Multiscale Simulations Using Particles

Petros Koumoutsakos

KITP

June 5, 2012

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Troy, The movie

MULTISCALING as CAUSALITY

MULTISCALING as CAUSALITY

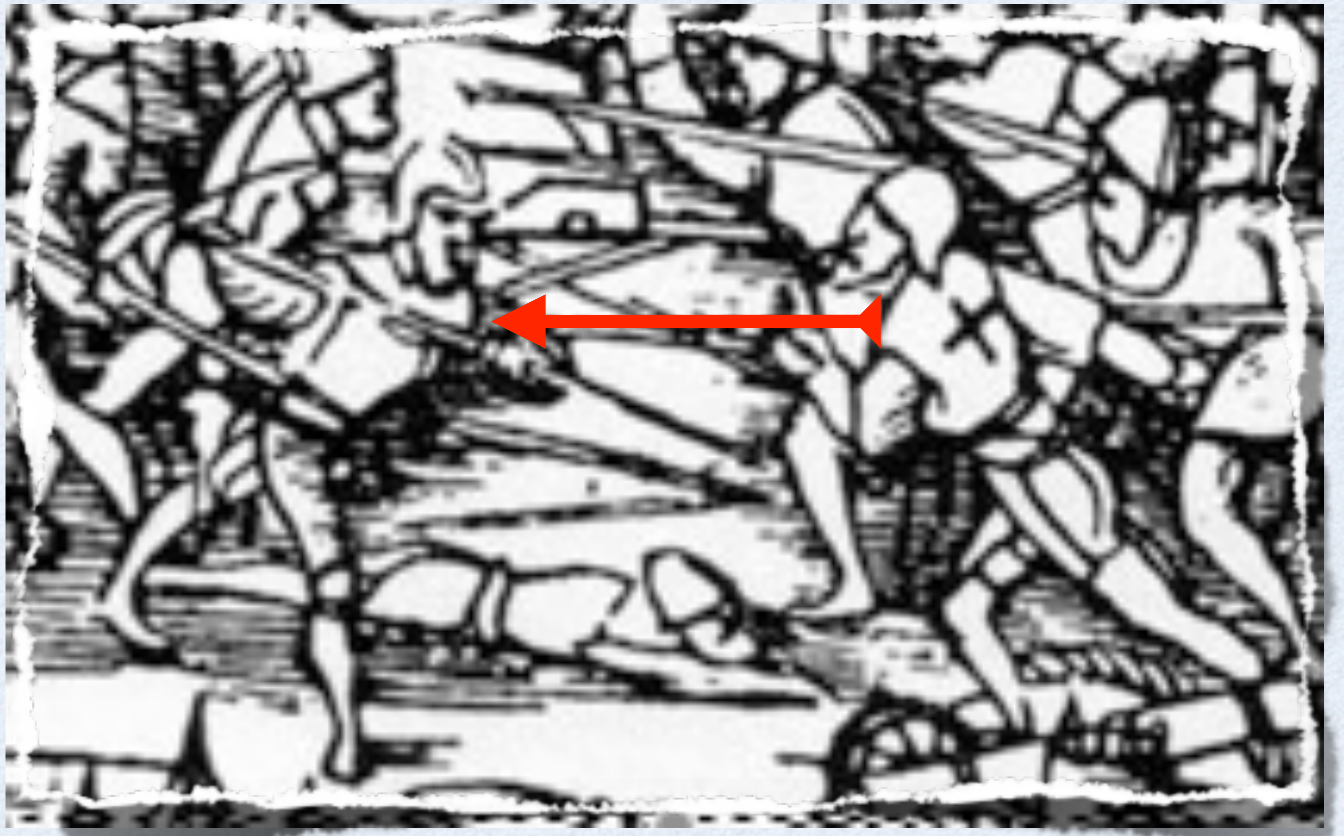
adapted from Robert E. Ulanowicz

MULTISCALING as CAUSALITY

adapted from Robert E. Ulanowicz

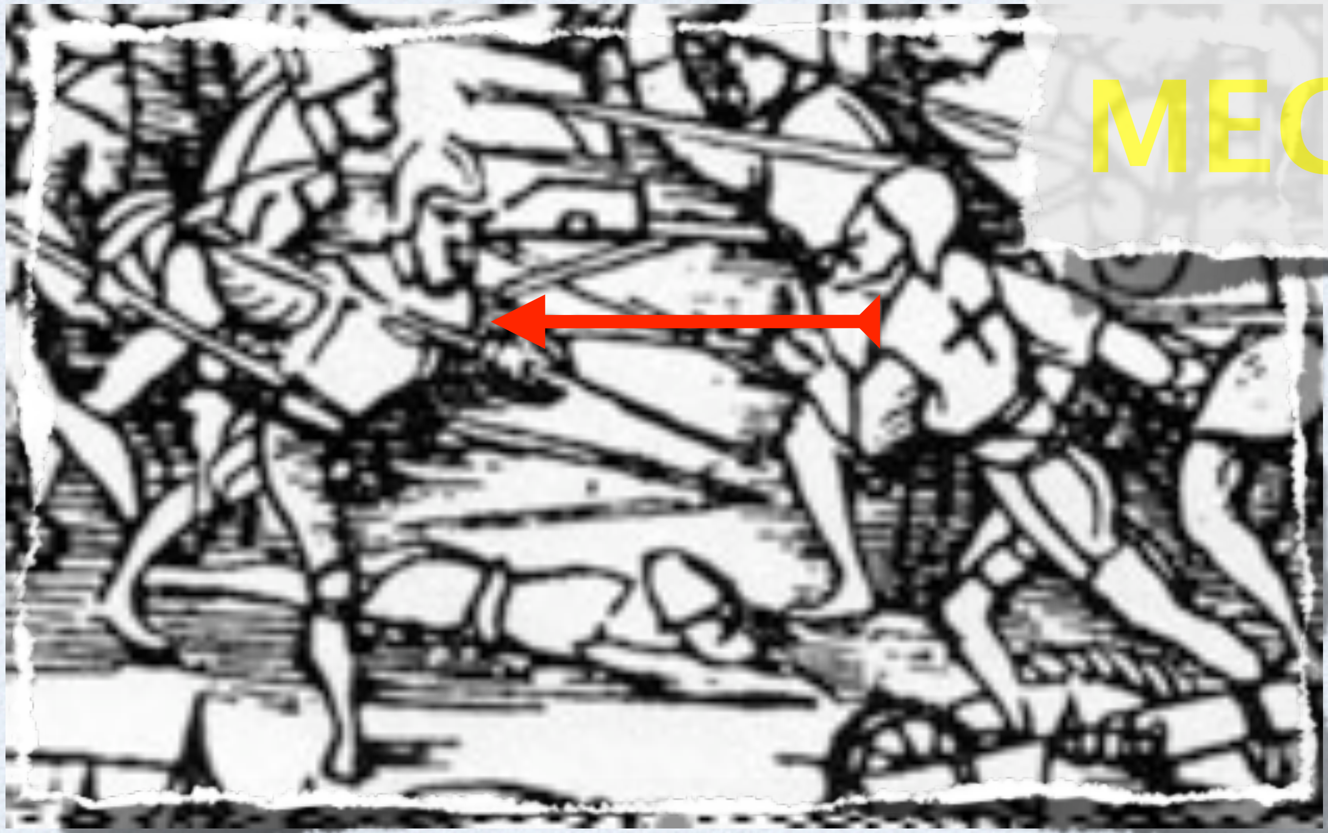
MULTISCALING as CAUSALITY

adapted from Robert E. Ulanowicz



MULTISCALING as CAUSALITY

adapted from Robert E. Ulanowicz

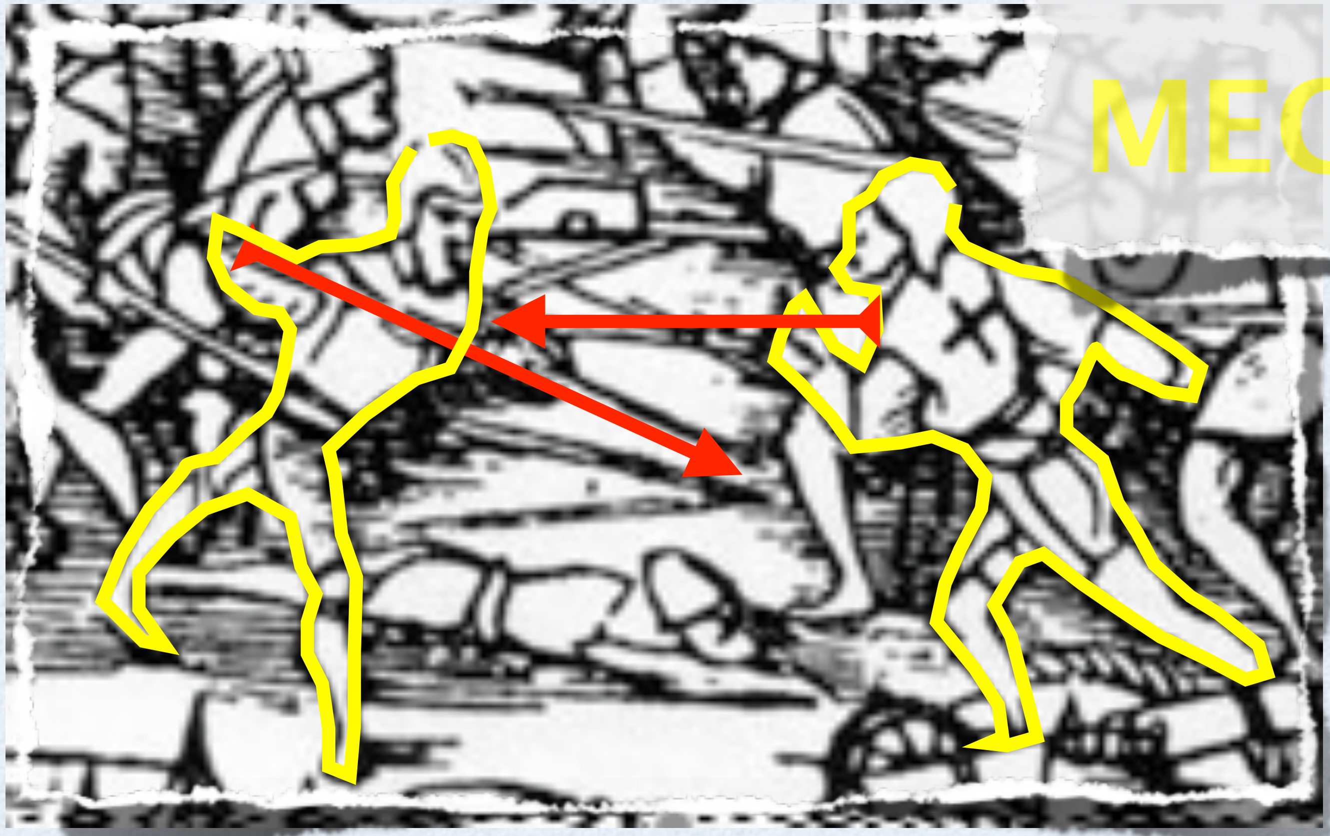


MATERIAL

MECHANICAL

MULTISCALING as CAUSALITY

adapted from Robert E. Ulanowicz



MATERIAL

MECHANICAL

MULTISCALING as CAUSALITY

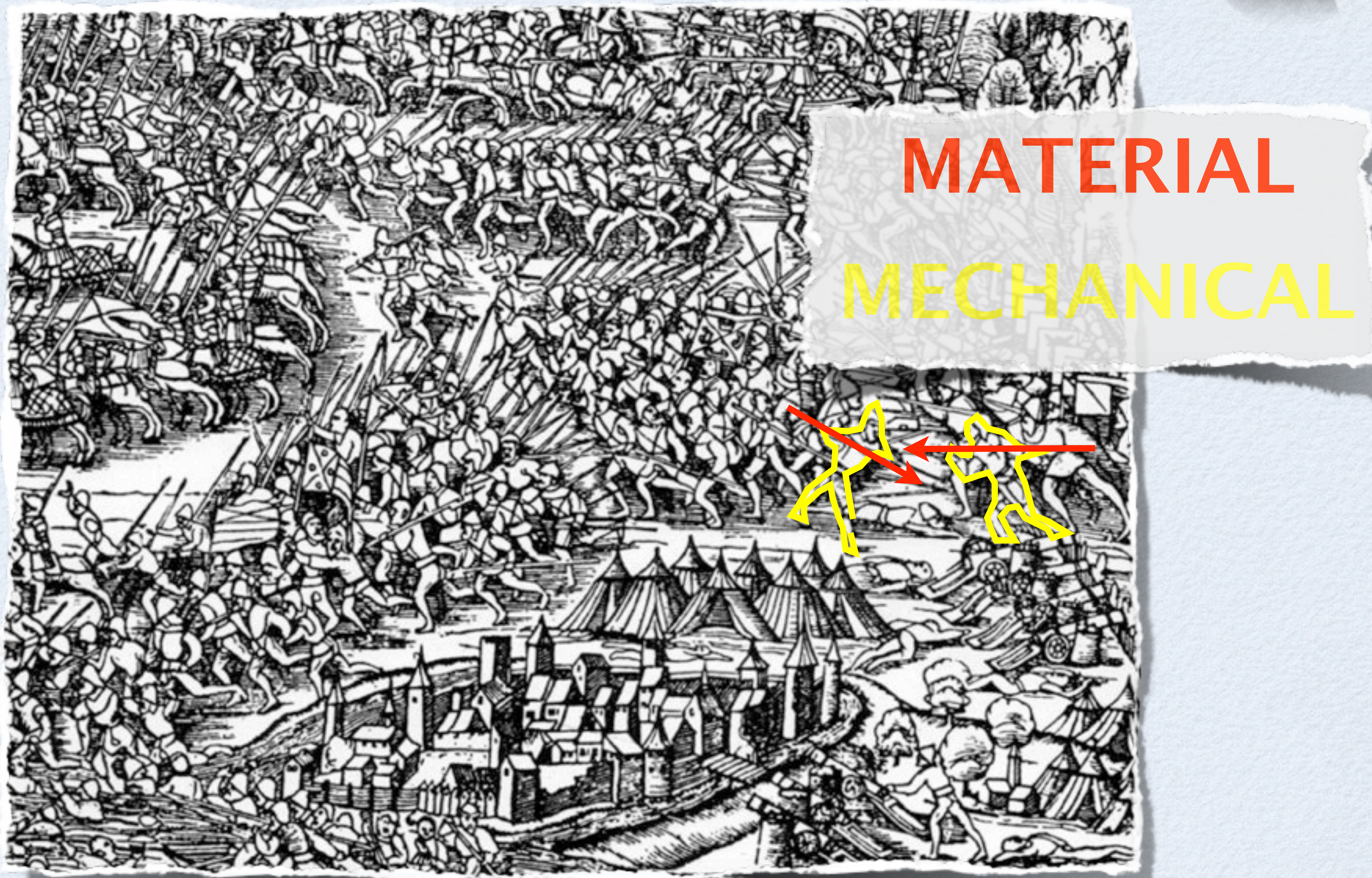
adapted from Robert E. Ulanowicz

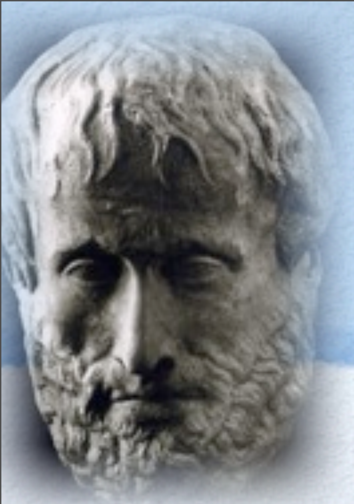
MATERIAL

MECHANICAL

MULTISCALING as CAUSALITY

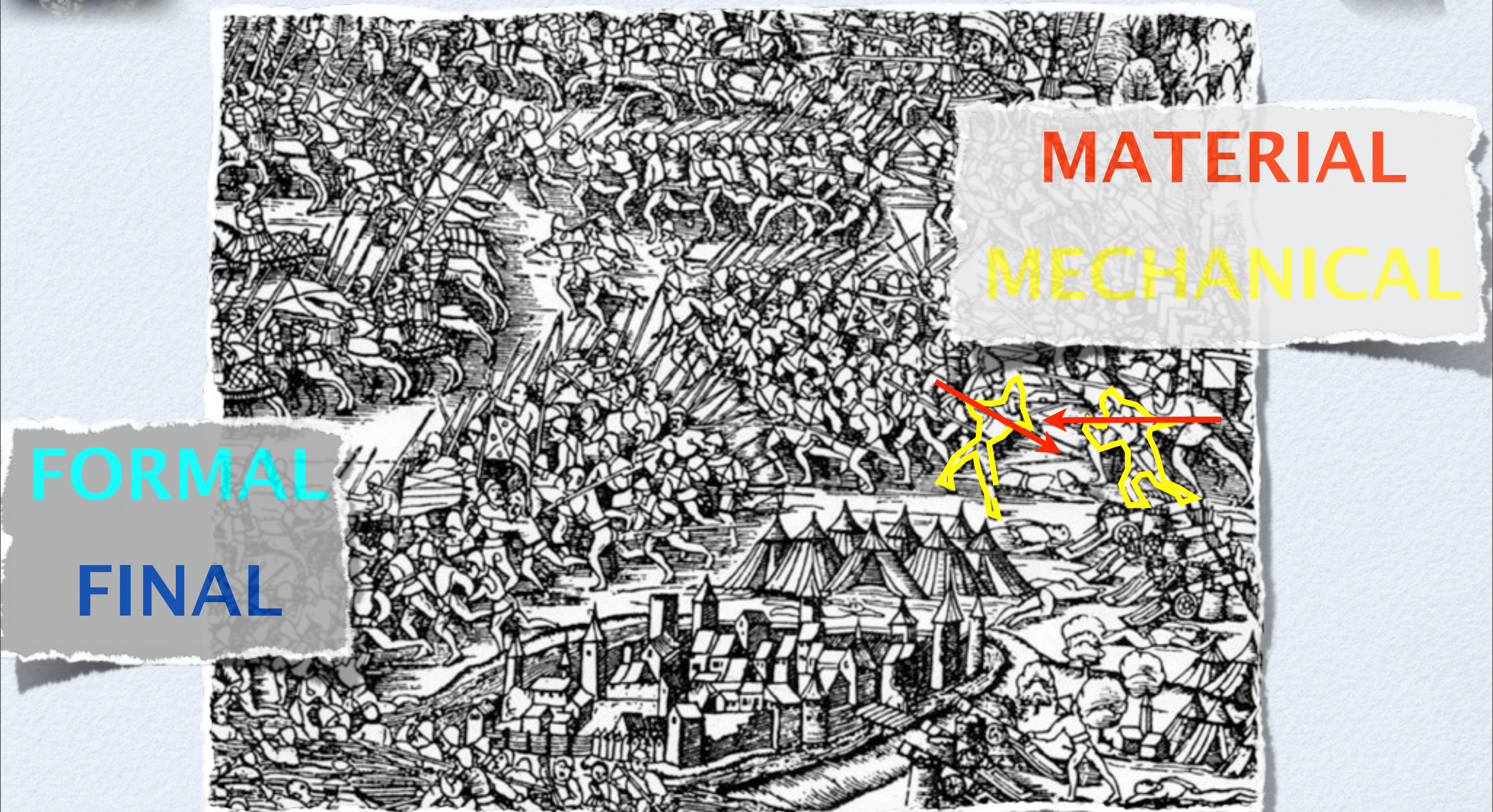
adapted from Robert E. Ulanowicz

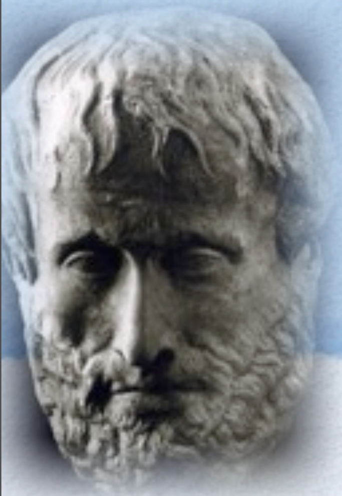




MULTISCALING as CAUSALITY

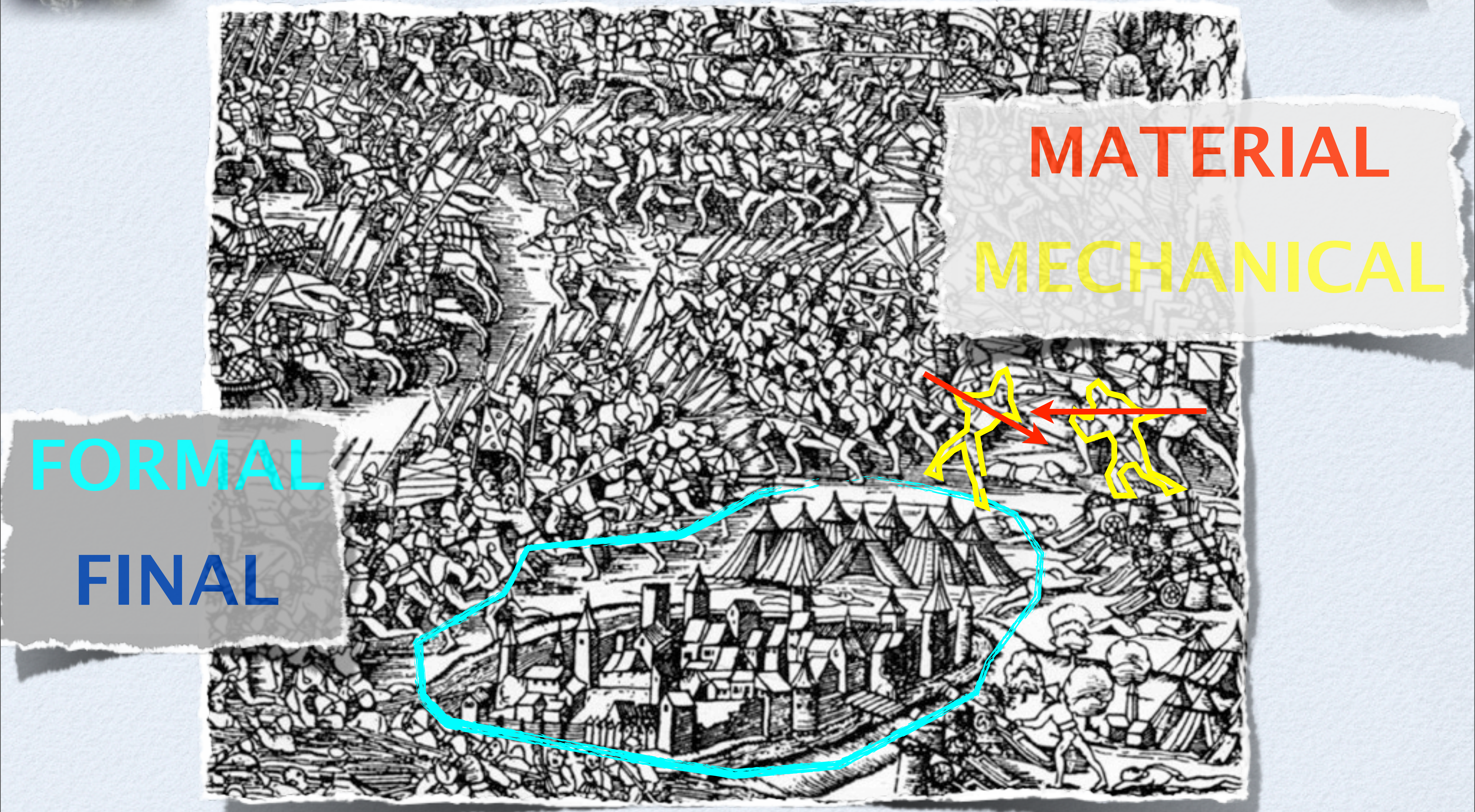
adapted from Robert E. Ulanowicz





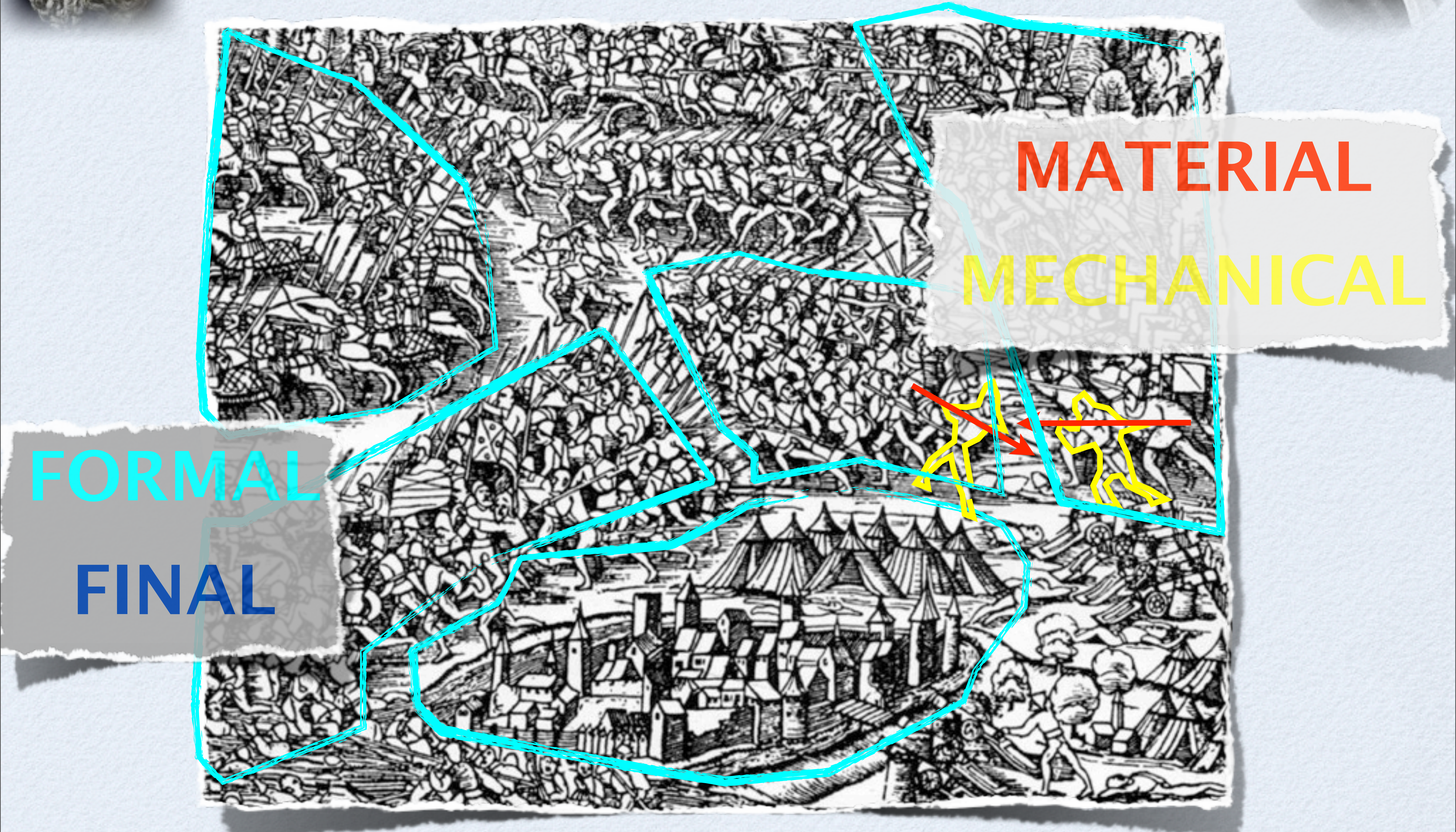
MULTISCALING as CAUSALITY

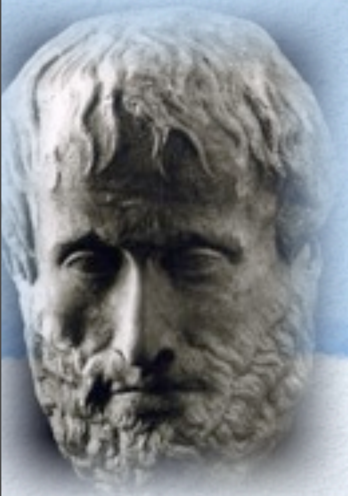
adapted from Robert E. Ulanowicz



MULTISCALING as CAUSALITY

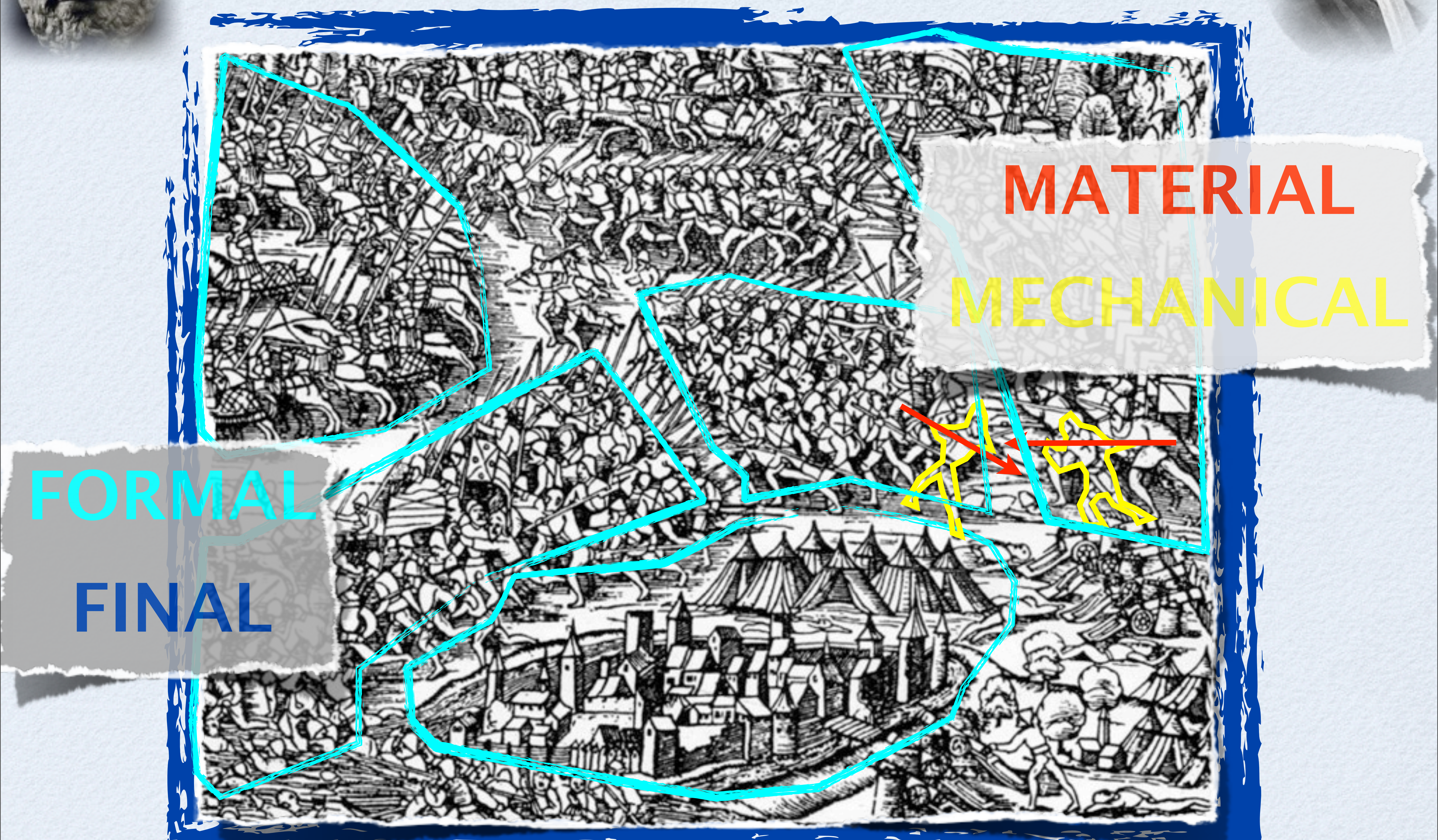
adapted from Robert E. Ulanowicz





MULTISCALING as CAUSALITY

adapted from Robert E. Ulanowicz



OUTLINE

OUTLINE

- INTRODUCTION

OUTLINE

- INTRODUCTION
  - Particles: a superset of Grids

OUTLINE

- INTRODUCTION
  - Particles: a superset of Grids
- PARTICLES AND MULTI-RESOLUTION

OUTLINE

- INTRODUCTION
  - Particles: a superset of Grids
- PARTICLES AND MULTI-RESOLUTION
  - Particles + Wavelets

OUTLINE

- INTRODUCTION
  - Particles: a superset of Grids
- PARTICLES AND MULTI-RESOLUTION
  - Particles + Wavelets
- BOUNDARIES AND INTERFACES

OUTLINE

