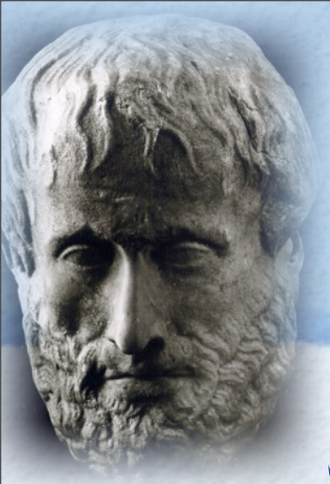
Adaptive Stochastic and Deterministic Simulations Using Particles

Petros Koumoutsakos

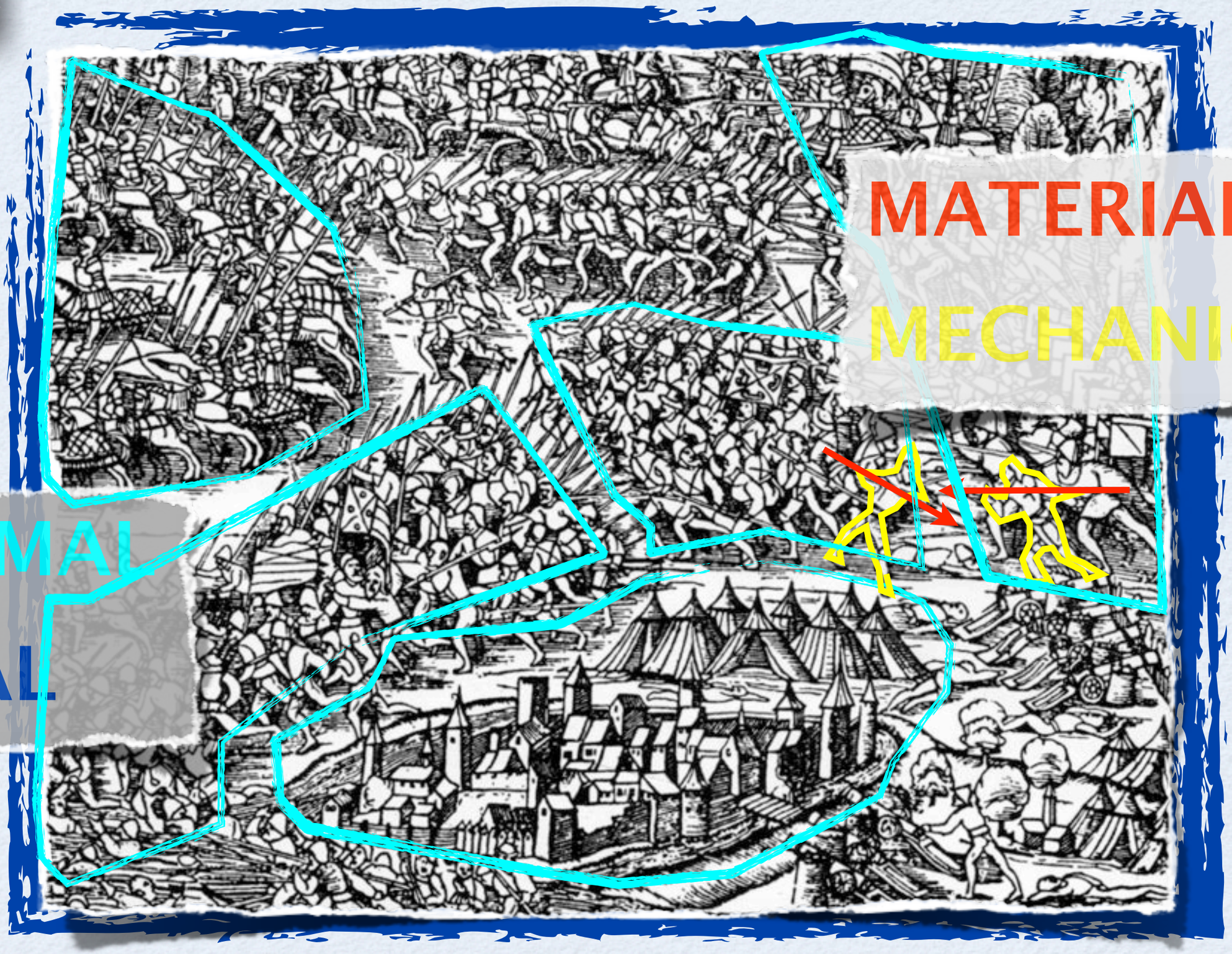
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

www.cse-lab.ethz.ch



MULTISCALING as CAUSALITY

adapted from Ulanowicz



MATERIAL – MECHANICAL

FORMAL

FINAL

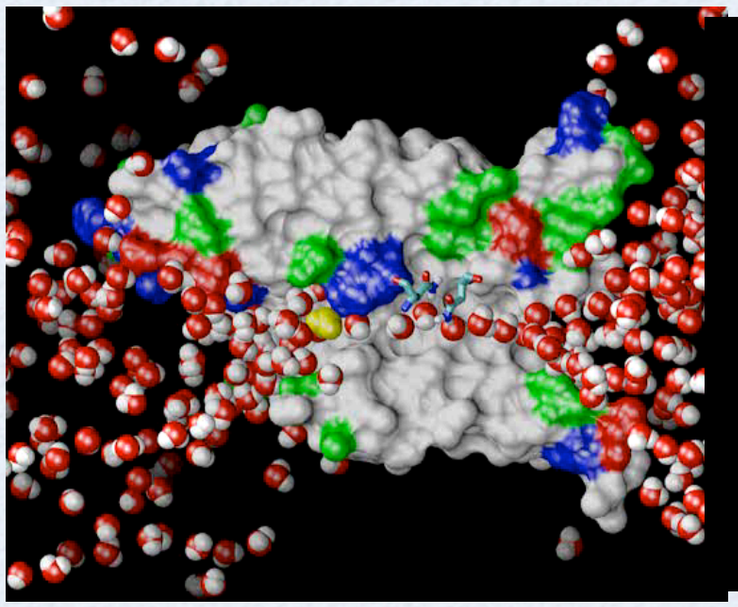
OUTLINE

- INTRODUCTION
- STOCHASTIC
  - R_Leaping + AMR_S
- DETERMINISTIC
  - Particles + Wavelets
- BOUNDARIES
- OUTLOOK

MULTIPLE SCALES + PARTICLES

Molecular Dynamics

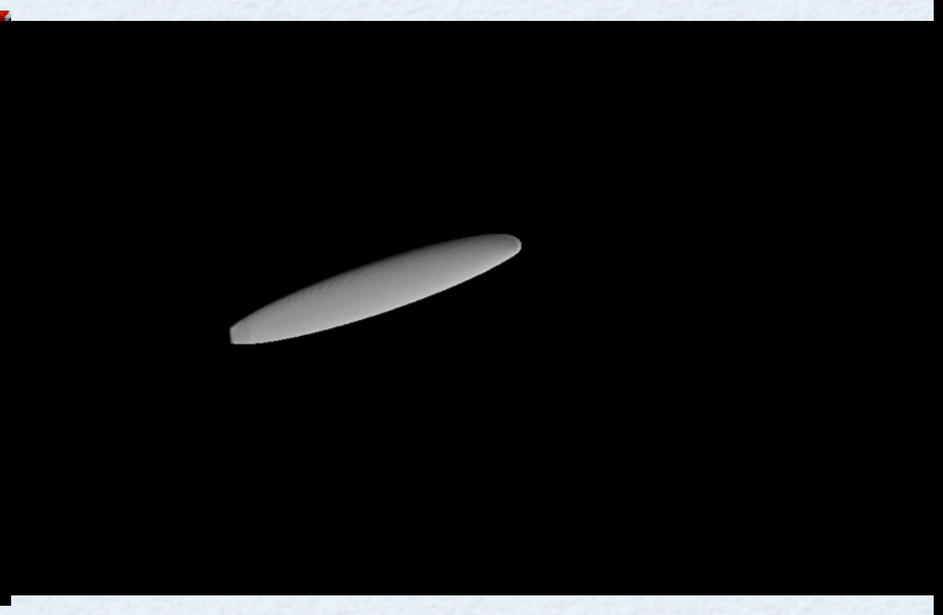
-9



Transport in aquaporins
Schulten Lab, UIUC

Vortex Methods

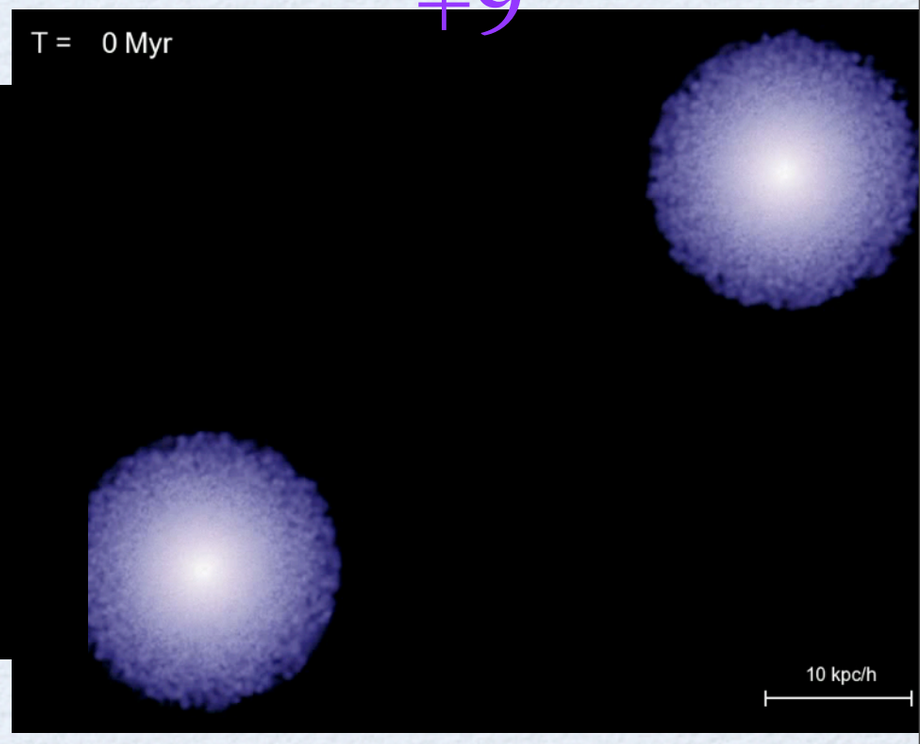
0



Vortex Dynamics
Koumoutsakos Lab, ETHZ

Smoothed Particle Hydrodynamics

+9



Growth of Black Holes
Springel, MPI - Hernquist, Harvard

PARTICLES : Lagrangian, Conservation and Other Laws

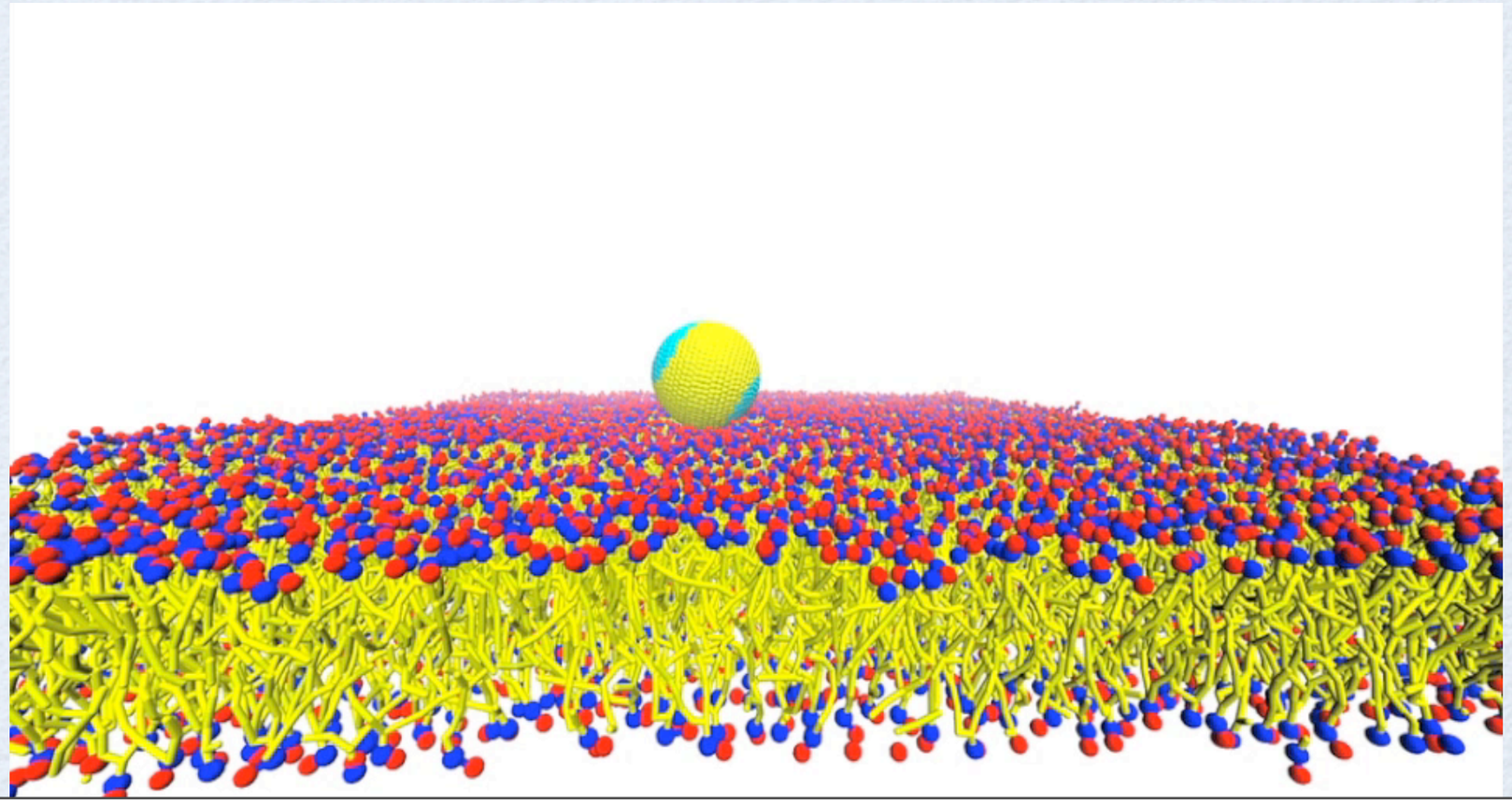
SPH, Vortex Methods

$$\rho_p \frac{D\mathbf{u_p}}{Dt} = (\nabla \cdot \sigma)_p$$

$$\frac{d\mathbf{x_p}}{dt} = \mathbf{u}_p$$

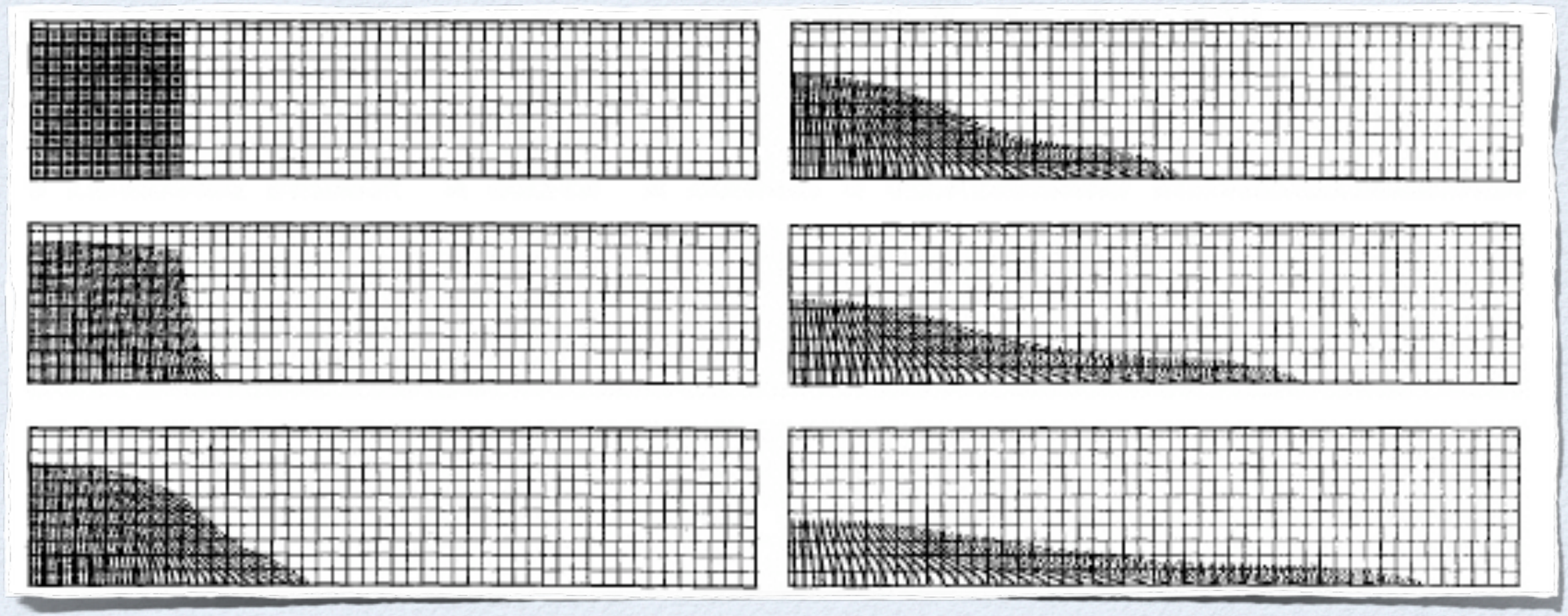
$$m\frac{d\mathbf{u_p}}{dt} = F_p$$

MD, DPD, CGMD

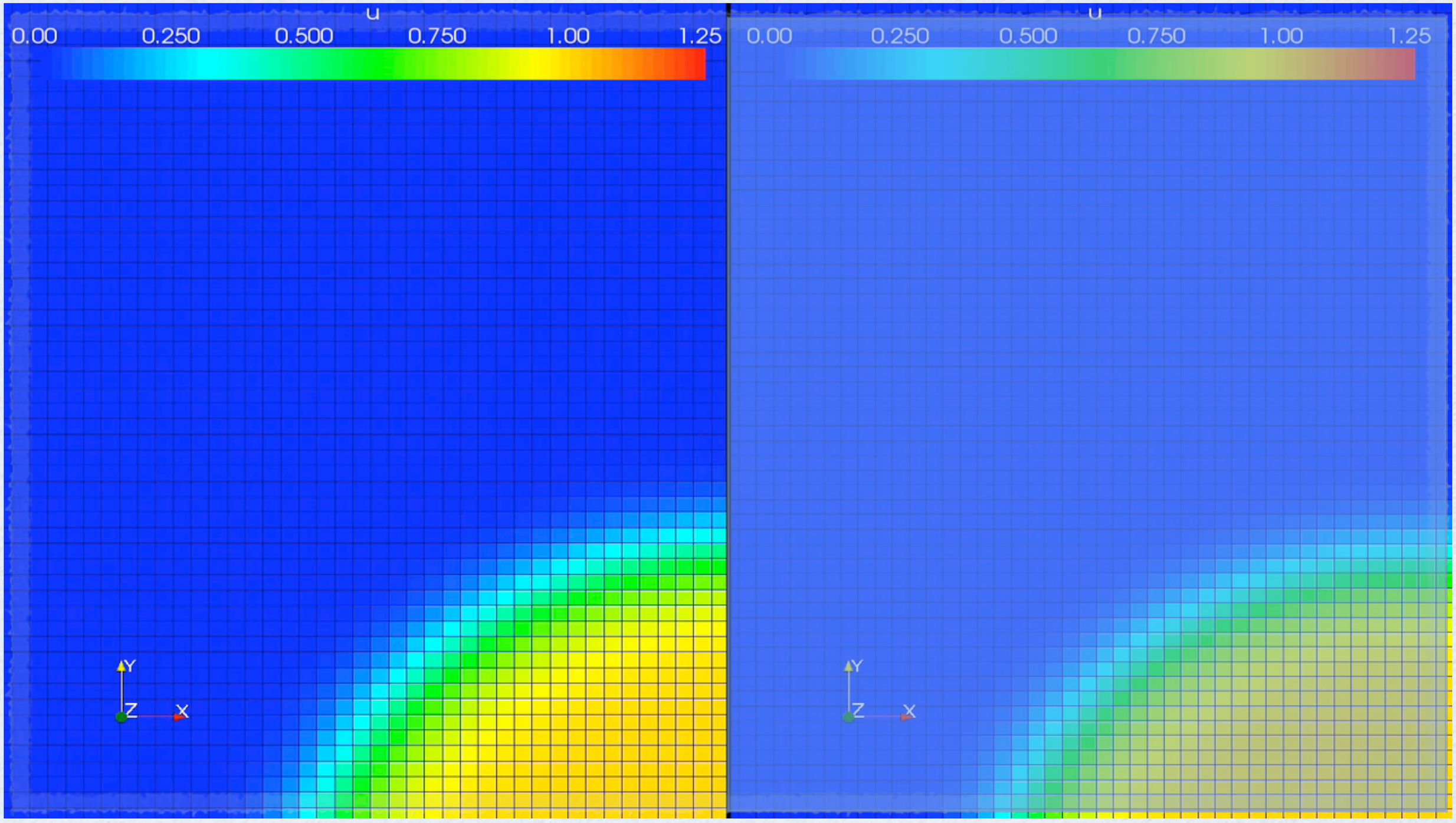


Particles and Grids

Marker and Cell (MAC) - F.H. Harlow and E.J. Welch



Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface,, Harlow, Francis H. and Welch, J. Eddie, Physics of Fluids, 1965

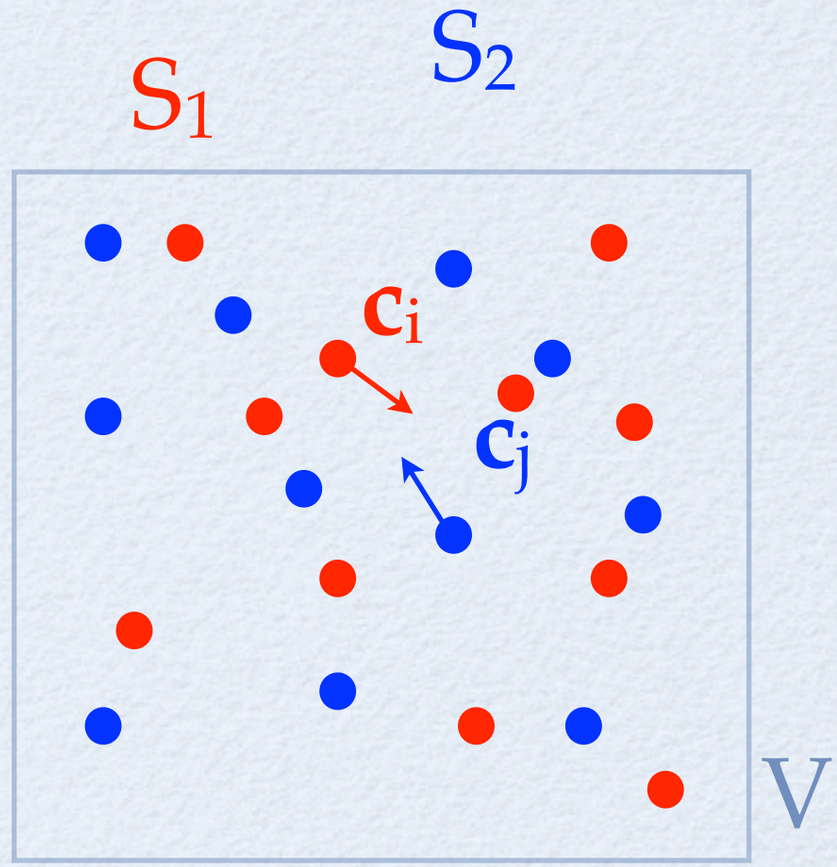


STOCHASTIC

DETERMINISTIC

Chemical kinetics : Set-up

- Well stirred reaction volume V
- Experiment length T
- N different species $S_1, S_2, \ldots, S_N$ in numbers $X_1, X_2, \ldots, X_N$
- random collisions and reactions through M channels $R_1, R_2, \ldots, R_M$



Kinetics + Space

Diffusion in 1-D

Deterministic

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$$

Stochastic

D. Bernstein,. *Phys. Rev. E*, 2005.

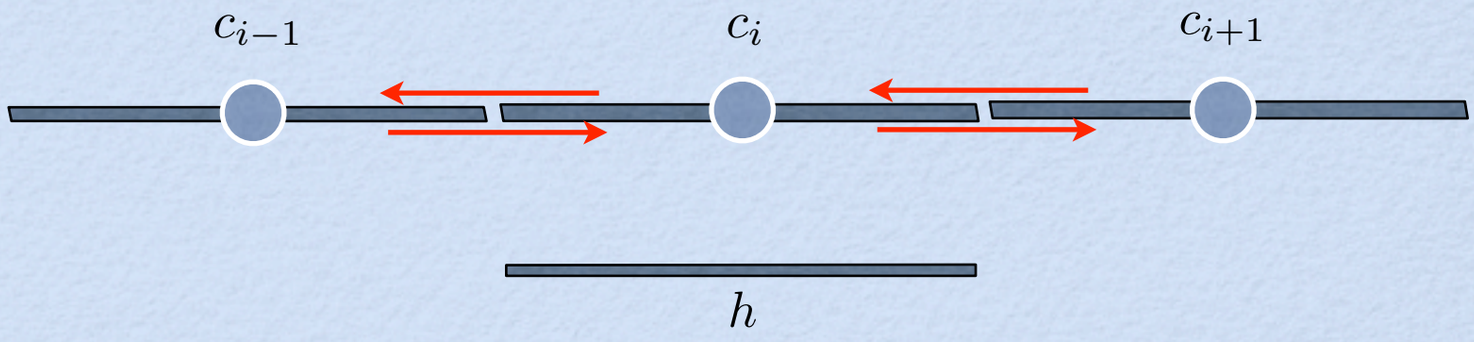
A species U, with elements labeled by an index i

Diffusion as reactions of the form:

$$U_i \xrightarrow{k_{i,i+1}} U_{i+1} \qquad U_{i+1} \xrightarrow{k_{i+1,i}} U_i$$

$$U_i \xrightarrow{k_{i,i-1}} U_{i-1} \qquad U_{i-1} \xrightarrow{k_{i-1,i}} U_i$$

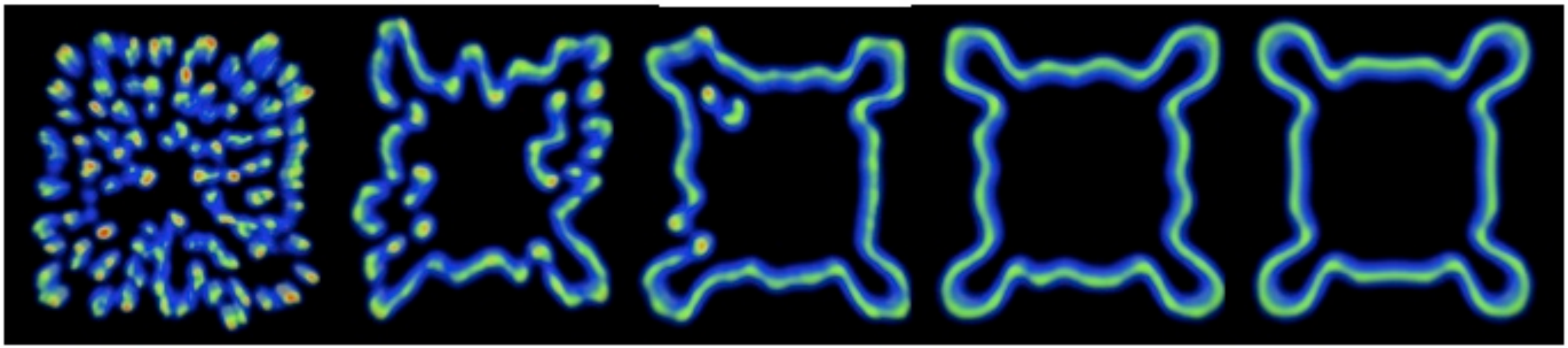
Uniform Cells: $k_{i,j} = \frac{D}{h^2}$



SPACE: 10^6 – 10^9 "Reactions"

Molecules per grid cell for a 300 x 300 grid

500 1000 5000 10000



Microscopic scale Macroscopic scale

$$U + 2V \rightarrow 3V,$$
$$V \rightarrow P$$

$$\frac{\partial u}{\partial t} = d_u \Delta u - uv^2 + F(1-u),$$

$$\frac{\partial v}{\partial t} = d_v \Delta v + uv^2 - (F+\kappa)v.$$

$F = 0.04, \kappa = 0.06, t = 1000$

Stochastic Simulation Algorithm

Gillespie,
J. Comp. Phys. 1977

- For M reactions, time until **any** reaction

$$\tau \sim \mathcal{E}(1/a_0) \qquad a_0 = \sum_{j=1}^{M} a_j$$

- Reaction **index** : point-wise distribution

$$p(j = l) = \frac{a_l}{a_0}$$

ONE STEP

- Sample τ
- Sample the index j
- Update the X_i, t=t+τ

The SSA simulates every reaction event !

