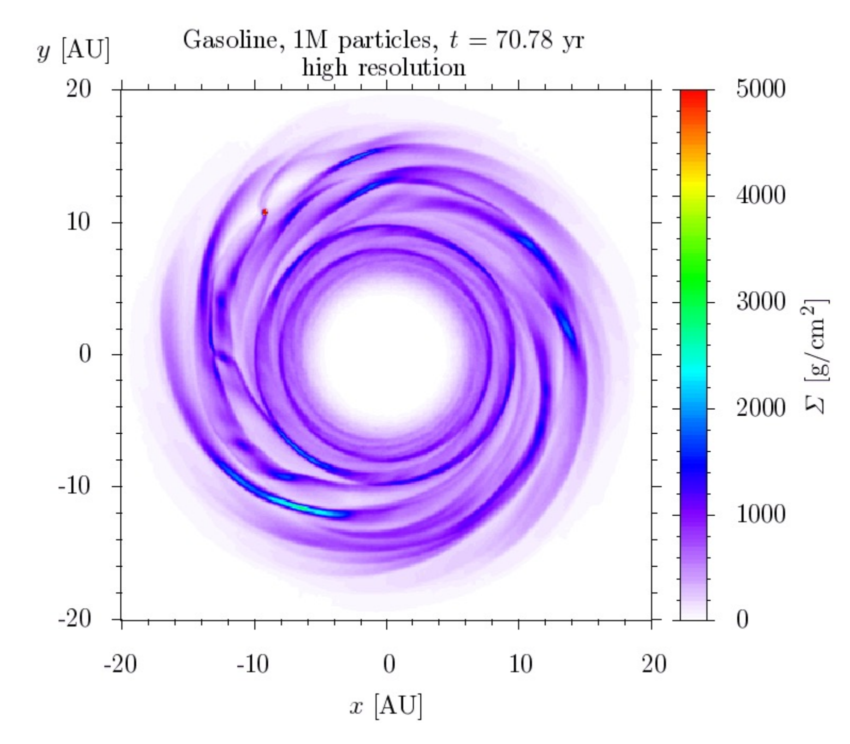
Self-gravitating disks

Gravitational instability

For short wavelength axisymmetric perturbation

$$\omega^2 = \kappa^2 - 2\pi G\Sigma|k_x| + c^2k_x^2,$$

$$Q = \kappa c/\pi G\Sigma < 1$$

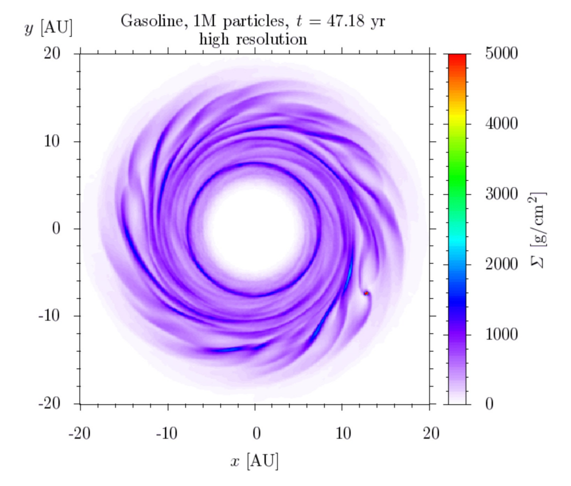


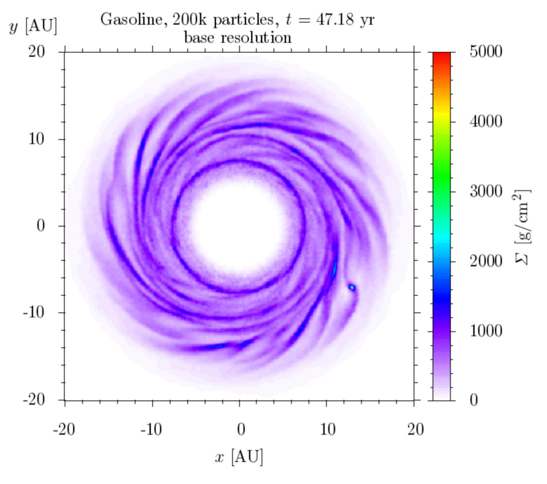
Theory for the growth of spirals (e.g. Deng & Ogilvie 2022)