- INTRODUCTION
  - Particles: a superset of Grids
- PARTICLES AND MULTI-RESOLUTION
  - Particles + Wavelets
- BOUNDARIES AND INTERFACES
  - A bridge between Flux and Domain Decomposition Algorithms

OUTLINE

- INTRODUCTION
  - Particles: a superset of Grids
- PARTICLES AND MULTI-RESOLUTION
  - Particles + Wavelets
- BOUNDARIES AND INTERFACES
  - A bridge between Flux and Domain Decomposition Algorithms
- RESULTS and OUTLOOK

PARTICLES : Lagrangian, Conservation and Other Laws

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

PARTICLES : Lagrangian, Conservation and Other Laws

SPH, Vortex Methods

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

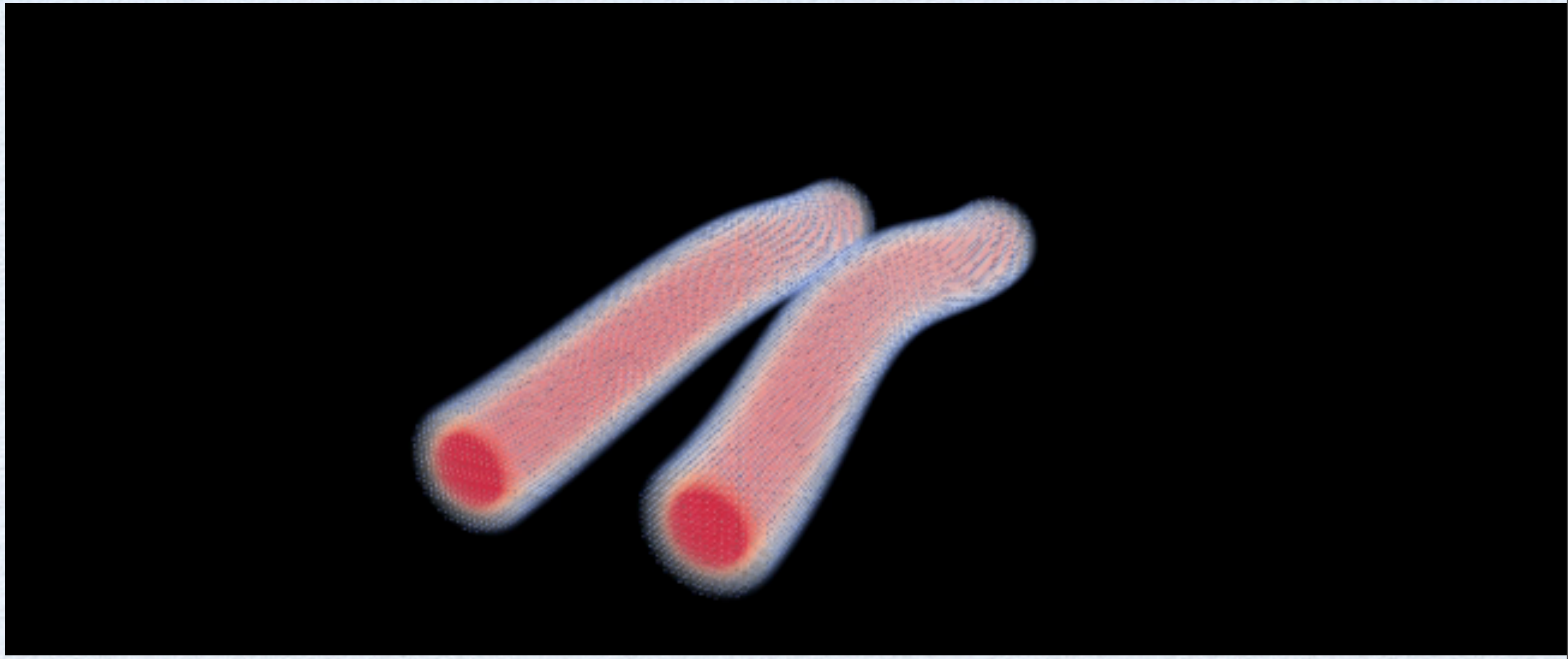
$$\frac{d\mathbf{x_p}}{dt} = \mathbf{u}_p$$

PARTICLES : Lagrangian, Conservation and Other Laws

SPH, Vortex Methods

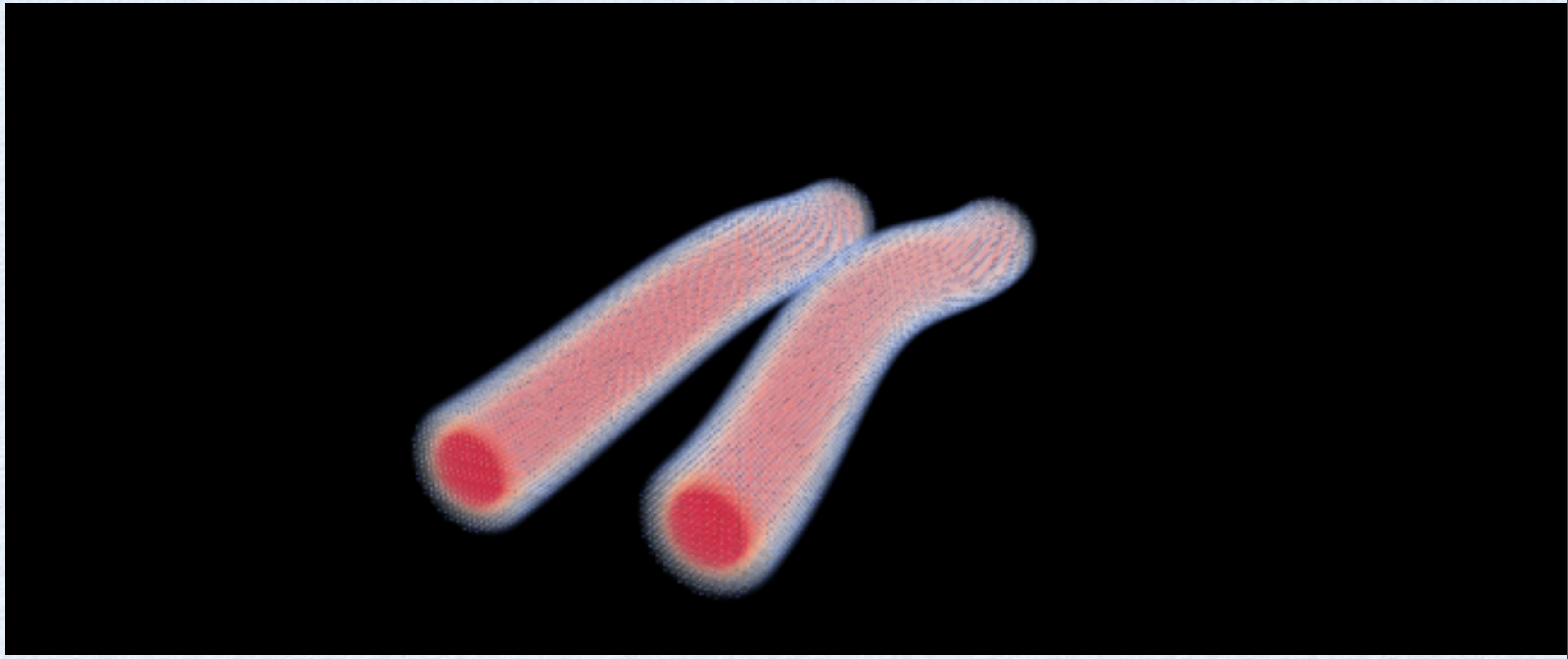
$$\rho_p \frac{D\mathbf{u_p}}{Dt} = (\nabla \cdot \sigma)_p$$

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



PARTICLES : Lagrangian, Conservation and Other Laws

SPH, Vortex Methods



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

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

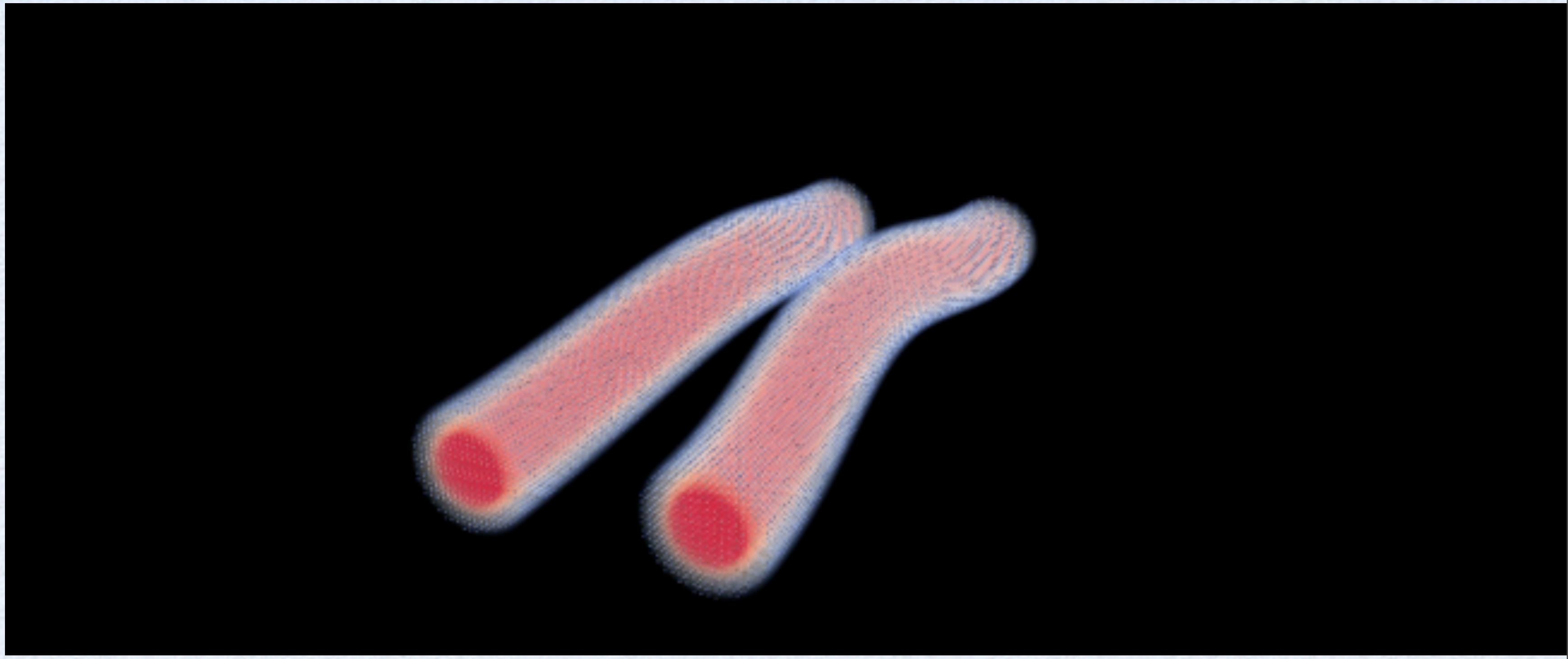
$$m \frac{d\mathbf{u_p}}{dt} = F_p$$

MD, DPD, CGMD

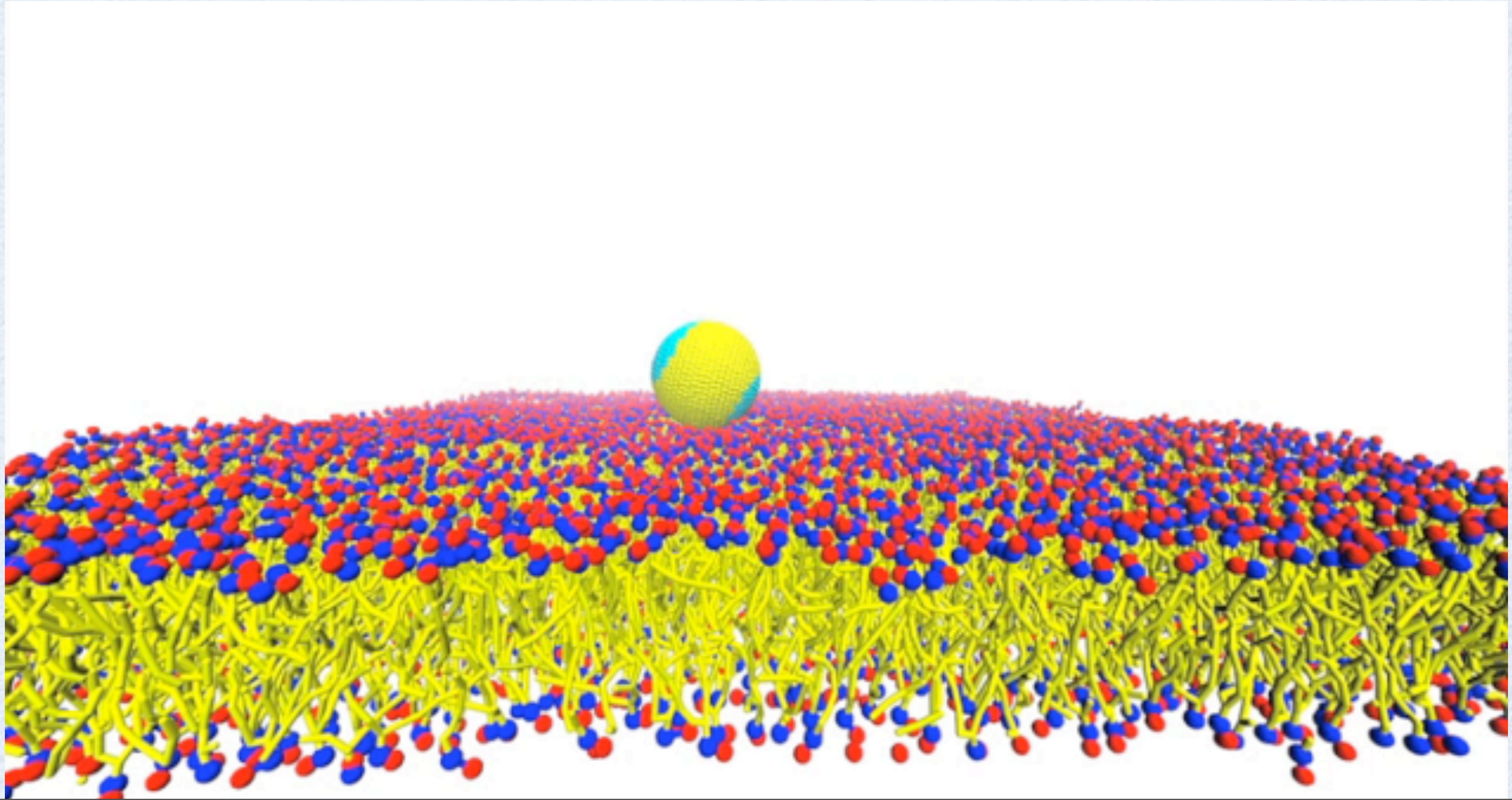
PARTICLES : Lagrangian, Conservation and Other Laws

SPH, Vortex Methods

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



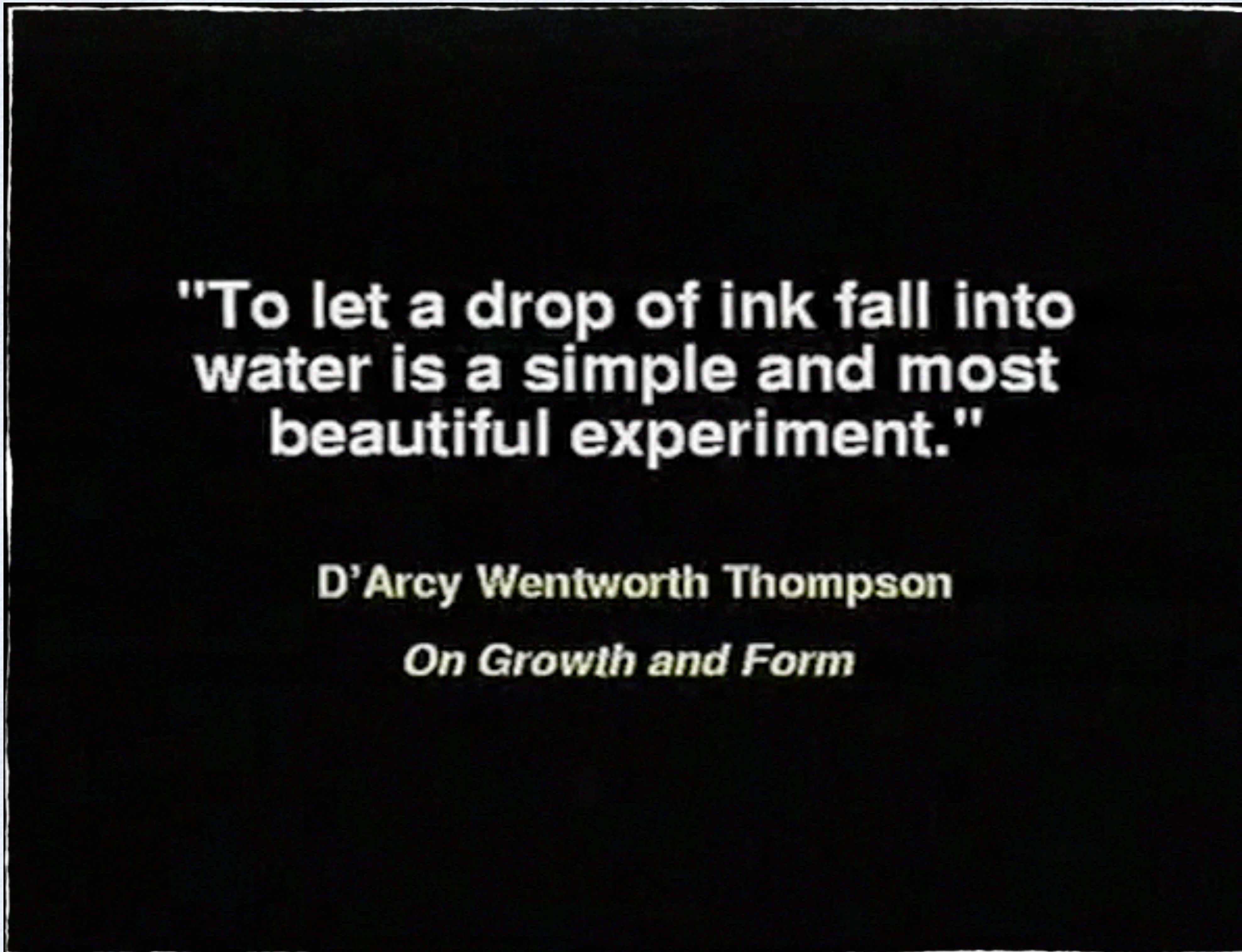
$$\frac{d\mathbf{x_p}}{dt} = \mathbf{u}_p$$



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

MD, DPD, CGMD

MODELING – APPROXIMATION

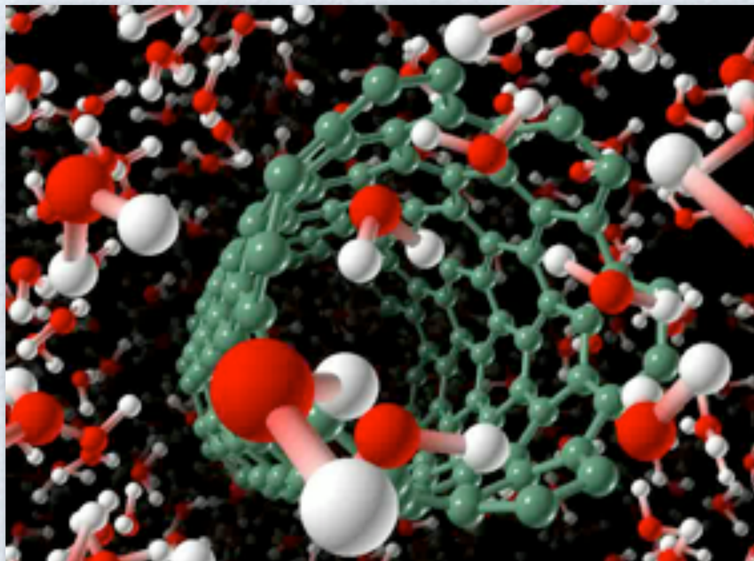


J. H. Walther, P. Koumoutsakos, Three-dimensional vortex methods for particle-laden flows with two-way coupling, **J. Comput. Phys.**, 167, 39-71, 2001 multi Scale **Simulations using Particles**