Exact but SLOW

ACCELERATING SSA : τ leaping

Gillespie,
J. Chem. Phys. 2001

τ leaping : several reaction events over one time step

ASSUMPTION : reaction propensities a_i remain essentially constant over τ, in spite of several firings

- Over this given τ, the number of reaction firings K^P_j is governed by a Poisson distribution

$$K_j^{\mathcal{P}} \sim \mathcal{P}(a_j \tau)$$
$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum_{j=1}^{M} K_j^{\mathcal{P}} \boldsymbol{\nu}_j.$$

Cost ~ M Poisson samplings

SPEEDUP ~ 100 X SSA

τ-leaping : Consequences

τ leaping can generate negative populations

- Binomial τ leaping : Approximate the unbounded Poisson distributions with Binomial ones

Chatterjee et al., J. Chem. Phys. 2005

- Modified τ leaping

Tian & Burrage, J. Chem. Phys. 2004

  - Critical reactions, i.e. those likely to drive some populations negative, handled by SSA
  - Other reactions advanced by τ leaping

Cao et al., J. Chem. Phys. 2005

R-leaping : Accelerate SSA by reaction leaps

Leaps : number of firings L across all reaction channels

- Time increment τ_L is Gamma-distributed $\tau_L \sim \Gamma(L, 1/a_0(\mathbf{x}))$
- In this interval we will have K_m firings of channel R_m
- with : $\sum_{m=1}^{M} K_m = L$
- In R-leaping, as in SSA, the index j of every firing obeys a point-wise distribution

$$P(j = l) = \frac{a_l(\mathbf{x})}{a_0(\mathbf{x})} \quad \text{for } l = 1, \ldots, M.$$

Auger et al.,, J. Chem. Phys. 2006

R-leaping : One step

- Define L

$$\tau_L \sim \Gamma(L, 1/a_0(\mathbf{x}))$$

- Sample the index j

$$P(j = l) = \frac{a_l(\mathbf{x})}{a_0(\mathbf{x})} \quad \text{for } l = 1, \ldots, L.$$

- Number of reactions for channel m

$$K_m = \sum_{l=1}^{L} \delta_{l,m}$$

- Update species and time :

$$\mathbf{X}(t + \tau_L) = \mathbf{X}(t) + \sum_{j=1}^{M} K_j \boldsymbol{\nu}_j$$

Auger et al.,
J. Chem. Phys. 2006

R-Leaping Theorem

The distribution of K_1 is a binomial distribution :

$$\mathcal{B}(L, a_1(\mathbf{x})/a_0(\mathbf{x}))$$

and for every $m \in \{2, \ldots, M\}$ the conditional distribution of K_m

given the event $\{(K_1, \ldots, K_{m-1}) = (k_1, \ldots, k_{m-1})\}$ is

$$K_m \sim \mathcal{B}\left(L - \sum_{i=1}^{m-1} k_i, \frac{a_m(\mathbf{x})}{a_0(\mathbf{x}) - \sum_{i=1}^{m-1} a_i(\mathbf{x})}\right) .$$

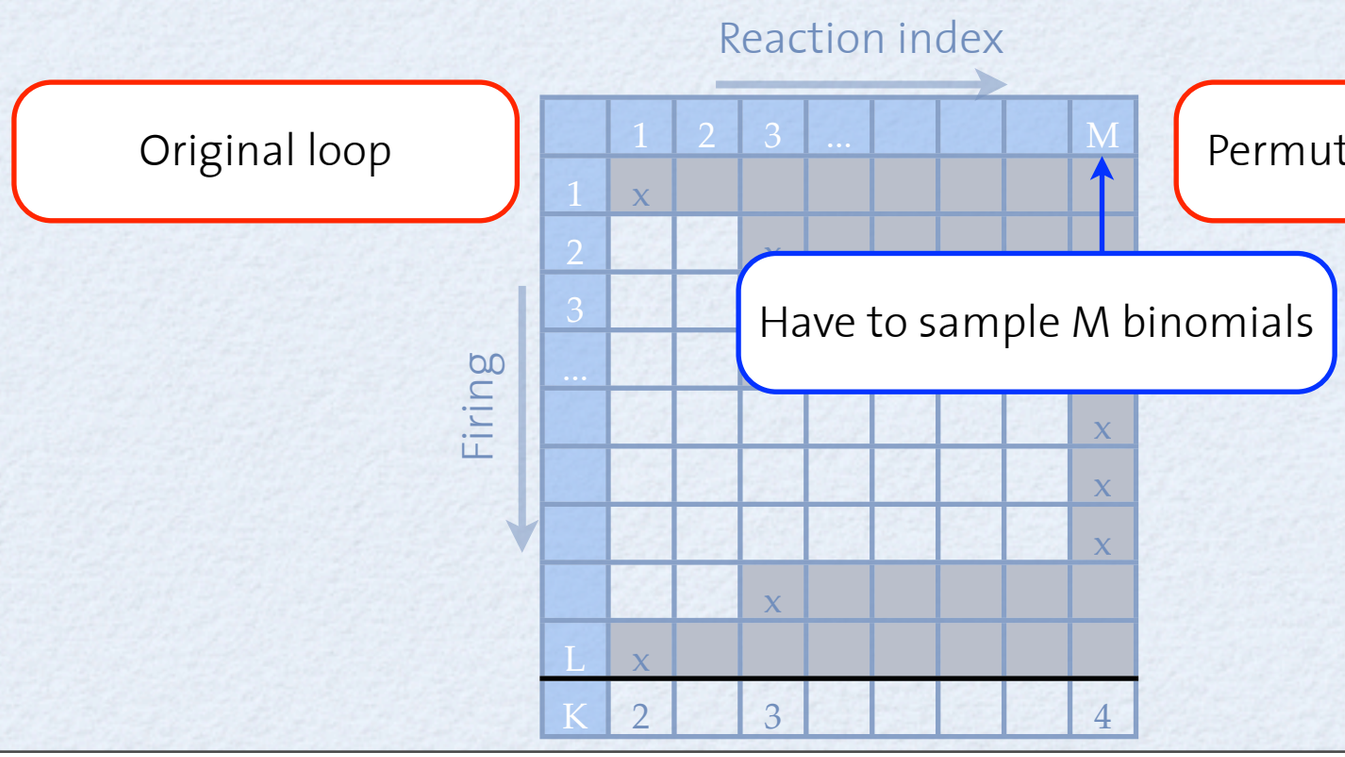
This result is invariant under any permutation of the indices

R-leaping : Sorting Reactions

- ***Sampling the M K_j efficiently***

$$\text{If } K_i = k_i,\ \forall i < m,\ K_m \sim \mathcal{B}\left(L - \sum_{i=1}^{m-1} k_i,\ \frac{a_m}{a_0 - \sum_{i=1}^{m-1} a_i}\right)$$

  - When $\sum_{i=1}^{m-1} k_i = L$, sampling is done!
  - Minimize the average m by a **permutation of the indices**, such that propensities are decreasing



Results

- LacZ/LacY genes expression and enzymatic/transport activities of LacZ/LacY proteins in E. Coli

Kierzek, Bioiformatics 2002

  - Moderately large system (M = 22)
  - Disparate rates
  - Scarce reactants and negative species

	Reaction Channel	Reaction rate
R_1	PLac + RNAP → PLacRNAP	0.17
R_2	PLacRNAP → PLac + RNAP	10
R_3	PLacRNAP → TrLacZ1	1
R_4	TrLacZ1 → RbsLacZ + PLac + TrLacZ2	1
R_5	TrLacZ2 → TrLacY2	0.015
R_6	TrLacY1 → RbsLacY + TrLacY2	1
R_7	TrLacY2 → RNAP	0.36
R_8	Ribosome + RbsLacZ → RbsRibosomeLacZ	0.17
R_9	Ribosome + RbsLacY → RbsRibosomeLacY	0.17
R_{10}	RbsRibosomeLacZ → Ribosome + RbsLacZ	0.45
R_{11}	RbsRibosomeLacY → Ribosome + RbsLacY	0.45
R_{12}	RbsRibosomeLacZ → TrRbsLacZ + RbsLacZ	0.4
R_{13}	RbsRibosomeLacY → TrRbsLacY + RbsLacY	0.4
R_{14}	TrRbsLacZ → LacZ	0.015
R_{15}	TrRbsLacY → LacY	0.036
R_{16}	LacZ → dgrLacZ	$6.42\text{x}10^{-5}$
R_{17}	LacY → dgrLacY	$6.42\text{x}10^{-5}$
R_{18}	RbsLacZ → dgrRbsLacZ	0.3
R_{19}	RbsLacY → dgrRbsLacY	0.3
R_{20}	LacZ + lactose → LacZlactose	$9.52\text{x}10^{-5}$
R_{21}	LacZlactose → product + LacZ	431
R_{22}	LacY → lactose + LacY	14

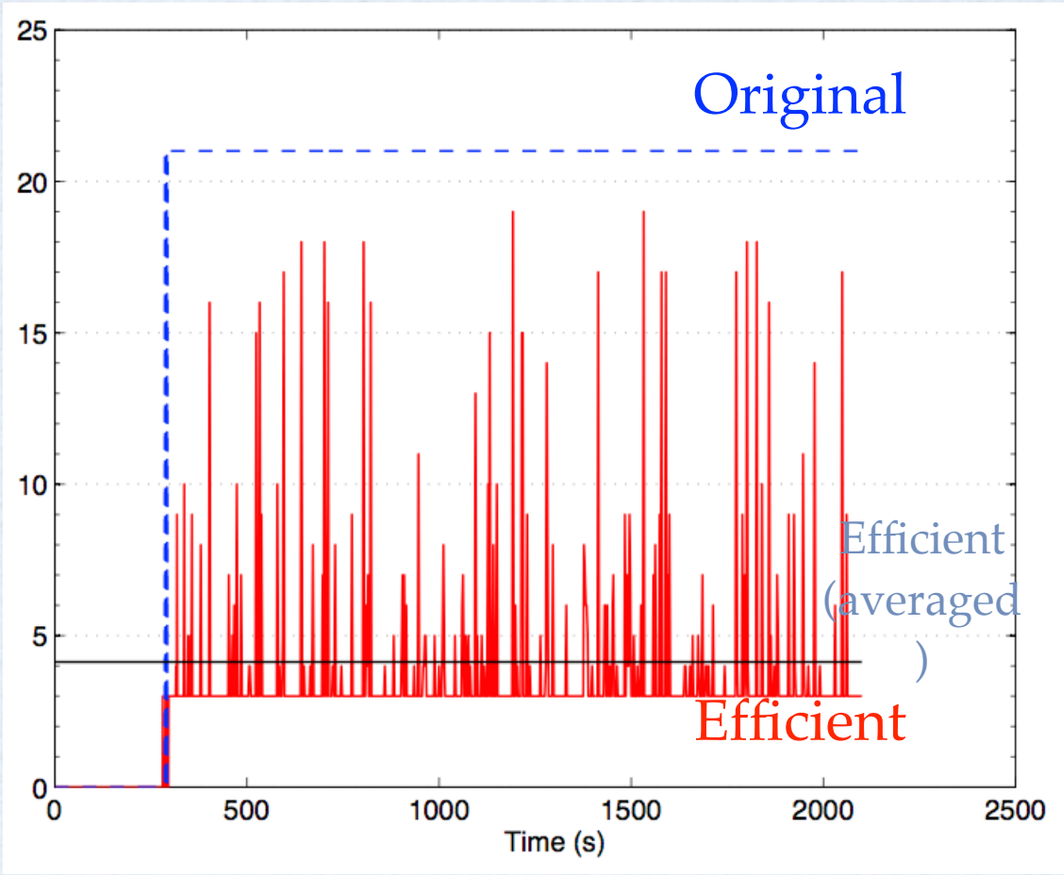
R-leaping : Sampling the M K_j efficiently

- **M** can be large (~10^2) for bio-chemical systems!
- M can be very large **(~10^6)** for diffusion
- Efficient sampling effectively loops over a fraction of **M**.

The larger the system, the bigger the payoff.

The more disparate the reaction rates are, the smaller the fraction.

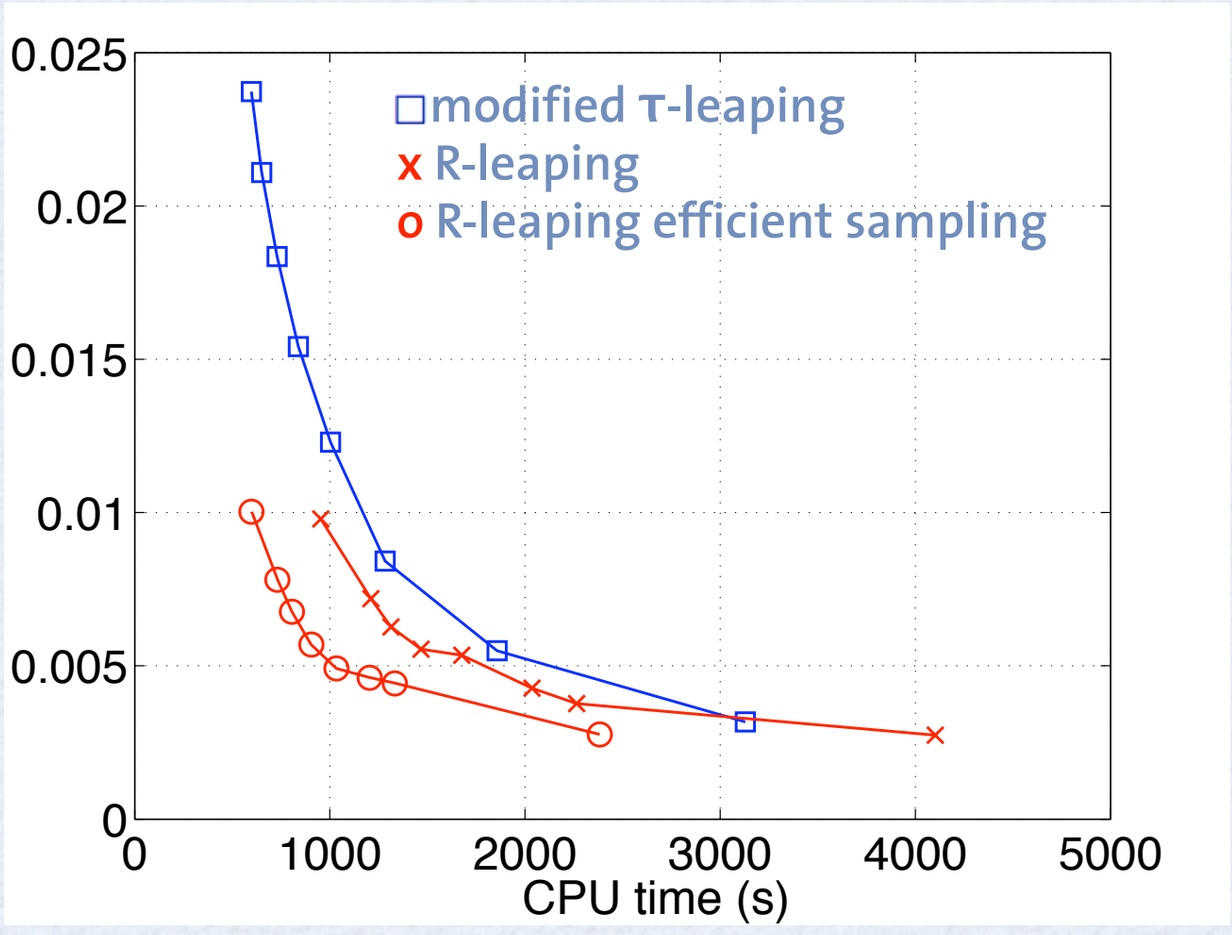
Price to pay: carry out re-ordering often enough

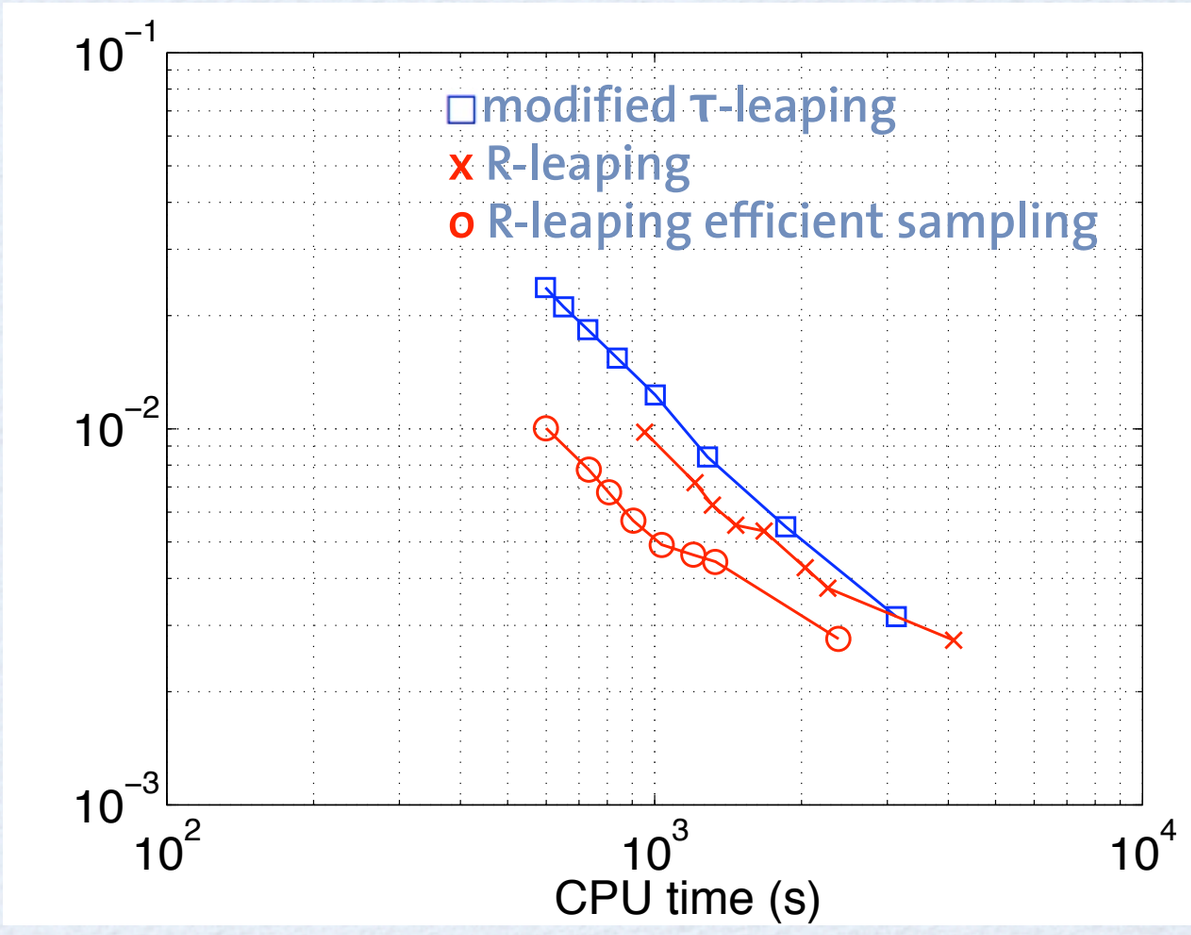


Number of binomial samples per time step
LacYLacZ activities in E. Coli., M=22

Histogram errors vs CPU time

- LacZ/LacY genes expression and enzymatic/transport activities of LacZ/LacY proteins in E. Coli





- $M = 22$: R-leaping **2X** faster than modified τ-leaping!

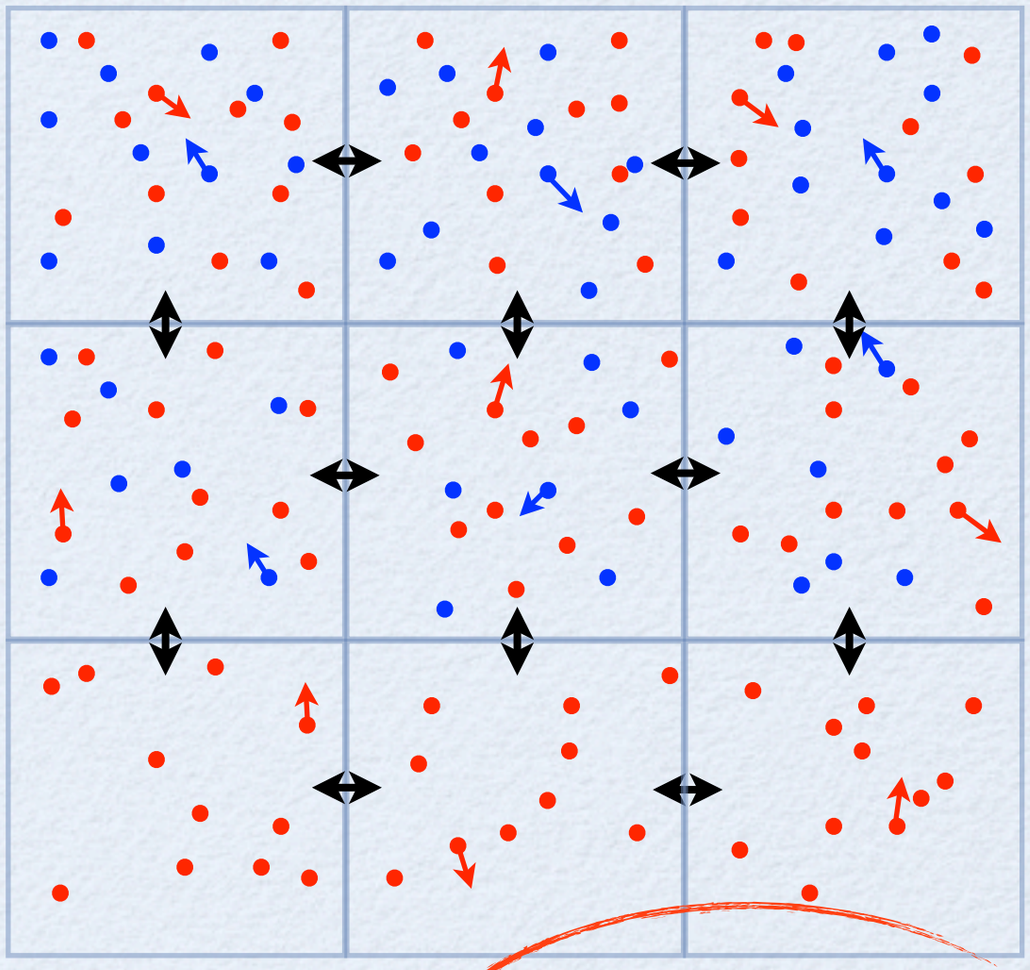
Reaction-Diffusion : SSA + AMR

- Collisions and reactions **within each** element

- For homogeneity :

Kuramato,Prog. Theor. Phys. 1974

$$\frac{\tau_R}{\tau_D} \gg 1$$



- τ_R mean free time for reactive collisions in a cell
- τ_D mean time during which a molecule will remain in element.