GI simulations: code comparison

The Wengen test: Enzo, Flash, Gasoline, CHYMERA

https://users.camk.edu.pl/gawrysz/test4/#movies



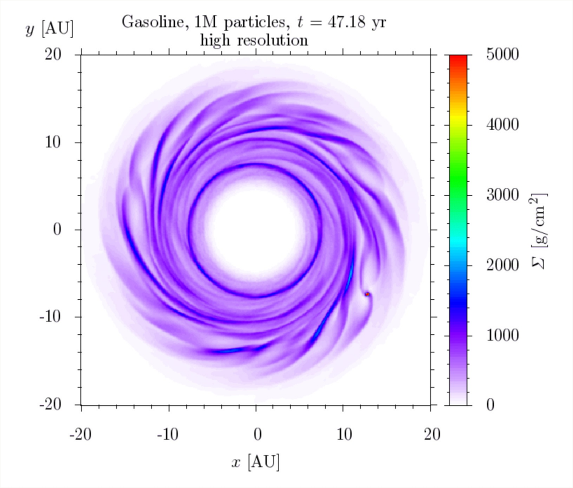


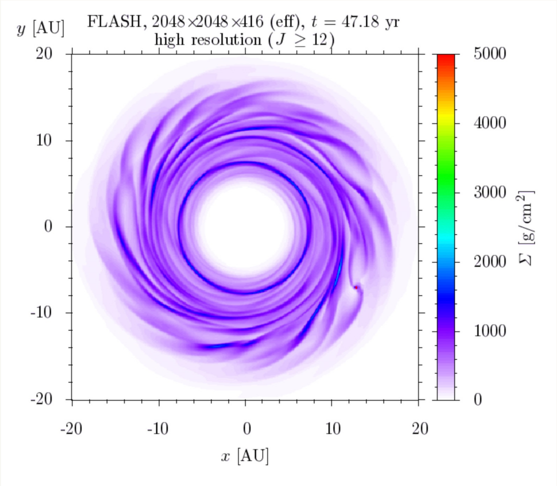
Consistency with as few as 200k particle in SPH simulation