PK, **Ann. Rev. Fluid Mechanics**, **2005**

multi Scale Simulations using Particles

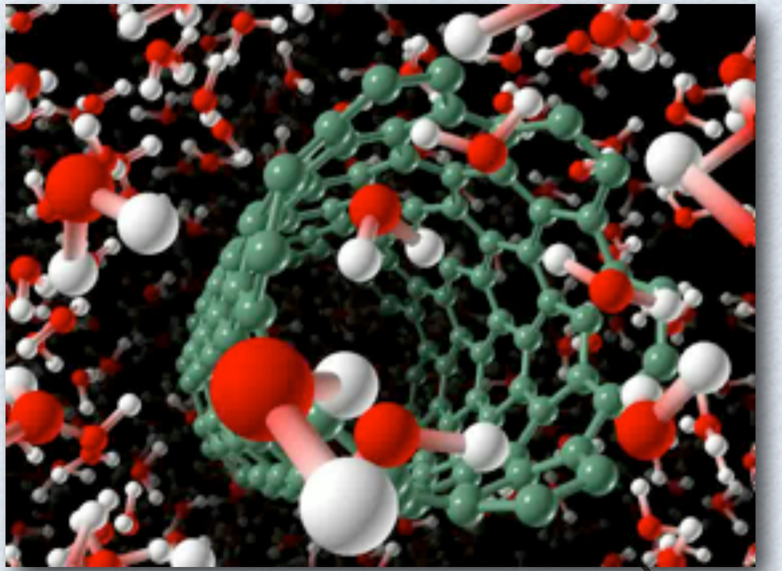
PK, **Ann. Rev. Fluid Mechanics**, **2005**



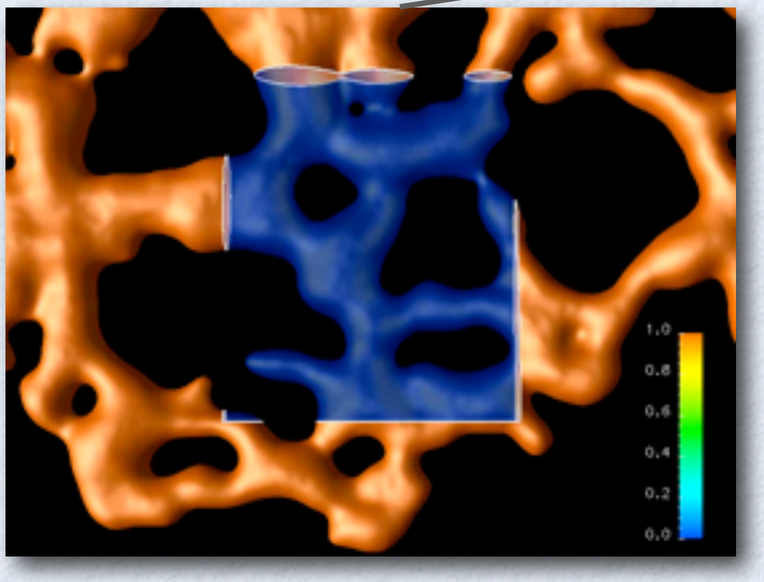
Water and CNTs

multi Scale Simulations using Particles

PK, **Ann. Rev. Fluid Mechanics**, **2005**



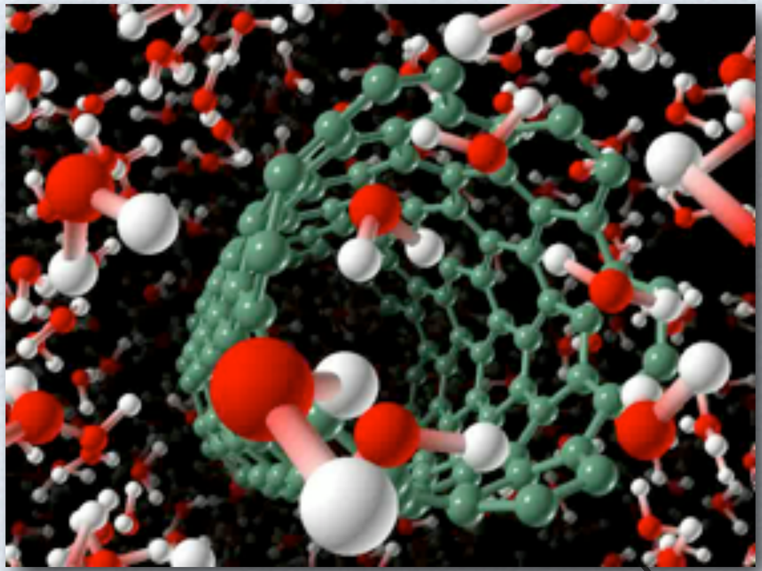
Water and CNTs



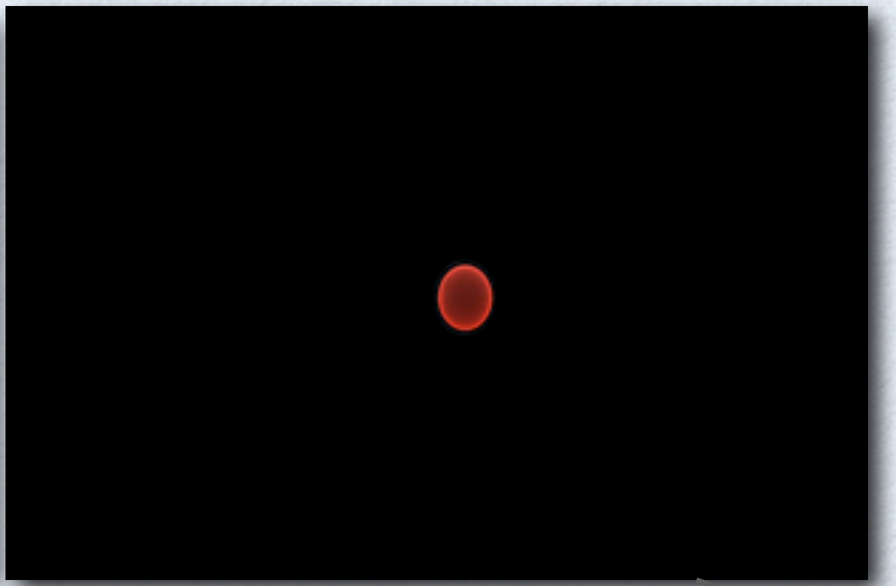
Diffusion in/on Cell Organelles

multi Scale Simulations using Particles

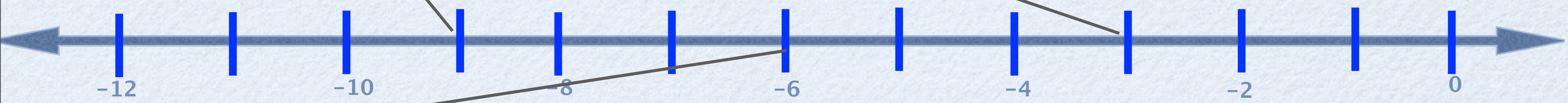
PK, **Ann. Rev. Fluid Mechanics, 2005**

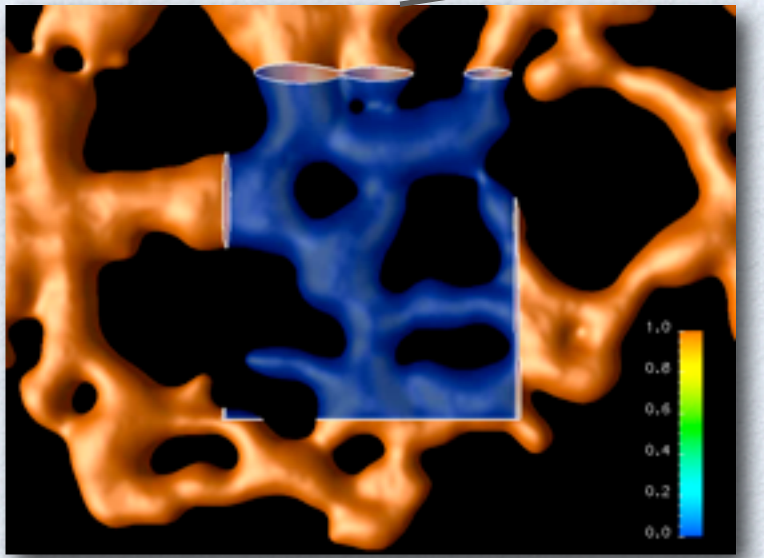


Water and CNTs



Cell Proliferation

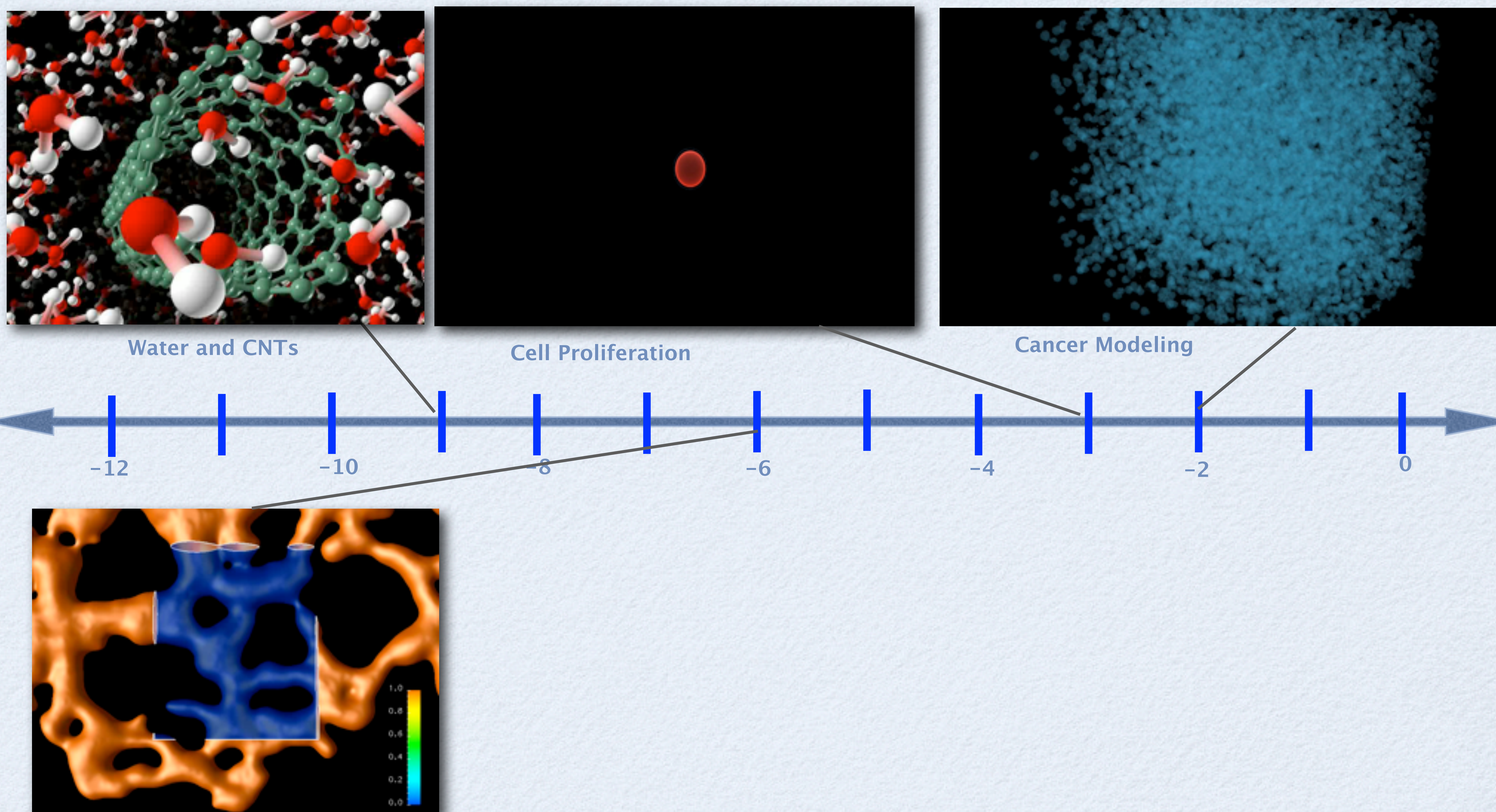




Diffusion in/on Cell Organelles

multi Scale Simulations using Particles

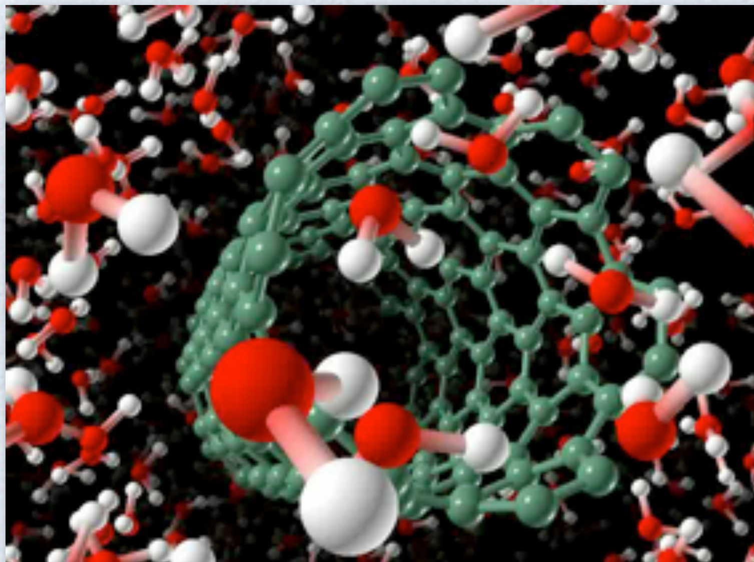
PK, **Ann. Rev. Fluid Mechanics**, **2005**



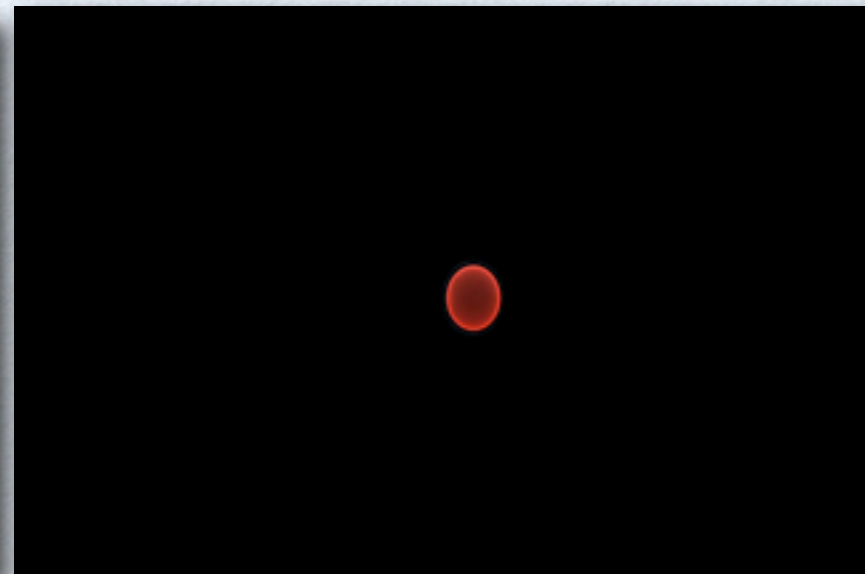
Diffusion in/on Cell Organelles

multi Scale Simulations using Particles

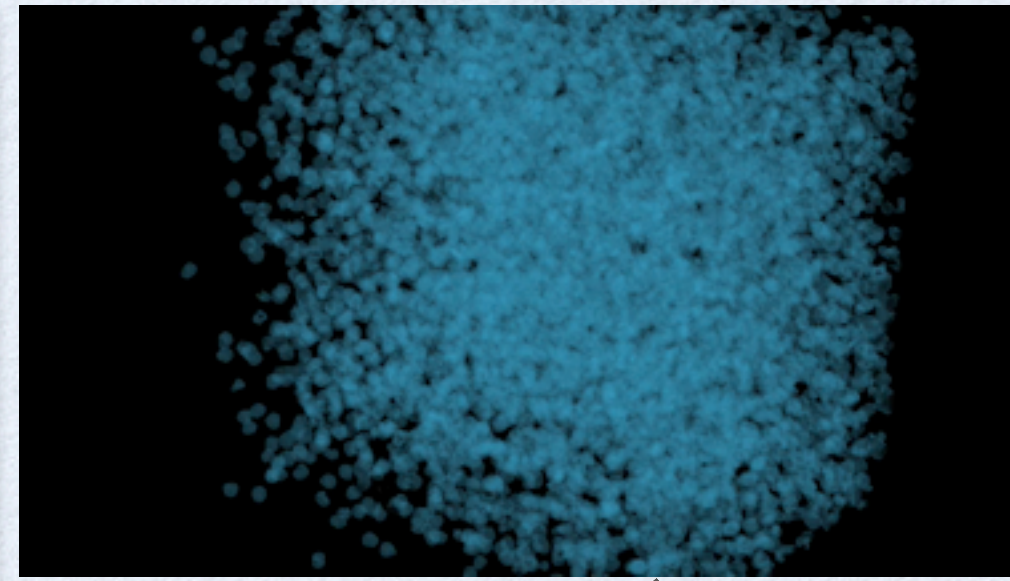
PK, **Ann. Rev. Fluid Mechanics**, **2005**



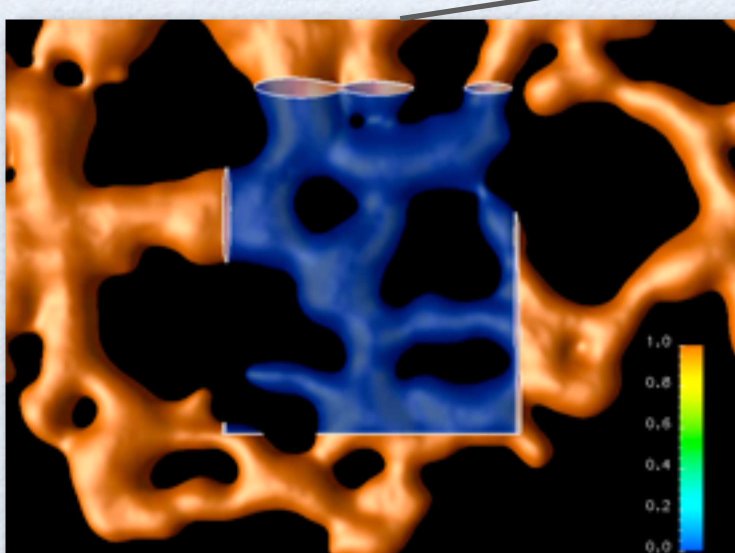
Water and CNTs



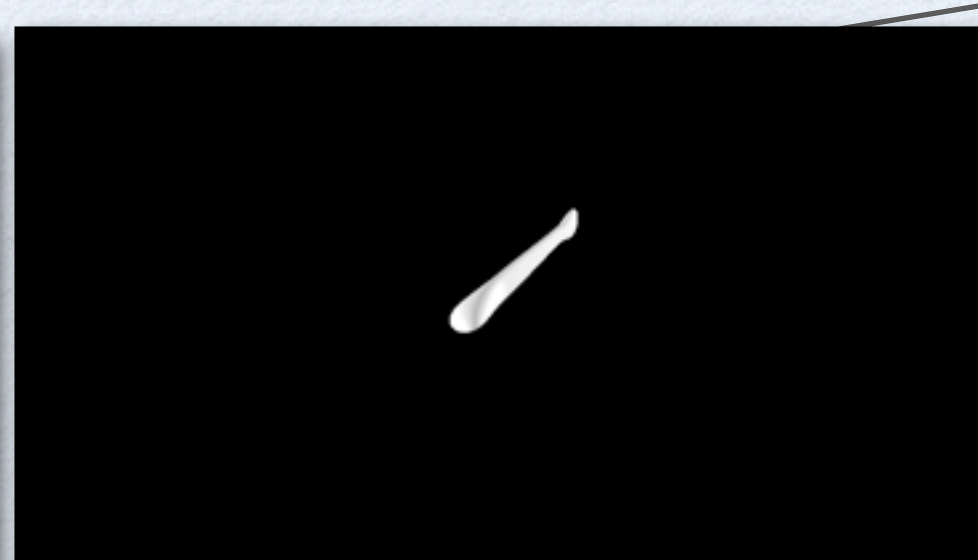
Cell Proliferation



Cancer Modeling



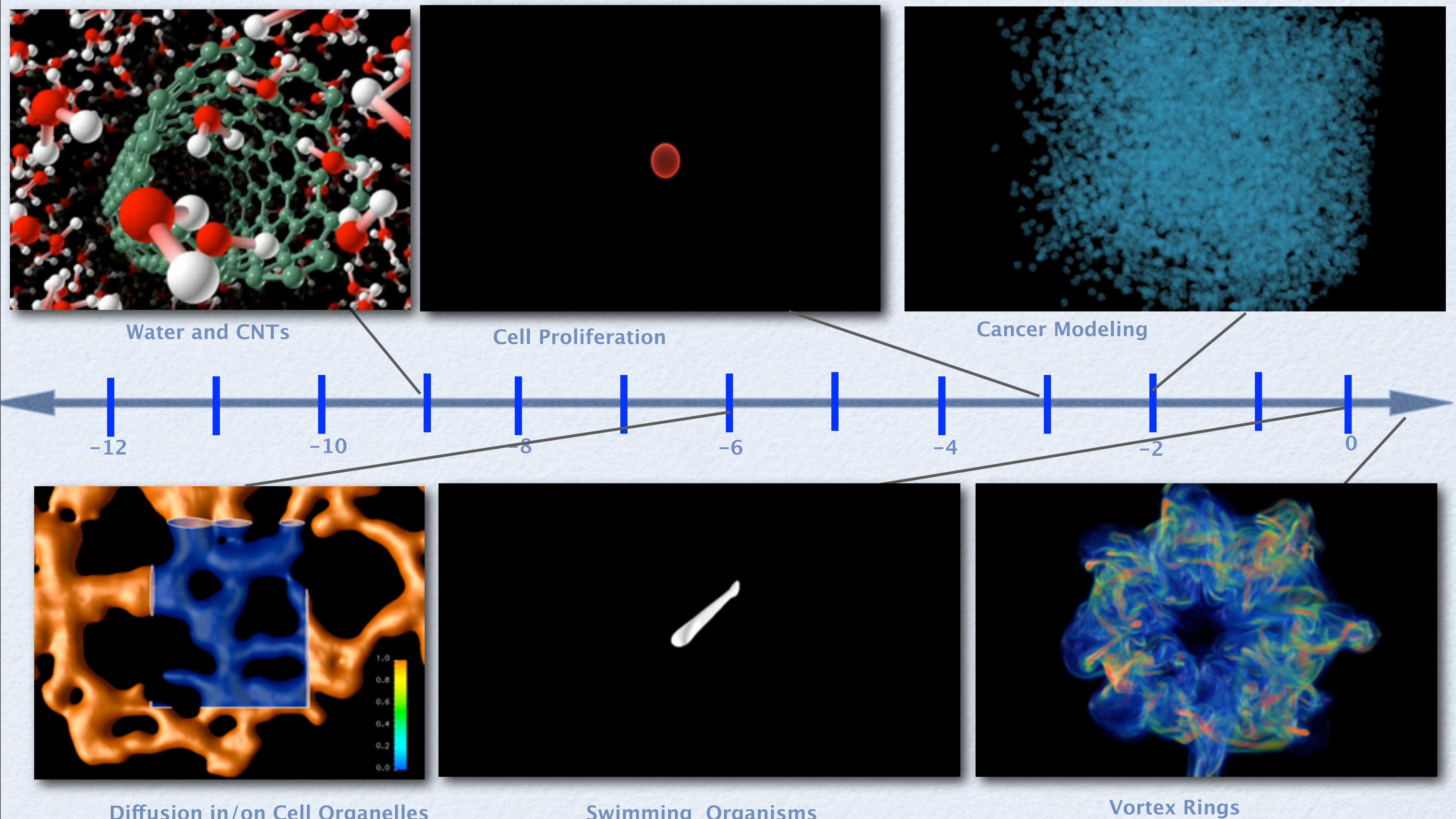
Diffusion in/on Cell Organelles



Swimming Organisms

multi Scale Simulations using Particles

PK, **Ann. Rev. Fluid Mechanics**, **2005**



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

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

FUNCTIONS and PARTICLES

Integral Function Representation

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

Function Mollification

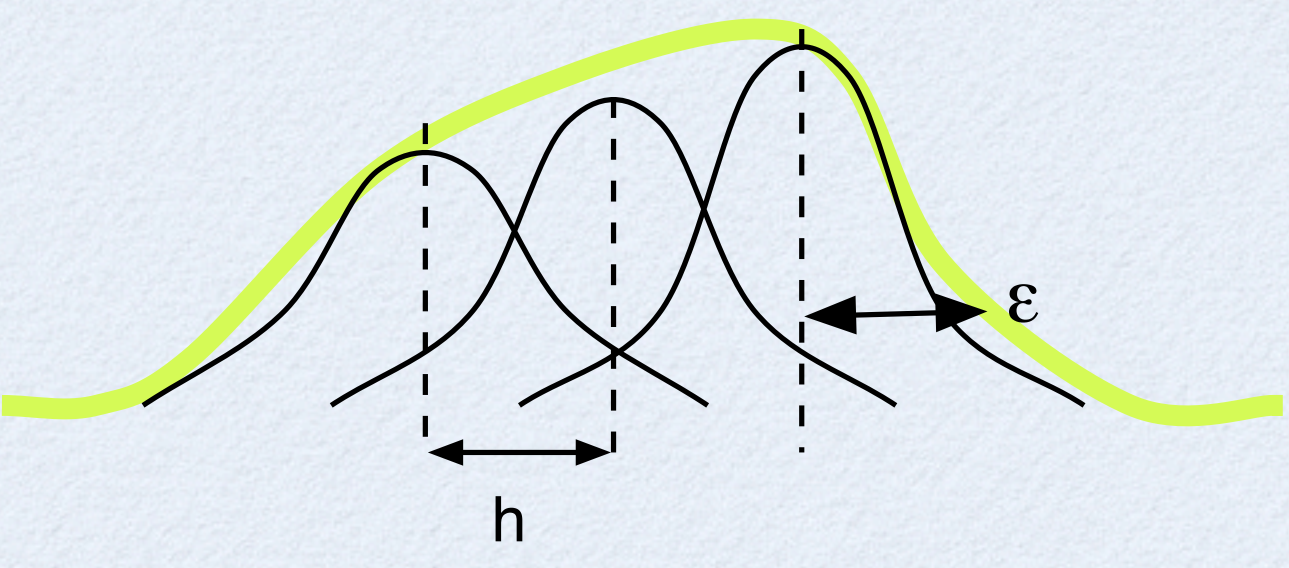
$$\Phi_\epsilon(x) \; = \int \Phi(y) \, \zeta_\epsilon(x-y) \, dy$$