- For a bimolecular reaction with rate k and diffusion coefficient D

$$\frac{\hat{\tau}_R}{\hat{\tau}_D} = \frac{D}{h^2 k}$$

***h* must be small for the discretization to be valid**

AMR + STOCHASTIC

i. PROPENSITIES FROM FV

ii. GRID REFINEMENT

iii. STOCHASTIC INTERPOLATION

iv. INTEGRATE: t/R Leaping

v. DATA STRUCTURES : OVERTURE

B. Bayati, et.al. , Phys. Chem. Chem. Phys., 2008

Bayati B., et.al., J. of Computational Physics, 2011

I. Propensities from FV schemes

D. Bernstein. Simulating mesoscopic reaction-diffusion systems using the Gillespie algorithm. *Phys. Rev. E*, 2005.

$\mathbf{J}\left(x, y+\frac{2}{3}\delta y\right)$ approximated by three Taylor series:

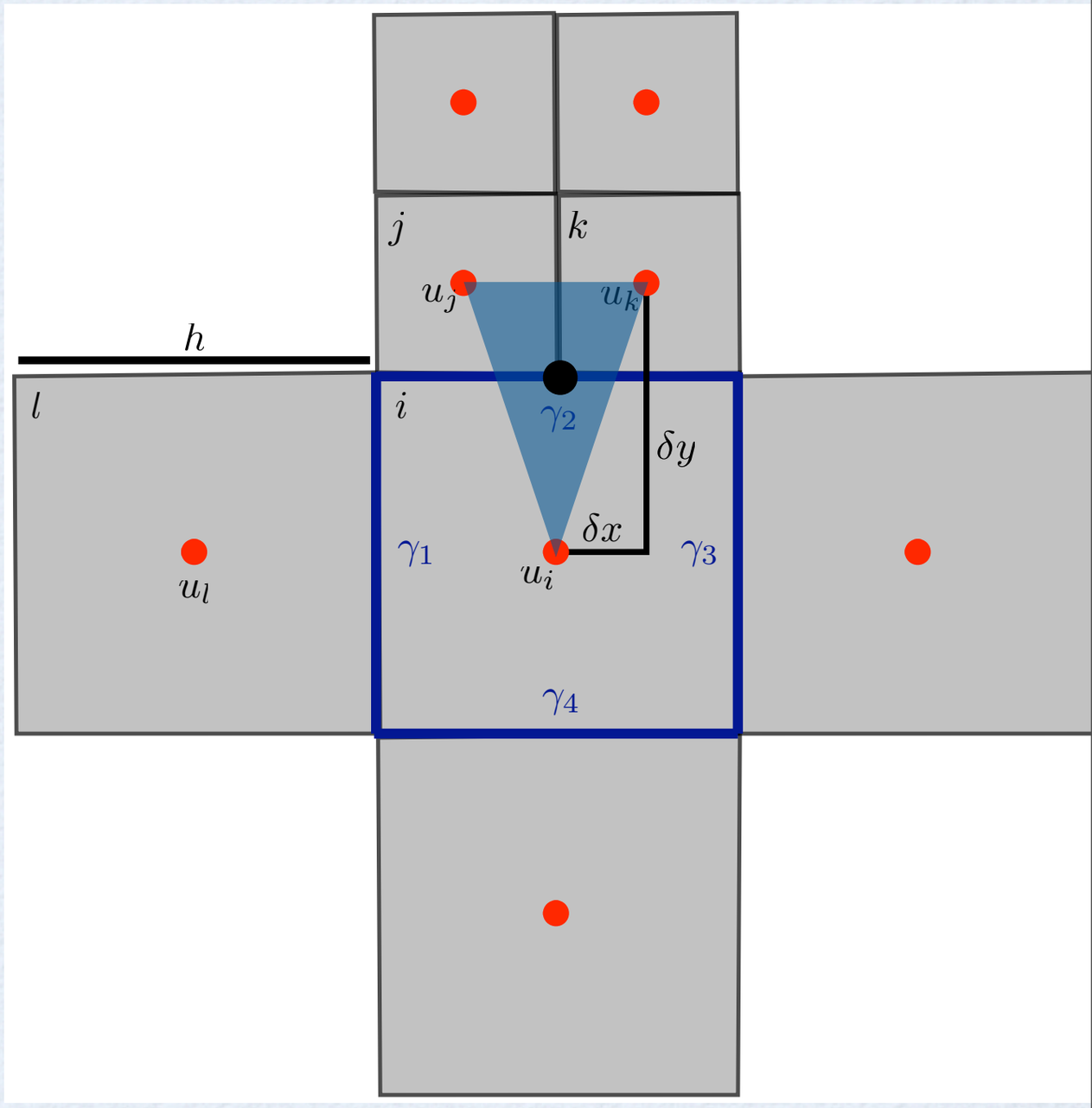
around the point $\left(x, y+\frac{2}{3}\delta y\right)$

$$\mathbf{J}\left(x, y+\frac{2}{3}\delta y\right) = -\frac{D}{2\delta y}\left(\bar{u}_j^{(s)} + \bar{u}_k^{(s)} - 2\bar{u}_i^{(s)}\right) + \mathcal{O}(h)$$

$$\frac{\mathrm{d}U_i^{(s)}}{\mathrm{d}t} = \frac{4D}{3h^2}\left(2(U_j^{(s)} + U_k^{(s)}) - U_i^{(s)}\right) + \mathcal{O}(h).$$

propensities for transitions between j & i

$$\begin{aligned} a_{i,j}^D &= \frac{4D}{3h^2}(U_j^{(s)} + U_k^{(s)}), \\ a_{j,i}^D &= \frac{2D}{3h^2}U_i^{(s)} \end{aligned}$$



II. Refinement Criteria

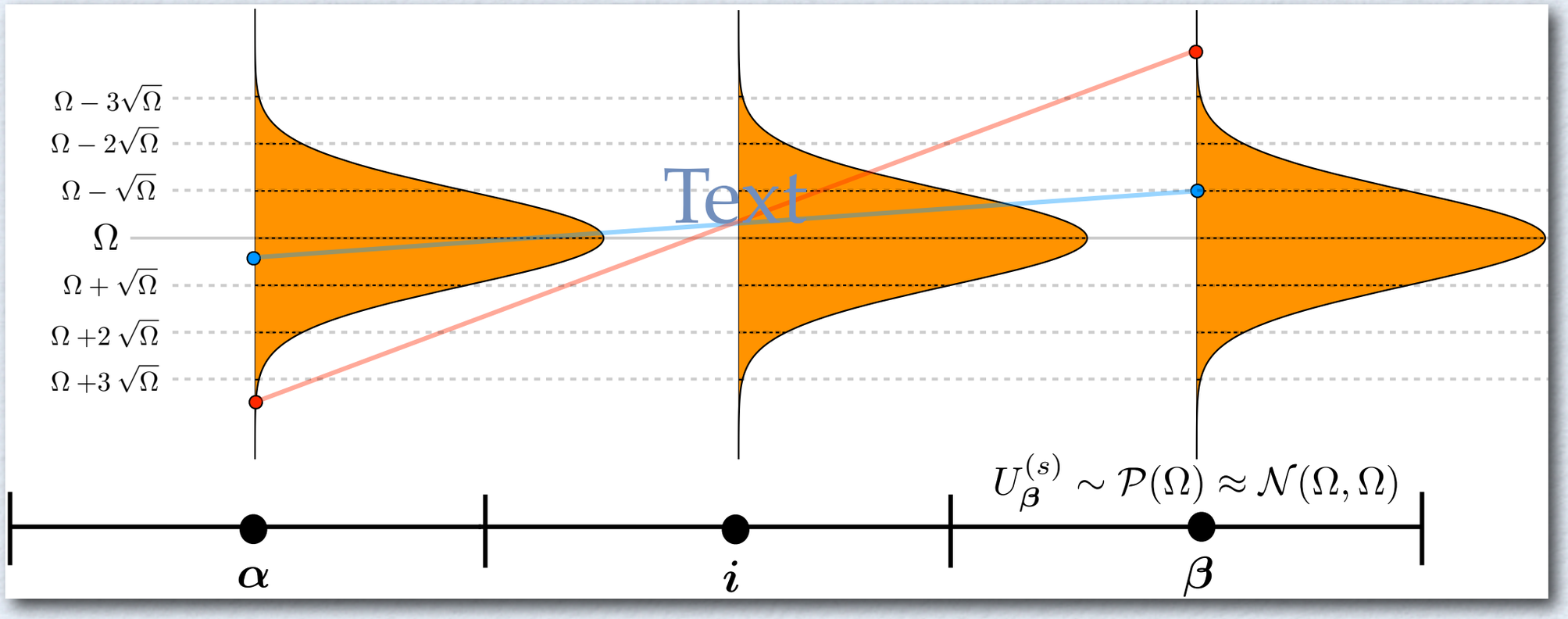
Bell et al., J. Comp. Phys., 2007

Distinguish between gradients and fluctuations,

refined

not refined, *probably background noise*

In equilibrium:



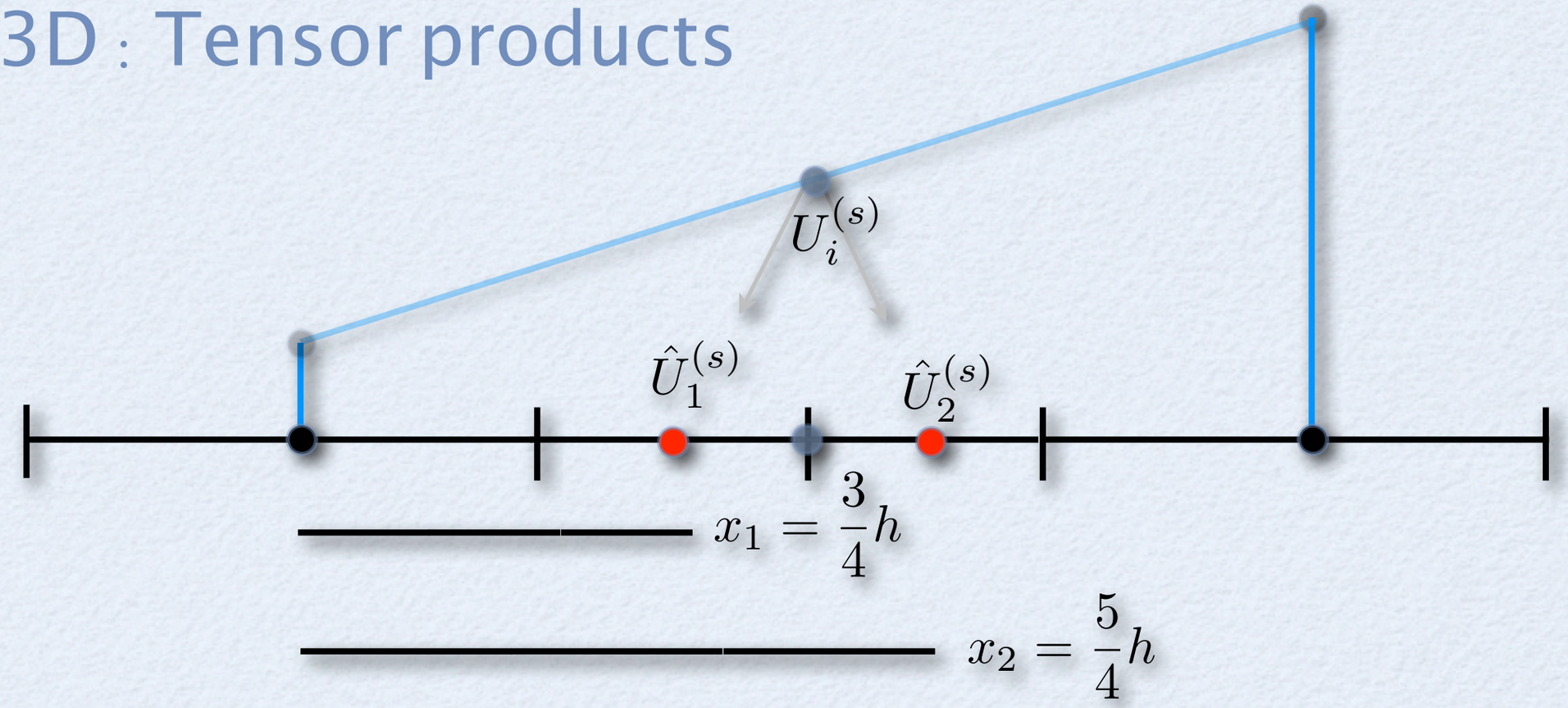
Refine volume element i if :

$$|U_{i+1}^{(s)} - U_{i-1}^{(s)}| > 2C\sqrt{\Omega}$$

III. Stochastic Interpolation

Conservative + Strictly Positive

2D, 3D : Tensor products

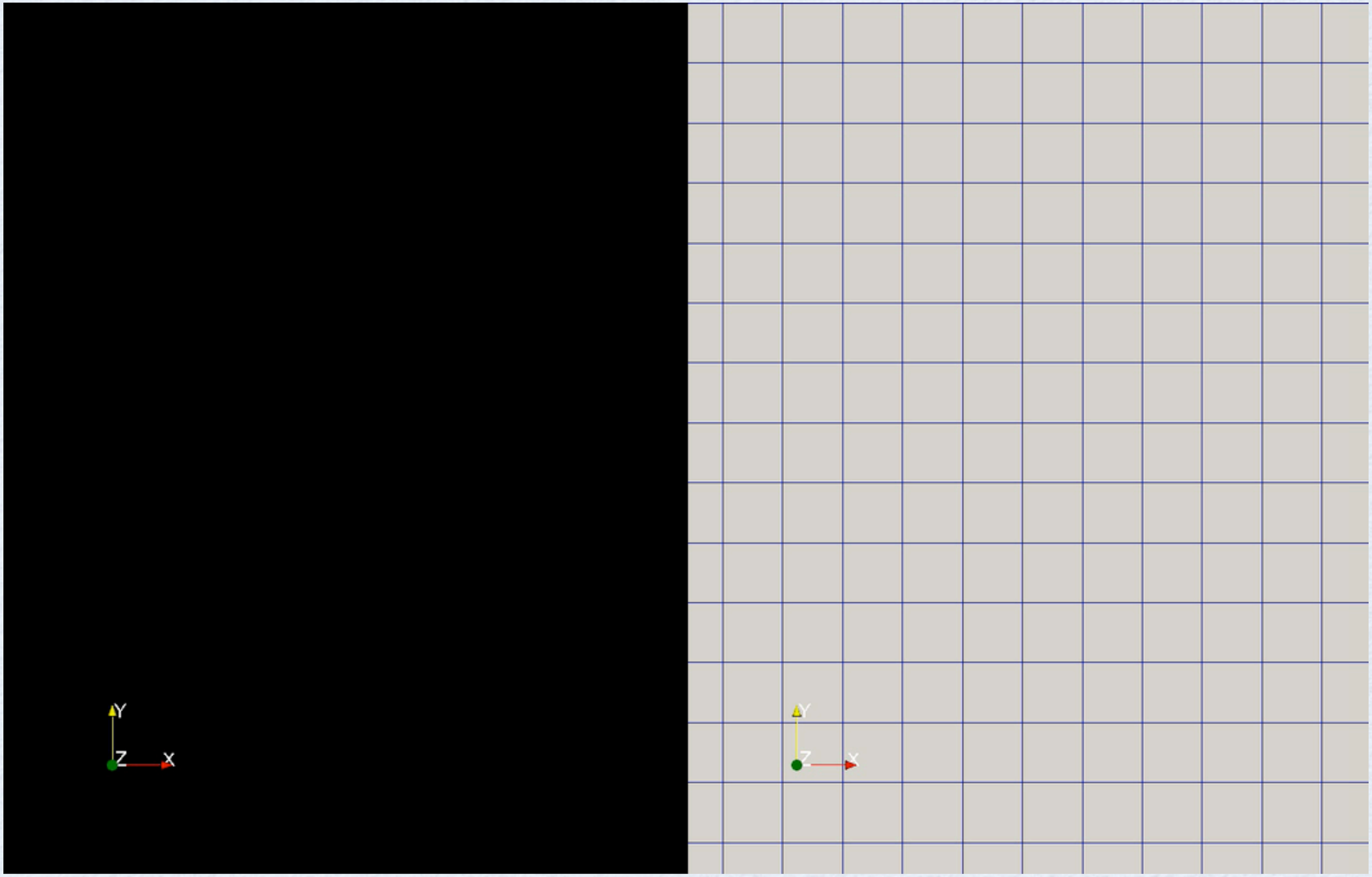


weights: $w_1 := \dfrac{mx_1 + a}{m(x_1 + x_2) + 2a}$ $\qquad$ $w_2 := \dfrac{mx_2 + a}{m(x_1 + x_2) + 2a}$

sampling: $\hat{U}_1^{(s)} \sim \mathcal{B}(U_i^{(s)}, w_1)$ $\qquad$ $\hat{U}_2^{(s)} = U_i^{(s)} - \hat{U}_1^{(s)}$

Fisher-Kolmogorov Reaction-Diffusion System in 2-D

$$U + V \xrightarrow{k} 2U$$

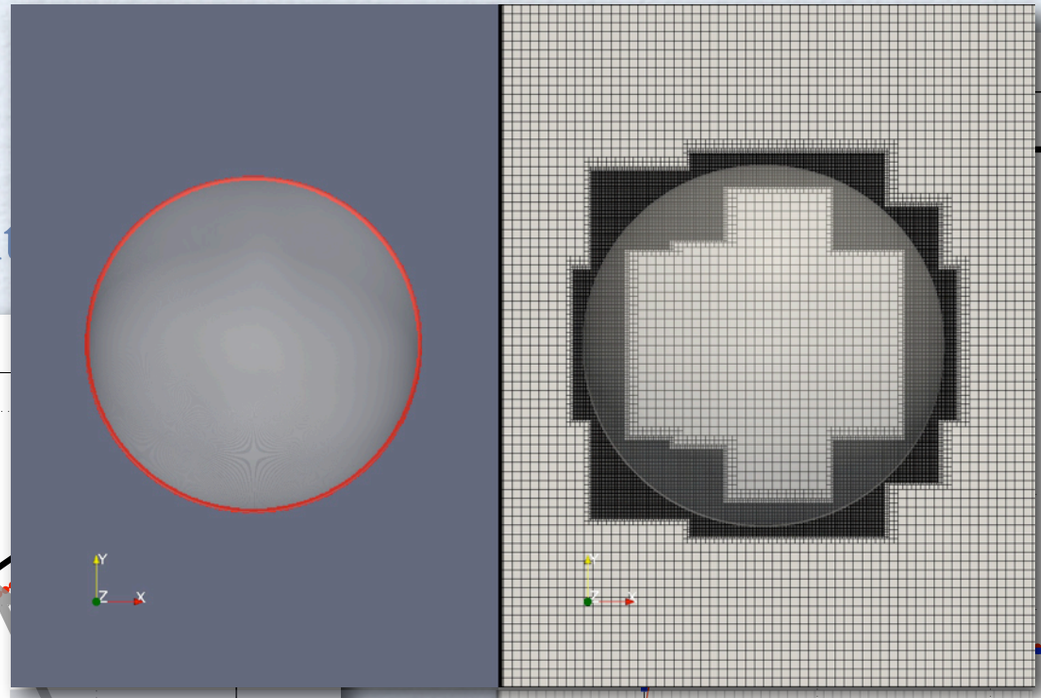


Fisher,
Ann. Eugenics 1937

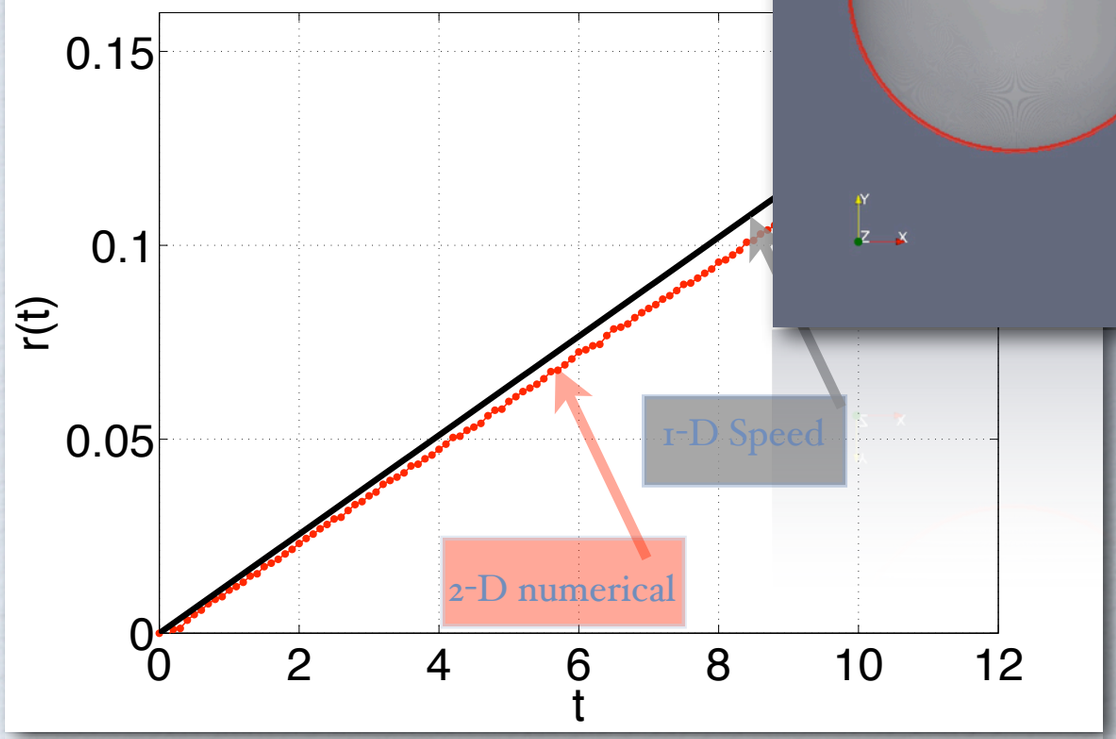
Wavespeed & Efficiency

Ratio of Cells vs Time

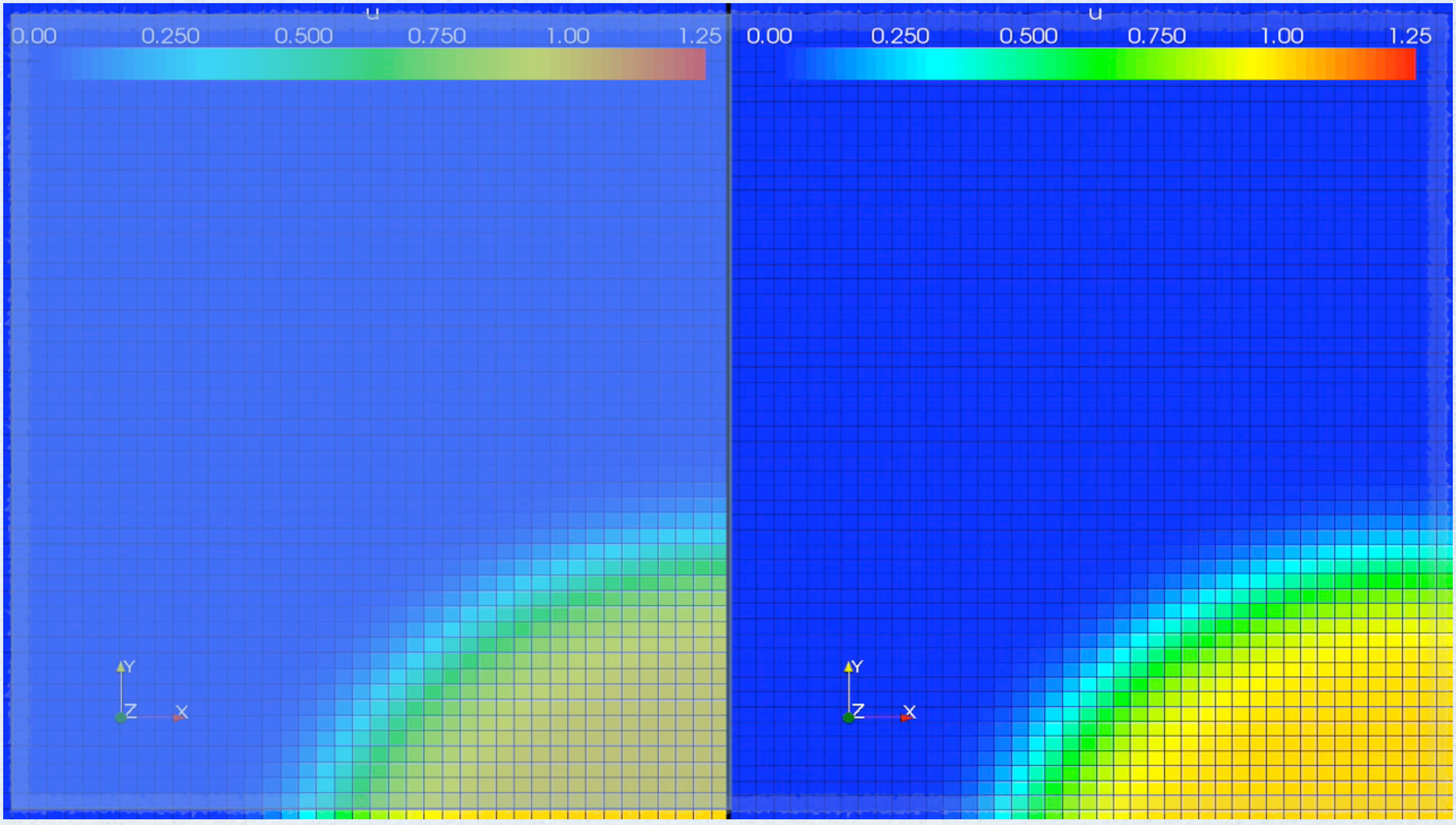
$$E(t') = \frac{L_{\mathrm{AMR}}(t')}{L_{h_{\min}}}$$