GI simulations: code comparison

The Wengen test: Enzo, Flash, Gasoline, CHYMERA

https://users.camk.edu.pl/gawrysz/test4/#movies



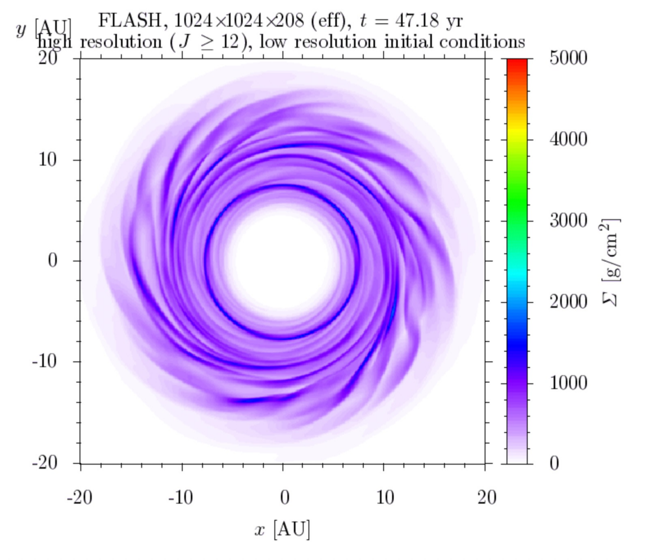


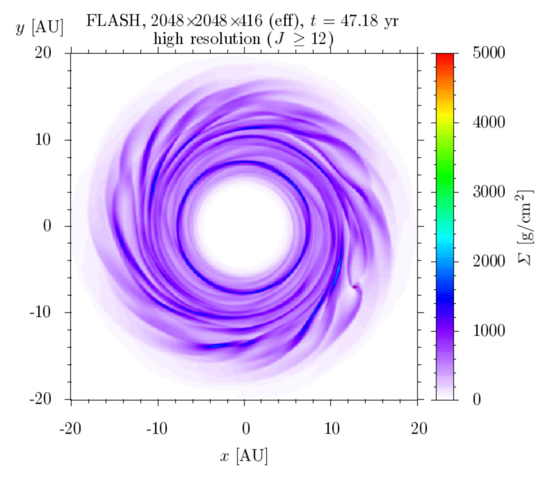
1M particles vs 1700M cells…

GI simulations: code comparison

The Wengen test: Enzo, Flash, Gasoline, CHYMERA

https://users.camk.edu.pl/gawrysz/test4/#movies



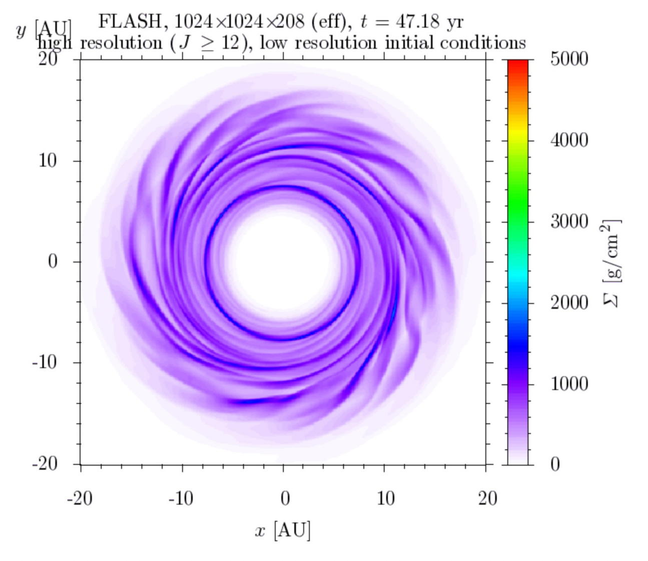


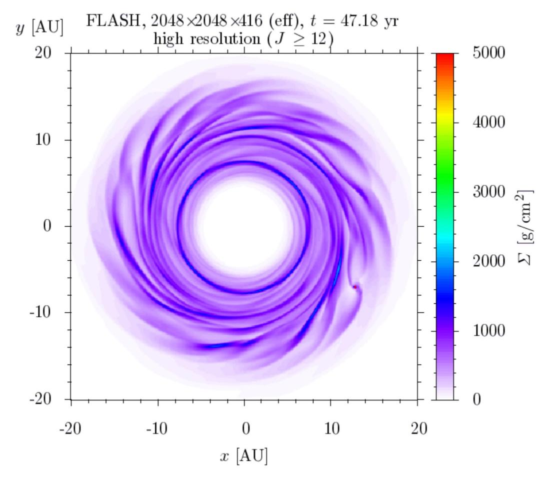
Inconsistency with even 218M cells

GI simulations: code comparison

The Wengen test: Enzo, Flash, Gasoline, CHYMERA

https://users.camk.edu.pl/gawrysz/test4/#movies





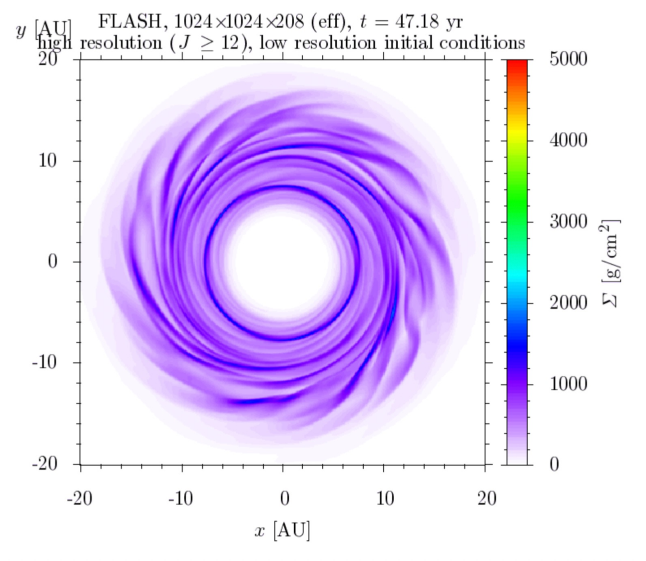
Inconsistency with even 218M cells

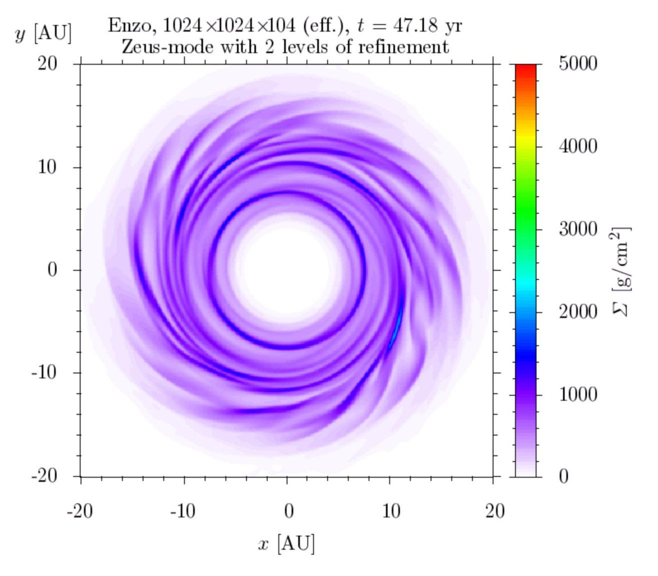
GI simulations: code comparison

The Wengen test: Enzo, Flash, Gasoline, CHYMERA

https://users.camk.edu.pl/gawrysz/test4/#movies

Seems very high resolution is needed for covergency, and to predict the fragment for some grid codes