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

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



FUNCTIONS and PARTICLES

Integral Function Representation

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

Function Mollification

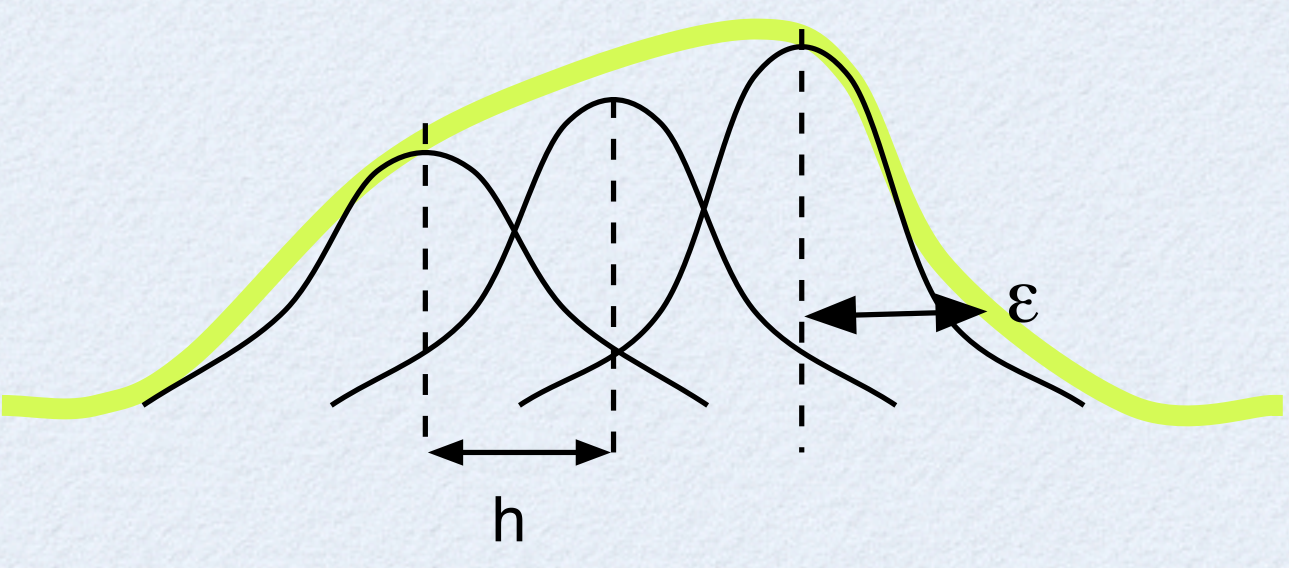
$$\Phi_\epsilon(x) \;=\; \int \Phi(y)\, \zeta_\epsilon(x-y)\, dy$$

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

FUNCTIONS and PARTICLES

Integral Function Representation

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

Function Mollification

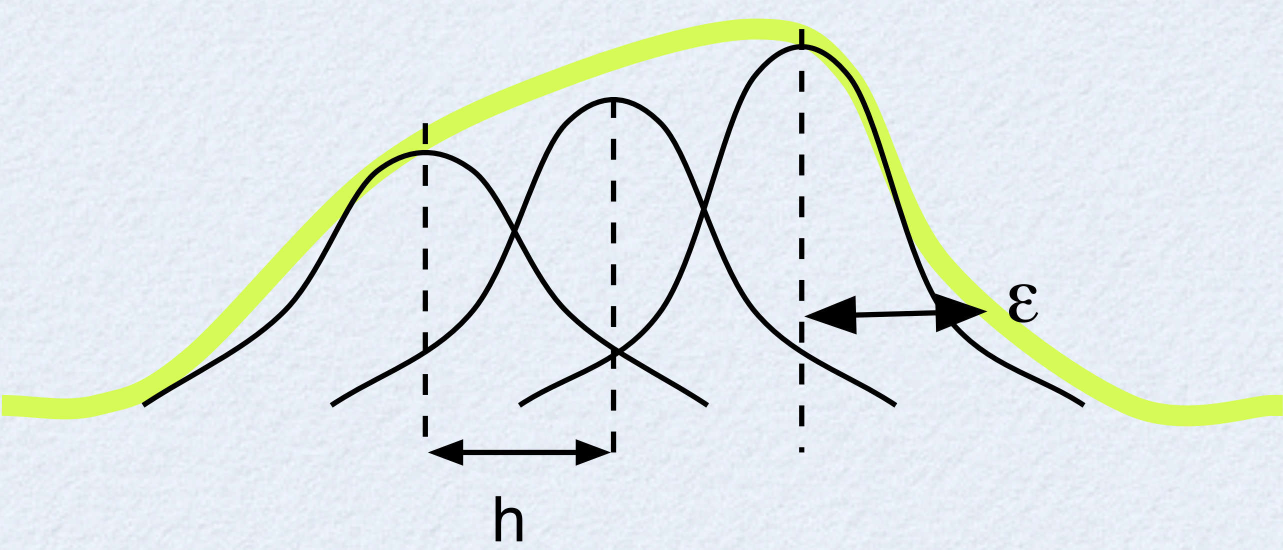
$$\Phi_\epsilon(x) \ = \int \Phi(y)\,\zeta_\epsilon(x-y)\,dy$$

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

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



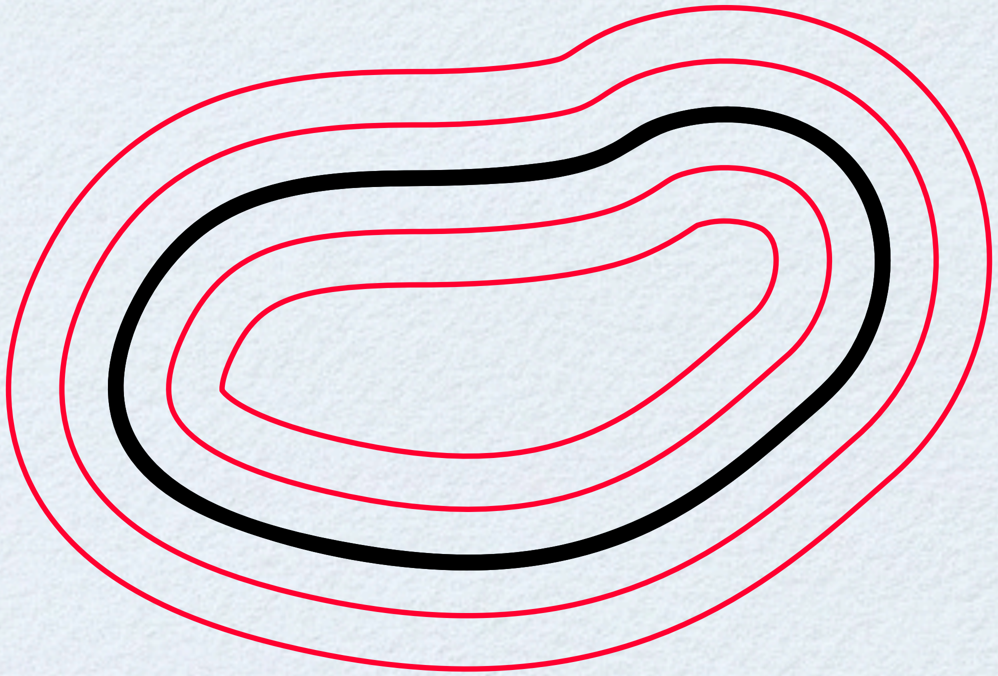
?

Particles are "mesh" free

EXAMPLE : LEVEL SETS

EXAMPLE : LEVEL SETS

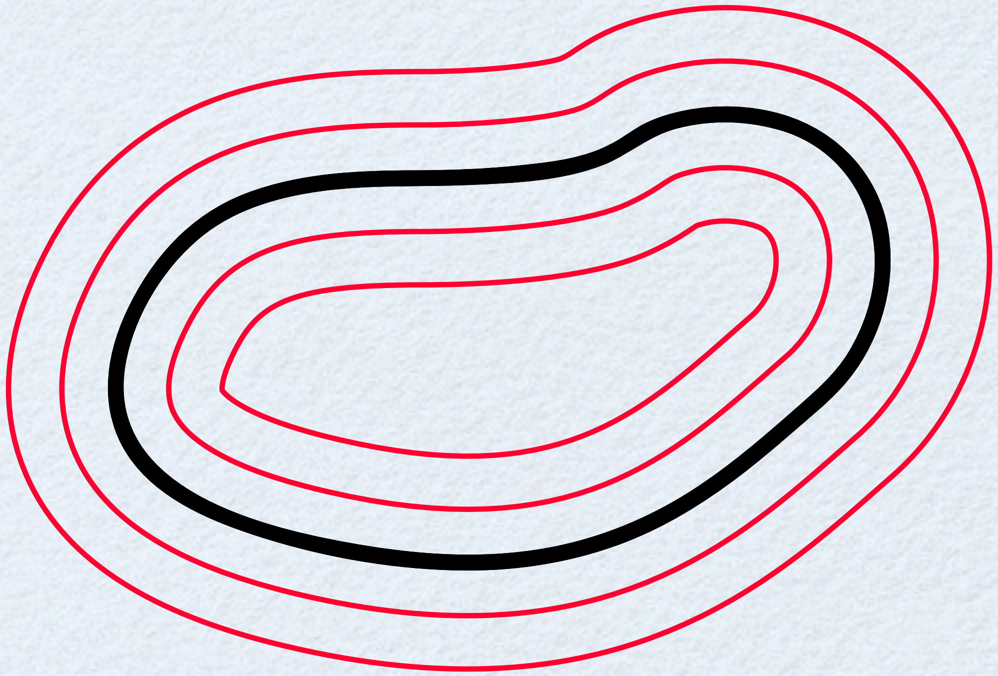
EXAMPLE : LEVEL SETS



EXAMPLE : LEVEL SETS

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

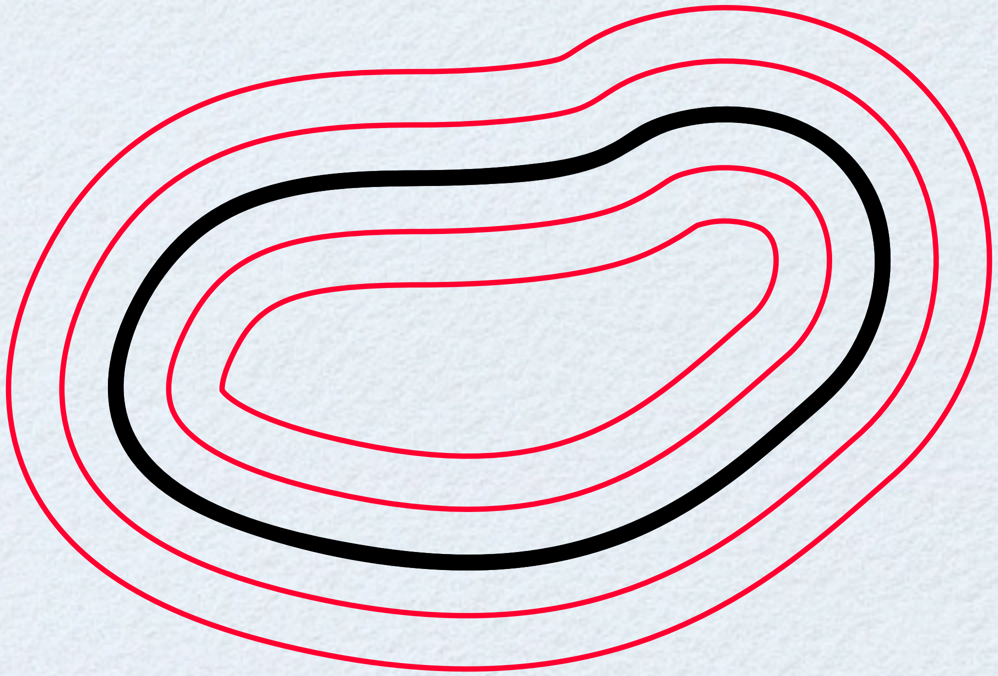
$$|\nabla \phi| = 1$$



EXAMPLE : LEVEL SETS

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

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



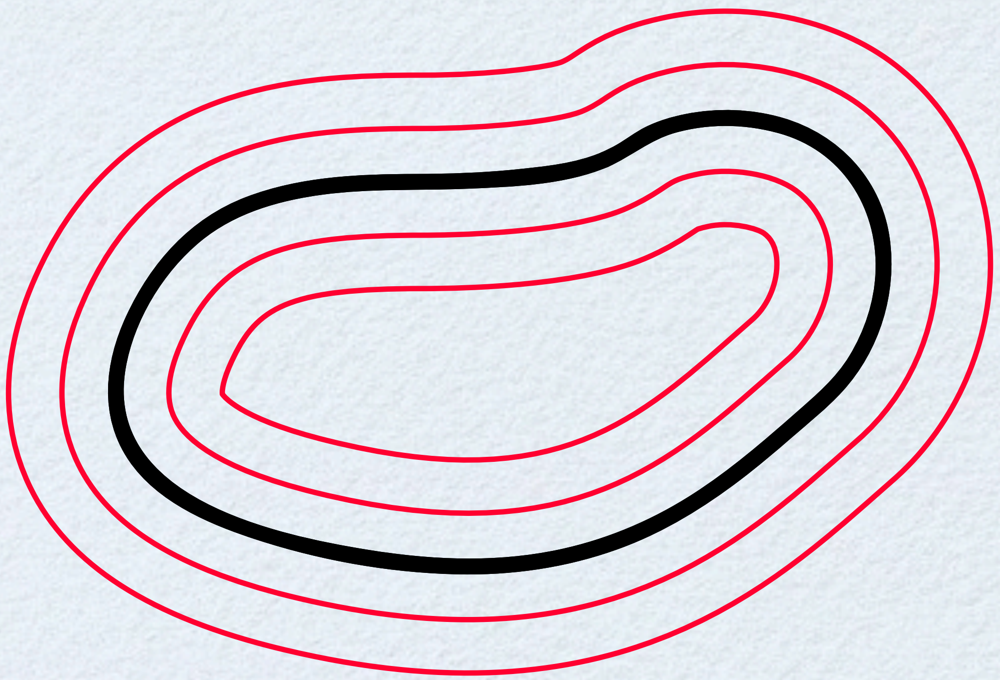
EVOLVING THE LEVEL SETS

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

EXAMPLE : LEVEL SETS

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

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



EVOLVING THE LEVEL SETS

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

PARTICLES

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

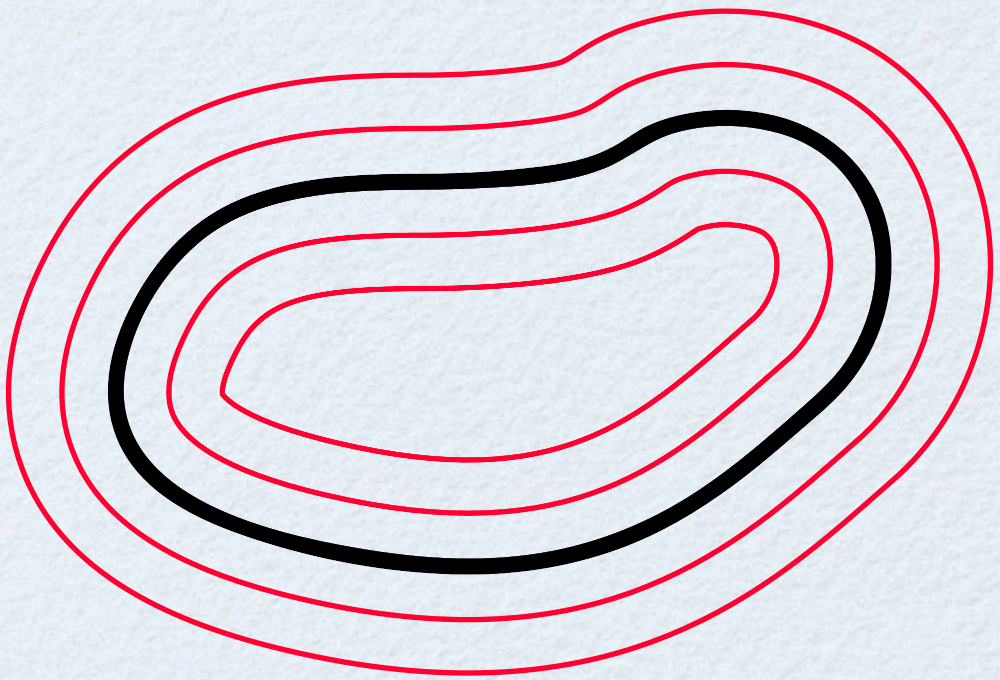
Lagrangian Surface Transport

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

EXAMPLE : LEVEL SETS

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

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



EVOLVING THE LEVEL SETS

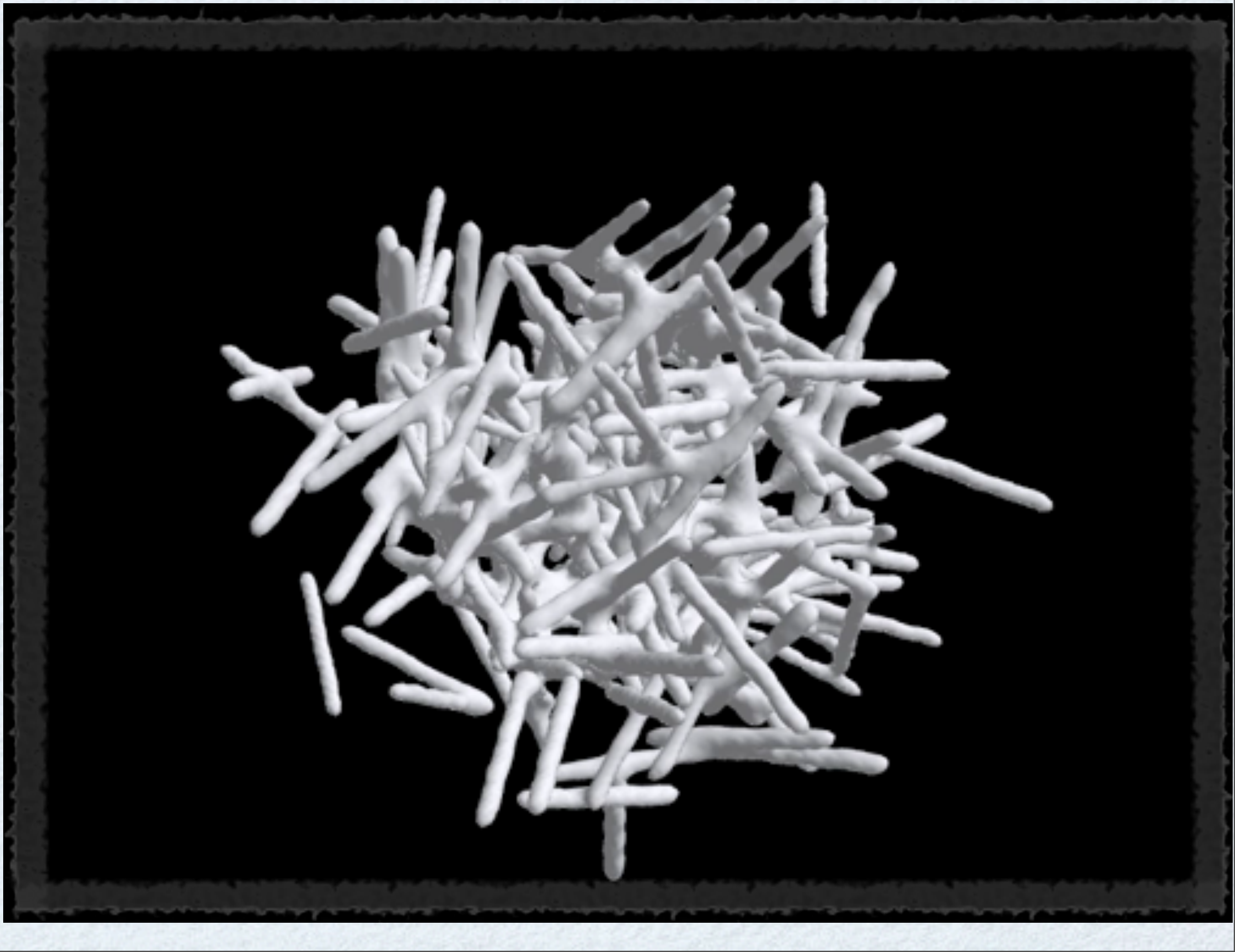
$$\frac{\partial \Phi}{\partial t} + u \cdot \nabla \Phi = 0$$

PARTICLES

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

Lagrangian Surface Transport

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



Lagrangian vs Eulerian Descriptions

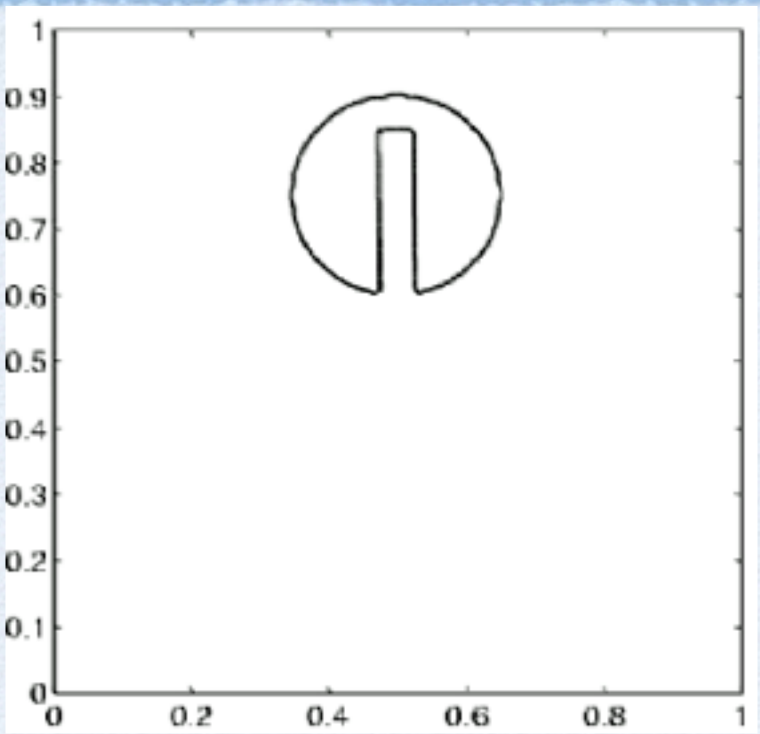
S. E. Hieber and P. Koumoutsakos. A Lagrangian particle level set method. **J. Computational Physics**, 210:342-367, 2005

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

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

Lagrangian vs Eulerian Descriptions

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

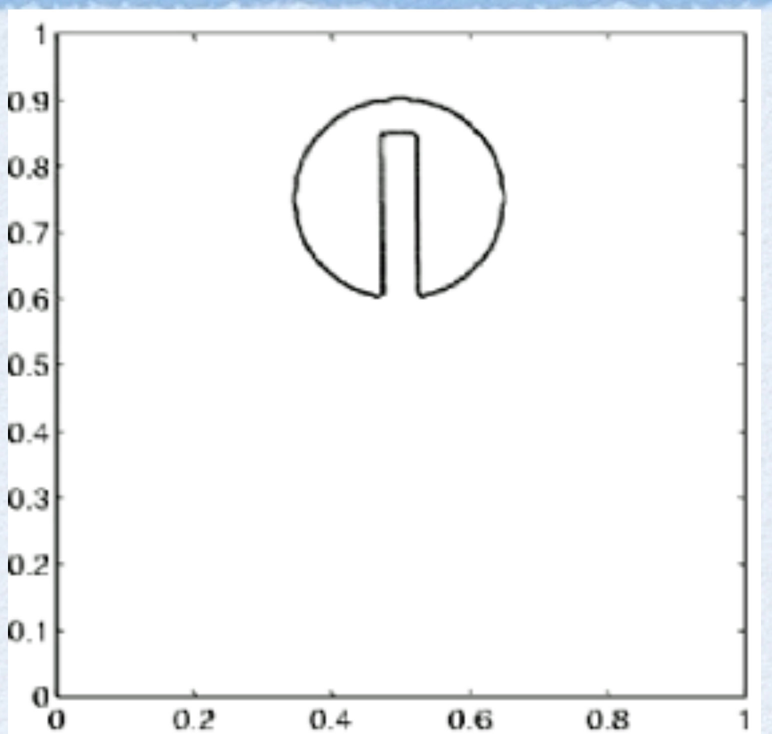


$$\Phi(\mathbf{x}, t) = \Phi_0(\mathbf{x} - \mathbf{u}t)$$

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

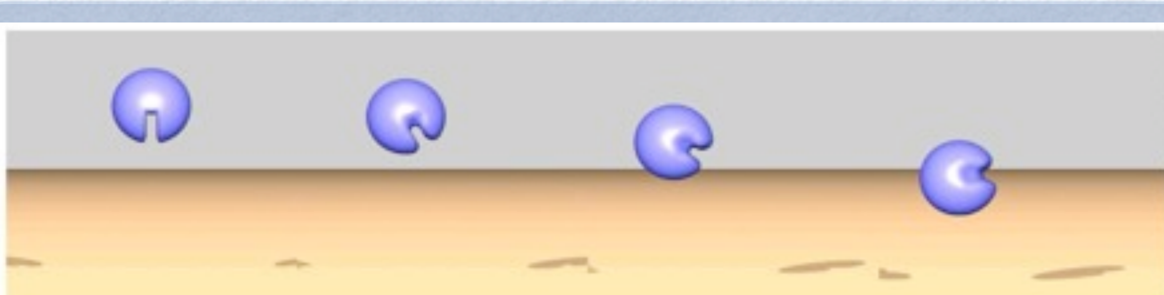
Lagrangian vs Eulerian Descriptions

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

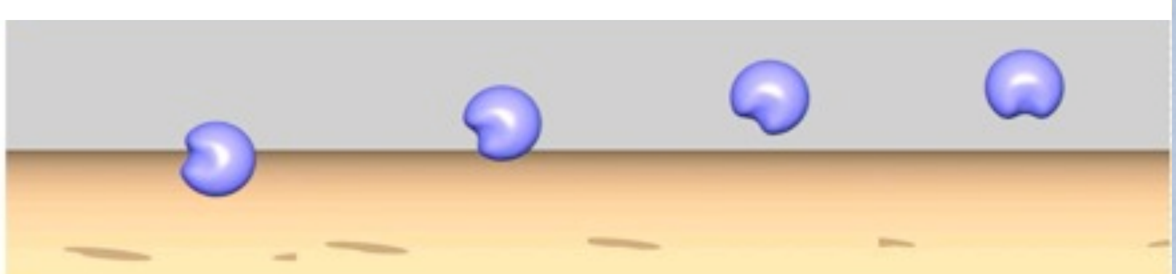


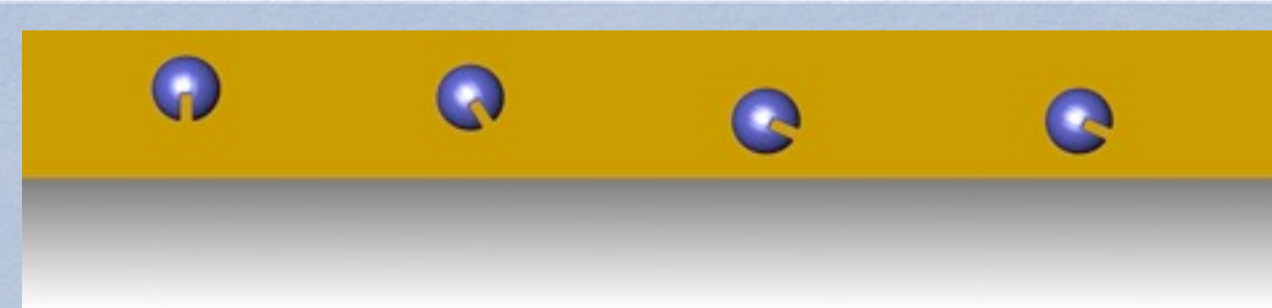
$$\Phi(\mathbf{x}, t) = \Phi_0(\mathbf{x} - \mathbf{u}t)$$