Fisher:Wavefront



Bayati B., et.al., J. of Comp. Phys., 2011

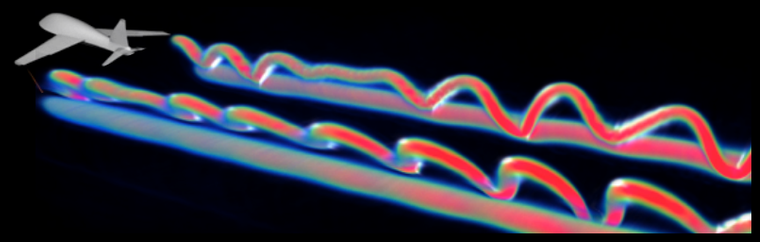


STOCHASTIC

DETERMINISTIC

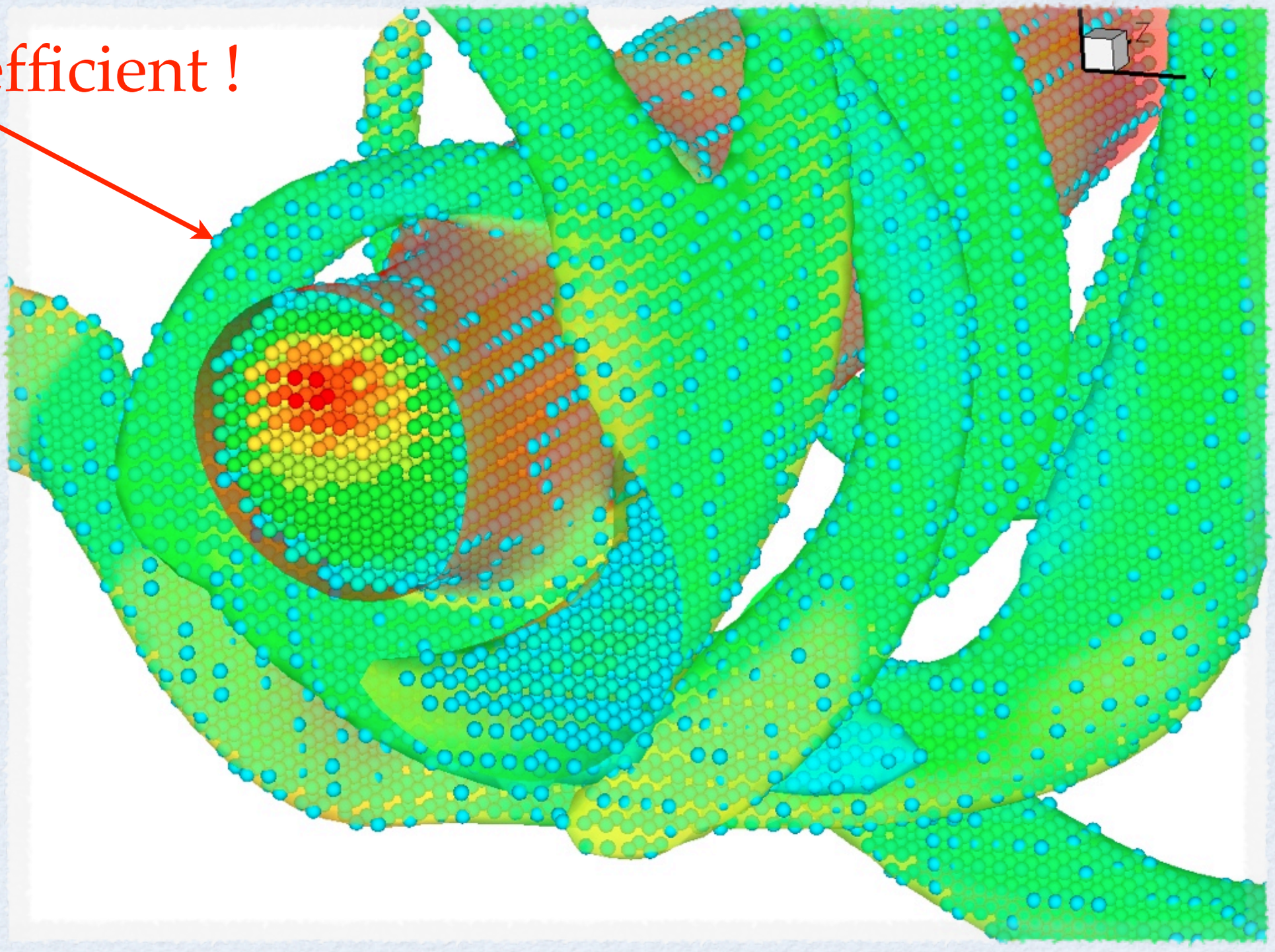
16384 Cores - 10 Billion Particles - 60% efficiency

Runs at IBM Watson Center - BLue Gene/L



Chatelain P., Curioni A., Bergdorf M., Rossinelli D., Andreoni W., Koumoutsakos P., Billion Vortex Particle Direct Numerical Simulations of Aircraft Wakes, Computer Methods in Applied Mech. and Eng. 197/13-16, 1296-1304, 2008

PARTICLES ARE ADAPTIVE



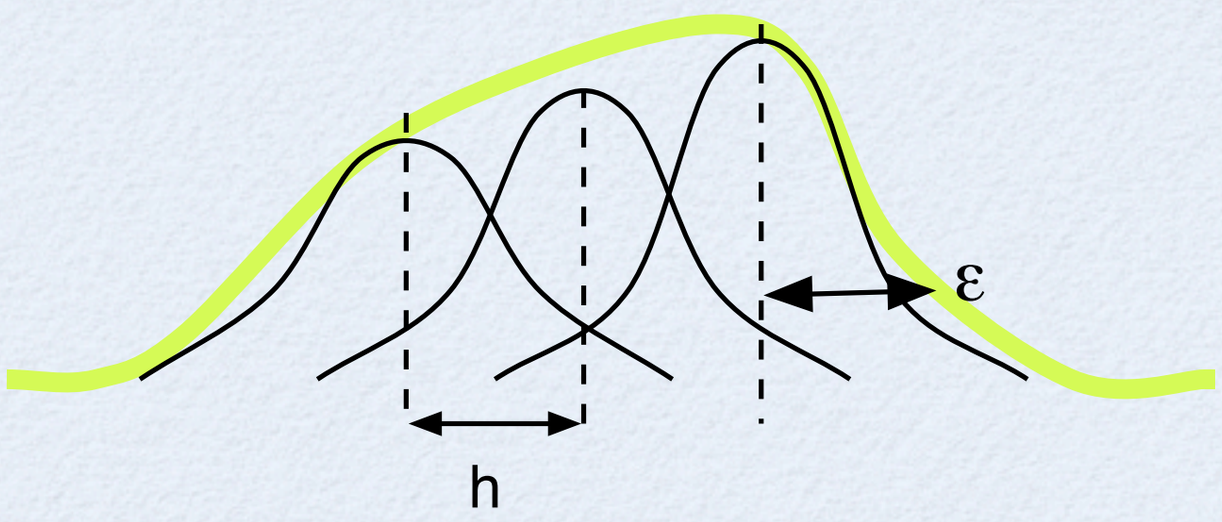
FUNCTIONS and PARTICLES

Integral Function Representation

$$\Phi(x) = \int \Phi(y)\,\delta(x-y)\,dy$$

Function Mollification

$$\Phi_\epsilon(x) = \int \Phi(y)\,\zeta_\epsilon(x-y)\,dy$$

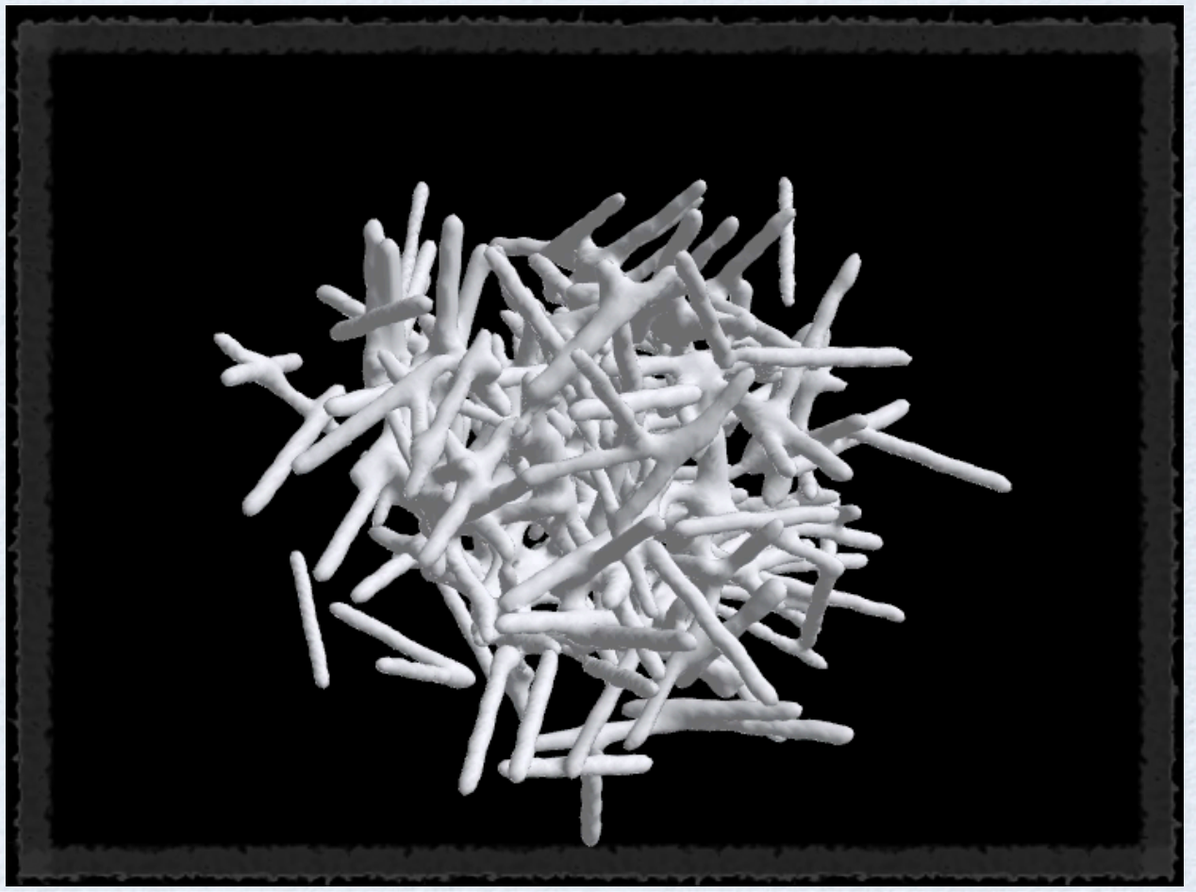


Point Particle Quadrature

$$\Phi^h(x,t) = \sum_{p=1}^{N_p} h_p^d\,\Phi_p(t)\,\delta(x-x_p(t))$$

Smooth Particle Quadrature

$$\Phi_\epsilon^h(x,t) = \sum_{p=1}^{N_p} h_p^d\,\Phi_p(t)\,\zeta_\epsilon(x-x_p(t))$$

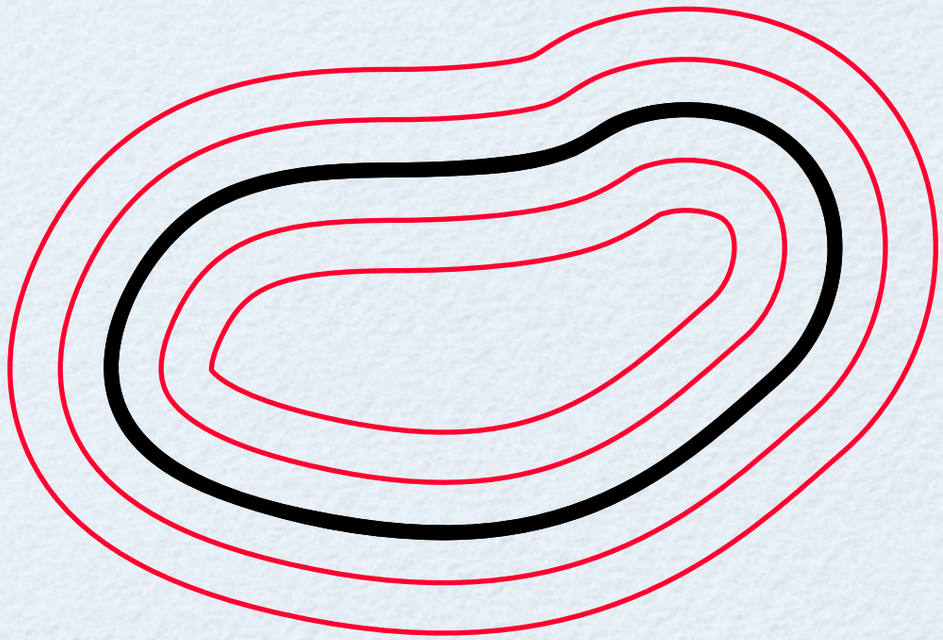


Particles are “mesh” free

SURFACES AS LEVEL SETS

$$\Gamma(t) = \{\mathbf{x} \in \Omega \,|\, \phi(\mathbf{x}, t) = 0\}$$

$$|\nabla \phi| = 1$$



EVOLVING THE LEVEL SETS

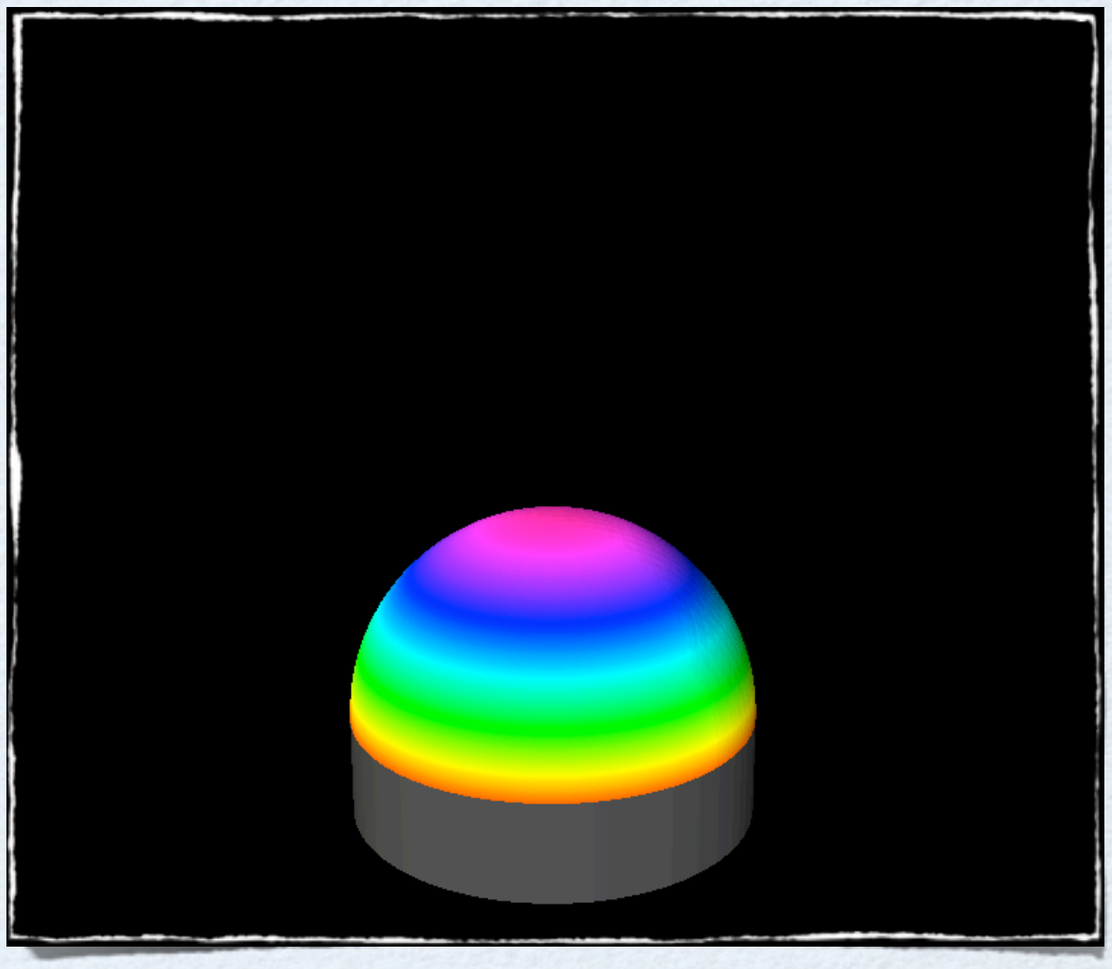
$$\frac{\partial \Phi}{\partial t} + u \cdot \nabla \Phi = 0$$

PARTICLE APPROXIMATION

$$\Phi_\epsilon^h(x,t) = \sum_{p=1}^{N_p} h_p^d \, \Phi_p(t) \, \zeta_\epsilon(x - x_p(t))$$

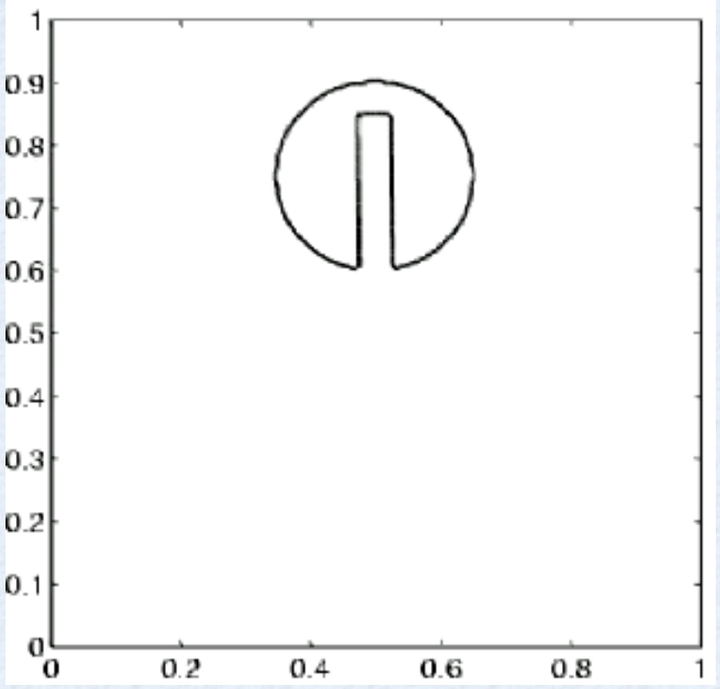
Lagrangian Surface Transport

$$\frac{dx_p}{dt} = \mathbf{u_p} \qquad \frac{D\Phi_p}{Dt} = 0$$



S. E. Hieber and P. Koumoutsakos. A Lagrangian particle level set method. **J. Computational Physics**, 210:342-367, 2005

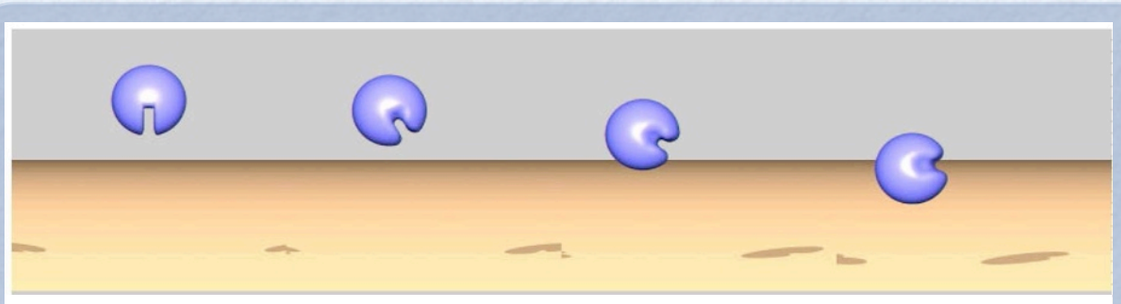
Lagrangian vs Eulerian Descriptions



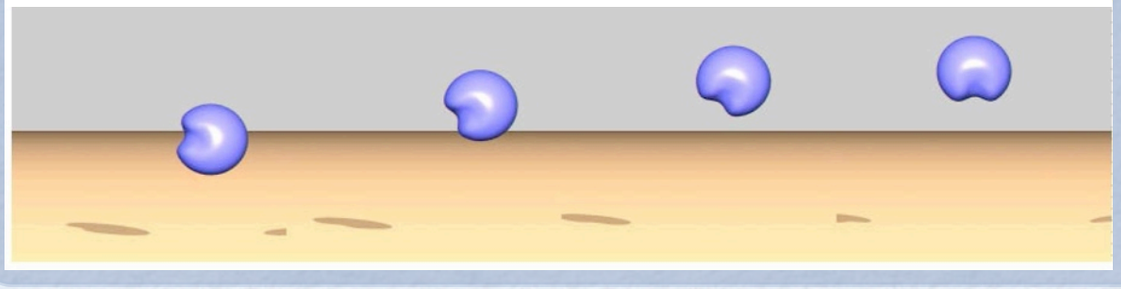
- **PARTICLE LEVEL SETS** exact for rigid body motion

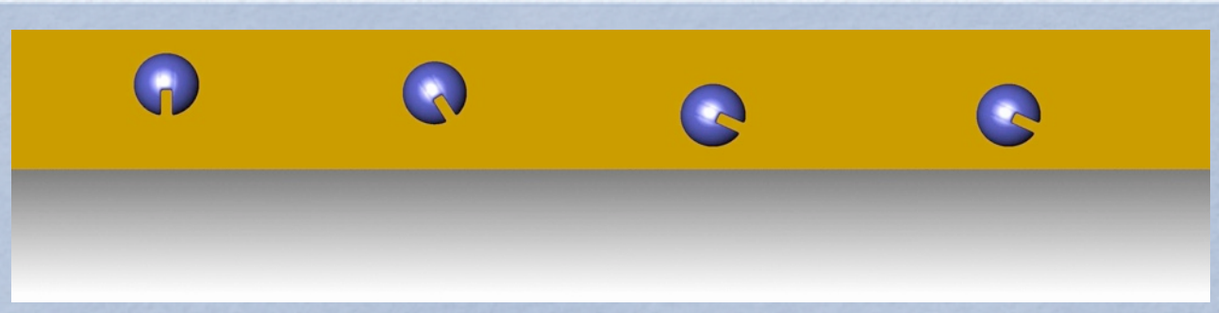
$$\Phi(\mathbf{x}, t) = \Phi_0(\mathbf{x} - \mathbf{u}t)$$

Lagrangian Particle methods
good for **linear** advection

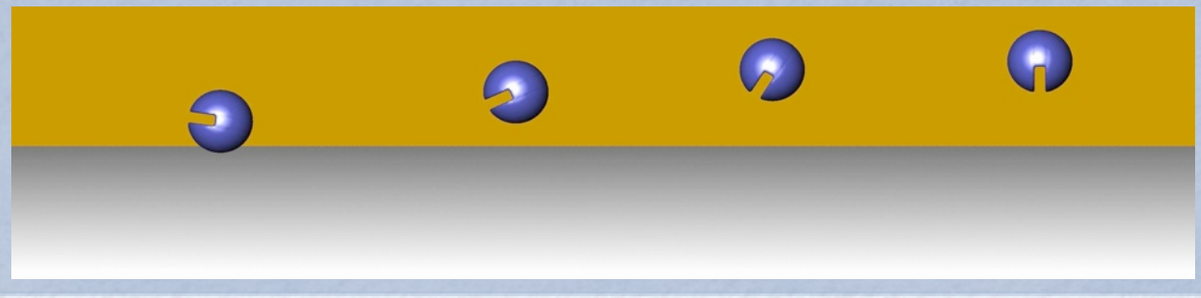


Hubrid Particle-Grid Level Sets
(Enright and Fedkiw, 2002)

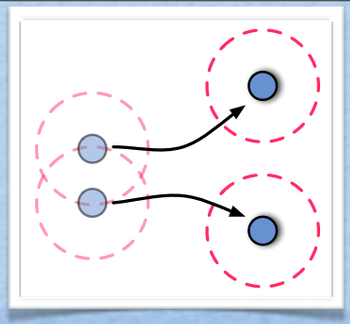




Lagrangian Particle Level Sets
(Hieber and Koumoutsakos, 2005)