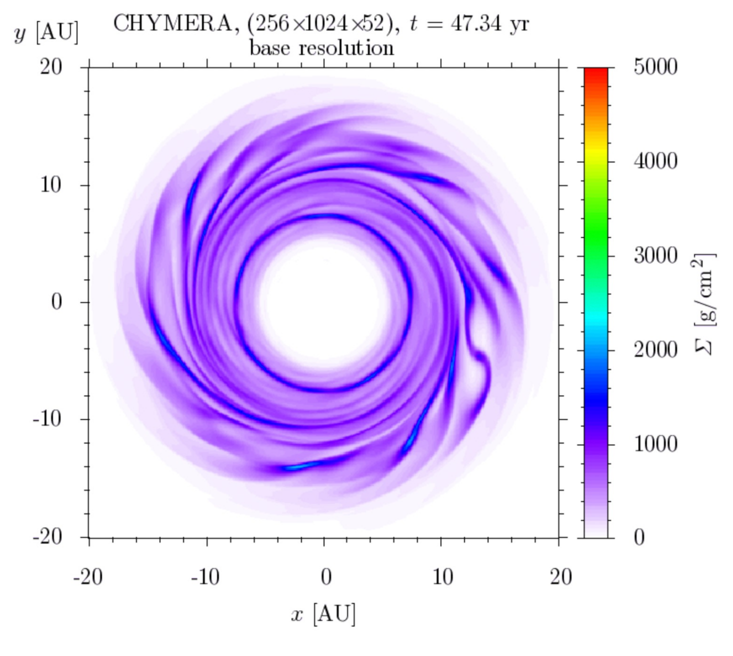


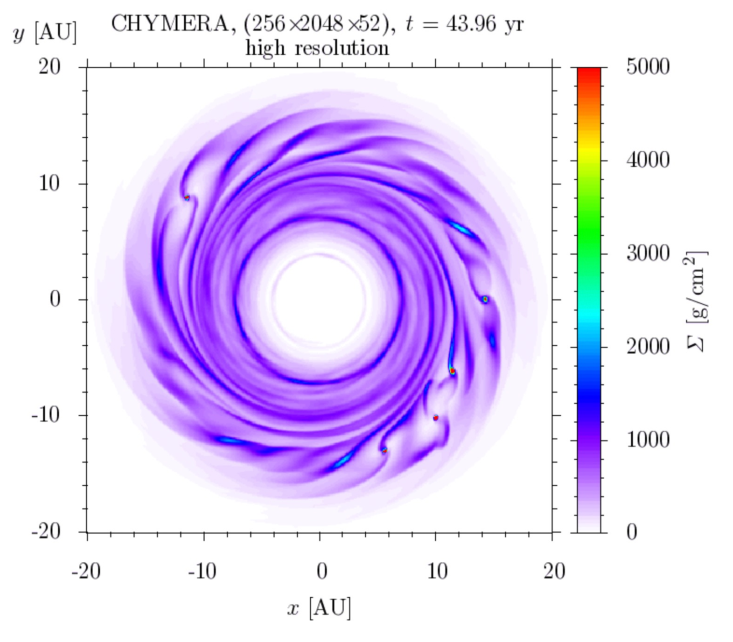


GI simulations: code comparison

The Wengen test: Enzo, Flash, Gasoline, CHYMERA