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

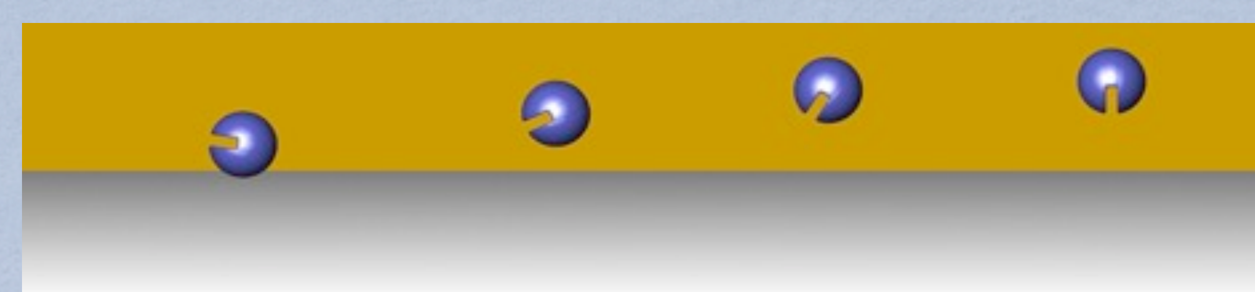


Hubrid Particle-Grid Level Sets
(Enright and Fedkiw, 2002)



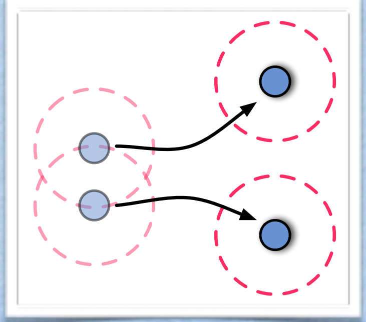


Lagrangian Particle Level Sets



LAGRANGIAN DISTORTION

- loss of overlap -> loss of convergence



Particles follow flow trajectories - Location distortion

EXAMPLE :

Incompressible 2D Euler Equations

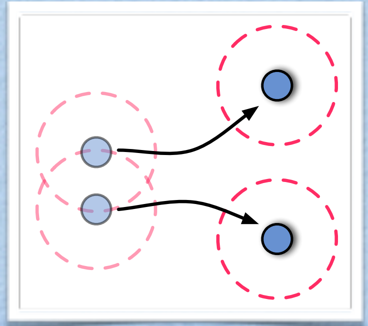
$$\omega = \nabla \times \mathbf{u} \quad \nabla \cdot \mathbf{u} = 0$$

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

There is an exact axisymmetric solution

LAGRANGIAN DISTORTION

- loss of overlap -> loss of convergence



Particles follow flow trajectories - Location distortion

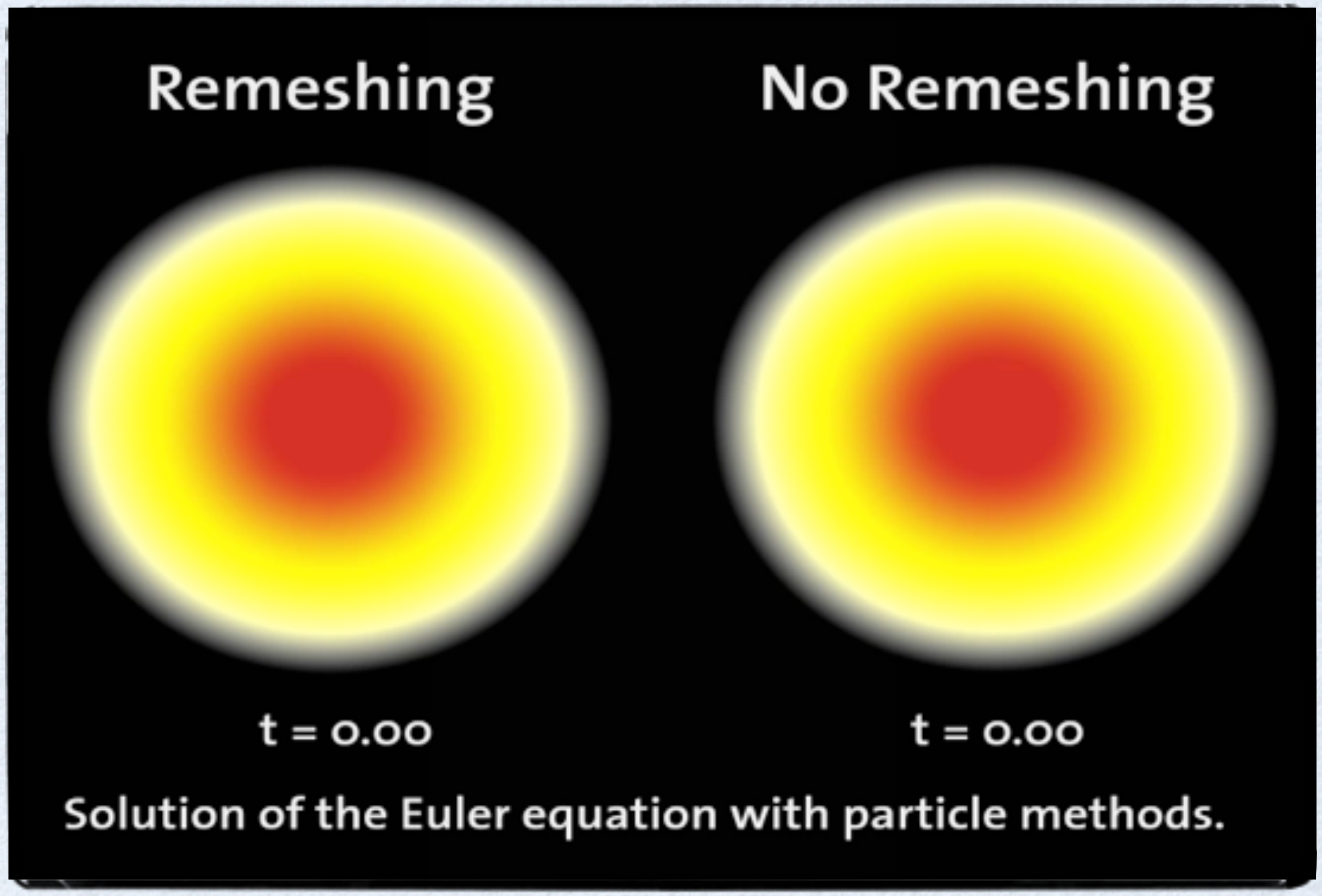
EXAMPLE :

Incompressible 2D Euler Equations

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

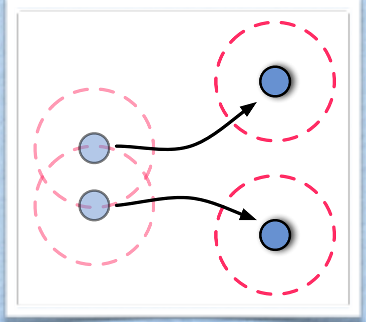
$$\frac{D\omega}{Dt} = 0$$

There is an exact axisymmetric solution



LAGRANGIAN DISTORTION

- loss of overlap -> loss of convergence



Particles follow flow trajectories - Location distortion

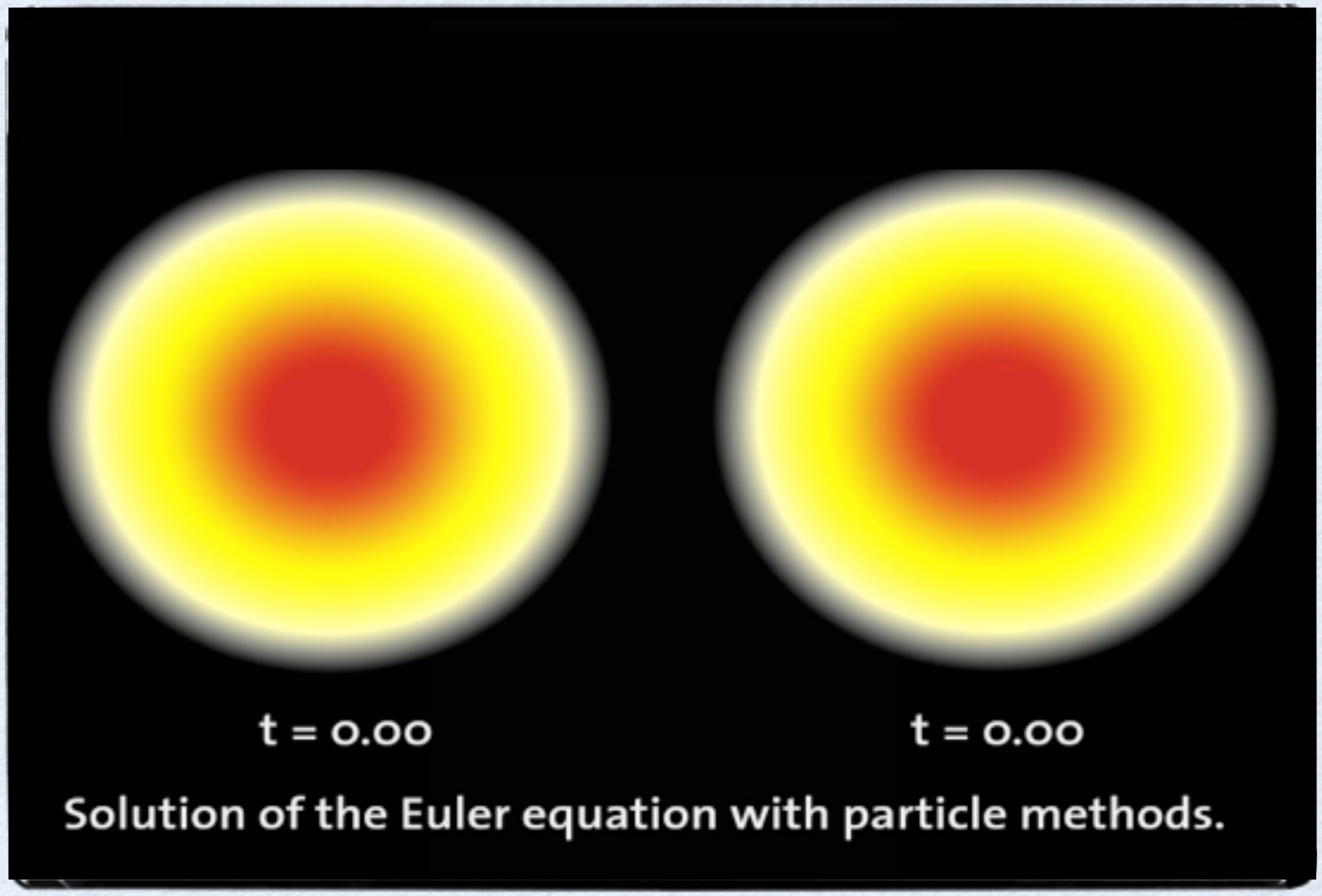
EXAMPLE :

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

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

Function Mollification

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

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

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

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

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

$$\begin{aligned} ||\Phi - \Phi_\epsilon^h|| &\le ||\Phi - \Phi_\epsilon|| + ||\Phi_\epsilon - \Phi_\epsilon^h|| \\ &\le (C_1\, \epsilon^r + C_2\, (\frac{h}{\epsilon})^m)\, ||\Phi||_\infty \end{aligned}$$

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

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

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

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

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

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

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

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

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

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

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

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

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/ε < 1 for accuracy

PARTICLES MUST OVERLAP

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

Are Particle Methods Grid Free ?

Are Particle Methods Grid Free ?

How to fix it ?

- Modify the smoothing kernels (SPH - Monaghan)
- Re-distribute particles with Voronoi Meshes (ALE - Russo)
- Re-initialise particle strengths (WRKPM - Liu, Belytchko)

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

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

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

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 “tools” of mesh based methods

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 “tools” of mesh based methods
- High CFL

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 “tools” of mesh based methods
- High CFL
- Multiresolution & Multiscaling

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

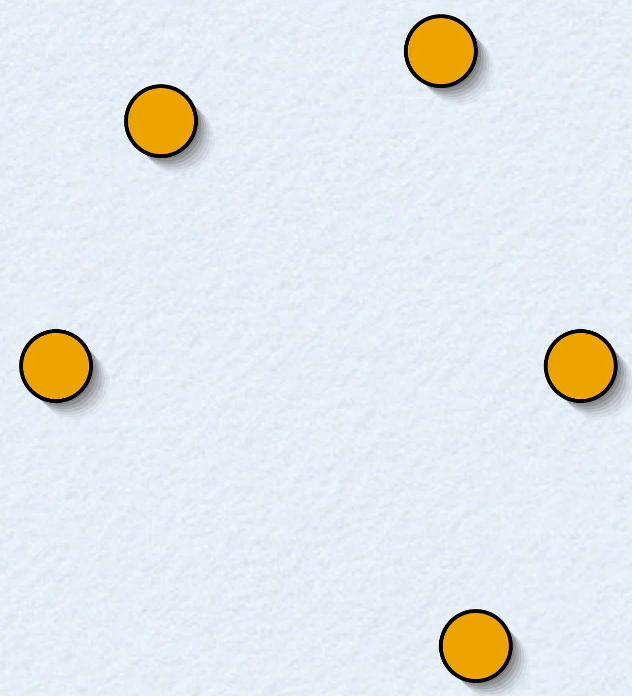
- NO MESH-FREE particle methods
- Can use all the “tools” of mesh based methods
- High CFL
- Multiresolution & Multiscaling
- ....