LAGRANGIAN DISTORTION



- loss of overlap -> loss of convergence

Particles follow flow trajectories - Location distortion

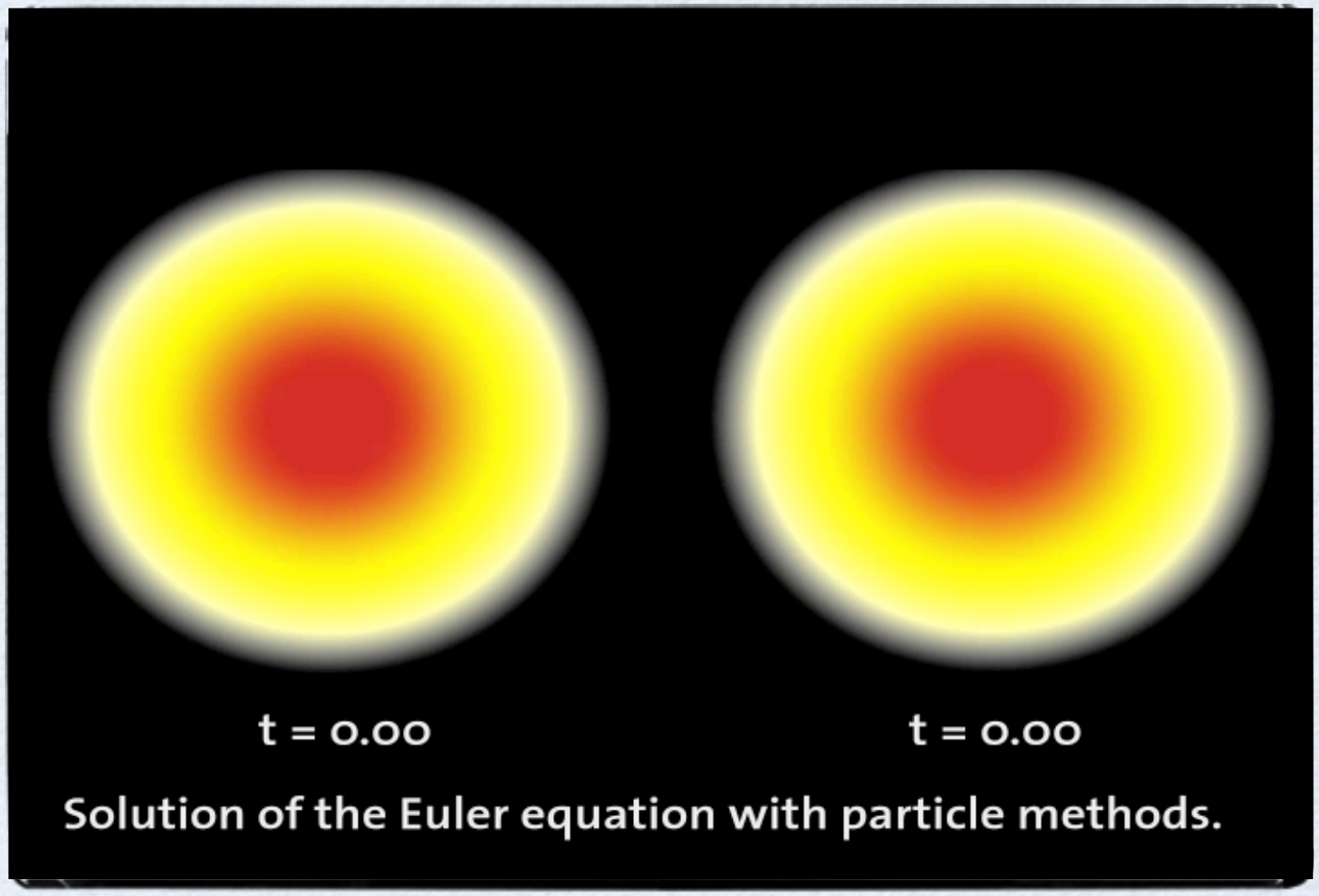
EXAMPLE :

Incompressible 2D Euler Equations

$$\omega = \nabla \times \mathbf{u} \quad \nabla \cdot \mathbf{u} = 0$$

$$\frac{D\omega}{Dt} = 0$$

There is an exact axisymmetric solution



SMOOTH PARTICLES MUST OVERLAP

Integral Function Representation

$$\Phi(x) = \int \Phi(y)\,\delta(x-y)\,dy$$

Point Particle Quadrature

$$\Phi^h(x,t) = \sum_{p=1}^{N_p} h_p^d\,\Phi_p(t)\,\delta(x-x_p(t))$$

Function Mollification

$$\Phi_\epsilon(x) = \int \Phi(y)\,\zeta_\epsilon(x-y)\,dy$$

Smooth Particle Quadrature

$$\Phi_\epsilon^h(x,t) = \sum_{p=1}^{N_p} h_p^d\,\Phi_p(t)\,\zeta_\epsilon(x-x_p(t))$$

$$\int \zeta\, x^\alpha\, dx = 0^\alpha \qquad 0 \le \alpha < r$$

TOTAL ERROR

$$||\Phi - \Phi_\epsilon^h|| \le ||\Phi - \Phi_\epsilon|| + ||\Phi_\epsilon - \Phi_\epsilon^h||$$

$$\le (C_1 \epsilon^r + C_2 (\frac{h}{\epsilon})^m)\,||\Phi||_\infty$$

Need $h/\varepsilon < 1$ for accuracy

PARTICLES MUST OVERLAP

Hald, Beale and Majda, (80's) Anderson, Cottet (90's)

Are Particle Methods Grid Free ?

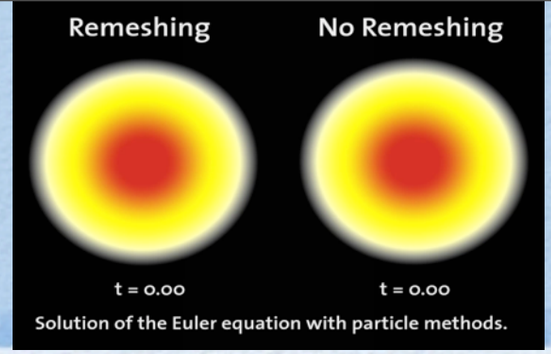
How to fix it ?

- Modify the smoothing kernels (SPH - Monaghan) — DOES NOT WORK
- Re-distribute particles with Voronoi Meshes (ALE - Russo) — EXPENSIVE - UNSTABLE
- Re-initialise particle strengths (WRKPM - Liu, Belytchko) — EXPENSIVE

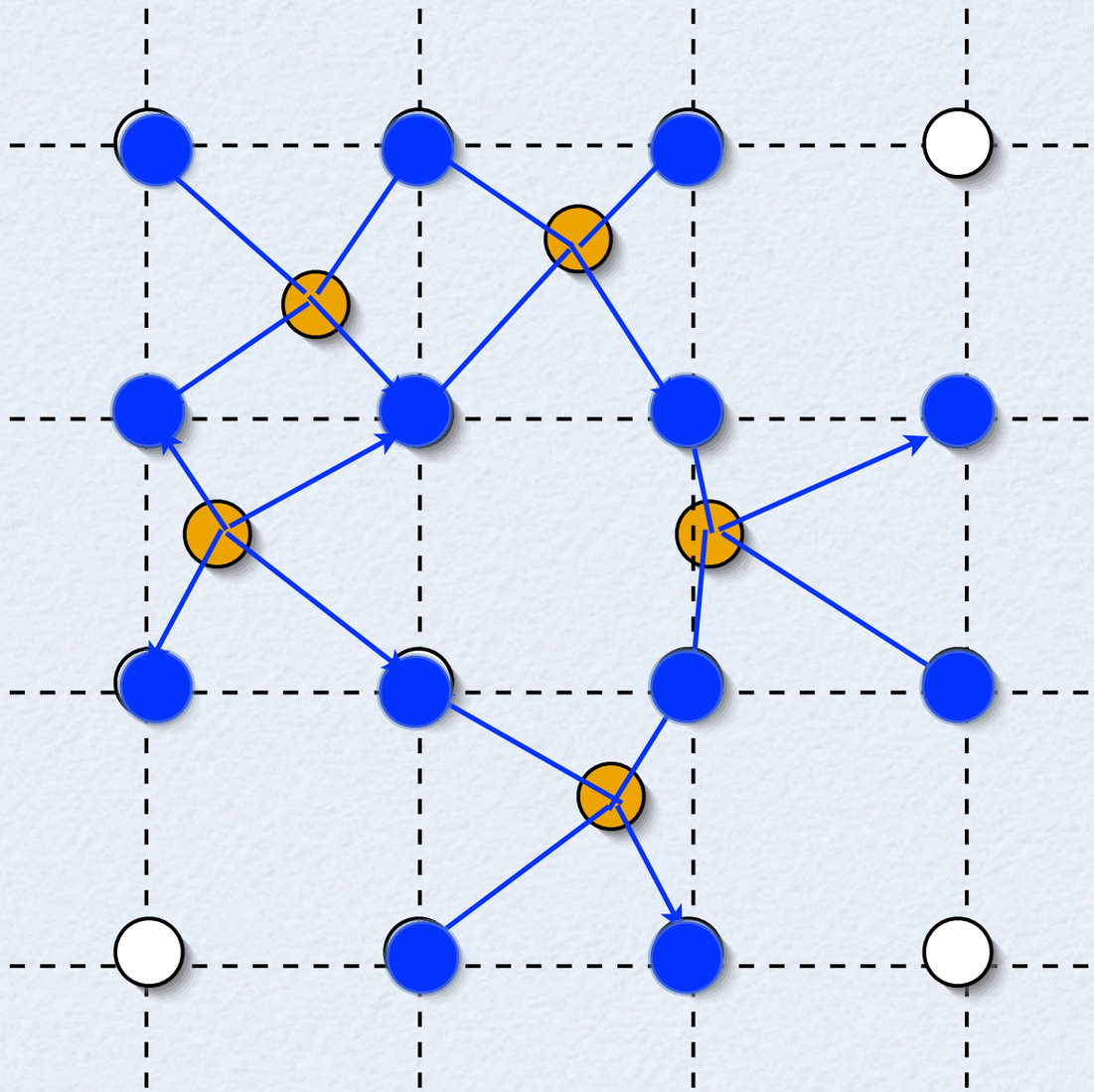
REMESHING : Re-project particles on a mesh

- NO MESH-FREE particle methods
- Can use all the “tricks” of mesh based methods
- High CFL
- Multiresolution & Multiscaling
- .....

Particle Remeshing



Koumoutsakos, JCP, 1997



Moment Conserving Interpolation : $Q_p^{\text{new}} = \sum_{p'} Q_{p'} M(j\,h - x_{p'})$

REMESHED PARTICLE METHODS

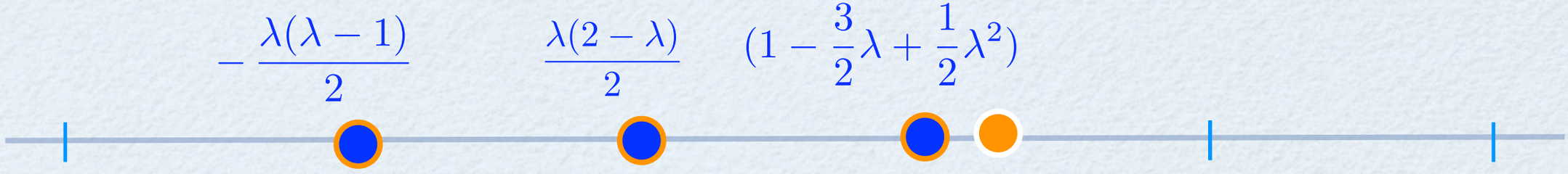
1.ADVECT : Particles ->Large CFL

2.REMESH : Particles to Mesh -> Gather/Scatter

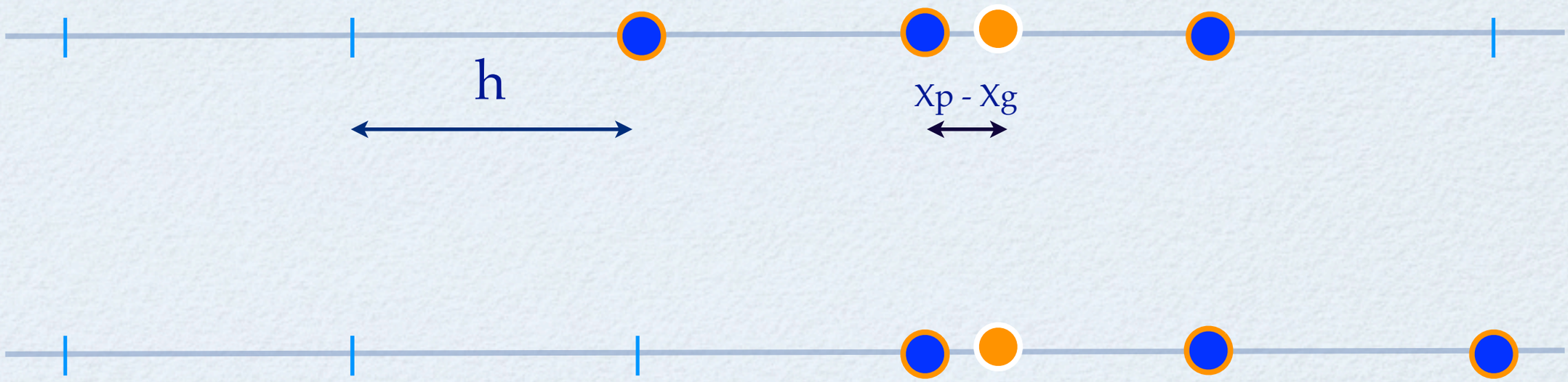
3.SOLVE:Poisson/Derivatives on Mesh ->FFTw/Ghosts

4:RESAMPLE: Mesh Nodes BECOME Particles

Remeshing Stencils



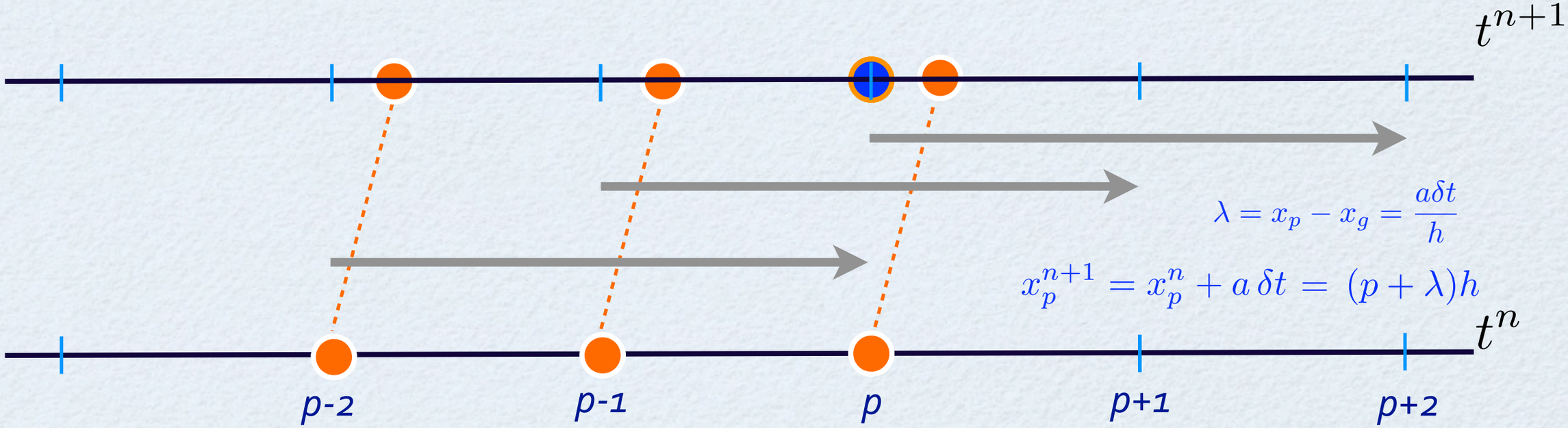
$$\lambda = \frac{X_p - X_g}{h}$$



Bergdorf et. al., MMS,2005

Cottet et.al., CRAS, 2008

$$u_p^{n+1} = -\frac{\lambda(\lambda-1)}{2}u_{p-2}^n + \frac{\lambda(2-\lambda)}{2}u_{p-1}^n + (1-\frac{3}{2}\lambda+\frac{1}{2}\lambda^2)u_p^n$$



$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \qquad u_p = u(x_p)h$$

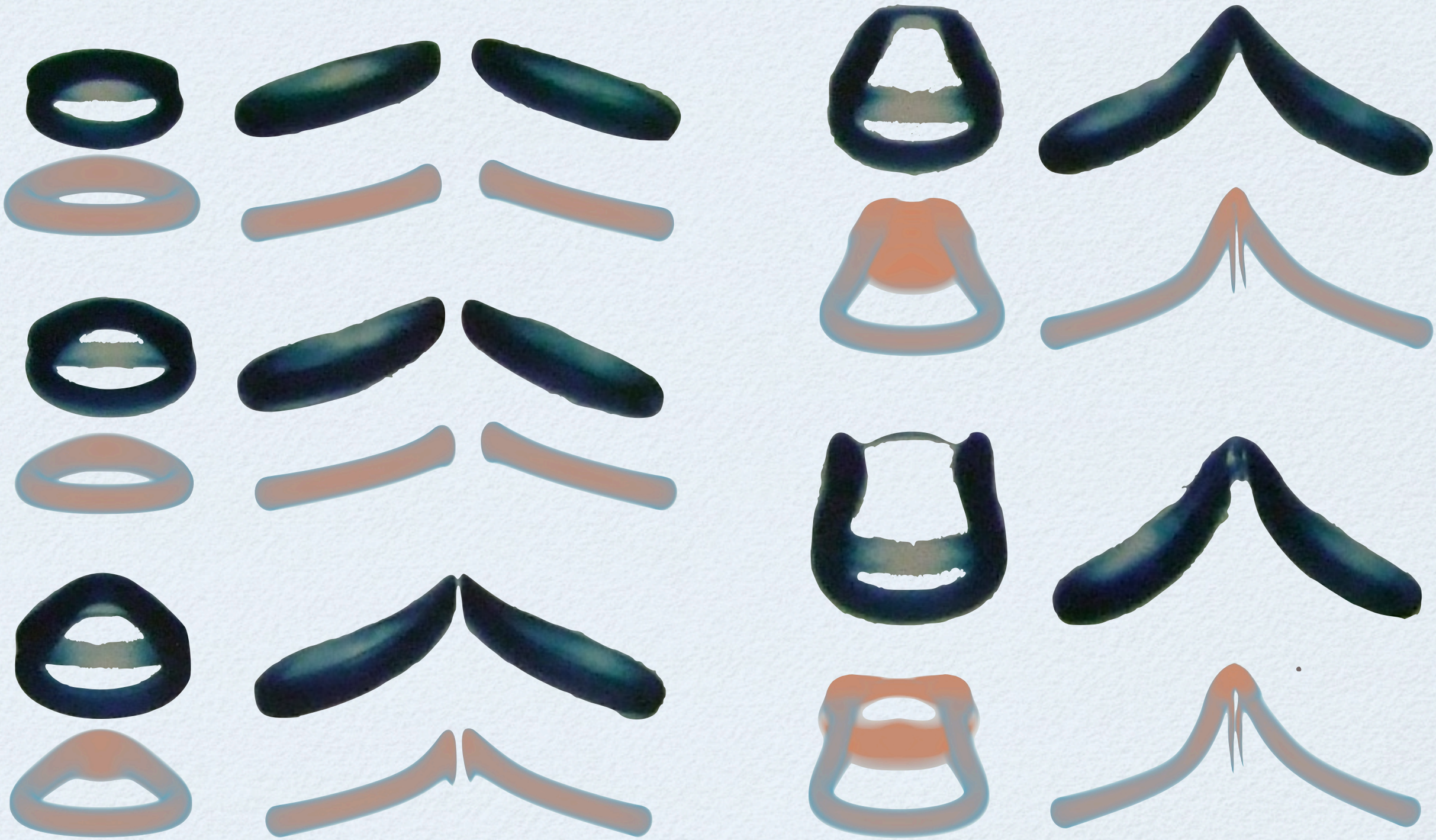
$$\frac{du_p}{dt} = 0 \qquad \frac{dx_p}{dt} = a \qquad \textbf{+ REMESH}$$

$$u_p^{n+1} = u_p^n - \frac{\lambda}{2}(3u_p^n - 4u_{p-1}^n + 4u_{p-2}^n) + \frac{\lambda^2}{2}(u_p^n - 2u_{p-1}^n + u_{p-2}^n)$$

Euler Advect + One-sided Remesh = Beam-Warming FD

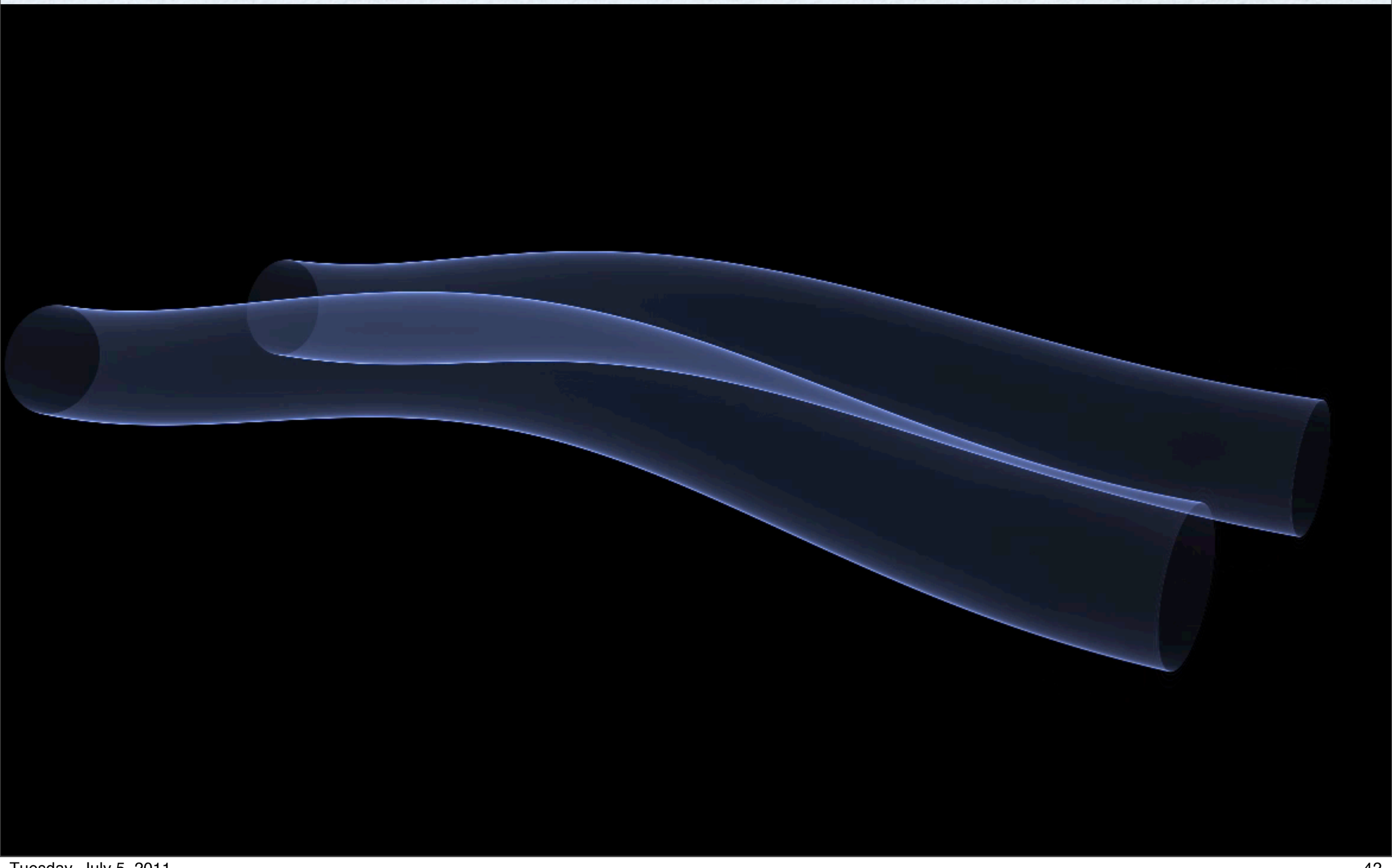
Euler Advect + Central Remesh = Lax - Wendroff FD .....