https://users.camk.edu.pl/gawrysz/test4/#movies

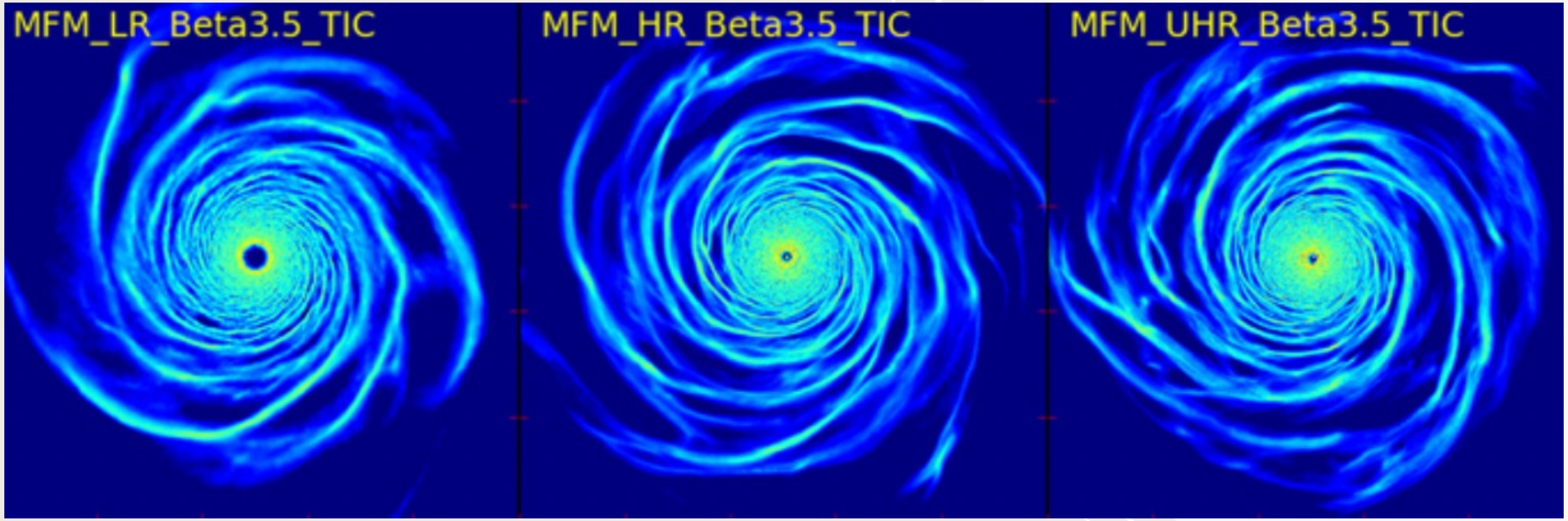




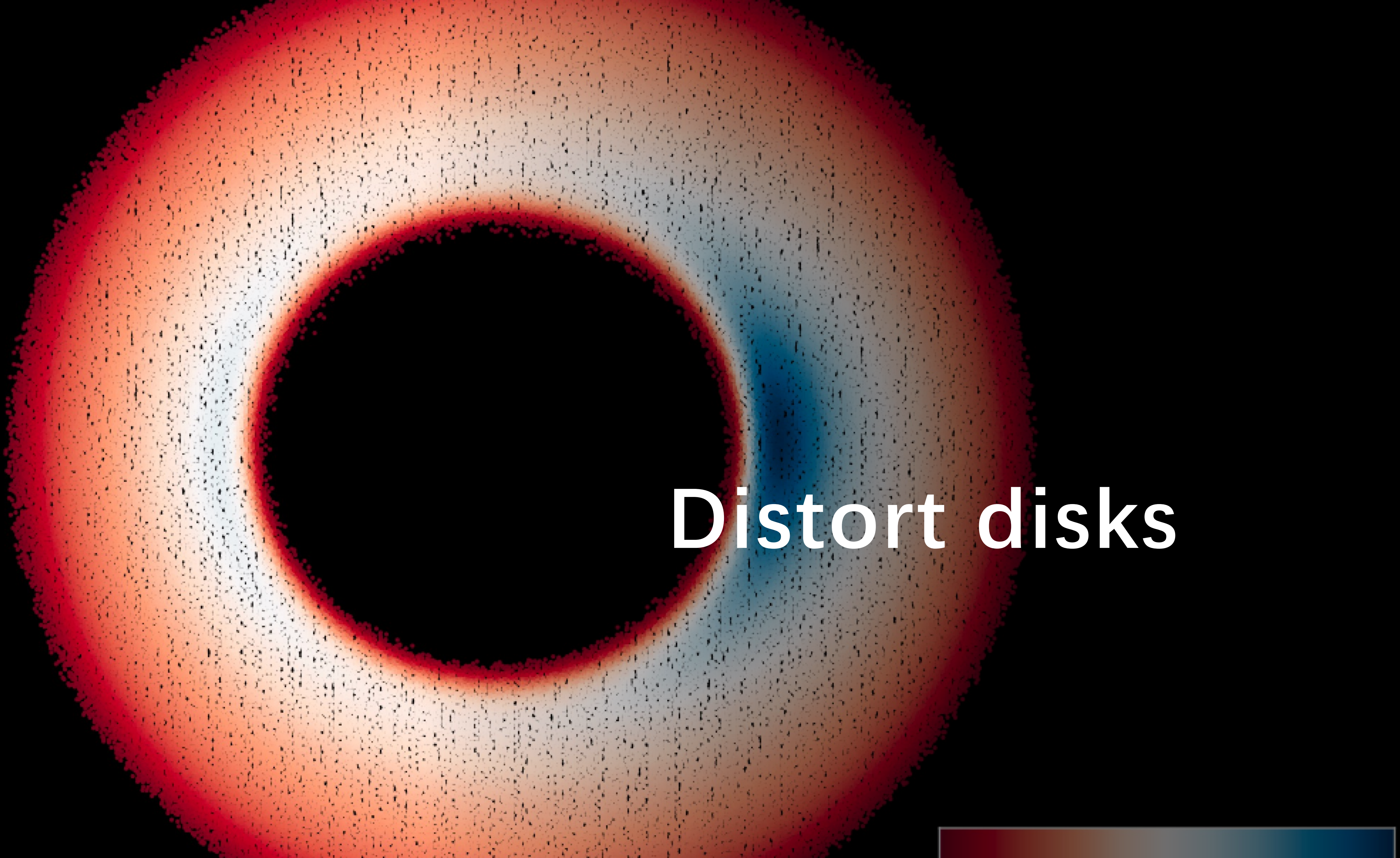
Grid code simulations, sometime too many fragments…