Particle Remeshing

P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997

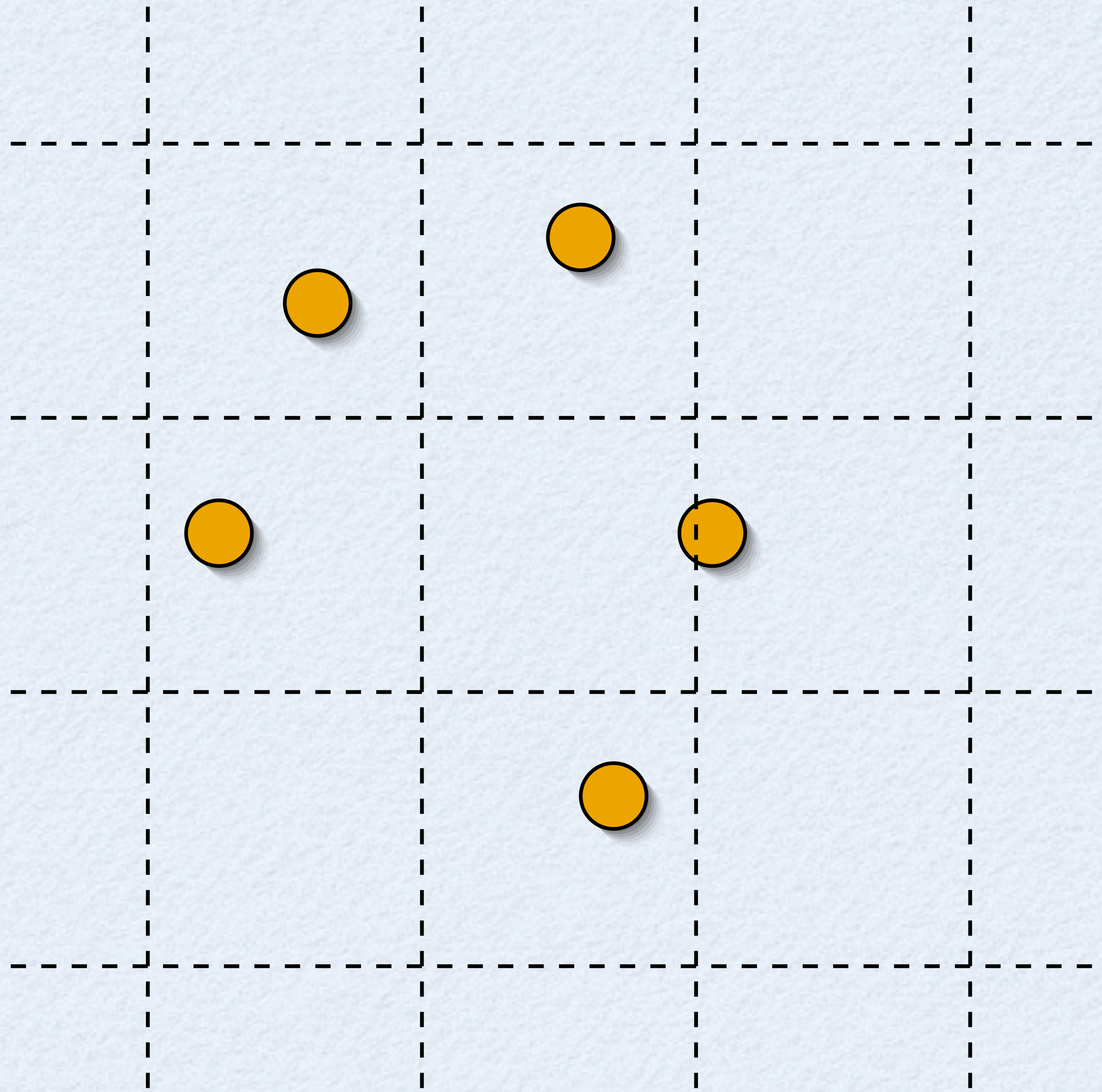
Particle Remeshing

P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997



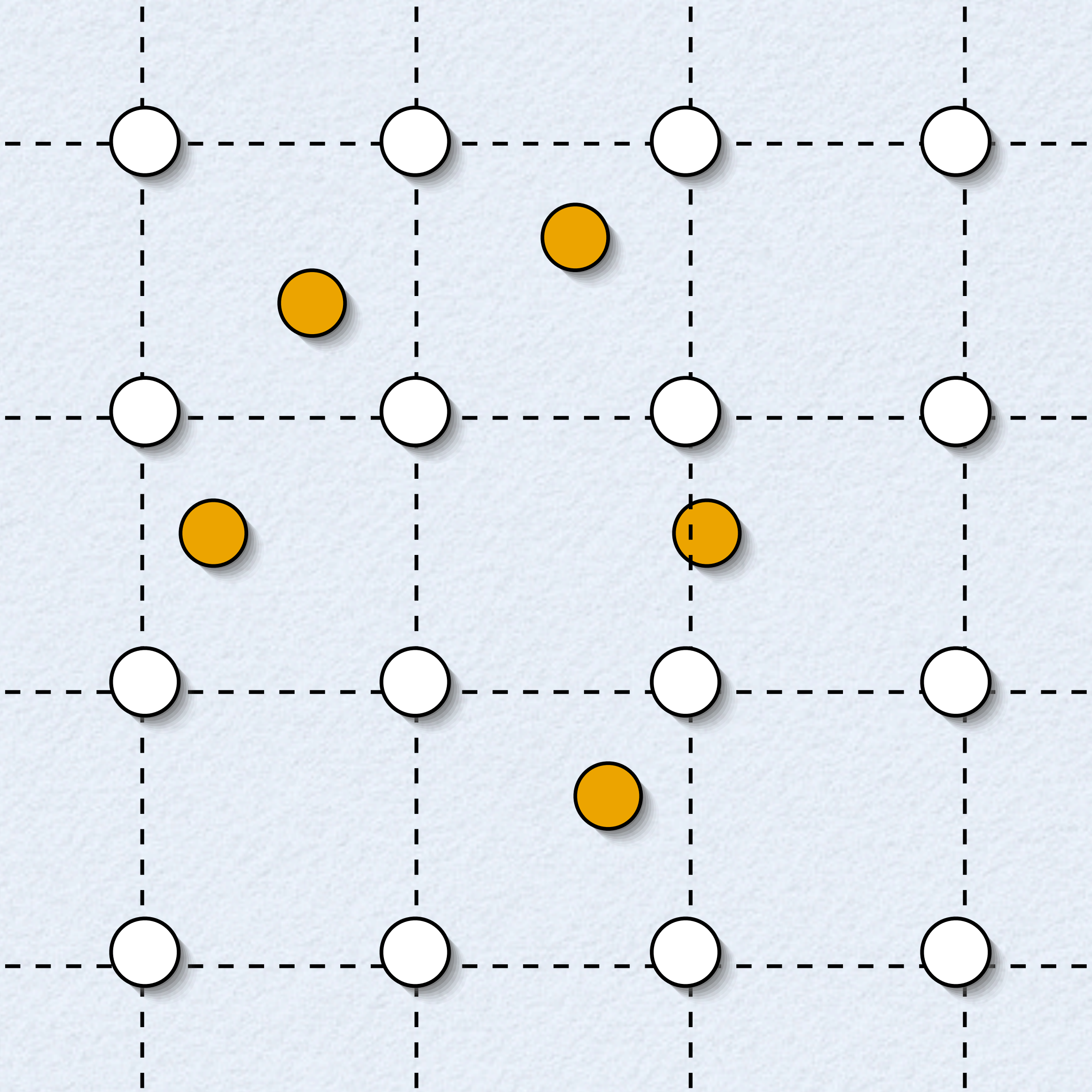
Particle Remeshing

P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997



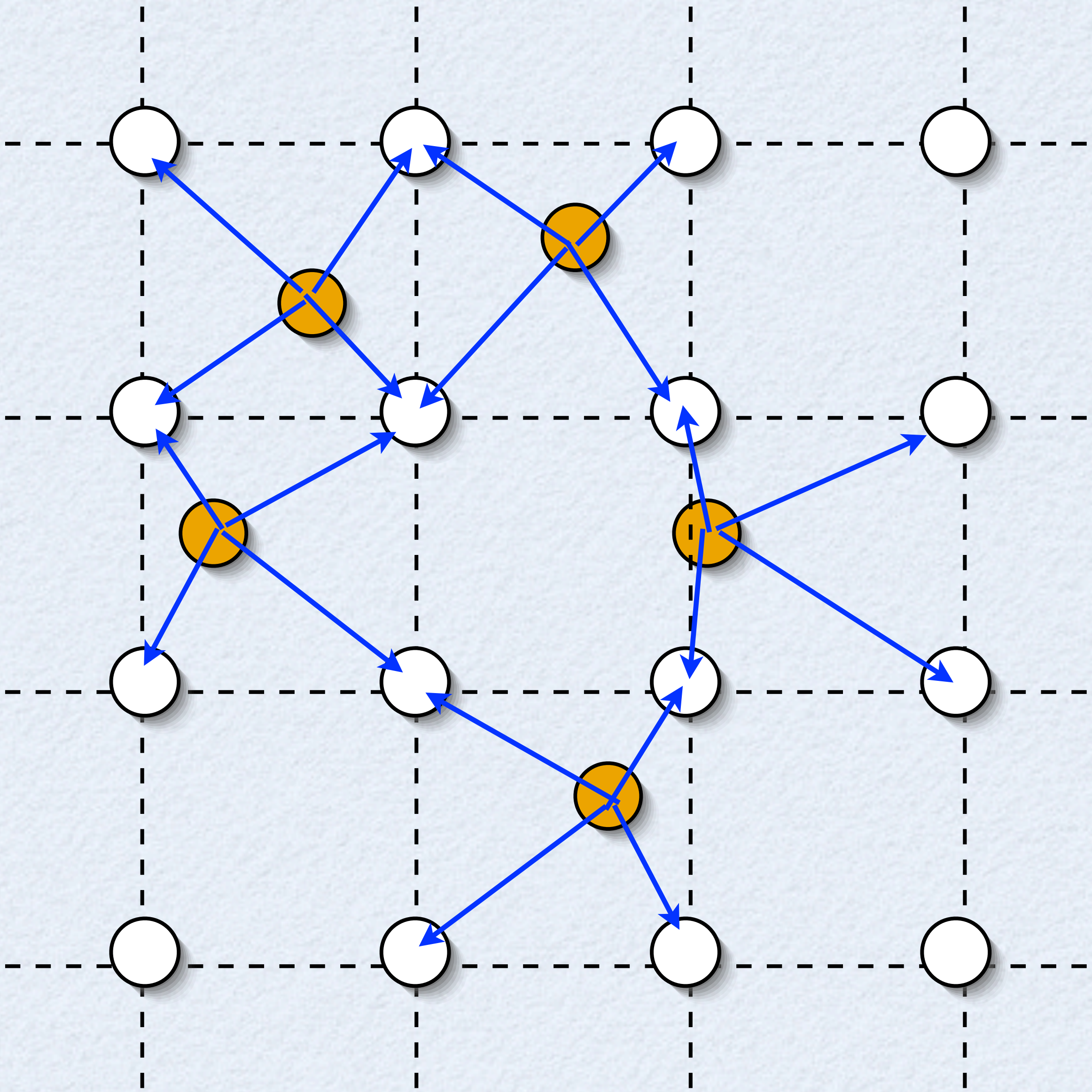
Particle Remeshing

P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997



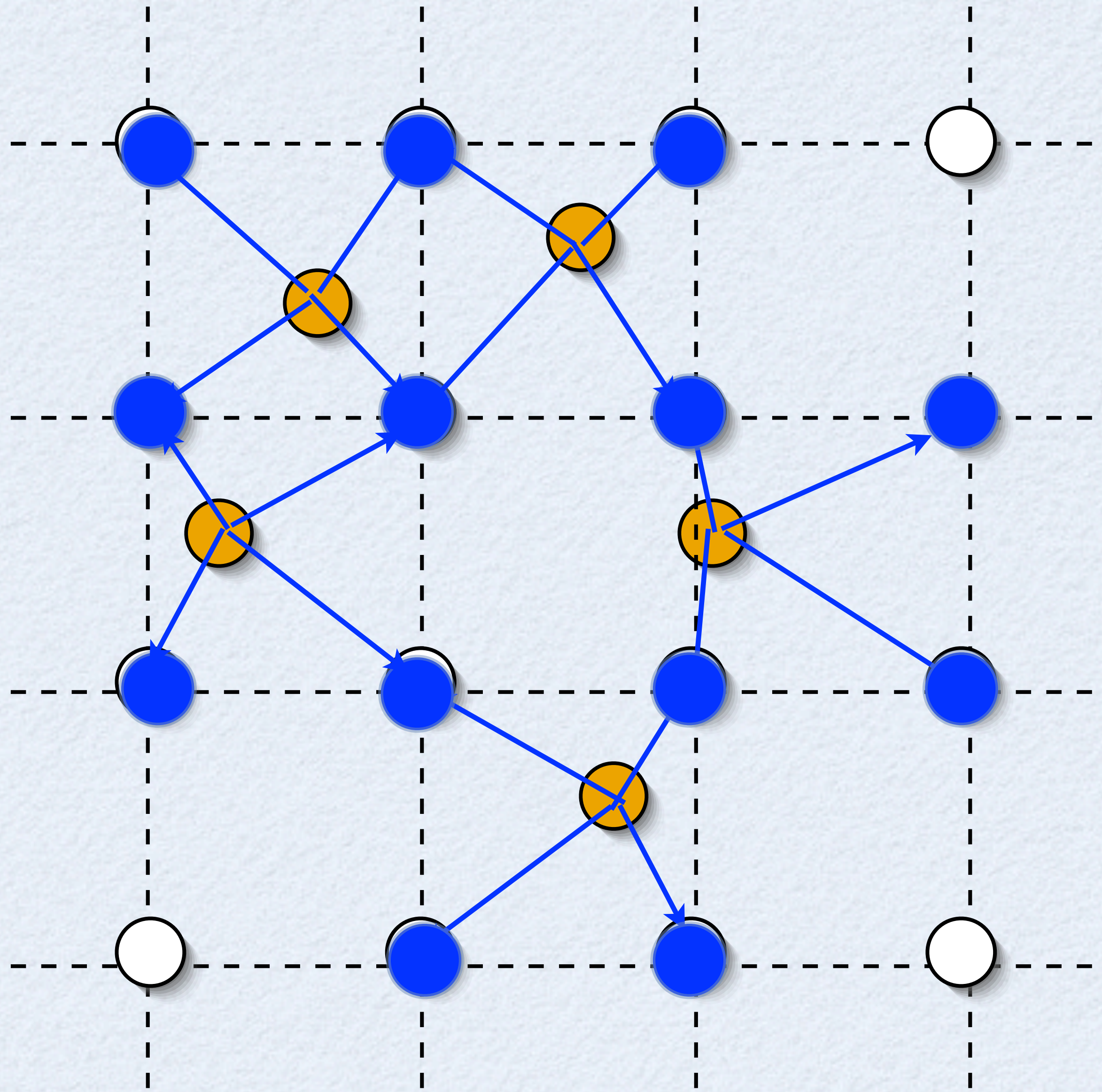
Particle Remeshing

P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997



Particle Remeshing

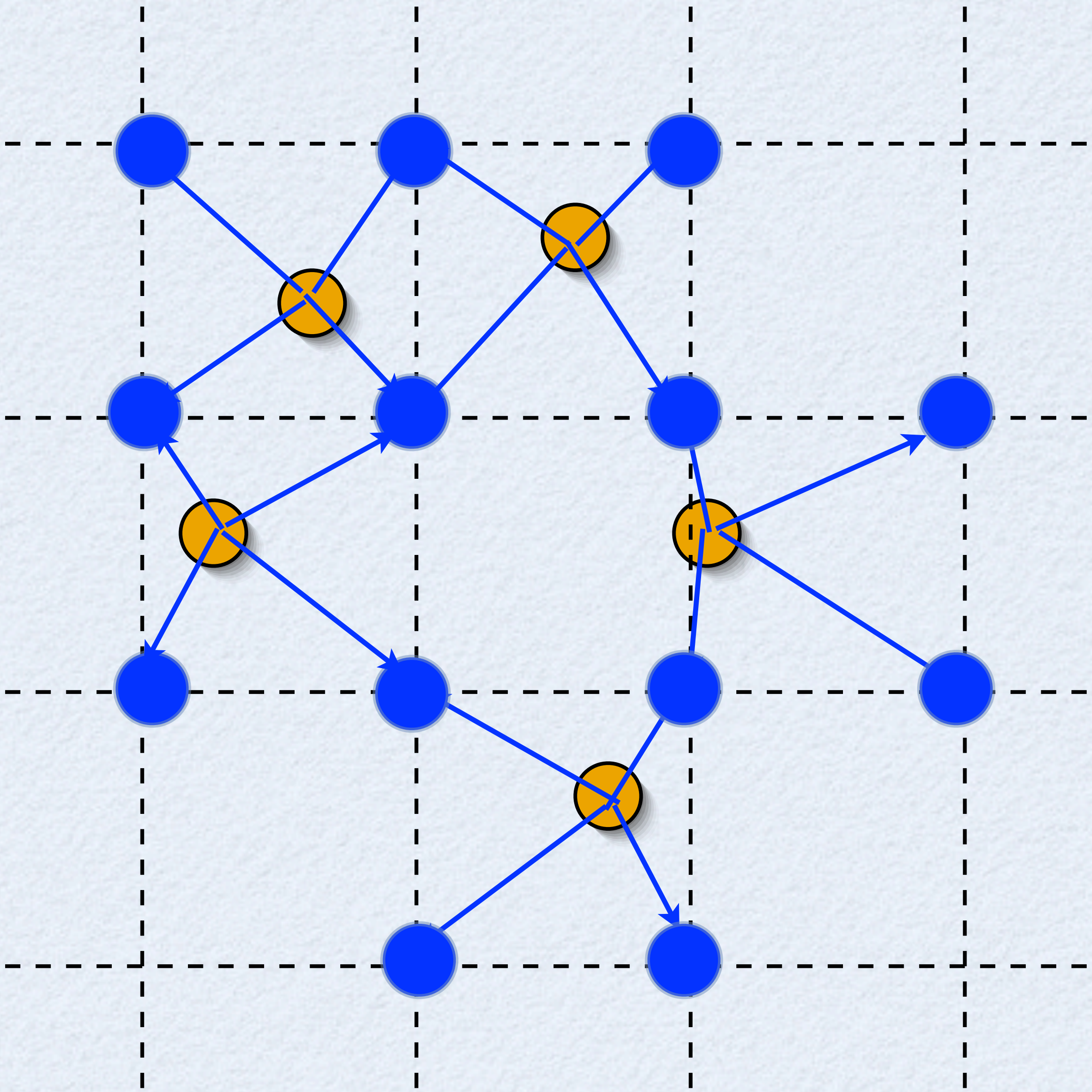
P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997



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

Particle Remeshing

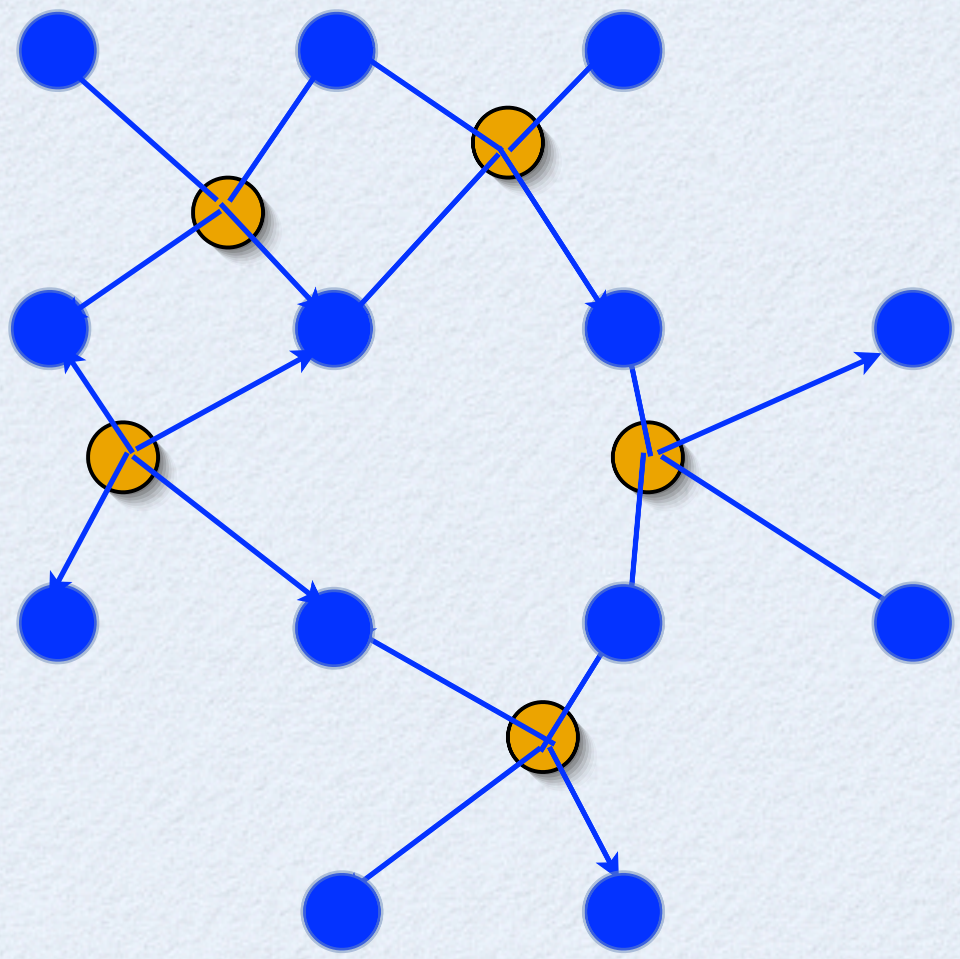
P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997



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

Particle Remeshing

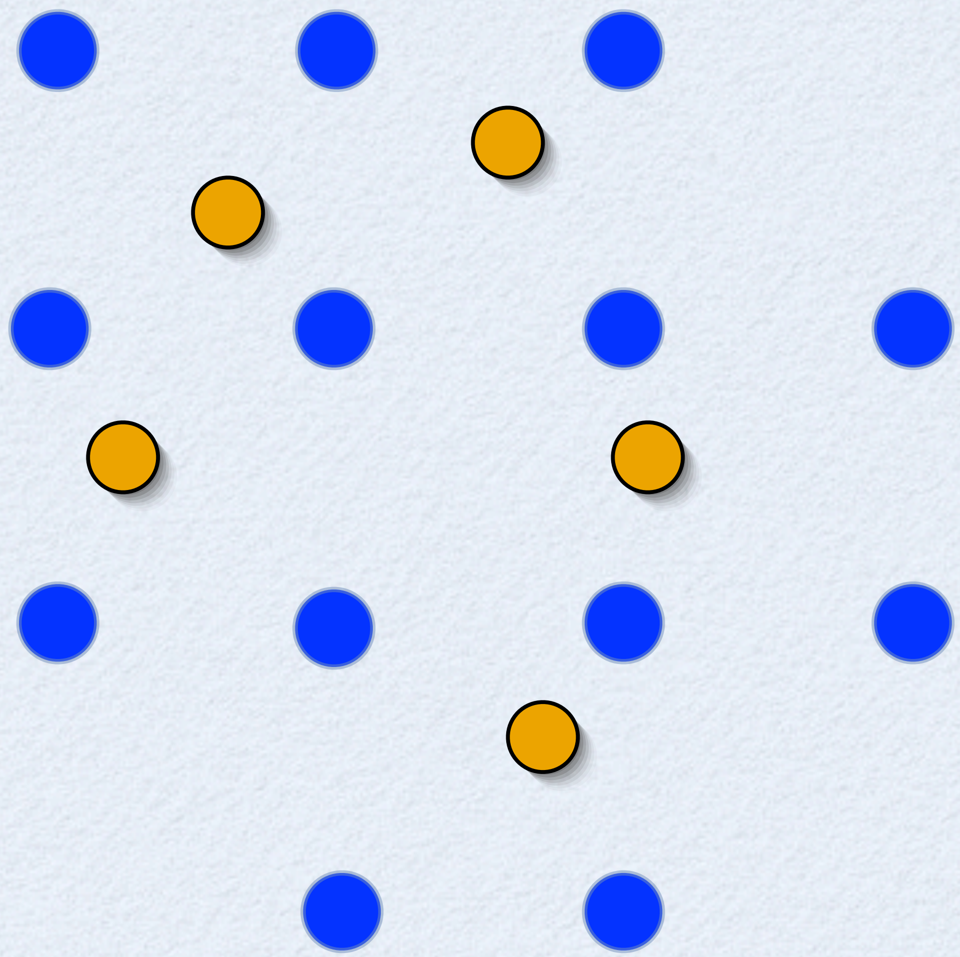
P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997



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

Particle Remeshing

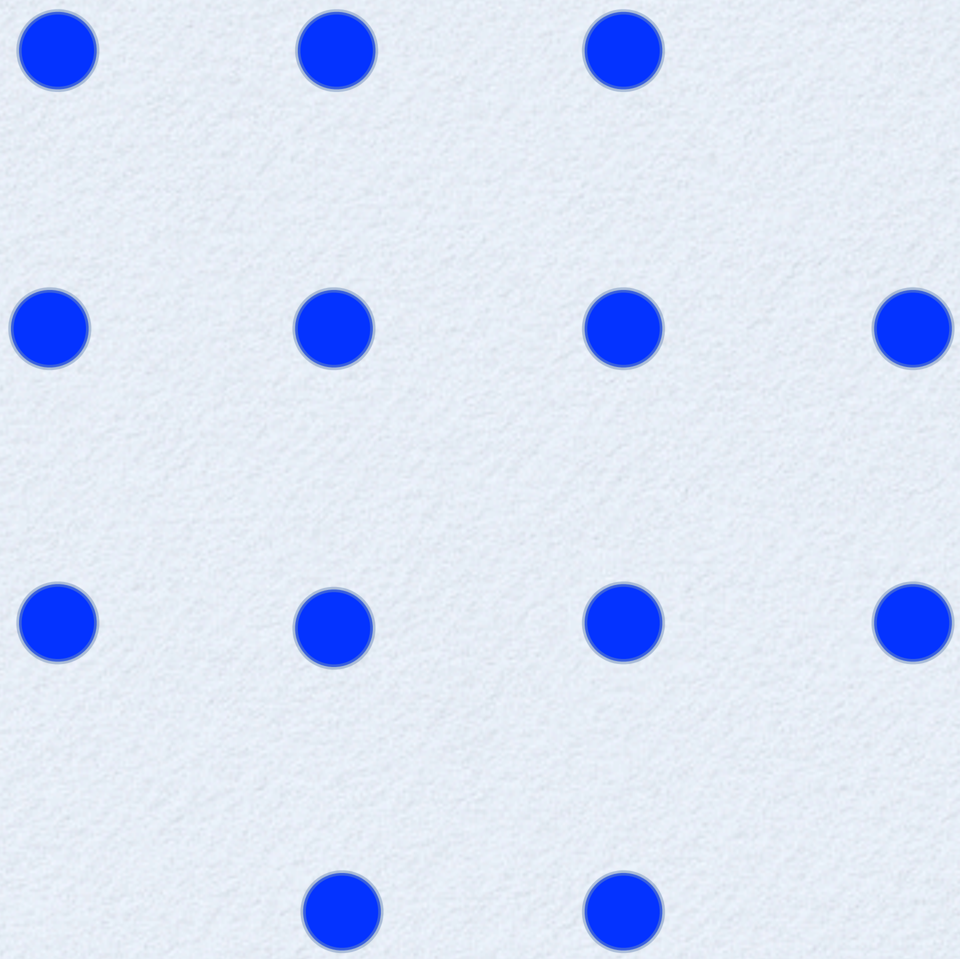
P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997



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

Particle Remeshing

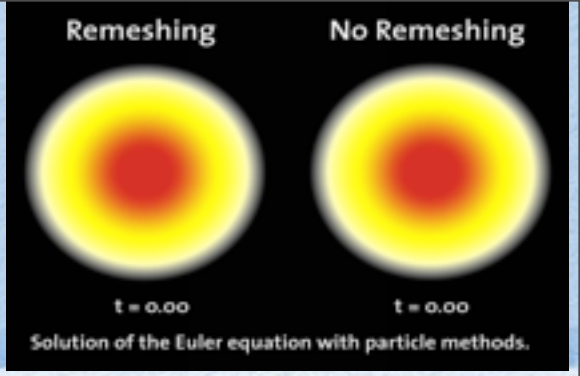
P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997

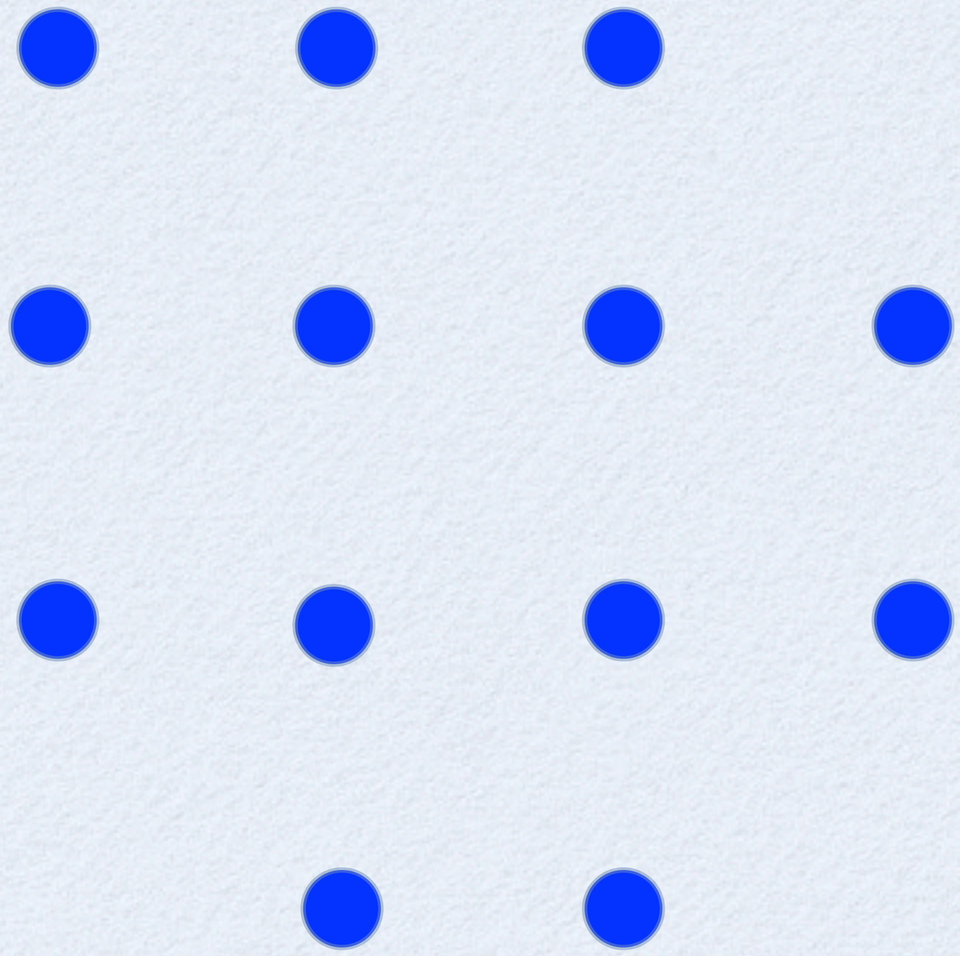


Moment Conserving Interpolation : $Q_p^{\text{new}} = \sum_{p'} Q_{p'} M(j\,h - x_{p'})$

Particle Remeshing

P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, **J. Comput. Phys.**, 1997





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

REMESHED PARTICLE METHODS

REMESHED PARTICLE METHODS

1. ADVECT : Particles ->Large CFL

REMESHED PARTICLE METHODS

1. ADVECT : Particles ->Large CFL

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

REMESHED PARTICLE METHODS

1. ADVECT : Particles ->Large CFL

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

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

REMESHED PARTICLE METHODS

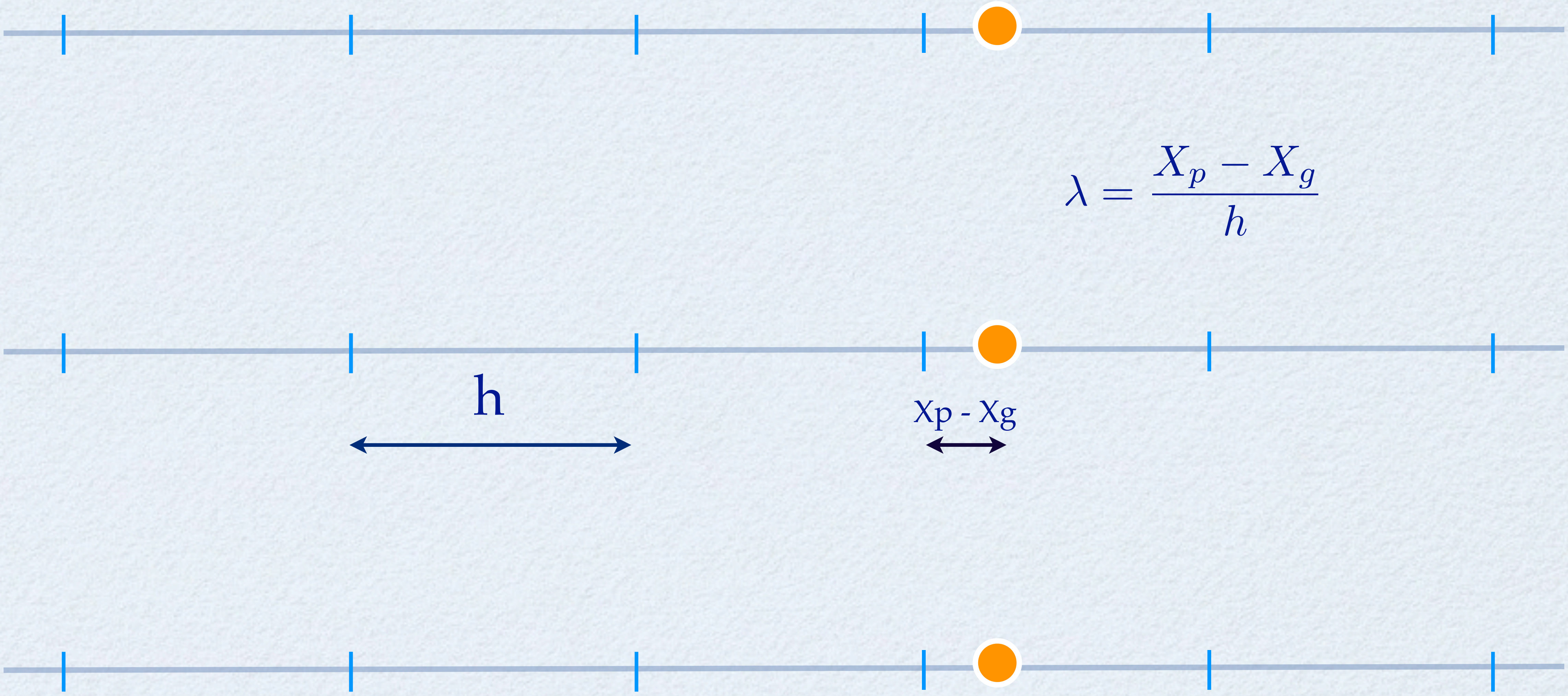
1. ADVECT : Particles ->Large CFL

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

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

4:RESAMPLE: Mesh Nodes BECOME Particles

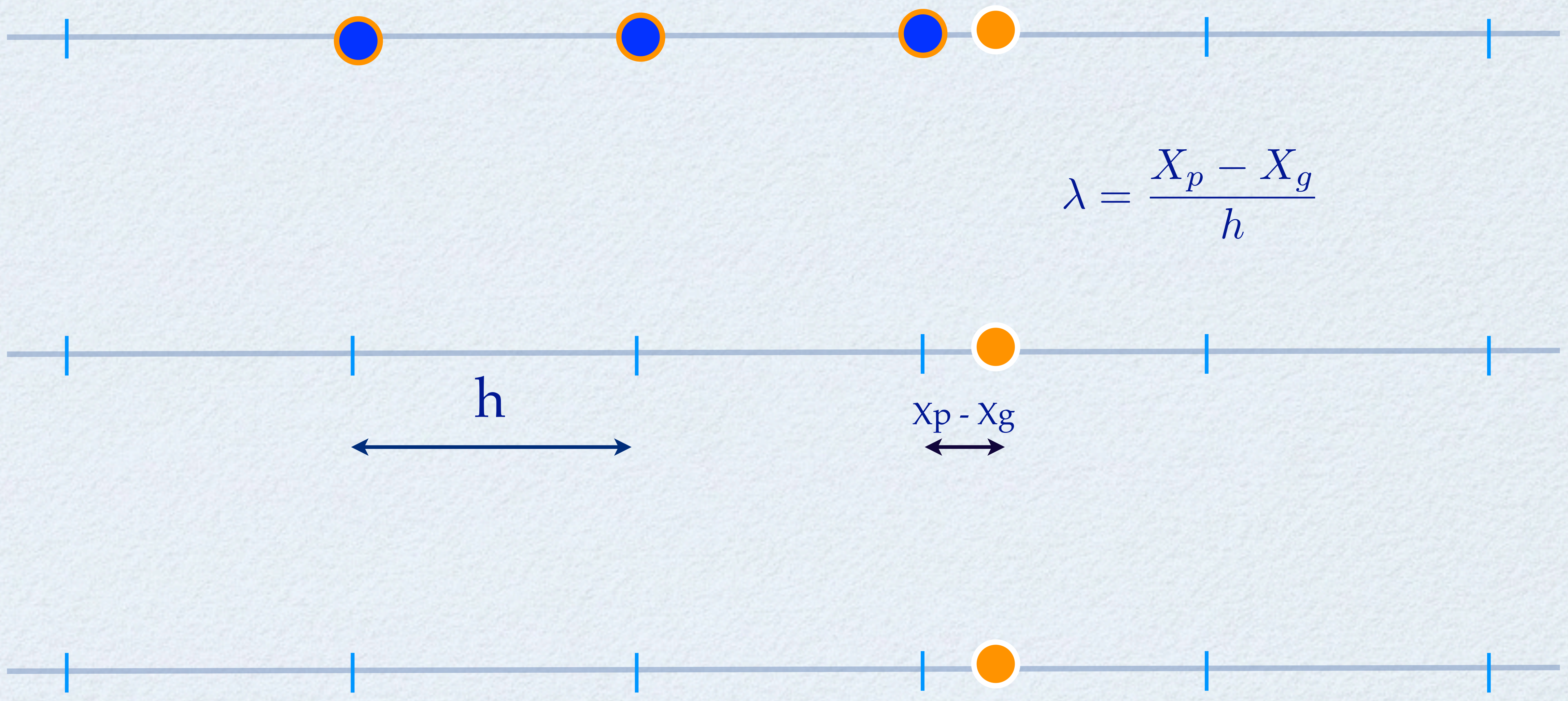
Size of Remeshing Stencil = # Conserved Moments