VORTEX RING COLLISION, Re = 1800

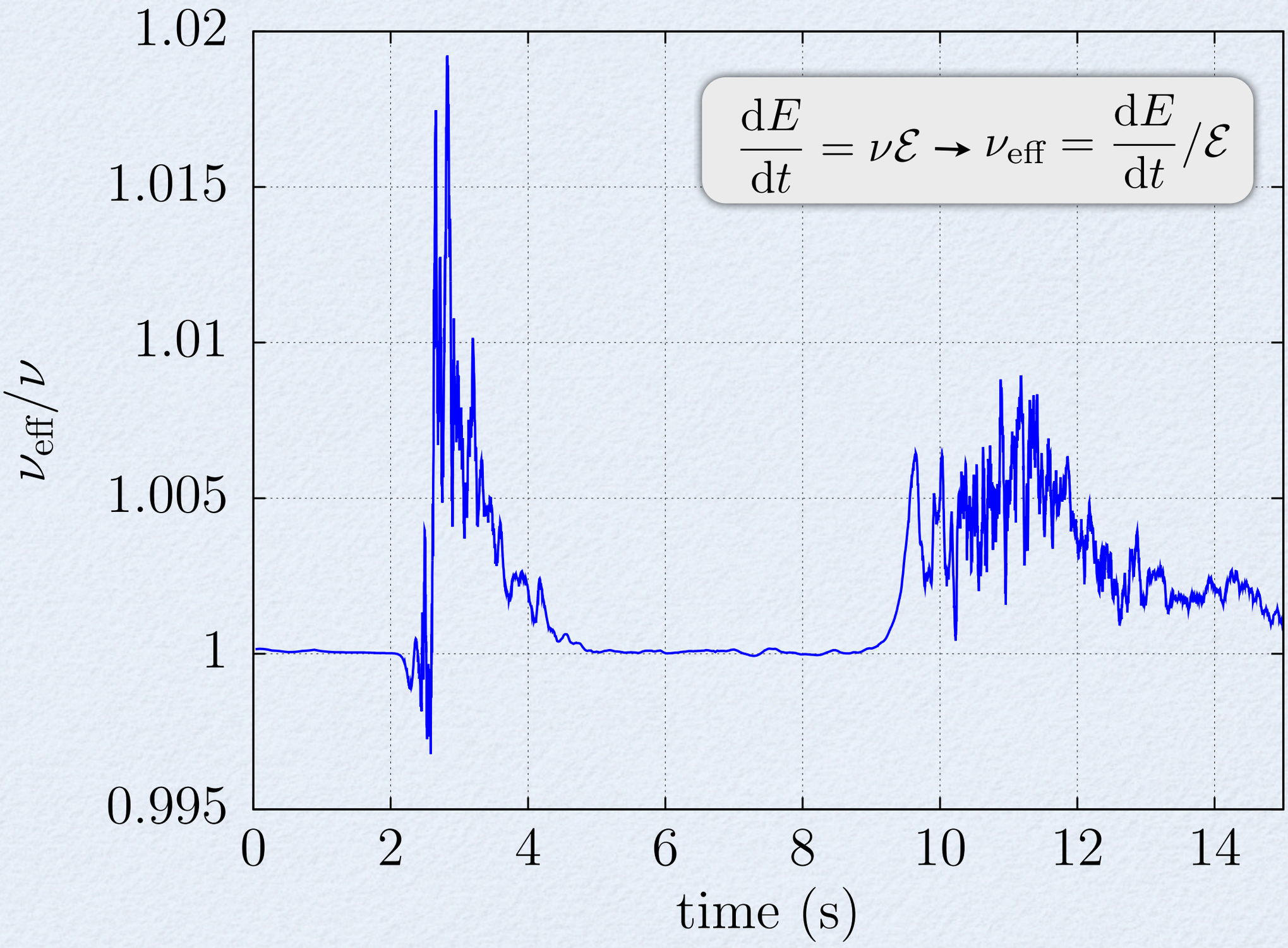


Experiments : P. Schatzle & D. Coles (1986)

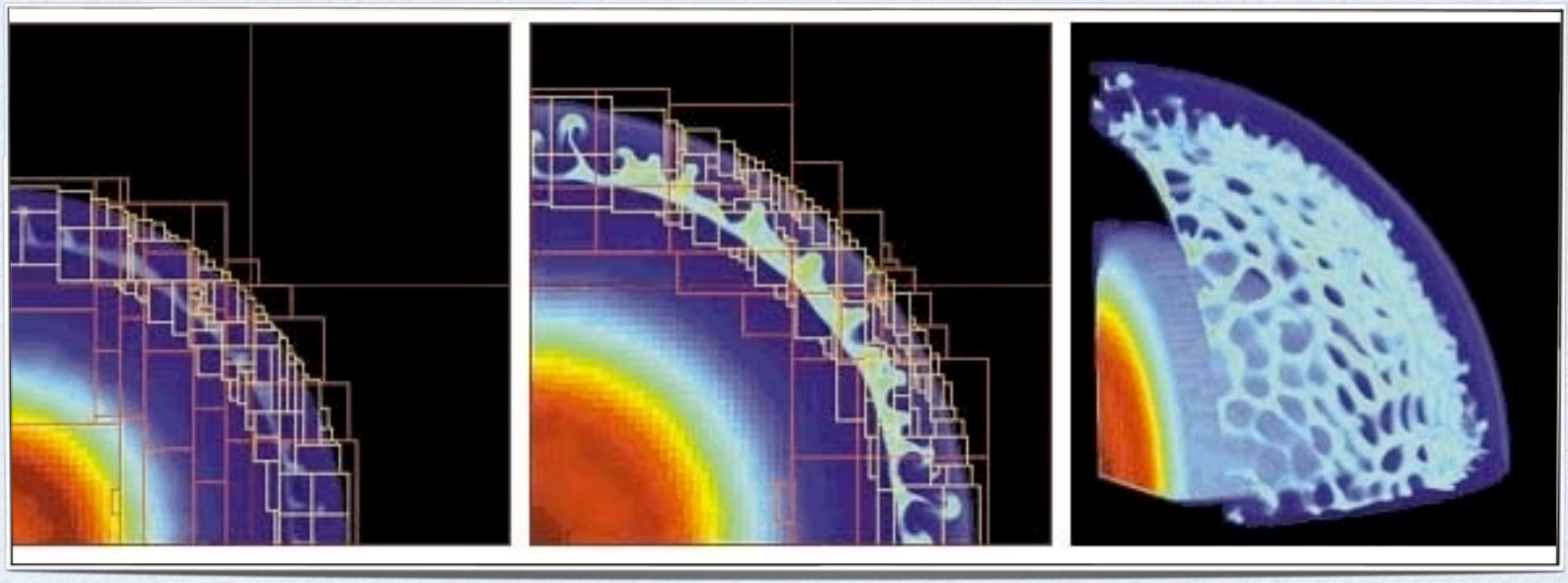
VORTEX DYNAMICS at High Re



Evolution of error in effective viscosity



Adaptive Mesh Refinement



- Support of unstructured grids
- Different mesh orientations

- Low compression rate (Gradient, curvature)
- No explicit control on the compression error

Berger, Colella,J. Comp. Phys., 1989

M. Bergdorf, P. Koumoutsakos. A Lagrangian Particle-Wavelet Method.
Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal, 5(3), 980-995, 2006

PARTICLETS : Particles and Wavelets

WAVELET PARTICLE METHOD

While particles are on grid locations

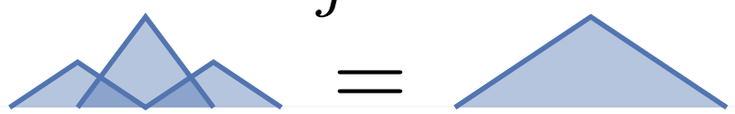
mollification kernel $\longleftrightarrow$ basis/scaling function

Multiresolution analysis (MRA) $\{\mathcal{V}^l\}_{l=0}^{L}$ of particle quantities

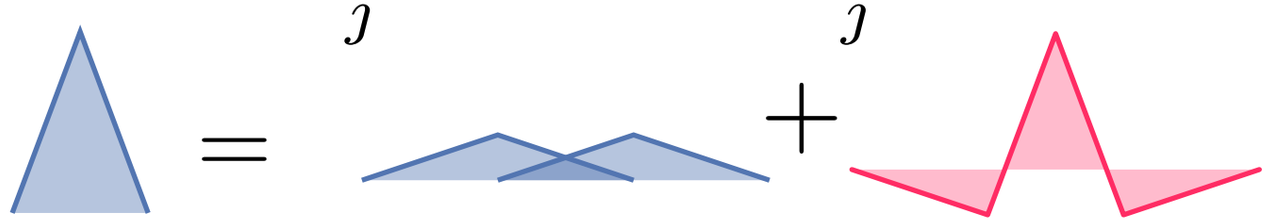
Refineable kernels
as basis functions of $\mathcal{V}^l$

Wavelets as basis functions of the complements $\mathcal{W}^l$

$$\zeta_k^l = \sum_j h_{j,k}^l \, \zeta_j^{l+1}$$



$$\zeta_k^{l+1} = \sum_j \tilde{h}_{j,k}^l \, \zeta_j^l + \sum_j \tilde{g}_{j,k}^l \, \psi_j^l$$



Multiresolution function representation:

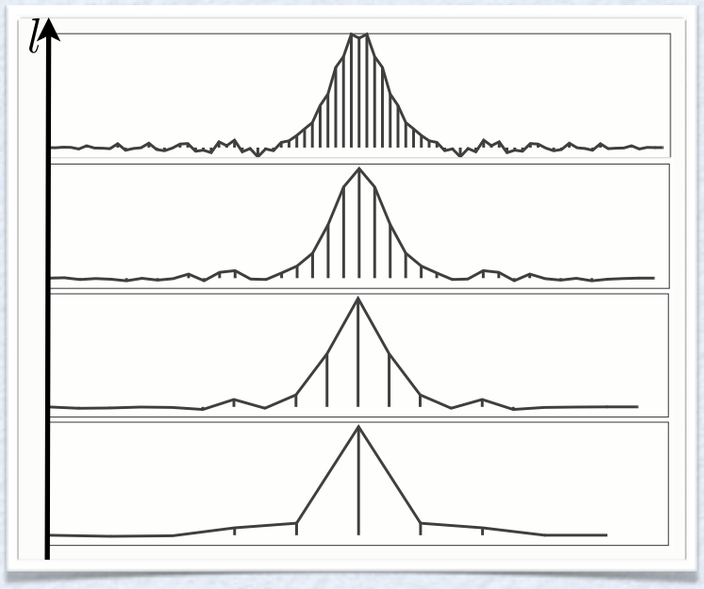
Analysis (collocation): d_k^l ~ | fine - Prediction(coarse) |

$$q^L = \sum_k c_k^0 \, \zeta_k^0 + \sum_{l<L} \sum_k d_k^l \, \psi_k^l$$

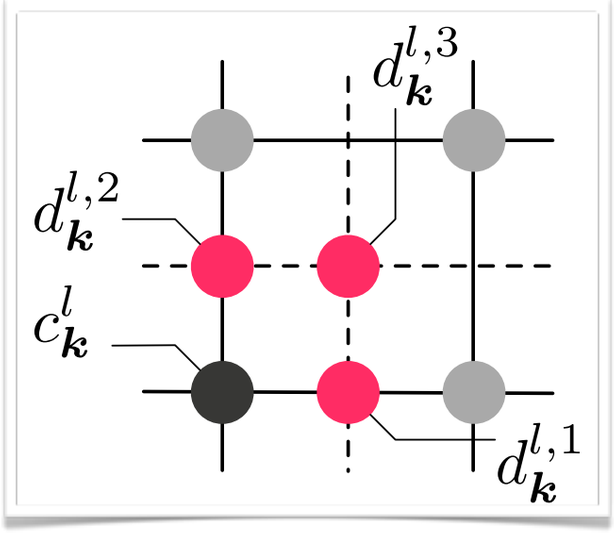
GROUND LEVEL

WAVELETS

DETAIL COEFFICIENTS



Each wavelet is associated with a specific grid point/particle (2D)



Compression / Adaptation:

Discard insignificant detail coefficients: $|d_{\boldsymbol{k}}^{l,m}| < \varepsilon$

Compressed function representation:

$\|q^L - q_{\geq}^L\| < \varepsilon$ → Adapted grid

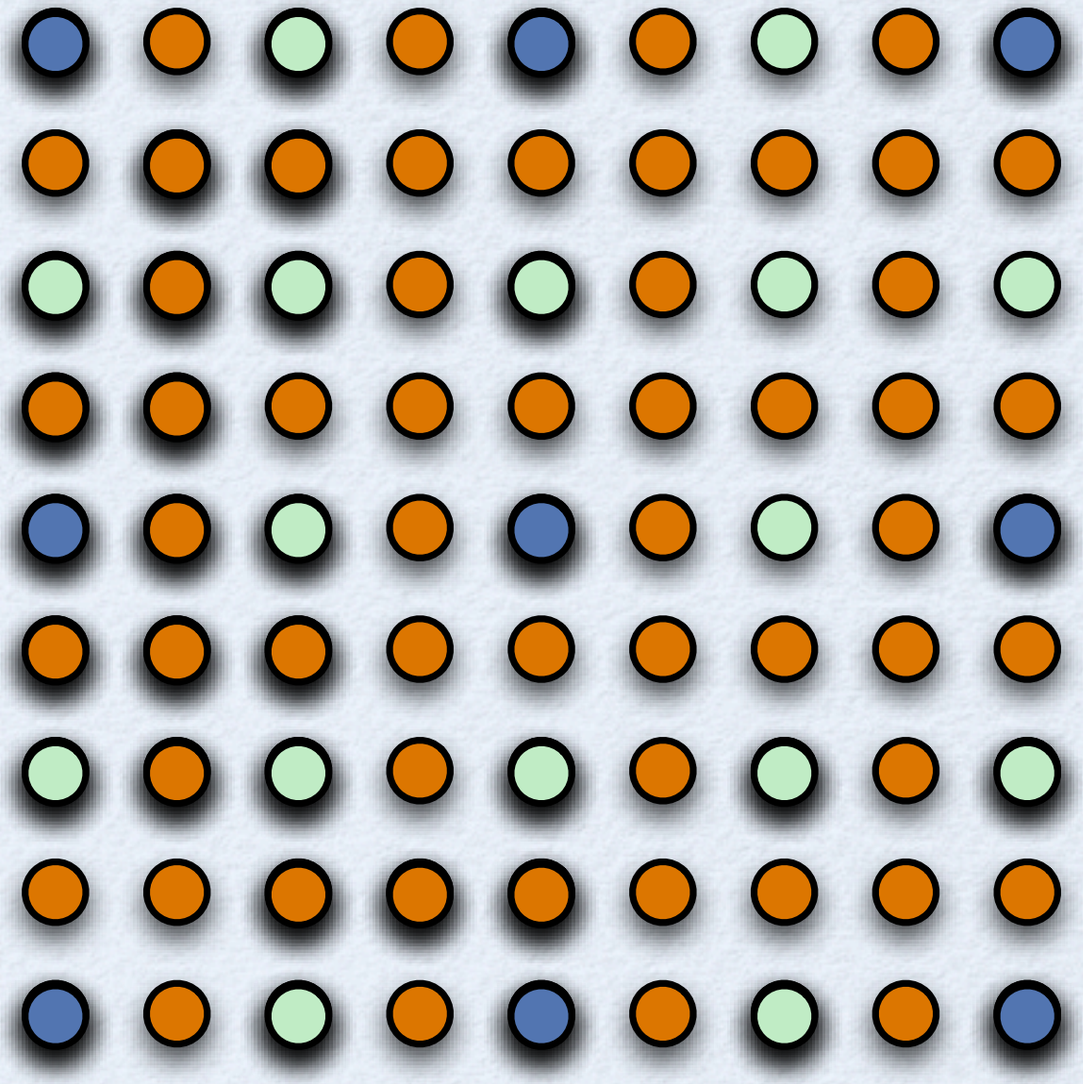
PARTICLETS : REMESHED PARTICLES + WAVELETS

$$q^L = \sum_k c_k^0 \, \zeta_k^0 + \sum_{l<L} \sum_k d_k^l \, \psi_k^l$$

"ground" level

detail coefficients

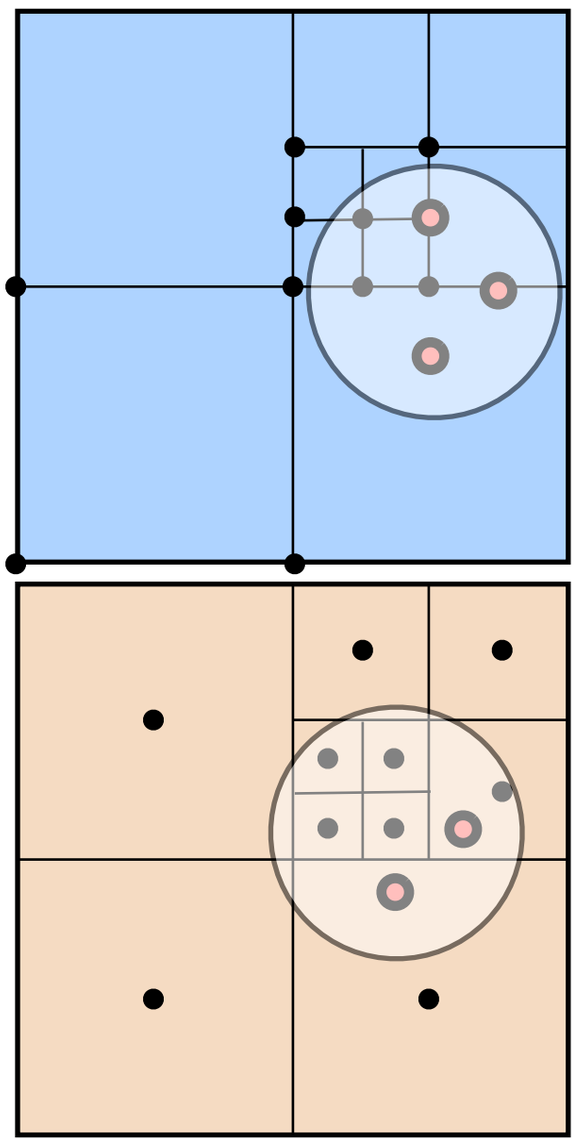
wavelets



1. Remesh
2. Wavelets- Compress/Adapt
3. Convect
4. Wavelets Reconstruct
5. GOTO 1

M. Bergdorf, P. Koumoutsakos. A Lagrangian Particle-Wavelet Method, Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal, 5(3), 980-995, 2006

Wavelet-adapted grids



PDE:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0$$

Spatial Differences = filtering operations:

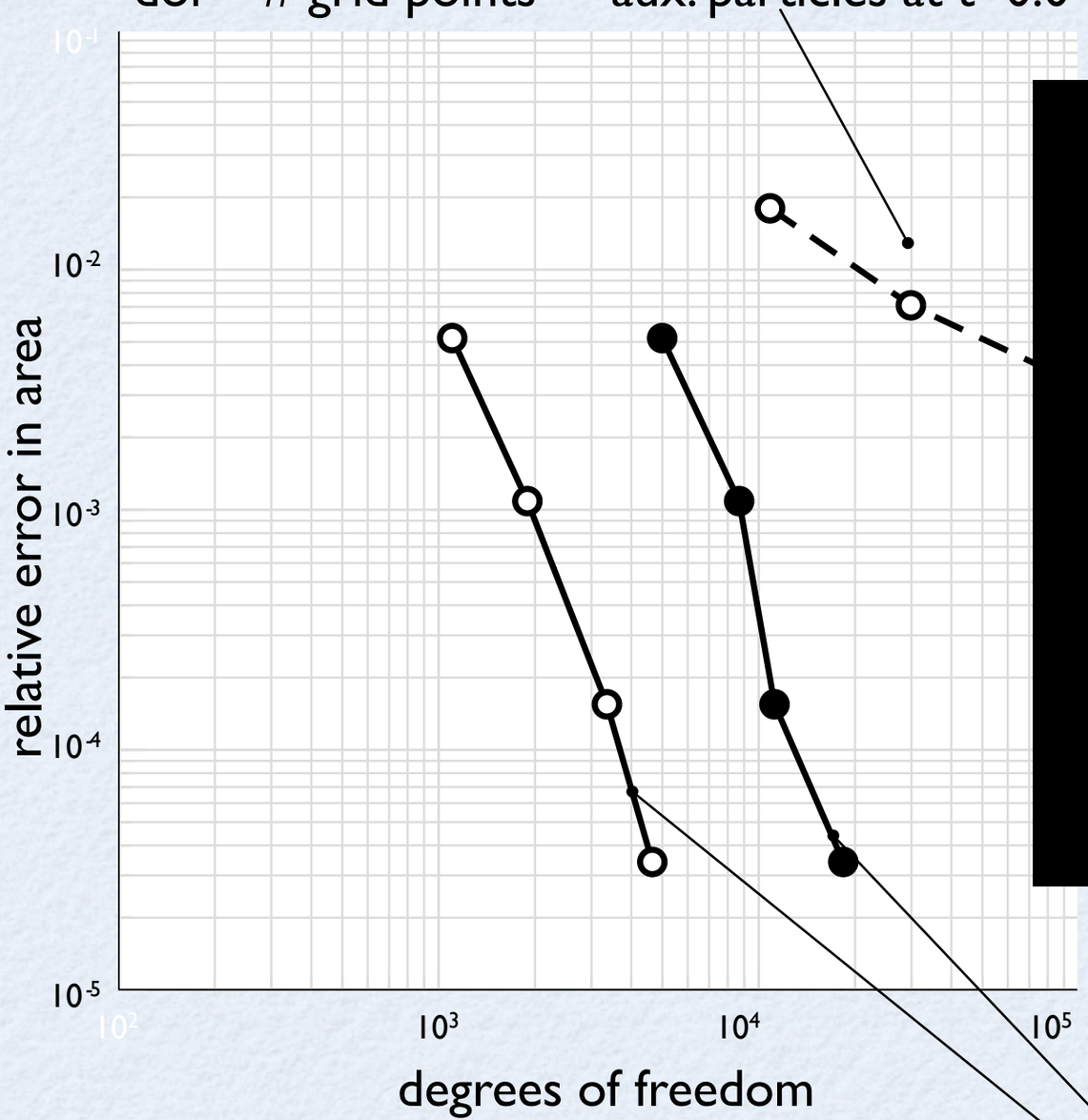
$$F(c_k^l) = \sum_{j=s_f}^{e_f - 1} c_{k+j}^l \beta_j^l, \quad \beta_j^l \text{ function of } \{c_m^l\}$$

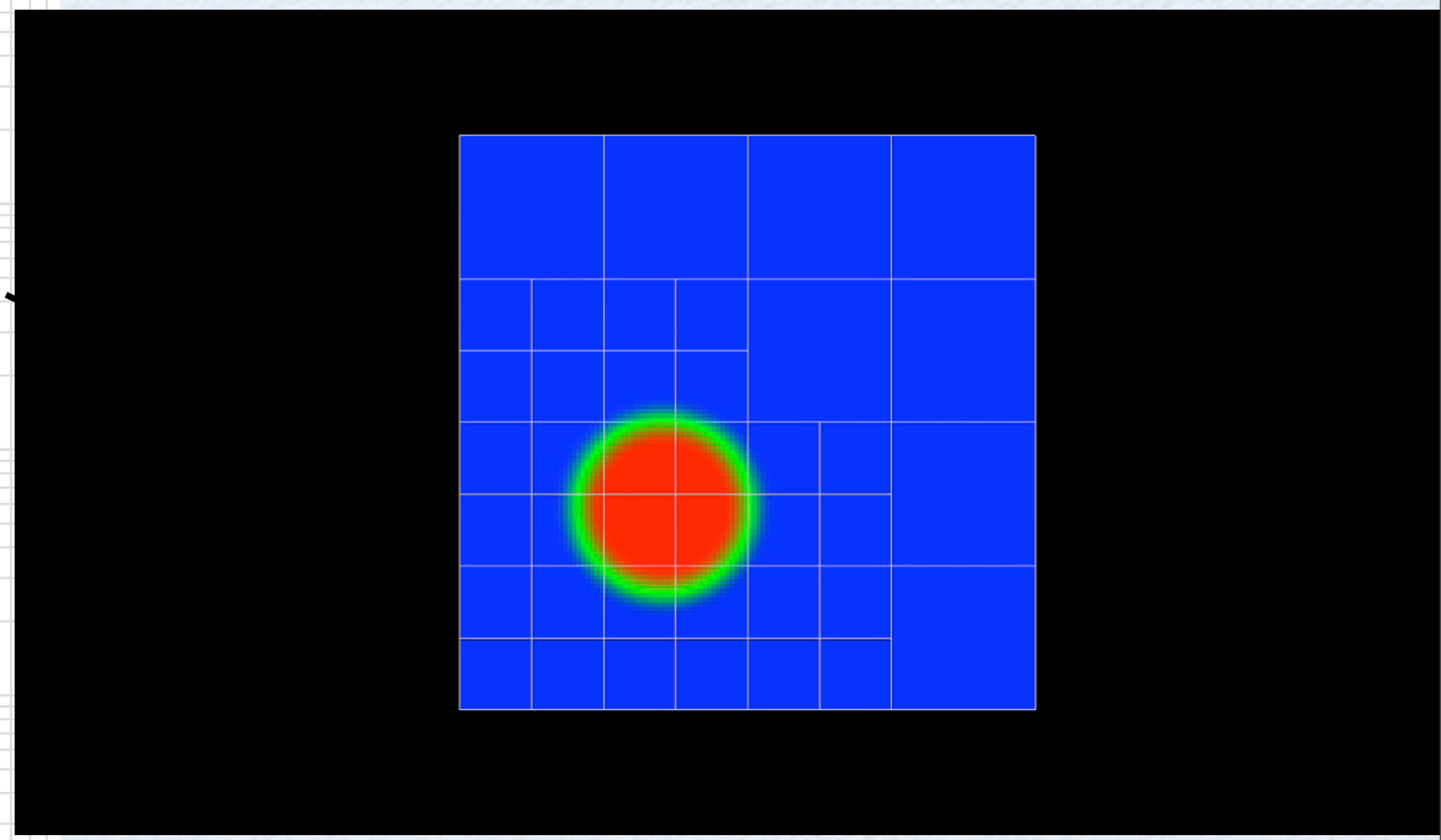
GHOSTS : easy to compute – (locally) uniform filtering of the grid

MULTIRESOLUTION LEVEL SETS

M. Bergdorf, P. Koumoutsakos. A Lagrangian Particle-Wavelet Method, Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal, 5(3), 980-995, 2006