GI simulations with moving mesh

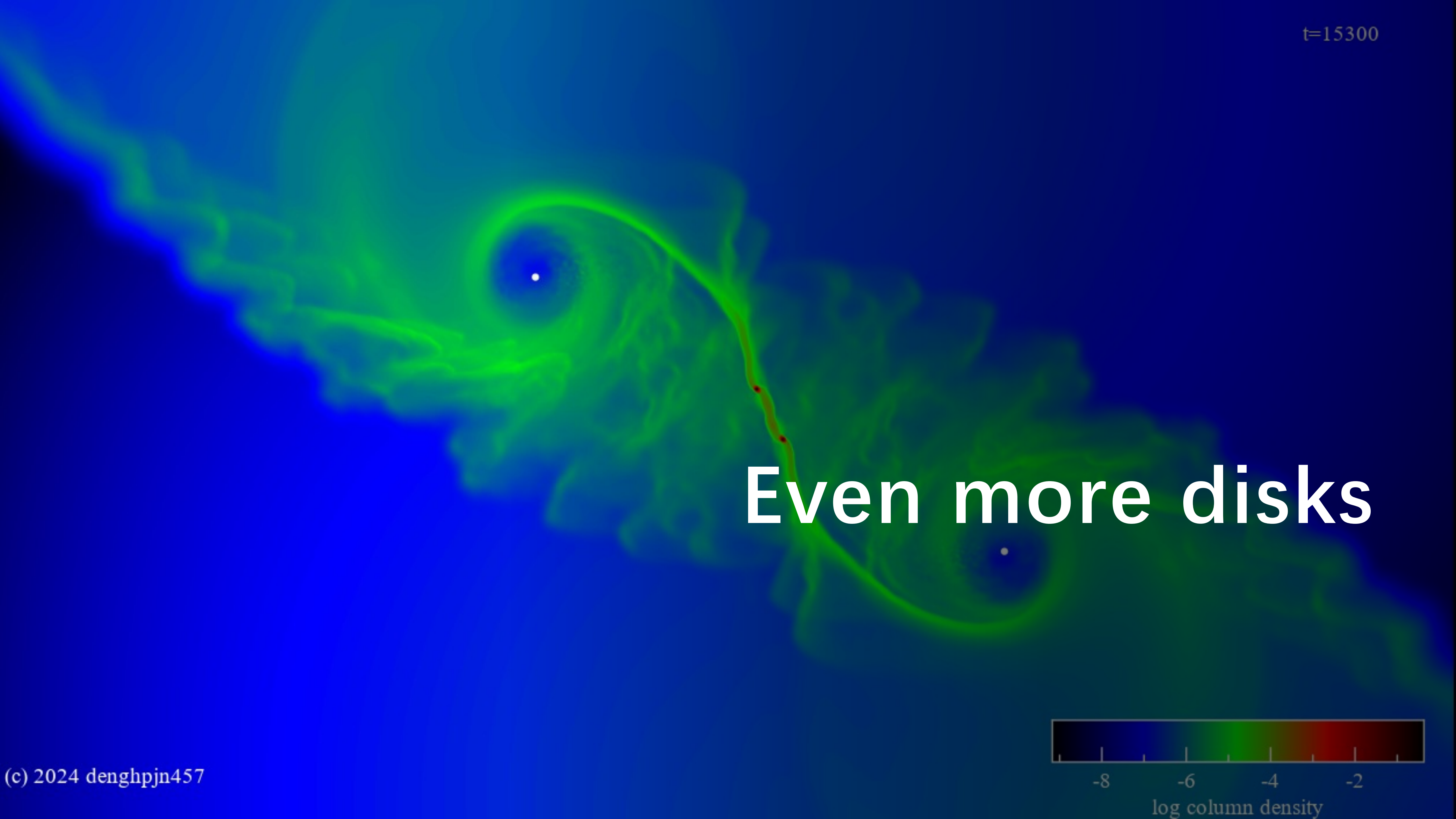
Moving mesh and the meshless method, with less numerical viscosity, should be better than SPH



e.g., the critical cooling for spiral to fragment (*Deng et al 2017*)


Distort disks

Even more disks



Practical issue: initial condition

How to sample a density field with particles?

Invert the probability function or Monte-Carlo sampling (rejection sampling)

The density field as the probability function of the particle distribution

Practical issue: initial condition

Assign other fields according to particles positions

Relax the initial condition by damping unwanted random velocity noise in the sampling process.

Try the GIZMO code (SPH+meshless) and other Lagrangian methods if you like. I am glad to help out.

projects

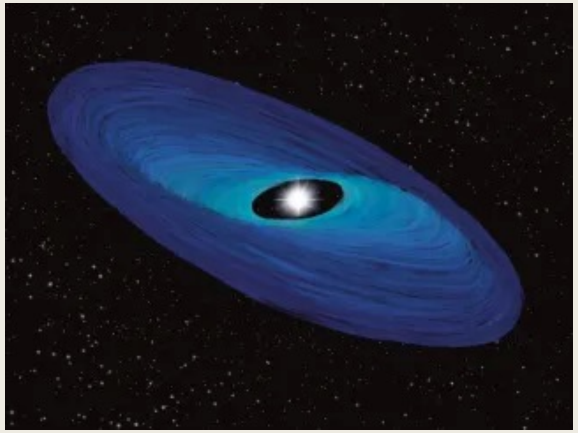
1. Solving for uniformly precessing warp disk structures (cf. Deng & Ogilvie 2022).