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

Bergdorf et. al., MMS,2005
Cottet et.al., CRAS, 2008

Size of Remeshing Stencil = # Conserved Moments

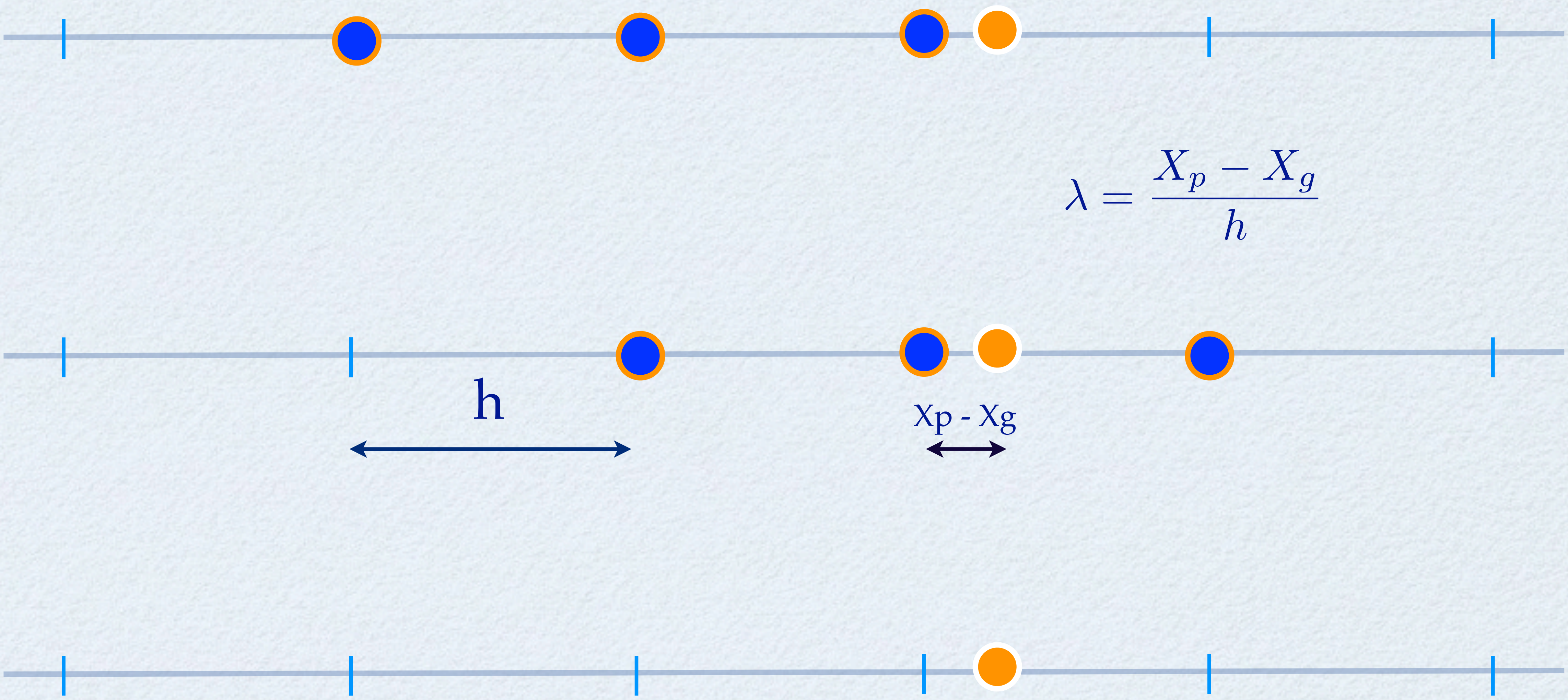


$$\lambda = \frac{X_p - X_g}{h}$$

Bergdorf et. al., MMS,2005

Cottet et.al., CRAS, 2008

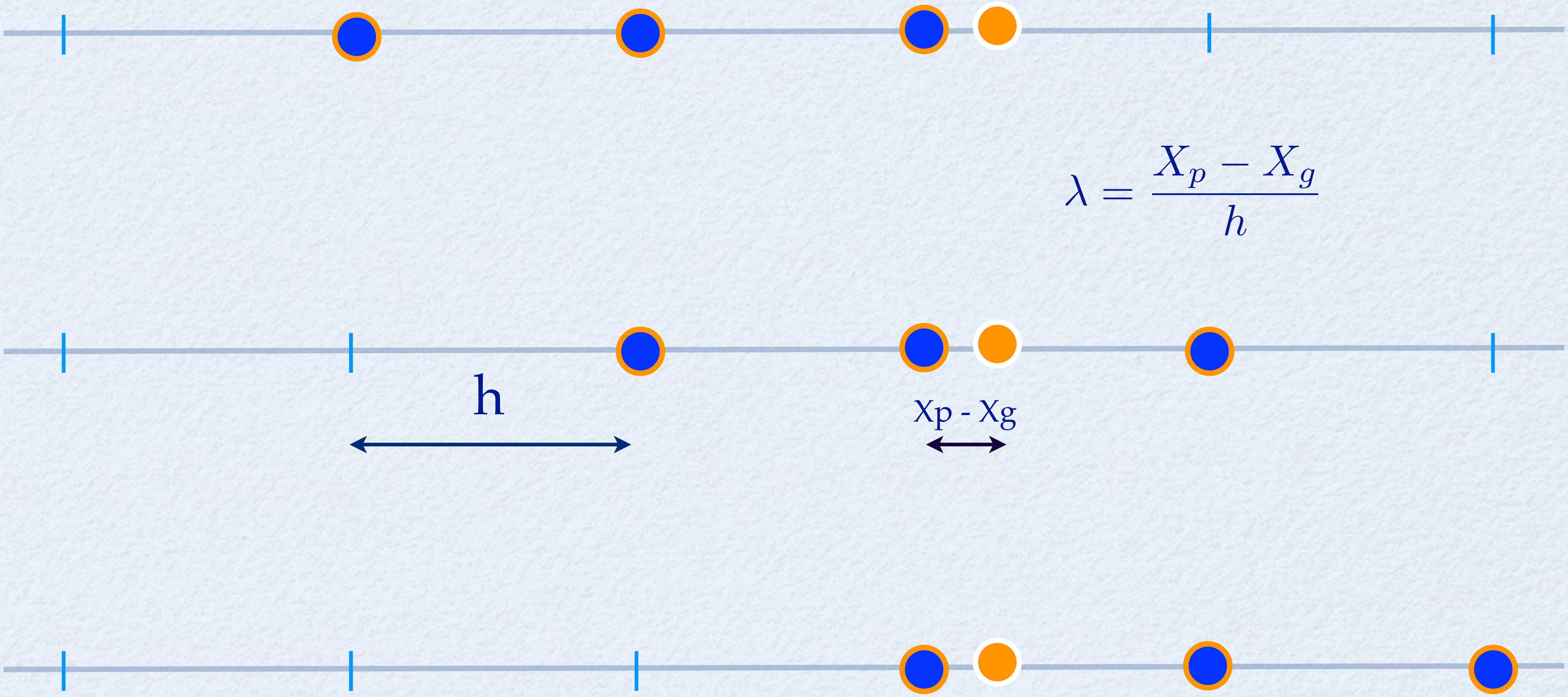
Size of Remeshing Stencil = # Conserved Moments



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

Bergdorf et. al., MMS,2005
Cottet et.al., CRAS, 2008

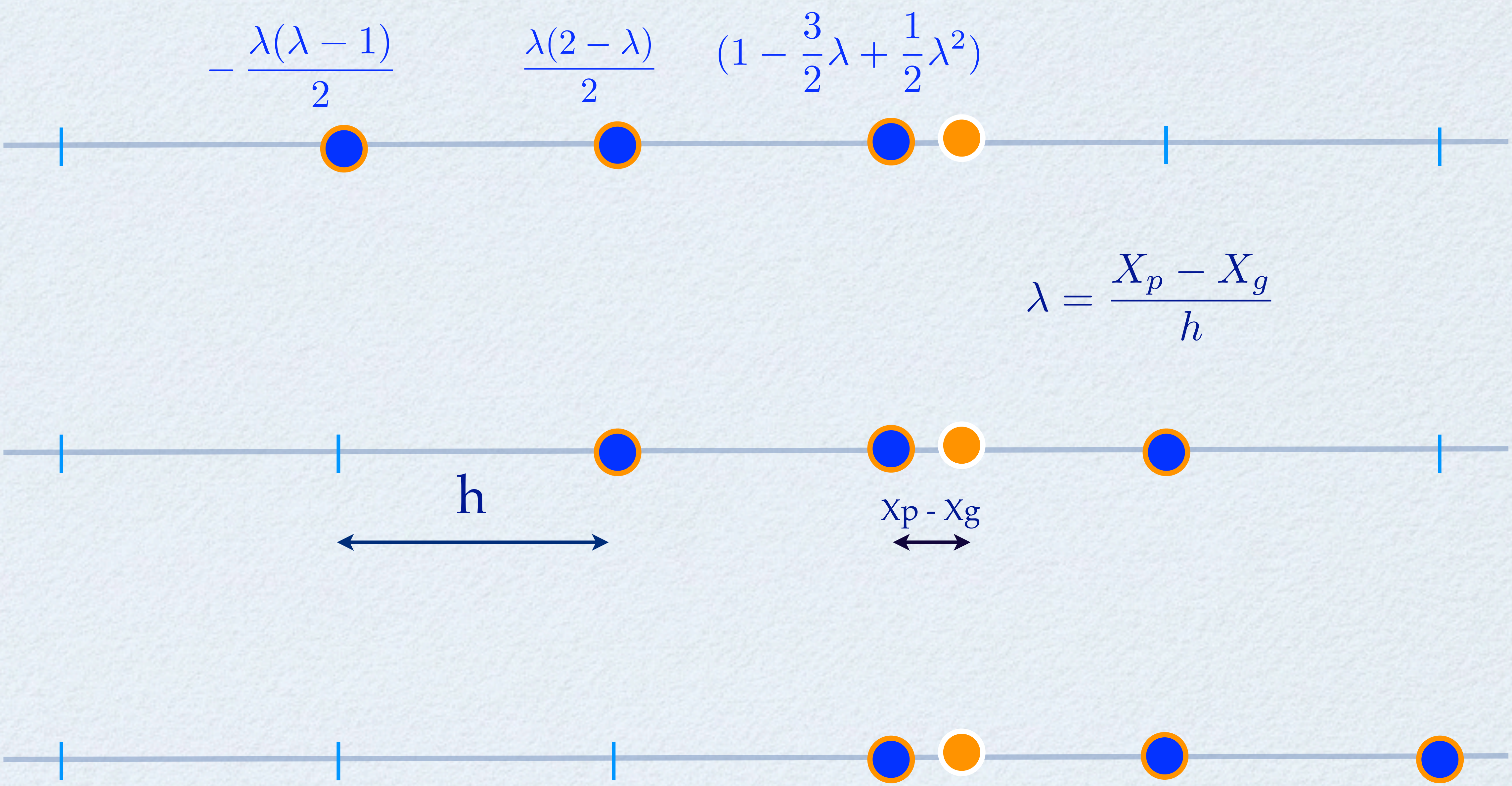
Size of Remeshing Stencil = # Conserved Moments



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

Bergdorf et. al., MMS,2005
Cottet et.al., CRAS, 2008

Size of Remeshing Stencil = # Conserved Moments



Bergdorf et. al., MMS,2005

Cottet et.al., CRAS, 2008

$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$

$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$

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

$$\frac{dx_p}{dt} = a$$

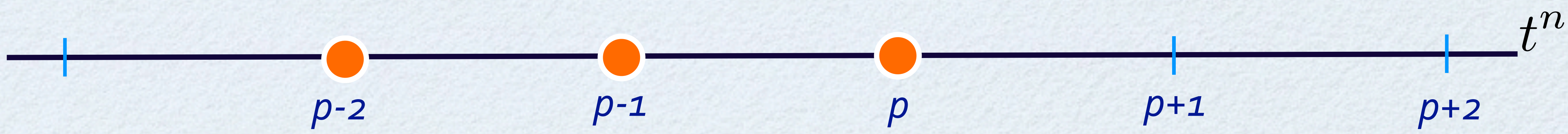
$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$ →

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

$$\frac{dx_p}{dt} = a$$

+ REMESH



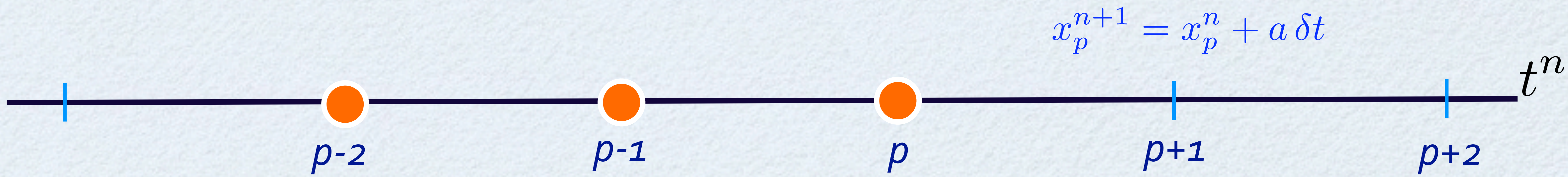
$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$

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

$$\frac{dx_p}{dt} = a$$

+ REMESH



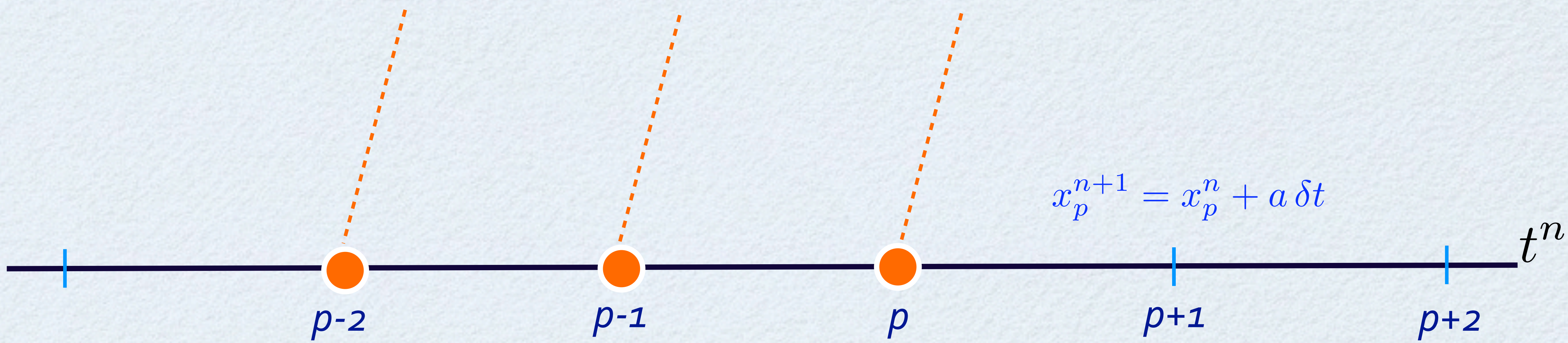
$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$

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

$$\frac{dx_p}{dt} = a$$

+ REMESH



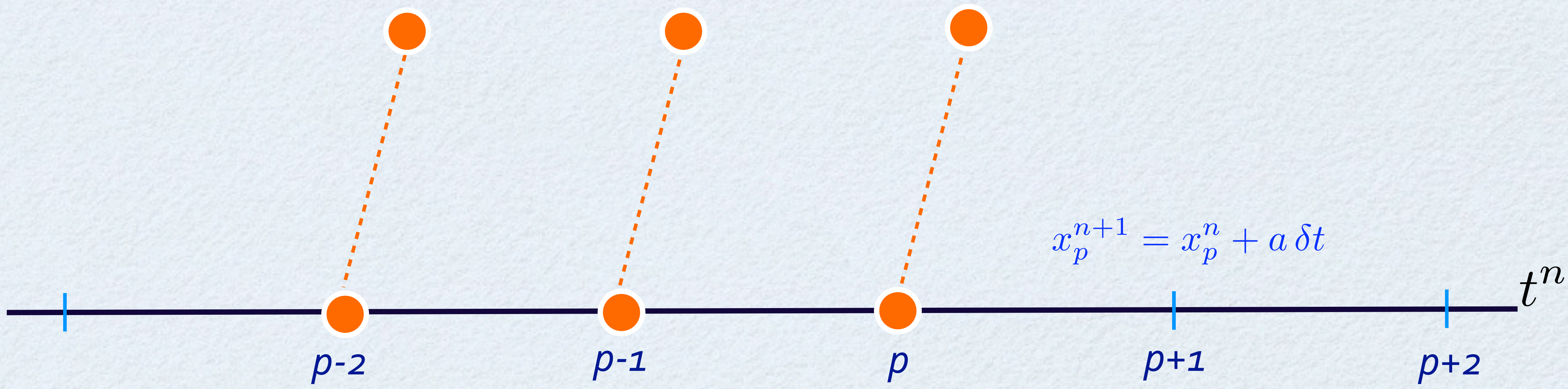
$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$ →

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

$$\frac{dx_p}{dt} = a$$

+ REMESH



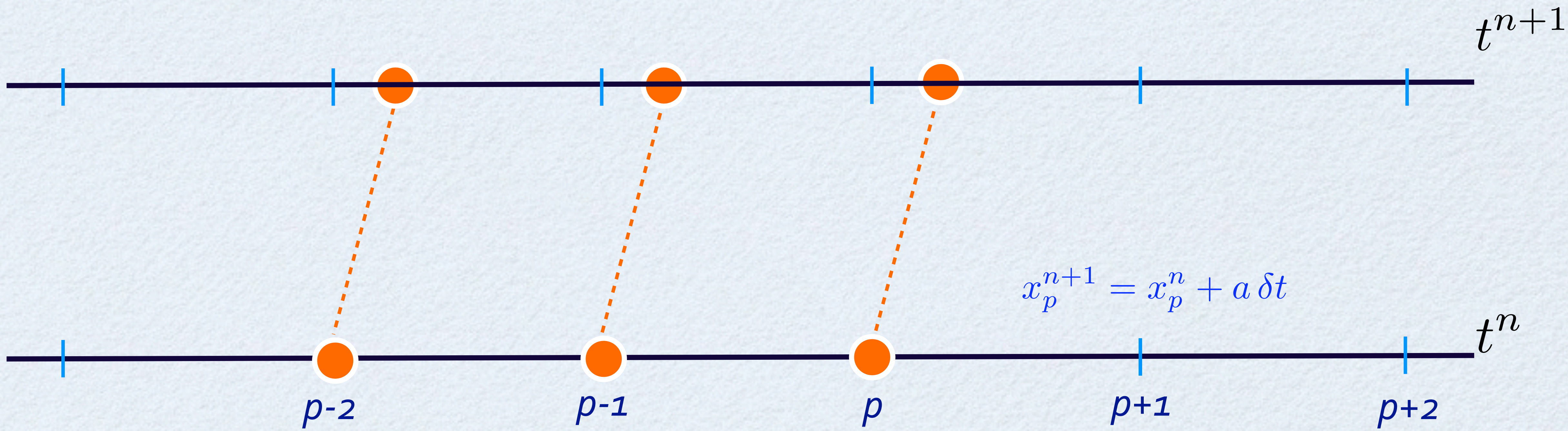
$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$

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

$$\frac{dx_p}{dt} = a$$

+ REMESH



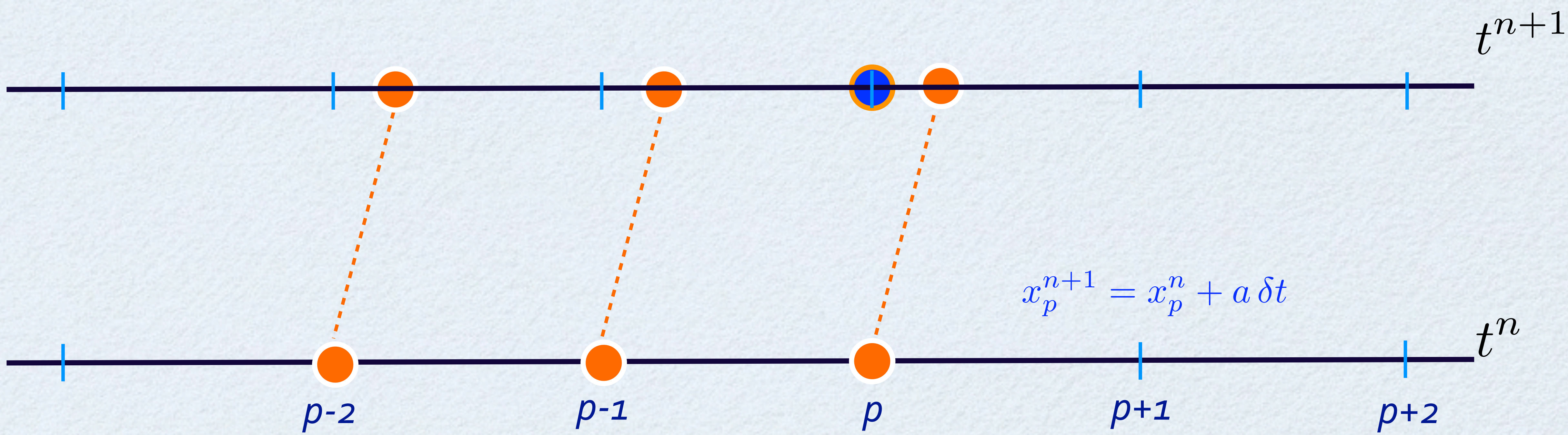
$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$