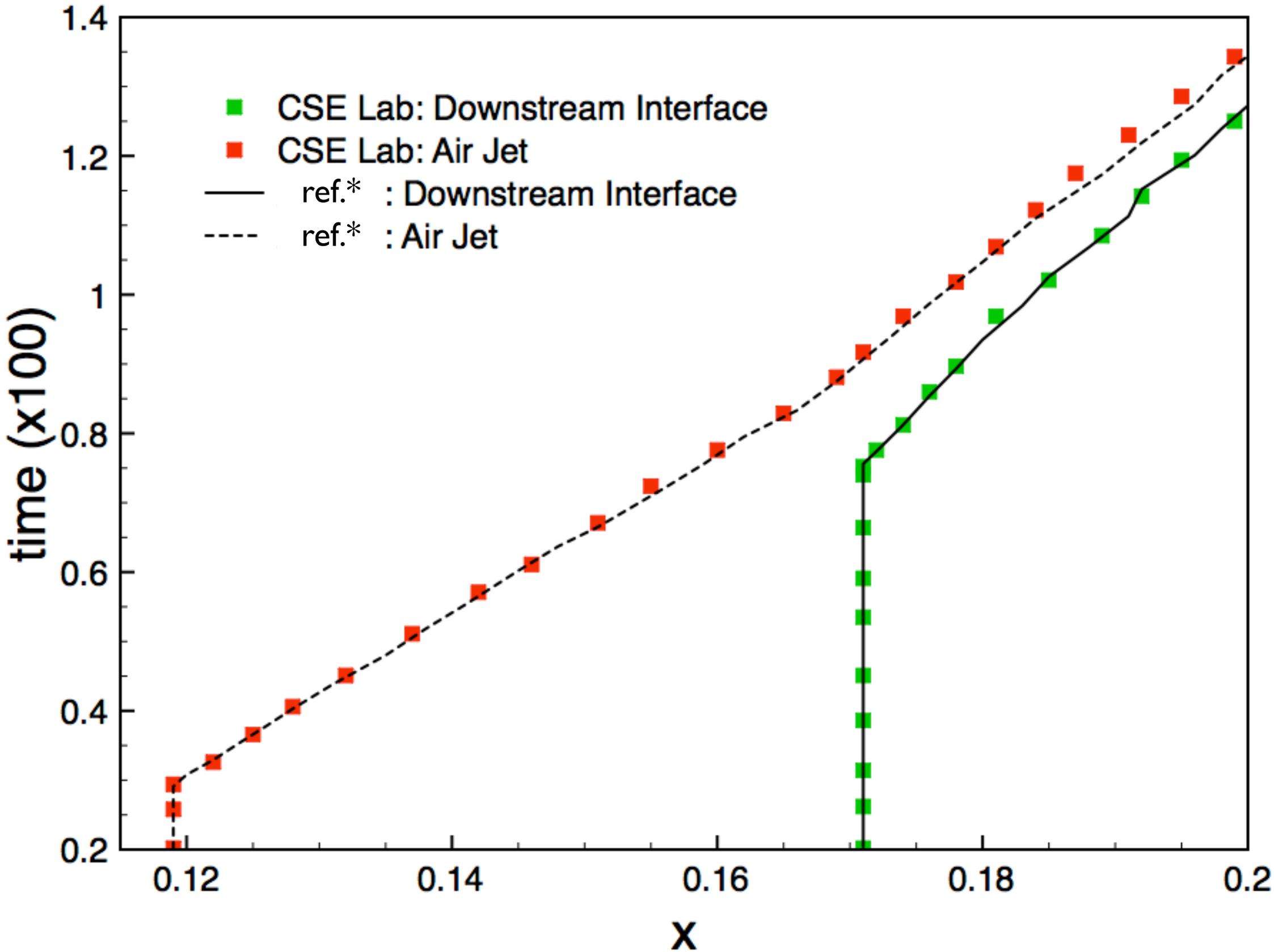
Enright, Fedkiw et al, 2002
dof = # grid points + aux. particles at t=0.0





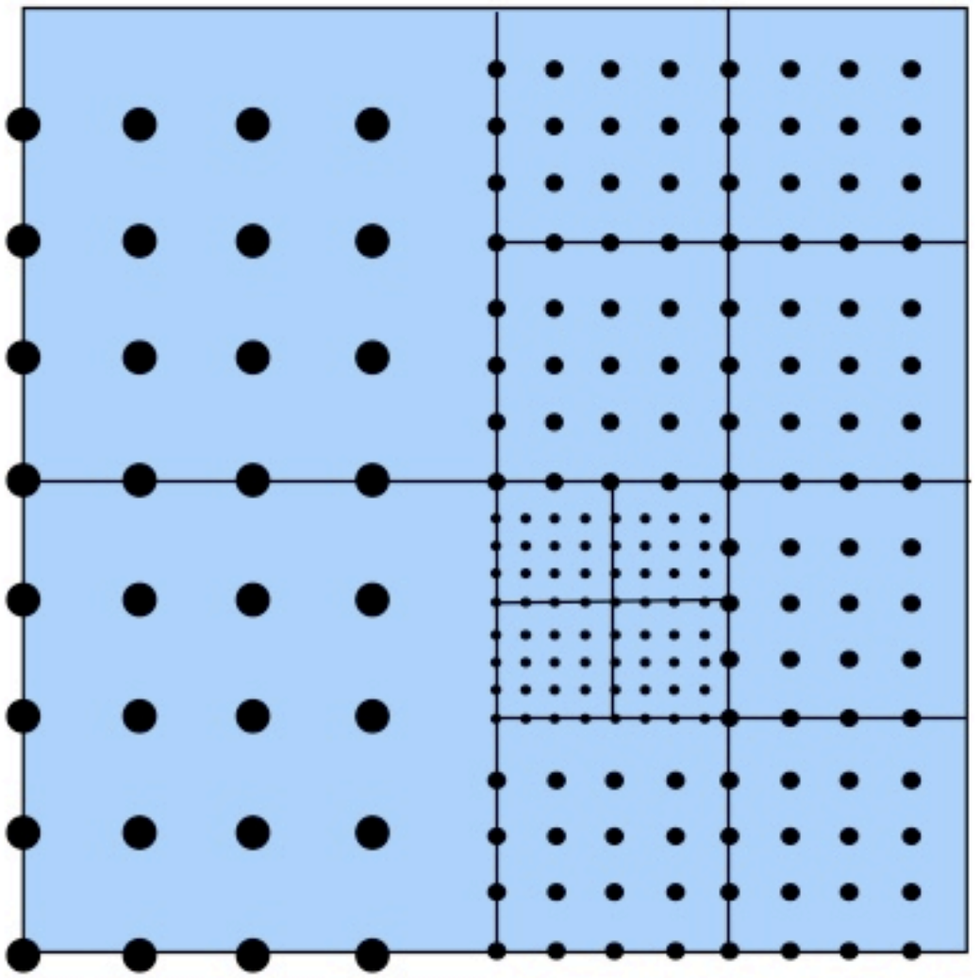
Present Method
dof = # active gp/particles at t=0.0
dof = # active gp/particles at final time

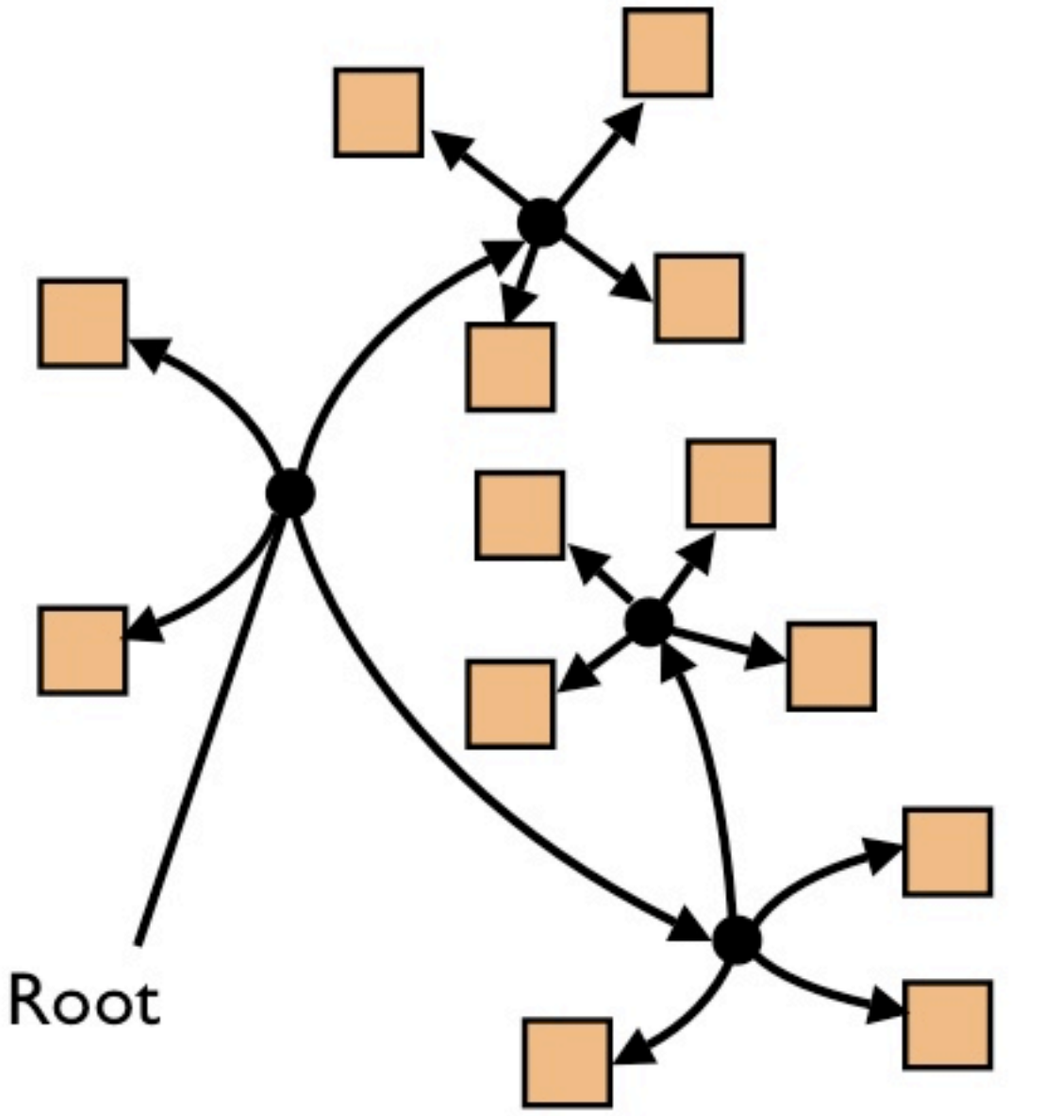
$CFL_{max} \sim 40$



*Hu, Khoo, Adams and Huang, 2006

Multi-core: Blocked Grid





- Neighbors look-up: less memory indirections
- Less #ghosts
- Within a block: random access

Multiresolution + MultiCore + GPU

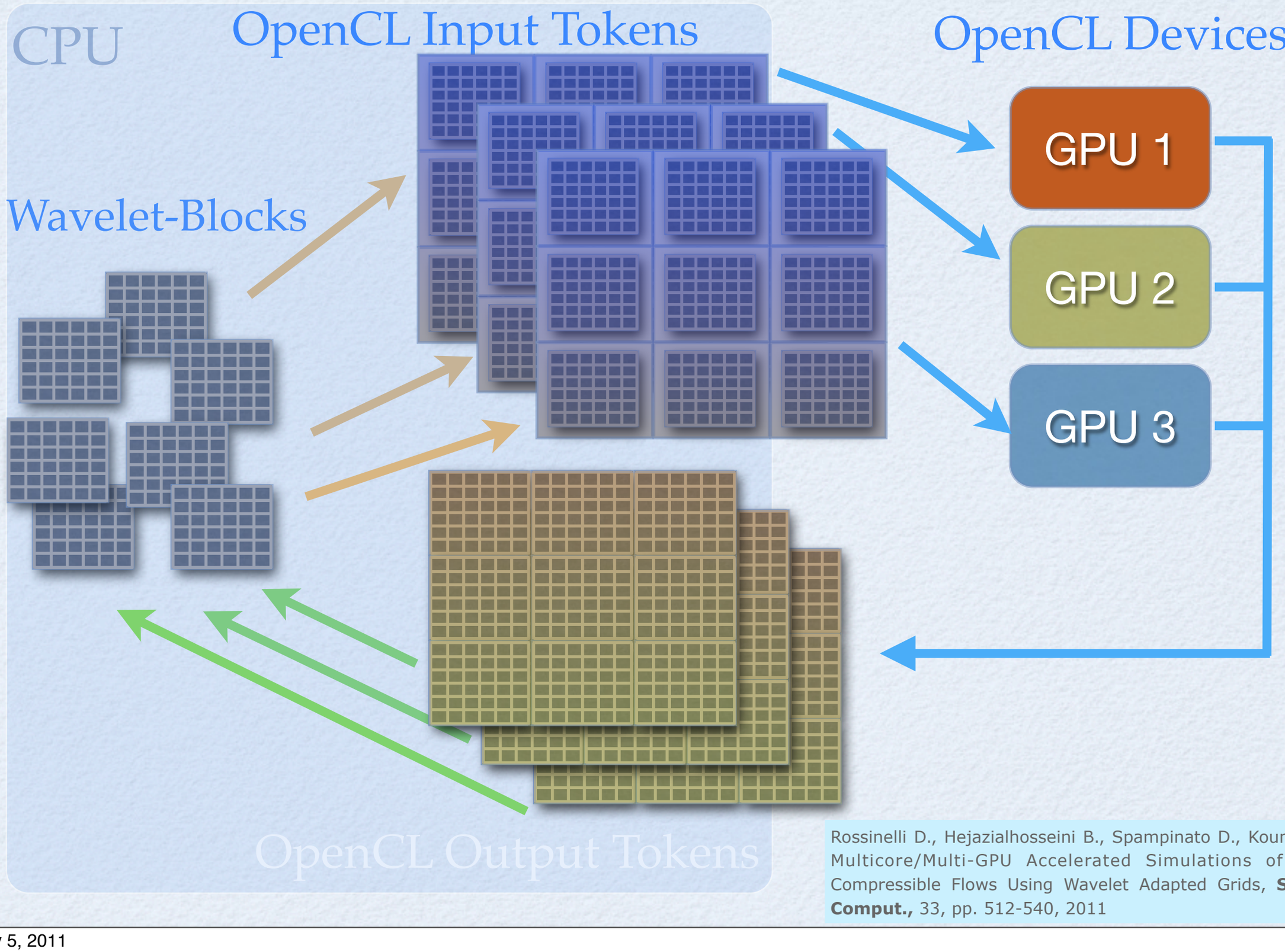
- DISTRIBUTED TASKS

  1.task parallel,ghost computing _> **multi-core**

  2.fine-grained data parallelism for RHS _> **GPUs**

$$\mathbf{q}^{\mathbf{new}} = \mathbf{q}^{\mathbf{old}} + \underbrace{\delta t \mathbf{F}\left(\mathbf{q}^{\mathbf{old}}, \nabla \mathbf{q}^{\mathbf{old}}\right)}_{\textbf{OpenCL/GPUs}}$$

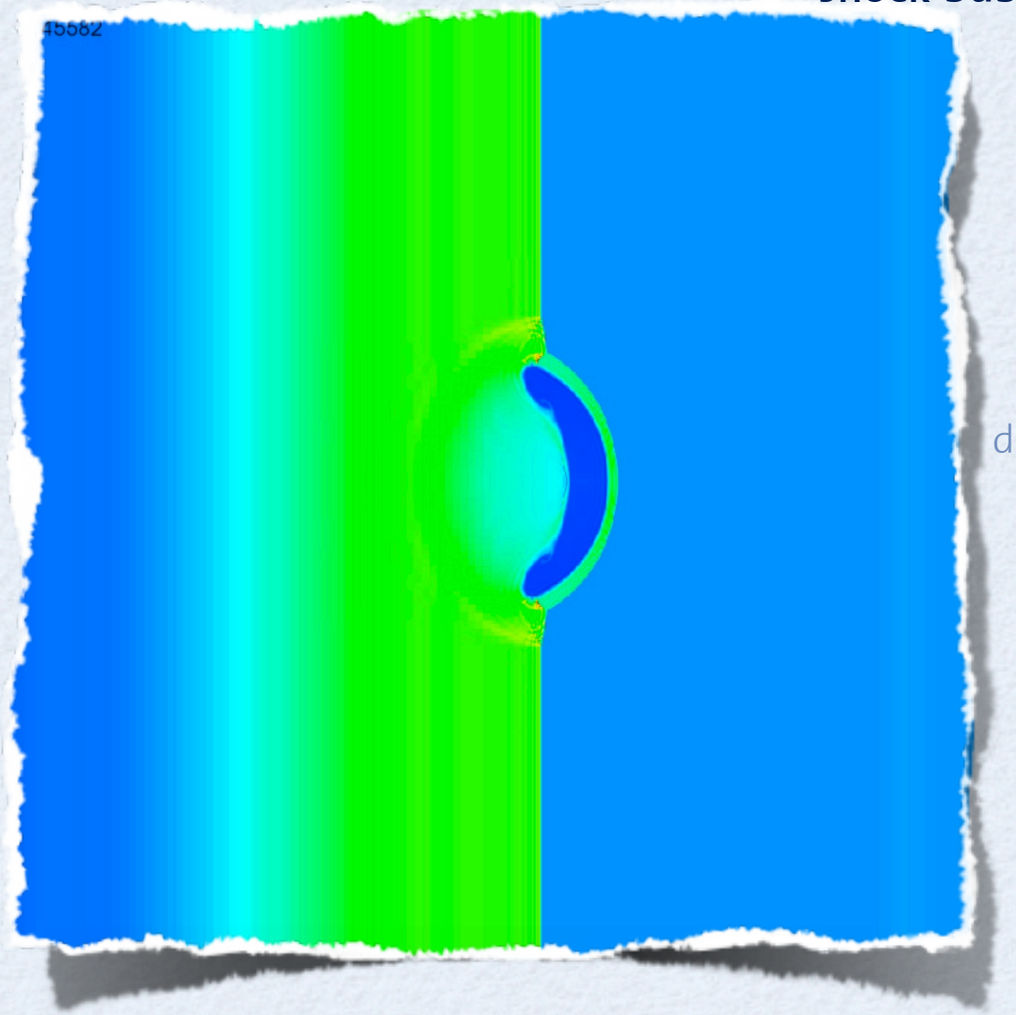
Wavelet Blocks on GPUs



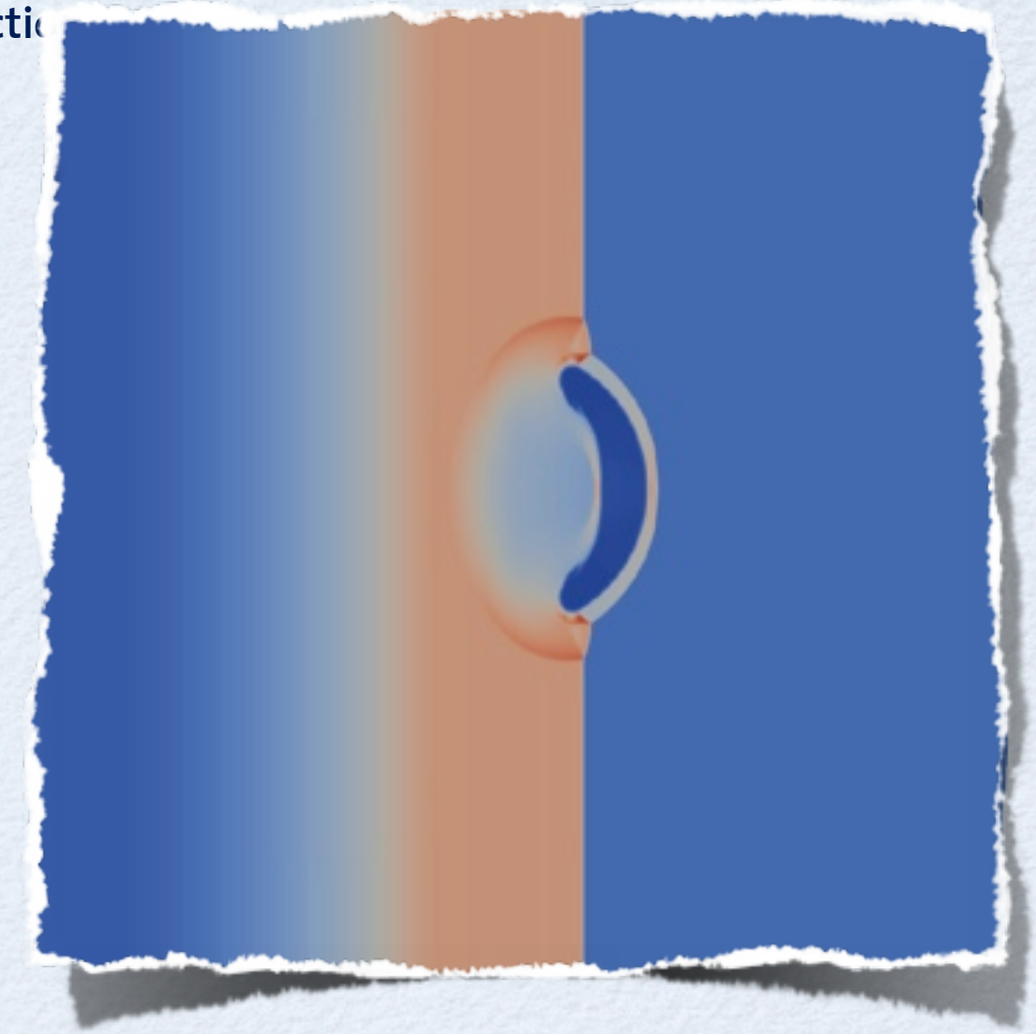
Rossinelli D., Hejazialhosseini B., Spampinato D., Koumoutsakos P., Multicore/Multi-GPU Accelerated Simulations of Multiphase Compressible Flows Using Wavelet Adapted Grids, **SIAM J. Sci. Comput.,** 33, pp. 512-540, 2011

A comparison of CHOMBO vs MRAG

shock-bubble interacti



density field



Chombo:
91 min, 230 MB

single-phase
2nd order PPM

MRAG (home grown, swiss quality stuff)**:**
56 min, 244 MB (+ 1 GPU: 7 min)

multi-phase
3rd order WENO scheme

Rossinelli D., Hejazialhosseini B., Spampinato D., Koumoutsakos P., Multicore/Multi-GPU Accelerated Simulations of Multiphase Compressible Flows Using Wavelet Adapted Grids, **SIAM J. Sci. Comput.,** 33, pp. 512-540, 2011

BOUNDARIES + ALGORITHMS

TIME : FLow AVeraging integratOR

Tao, Owhadi, & Marsden, Multiscale Model. Simul., 2010.