Variation of $\langle L \rangle$ with respect to β gives

$$\frac{\mathrm{d}}{\mathrm{d}R}\left[\Sigma_0 H_0 \left(-\frac{\mathrm{d}\bar{L}}{\mathrm{d}\chi}\right) e_1\right] = \frac{\Sigma_0}{R^2}\frac{Q^2 \cos\beta \sin\beta}{(1-e_1^2)^{3/2}} \tag{20}$$

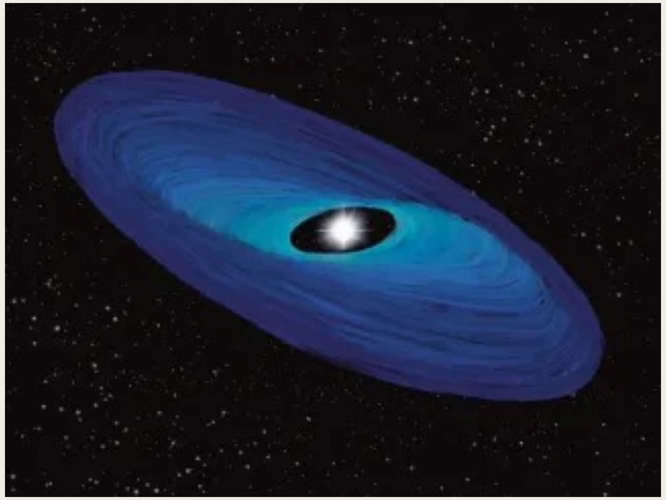
and variation of $\langle L \rangle$ with respect to e_1 gives

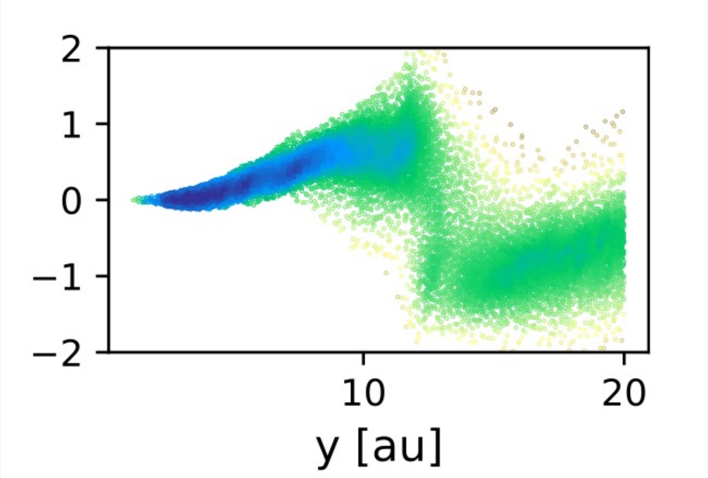
$$R^2 H_0 \left(-\frac{\mathrm{d}\bar{L}}{\mathrm{d}\chi}\right)\frac{\mathrm{d}\beta}{\mathrm{d}R} = \frac{Q^2\left(3\cos^2\beta - 1\right)e_1}{2(1-e_1^2)^{5/2}}. \tag{21}$$



projects

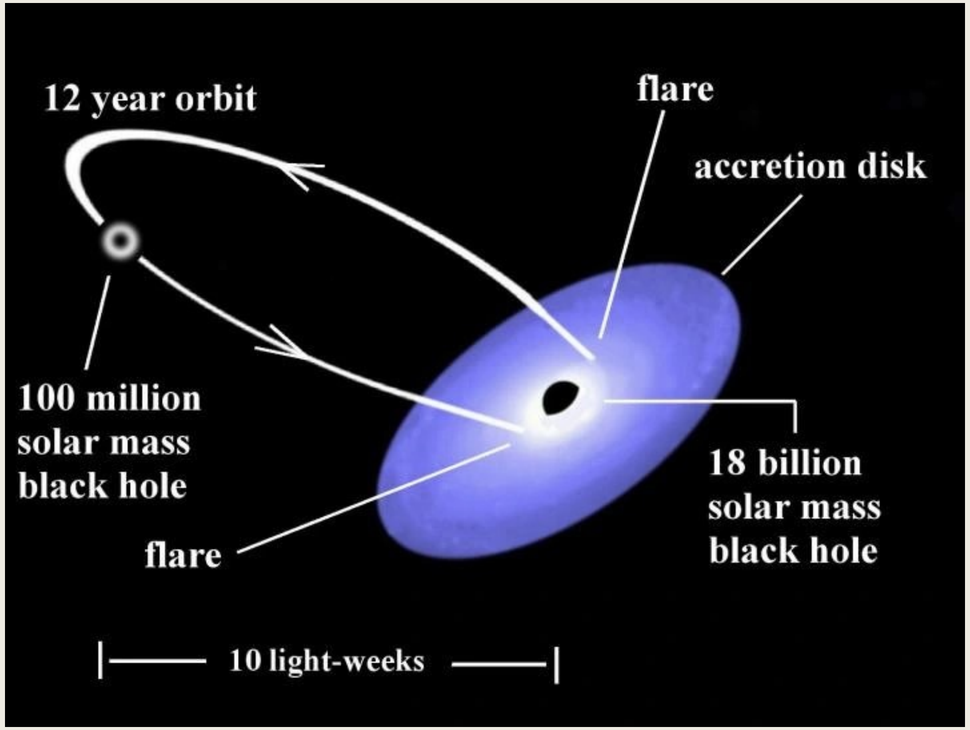
2. Break up a warped disk in hydrodynamic simulations