Stiff ODEs:

$$\dot{\mathbf{u}} = \mathbf{G}(\mathbf{u}) + \frac{1}{\epsilon}\mathbf{F}(\mathbf{u})$$

$$\epsilon \ll 1$$

with the *legacy* integration scheme:

$$\bar{\mathbf{u}}_{t+\tau} = \mathbf{\Phi}_{\tau}^{1/\epsilon}(\bar{\mathbf{u}}_t)$$

Small time-step: $\tau \ll \epsilon$

FLAVOR:

$$\bar{\mathbf{u}}_{t+h} = \left(\mathbf{\Phi}^{0}_{\frac{h}{M}-\tau} \circ \mathbf{\Phi}^{1/\epsilon}_{\tau}\right)^{M} (\bar{\mathbf{u}}_t)$$

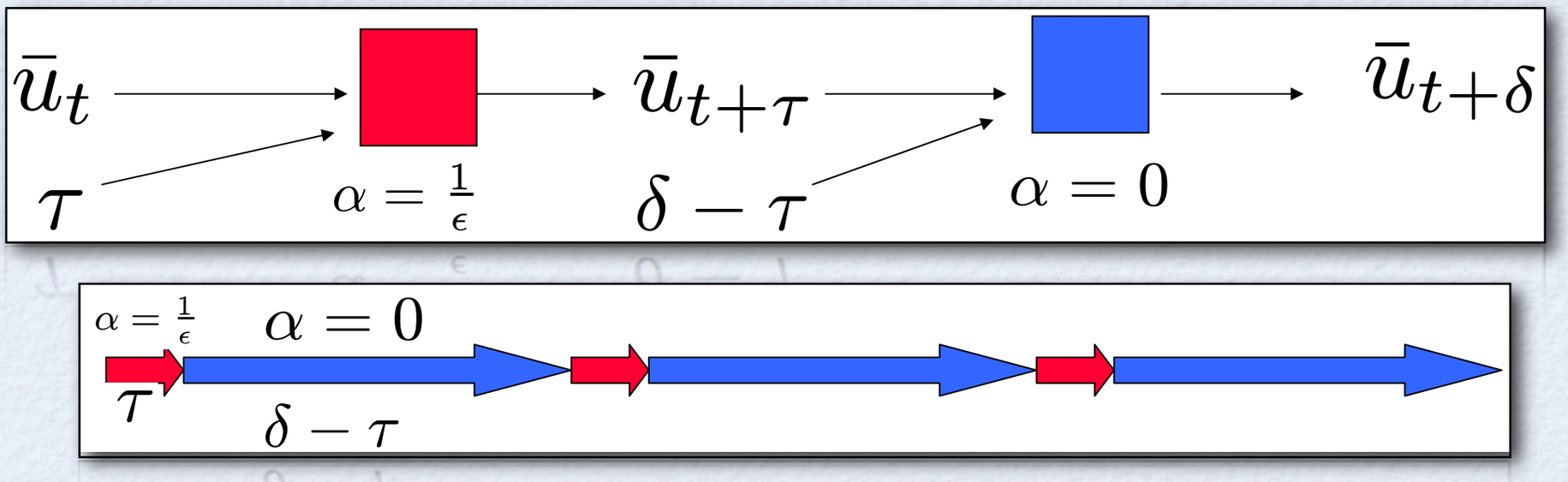
$\frac{1}{\epsilon}$:	OFF	ON

Large time-step: $\delta = \frac{h}{M}$

M regulates accuracy of large time-step

$$0 < \tau \ll \epsilon \ll \delta \ll 1$$

turn on/off large coefficients:



FLAVOR + Stiff Stochastics

SSA written as $(\mathbf{X}_n, t_n) = \left(\mathbf{\Phi}^{(1/\epsilon)}\right)(\mathbf{X}_{n-1}, t_{n-1})$

Define FLAVOR-SSA:

$$(\mathbf{X}_n, t_n) = \left(\mathbf{\Phi}^{(\xi/\epsilon)} \circ \mathbf{\Phi}^{(1/\epsilon)}\right)(\mathbf{X}_{n-1}, t_{n-1})$$

amounts to rescaling the total propensity every other iteration

$$\delta = \delta(\xi) \sim \mathcal{E}\left(\frac{1}{\hat{a}_0}\right)$$

$$\hat{a}_0(t, \xi) := \frac{\xi}{\epsilon} \sum_i \tilde{a}_i^{(fast)}(t) + \sum_j a_j^{(slow)}(t)$$

$$\xi \in [0, 1]$$

$\xi = 0 \implies$ Largest Speedup

$\xi = 1 \implies$ SSA

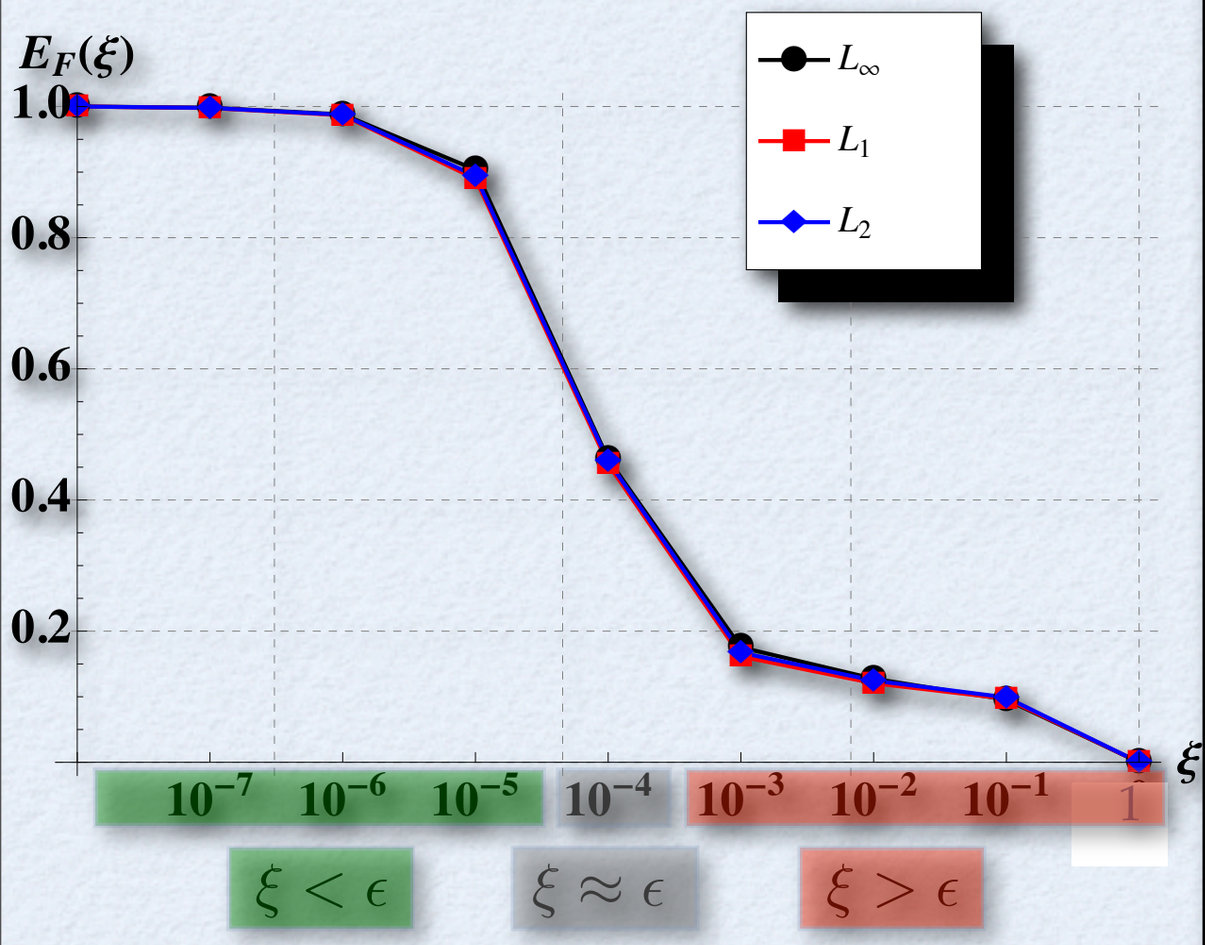
OPTIMAL ~value?

FLAVOR -S : A Cutoff Phenomenon

$$S_1 \overset{\mathcal{O}(1/\epsilon)}{\longleftrightarrow} S_2 \overset{\mathcal{O}(1)}{\longleftrightarrow} S_3$$

$$\epsilon = 10^{-4}$$

Error vs. ξ

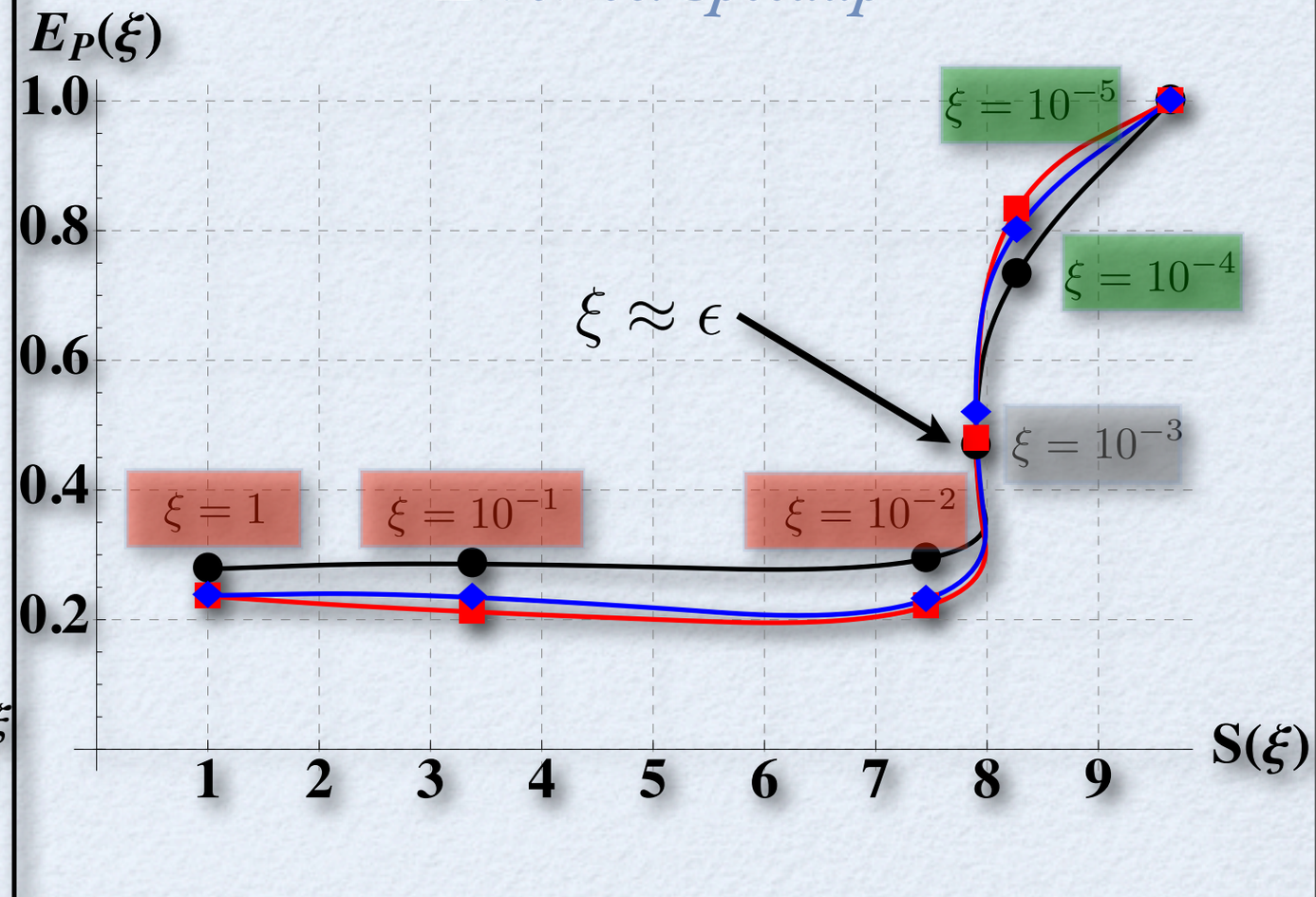


$$S_1 \longleftrightarrow S_2$$

$$S_1 \longleftrightarrow S_3$$

$$2S_2 + S_3 \longleftrightarrow 3S_4$$

Error vs. Speedup

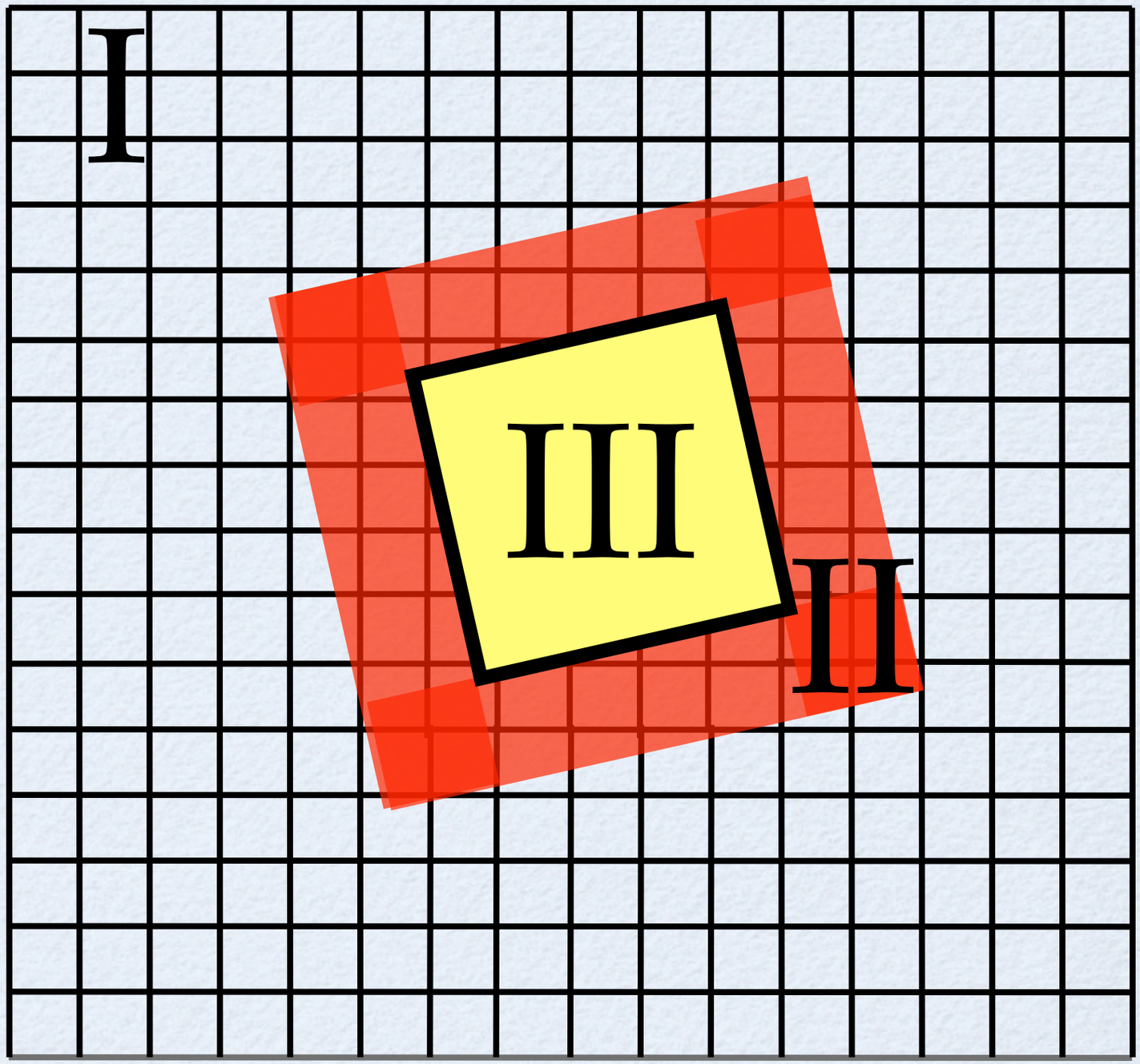


Bayati B., Owhadi H., Koumoutsakos P, J. Chem. Phys.,2010

Boundary Conditions = Coupling

COUPLING Different Physics in Space/Time

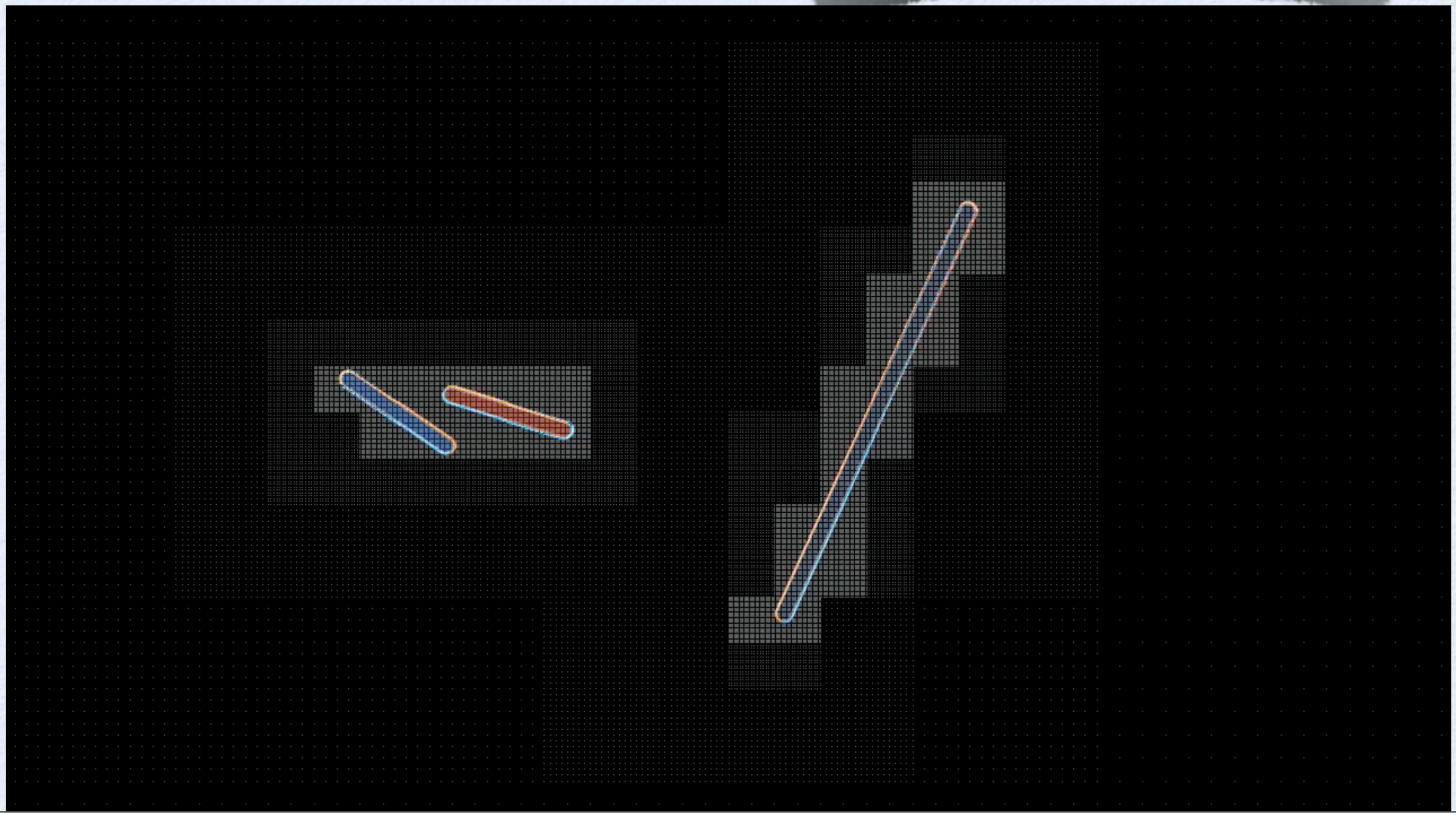
	I	II	III
No Slip	FLUID	Sharp Force	-
No Slip + PENALTY/IBM	Fluid	Smooth Force	-
Flow-Structure	Fluid	Smooth/Sharp Force	Solid
Multiscale I	Fluid	Fluid+MD	MD
Multiscale II	Fluid	Fluid+DPD	DPD+MD



COMPRESSIBLE FLOWS

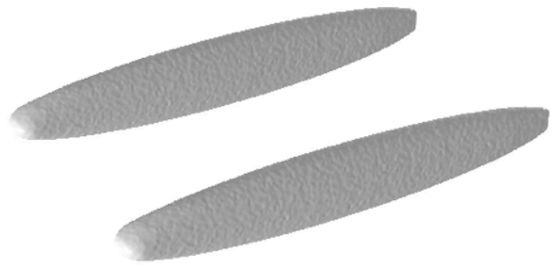
Moving Boundaries

Brinkman Penalization for Compressible Flow



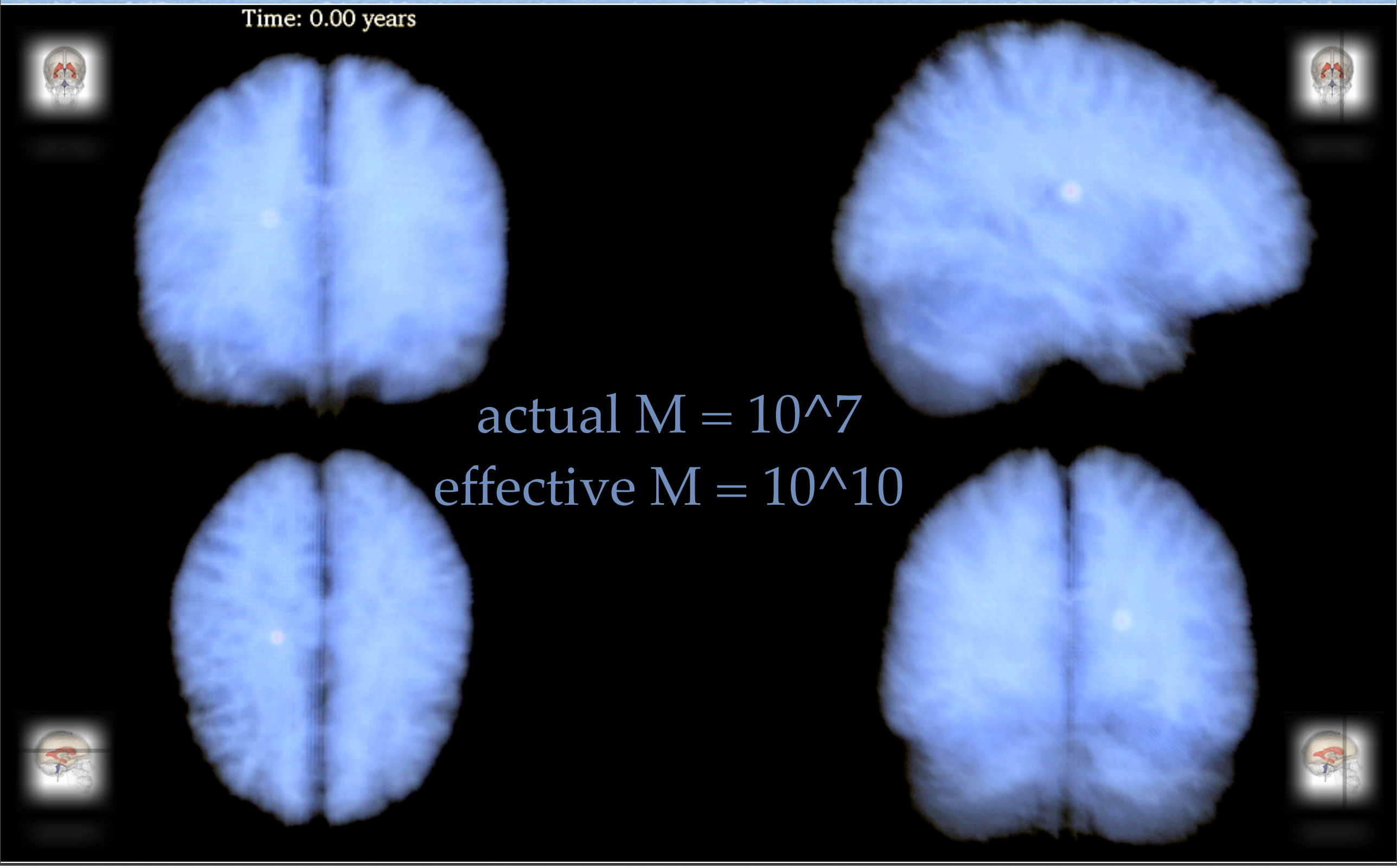
FISH SCHOOLING

1 FISH



2 FISH (OBVIOUSLY)

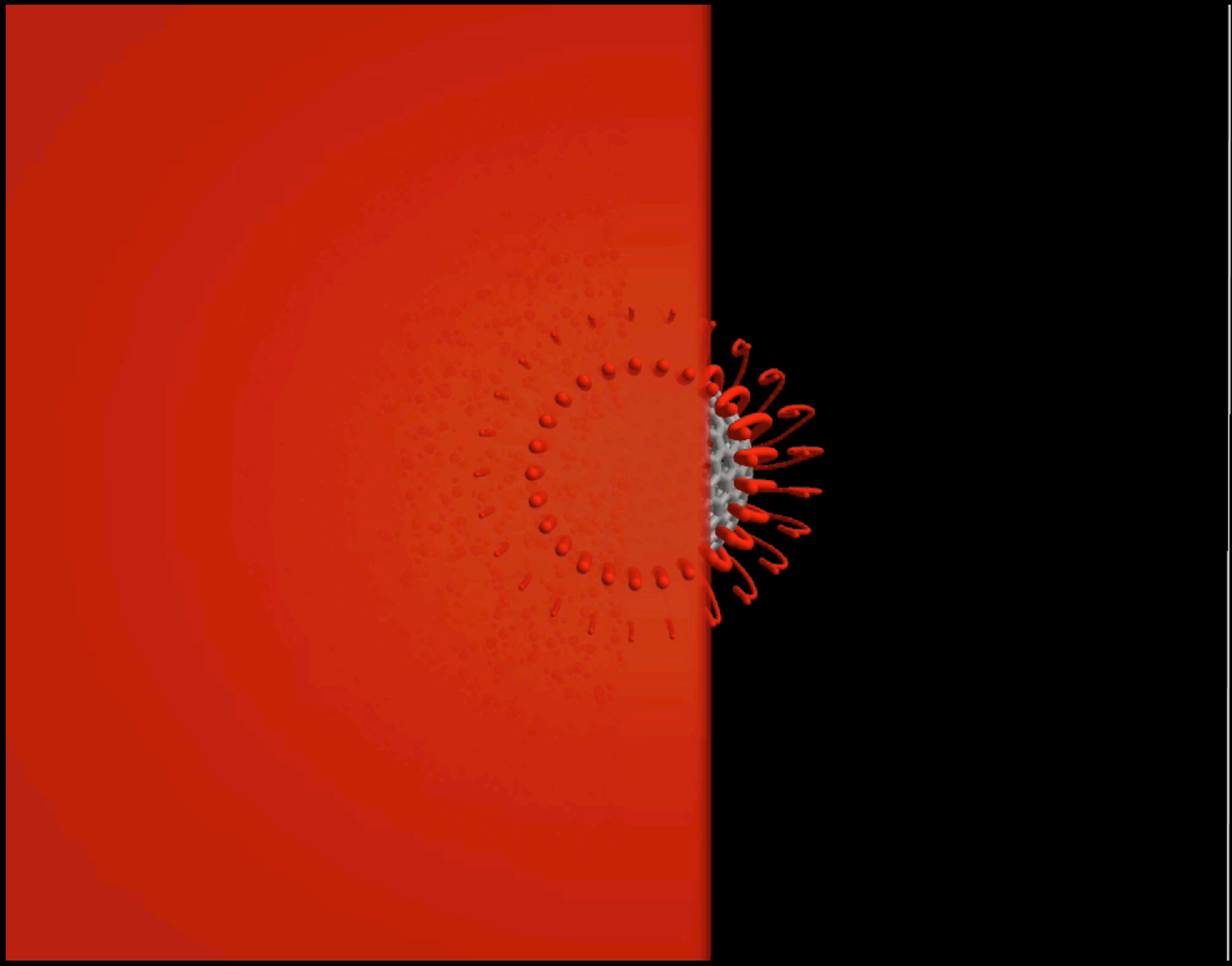
Simulations of Gliomas using MRAG



MULTISCALE METHODS

CORE CORE GPU
CORE CORE GPU

MD - Lattice-Boltzmann



COMPUTATIONAL Challenges

"Although **X** seem to be a small step from **Y** computation, in fact, they represent a huge step. They discard the most essential and appealing properties of **Y** computation:

understandability, predictability, and determinism.

X as a model of computation, are wildly nondeterministic, and the job of the programmer becomes one of pruning that nondeterminism."

-- *'The Problem with **X**, Edward A. Lee, UC Berkeley, 2006*

X = threads **Y** = sequential

X = Multi-scale **Y** = single-scale


Bergdorf


Bayati


Rossinelli


Chatelain


Hedjazialhosseini