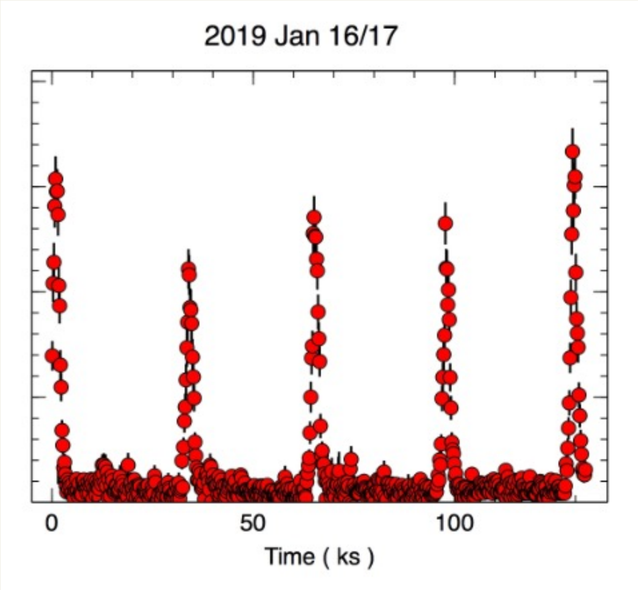




projects

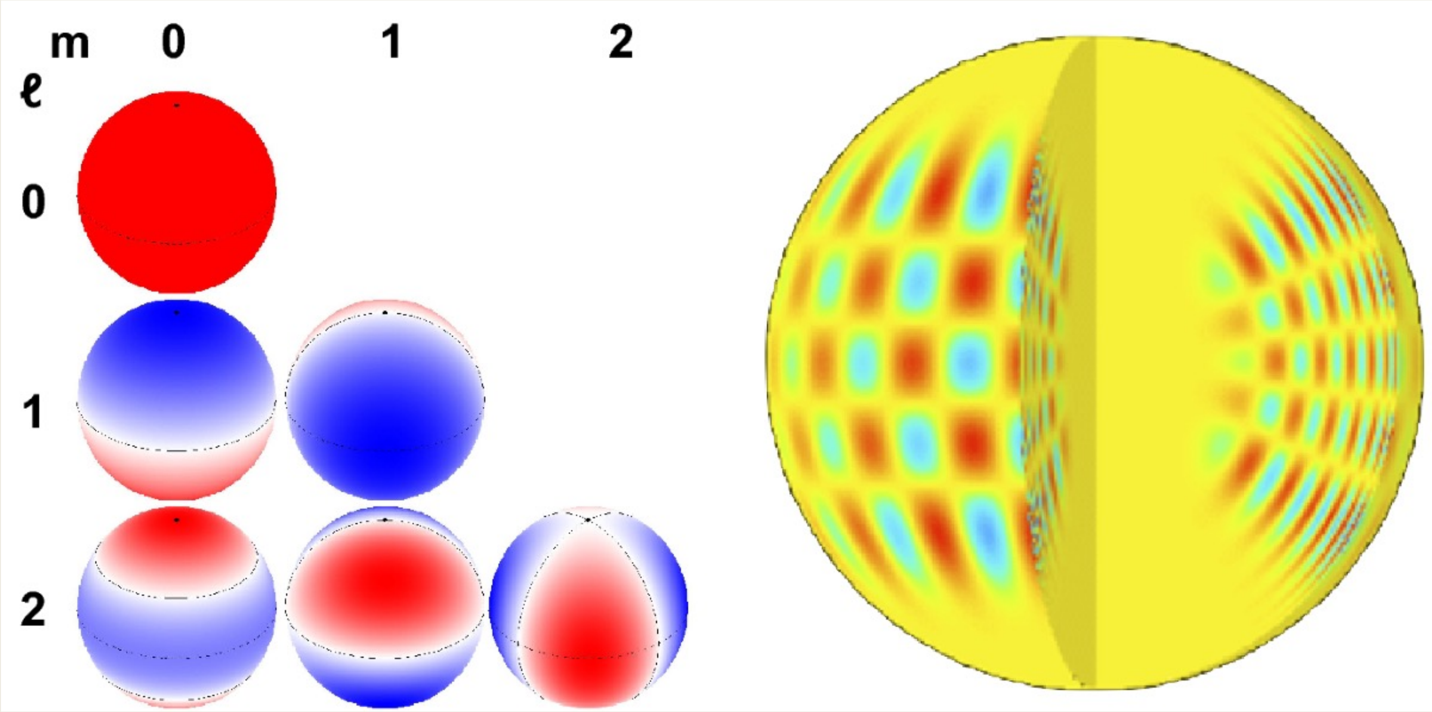
3. Punching a disk and eruptive events





projects

4. Hydrodynamic simulation of oscillating stars to pave the way towards direct simulations of conventions in tidally distorted stars.



projects

5.Applying moving mesh method in the AREPO code to planetary impact by merge our EOS library into the AREPO code.