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

$$\frac{dx_p}{dt} = a$$

+ REMESH



$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

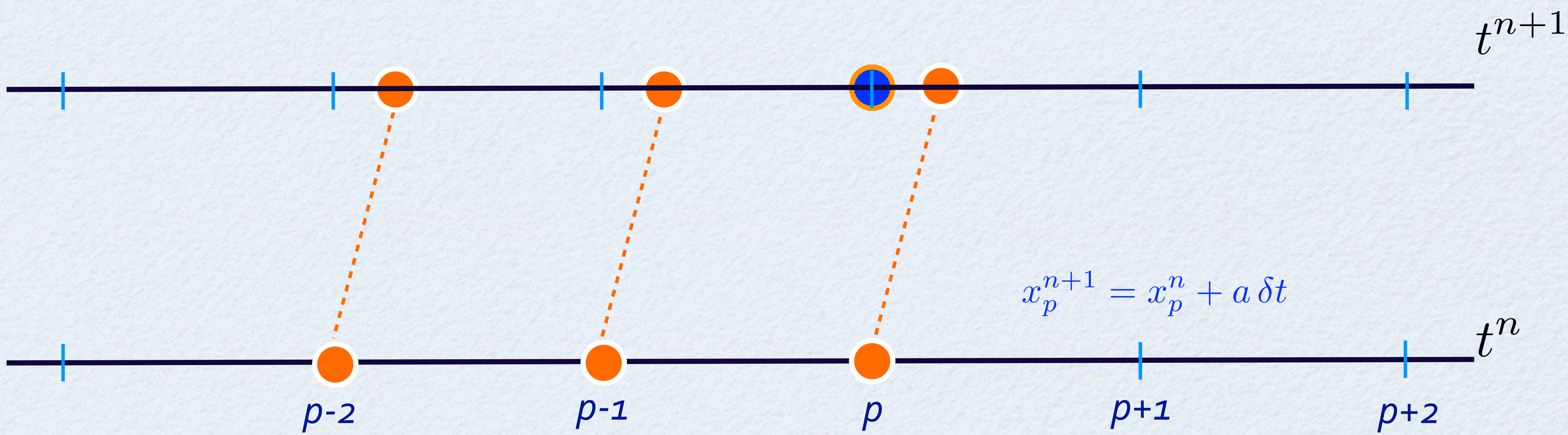
$u_p = u(x_p)h$ →

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

$$\frac{dx_p}{dt} = a$$

+ REMESH

$u_p^{n+1} =$



$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

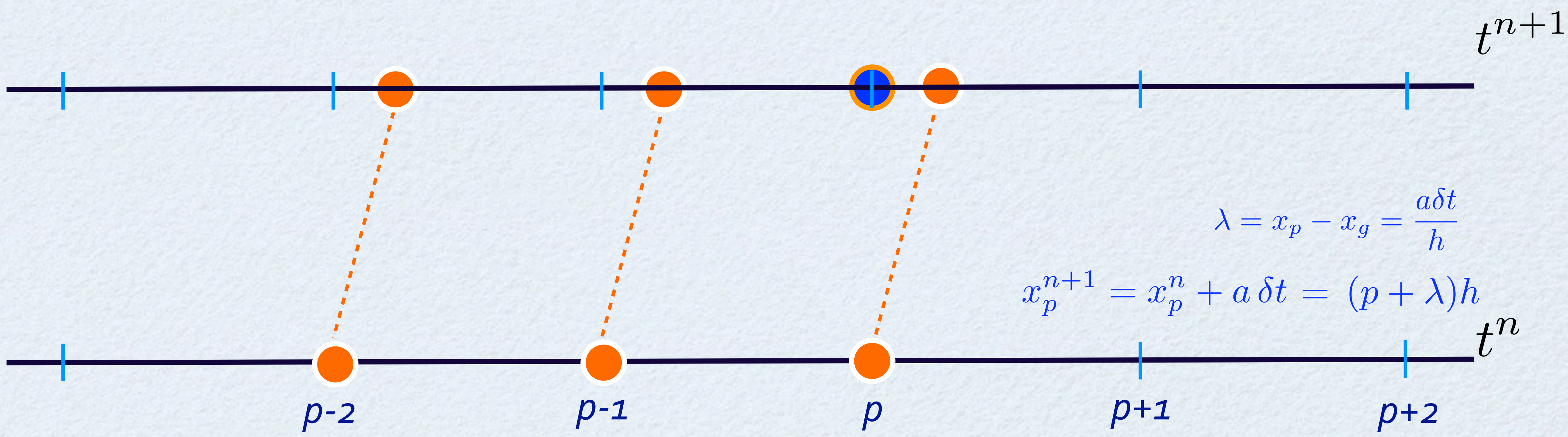
$u_p = u(x_p)h$ →

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

$$\frac{dx_p}{dt} = a$$

+ REMESH

$$u_p^{n+1} =$$



$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

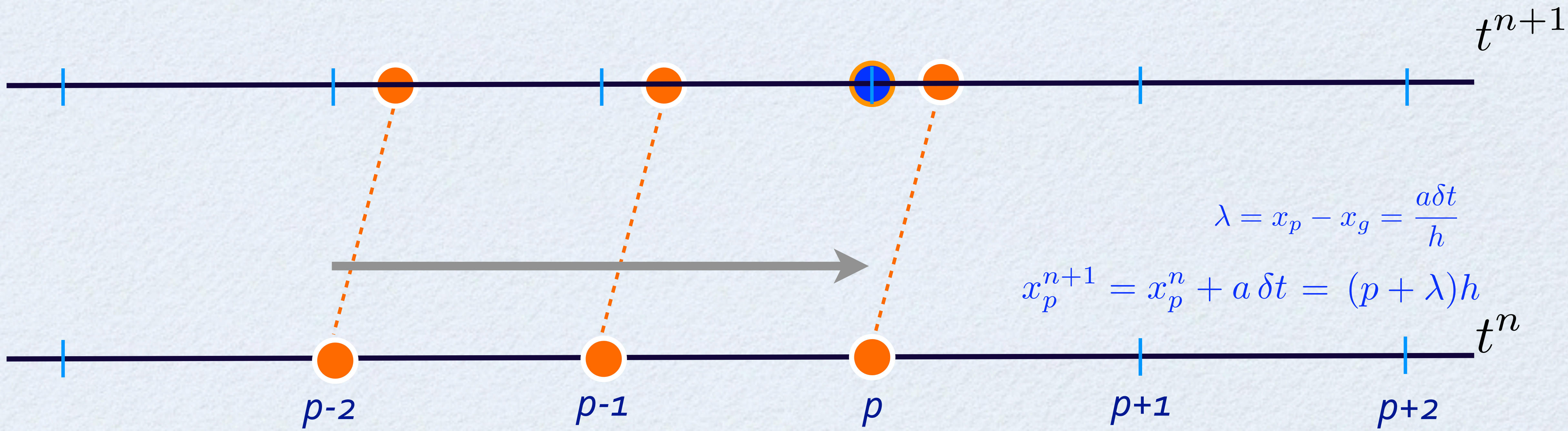
$u_p = u(x_p)h$ →

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

$$\frac{dx_p}{dt} = a$$

+ REMESH

$u_p^{n+1} \; =$



$$\lambda = x_p - x_g = \frac{a\delta t}{h}$$

$$x_p^{n+1} = x_p^n + a\,\delta t \;=\; (p+\lambda)h$$

$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

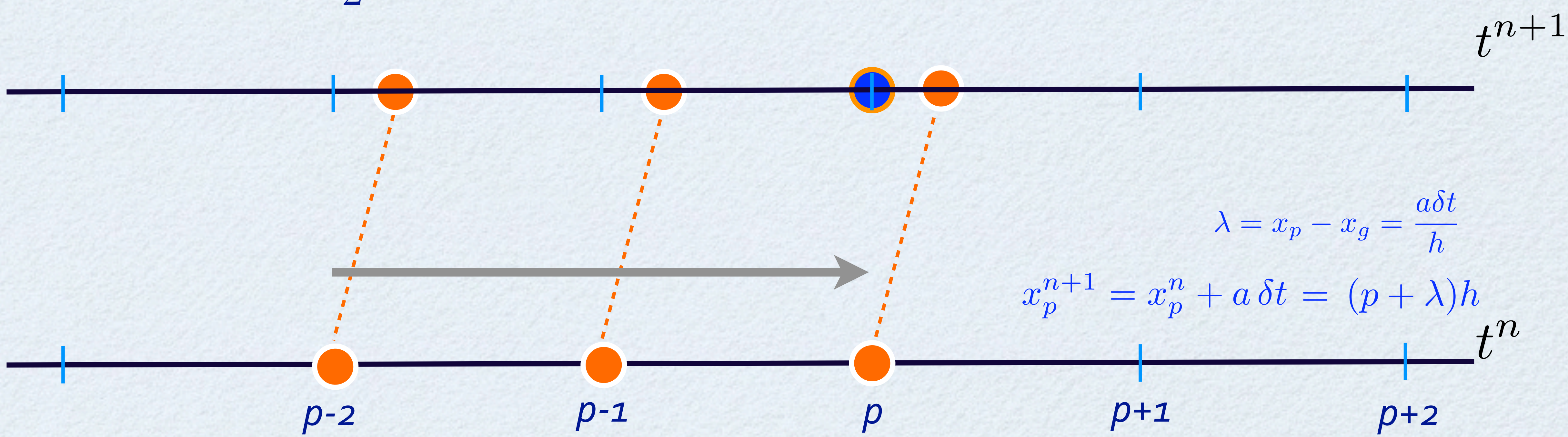
$u_p = u(x_p)h$ →

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

$$\frac{dx_p}{dt} = a$$

+ REMESH

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



$$\lambda = x_p - x_g = \frac{a\delta t}{h}$$

$$x_p^{n+1} = x_p^n + a\,\delta t = (p+\lambda)h$$

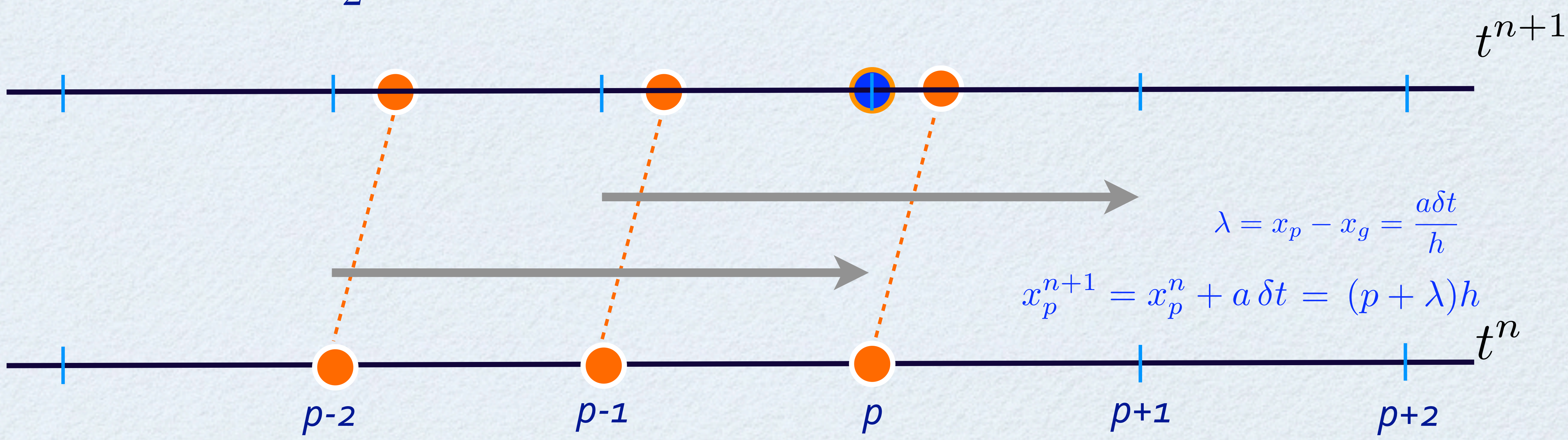
$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$ →

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

+ REMESH

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



$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$$

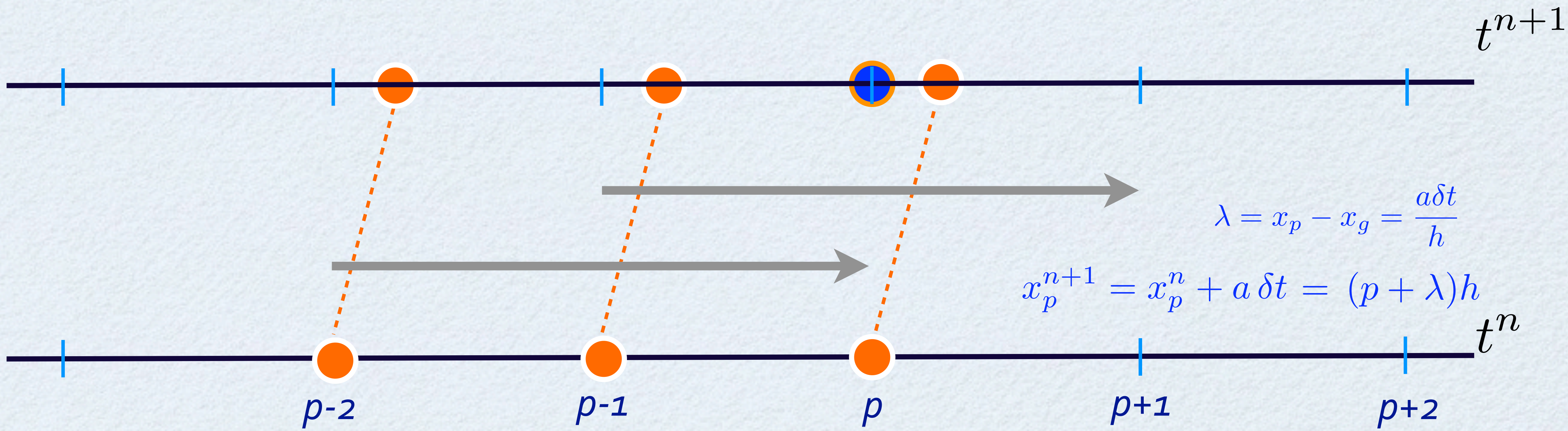
$u_p = u(x_p)h$ →

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

$$\frac{dx_p}{dt} = a$$

+ REMESH

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



$$\lambda = x_p - x_g = \frac{a\delta t}{h}$$

$$x_p^{n+1} = x_p^n + a\,\delta t = (p+\lambda)h$$

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$$

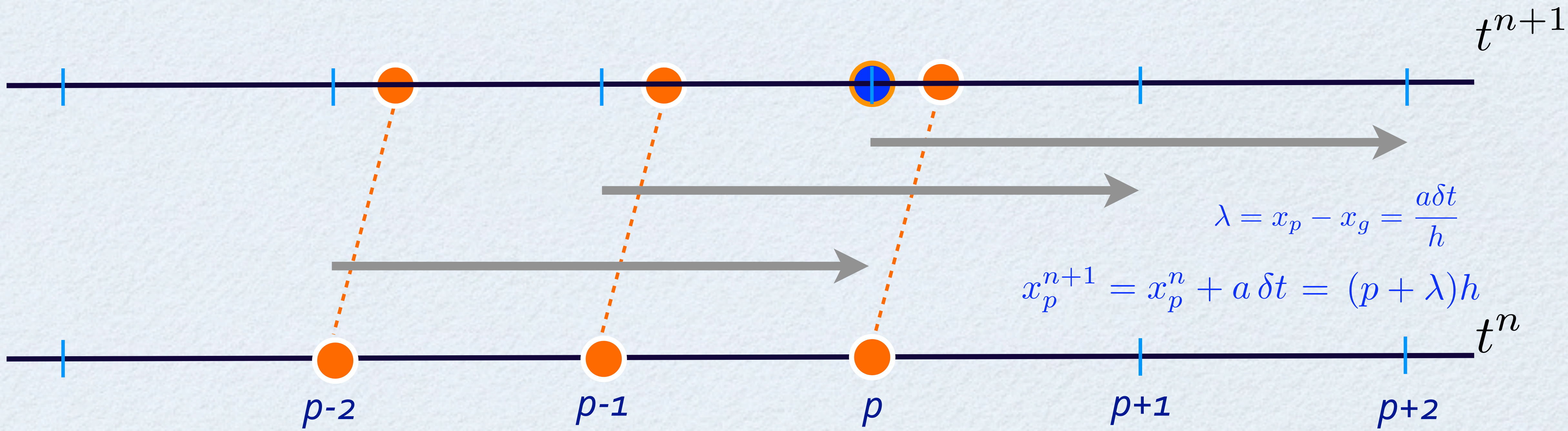
$u_p = u(x_p)h$

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

$$\frac{dx_p}{dt} = a$$

+ REMESH

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



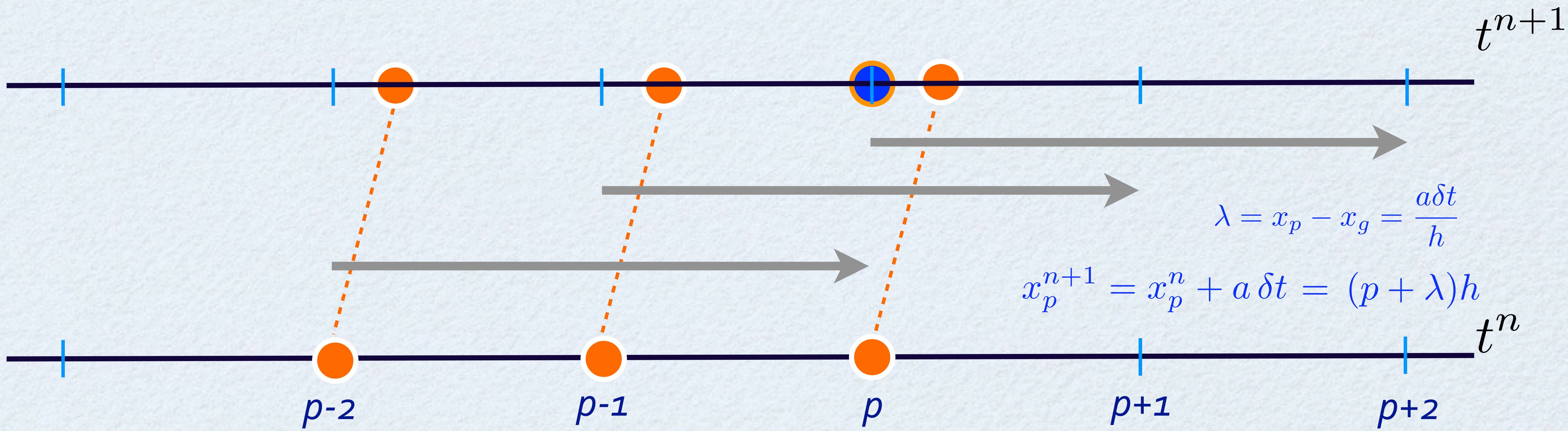
$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$ →

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

+ REMESH

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



$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

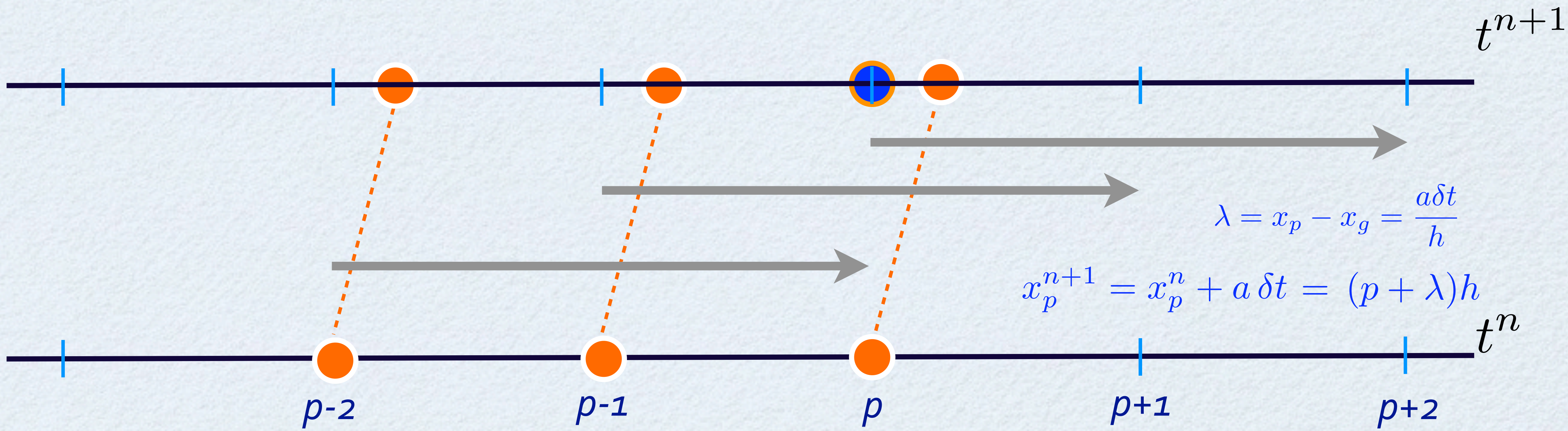
$u_p = u(x_p)h$ →

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

$$\frac{dx_p}{dt} = a$$

+ REMESH

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



$$\lambda = x_p - x_g = \frac{a\delta t}{h}$$

$$x_p^{n+1} = x_p^n + a\,\delta t = (p+\lambda)h$$

$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$ →

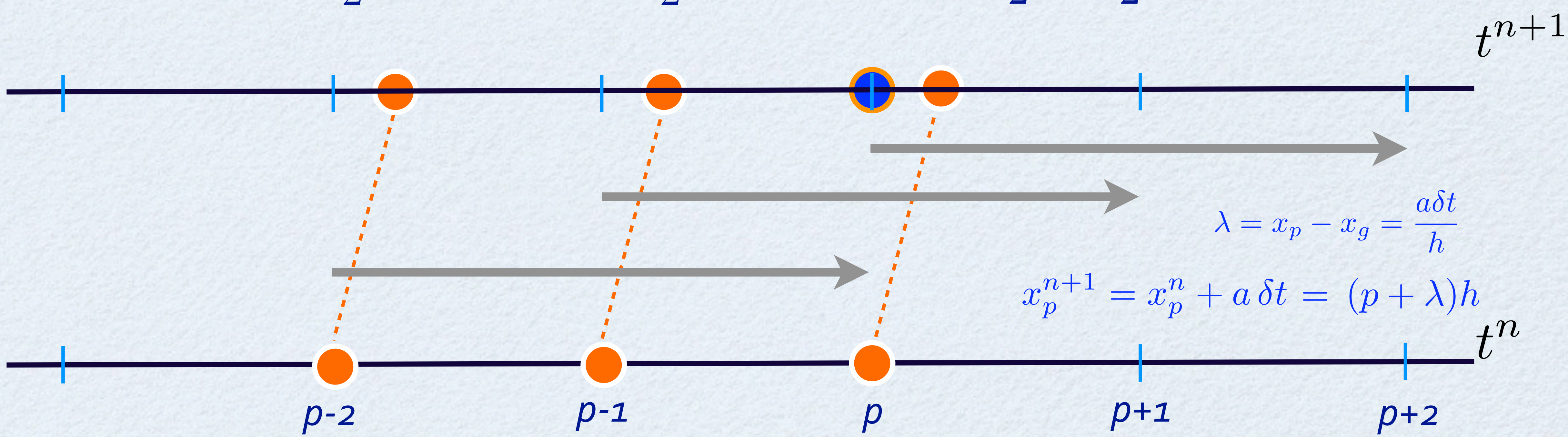
$$\frac{du_p}{dt} = 0$$

$$\frac{dx_p}{dt} = a$$

+ REMESH

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

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



$$\lambda = x_p - x_g = \frac{a\delta t}{h}$$

$$x_p^{n+1} = x_p^n + a\,\delta t = (p+\lambda)h$$

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$$

$$u_p = u(x_p)h$$

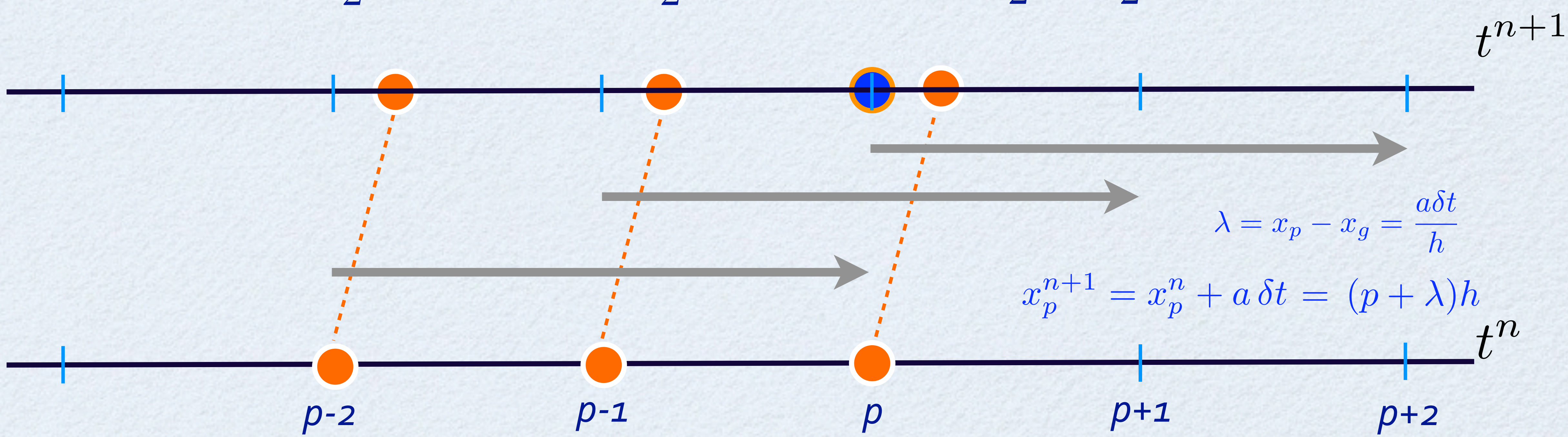
$$\frac{du_p}{dt} = 0 \qquad \frac{dx_p}{dt} = a$$

+ 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

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



$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$

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

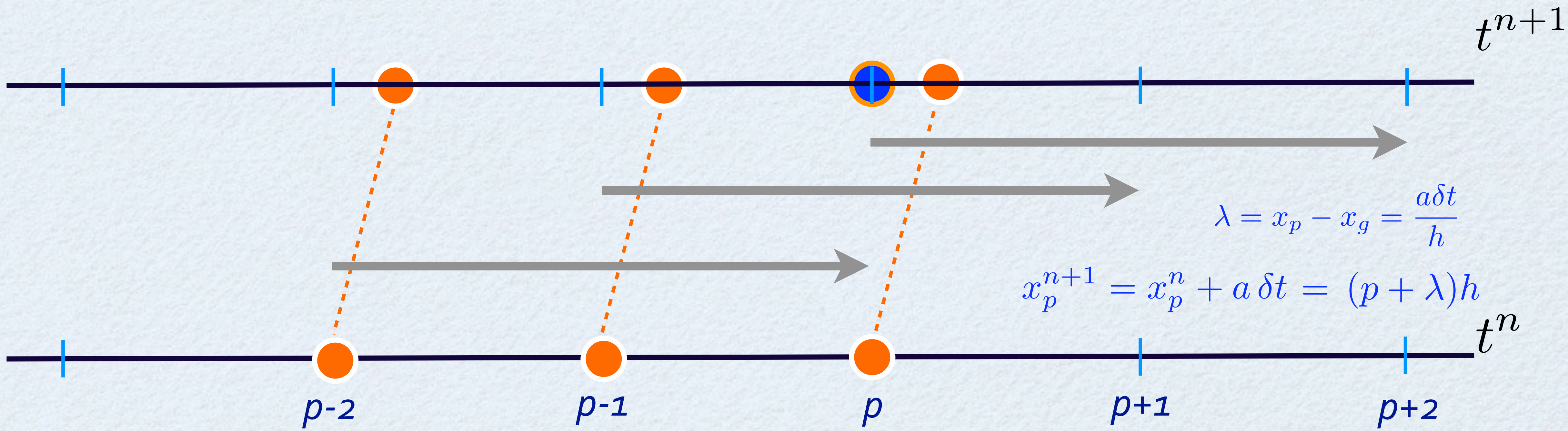
$$\frac{dx_p}{dt} = a$$

+ 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

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



$$\lambda = x_p - x_g = \frac{a\delta t}{h}$$

$$x_p^{n+1} = x_p^n + a\,\delta t = (p+\lambda)h$$

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$$

$u_p = u(x_p)h$ →

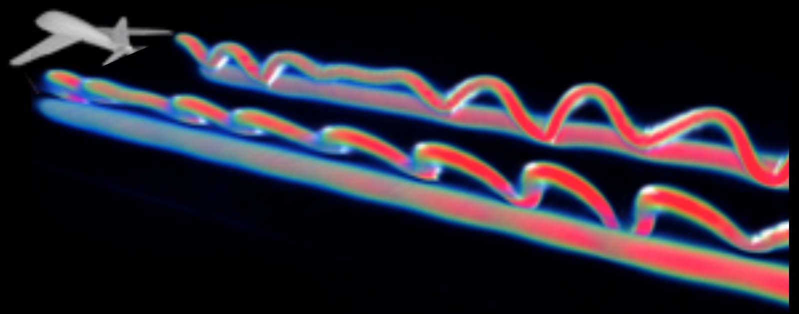
$$\frac{du_p}{dt} = 0 \qquad \frac{dx_p}{dt} = a$$

+ REMESH

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

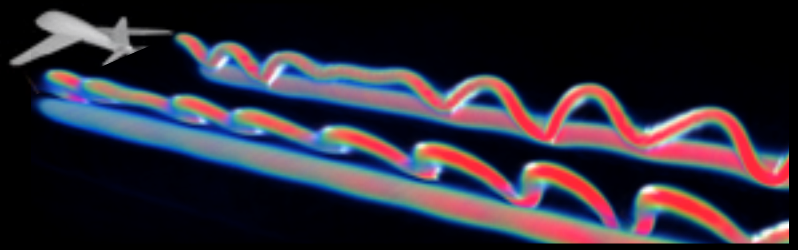
Euler Advect + One-sided Remesh = Beam-Warming FD

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



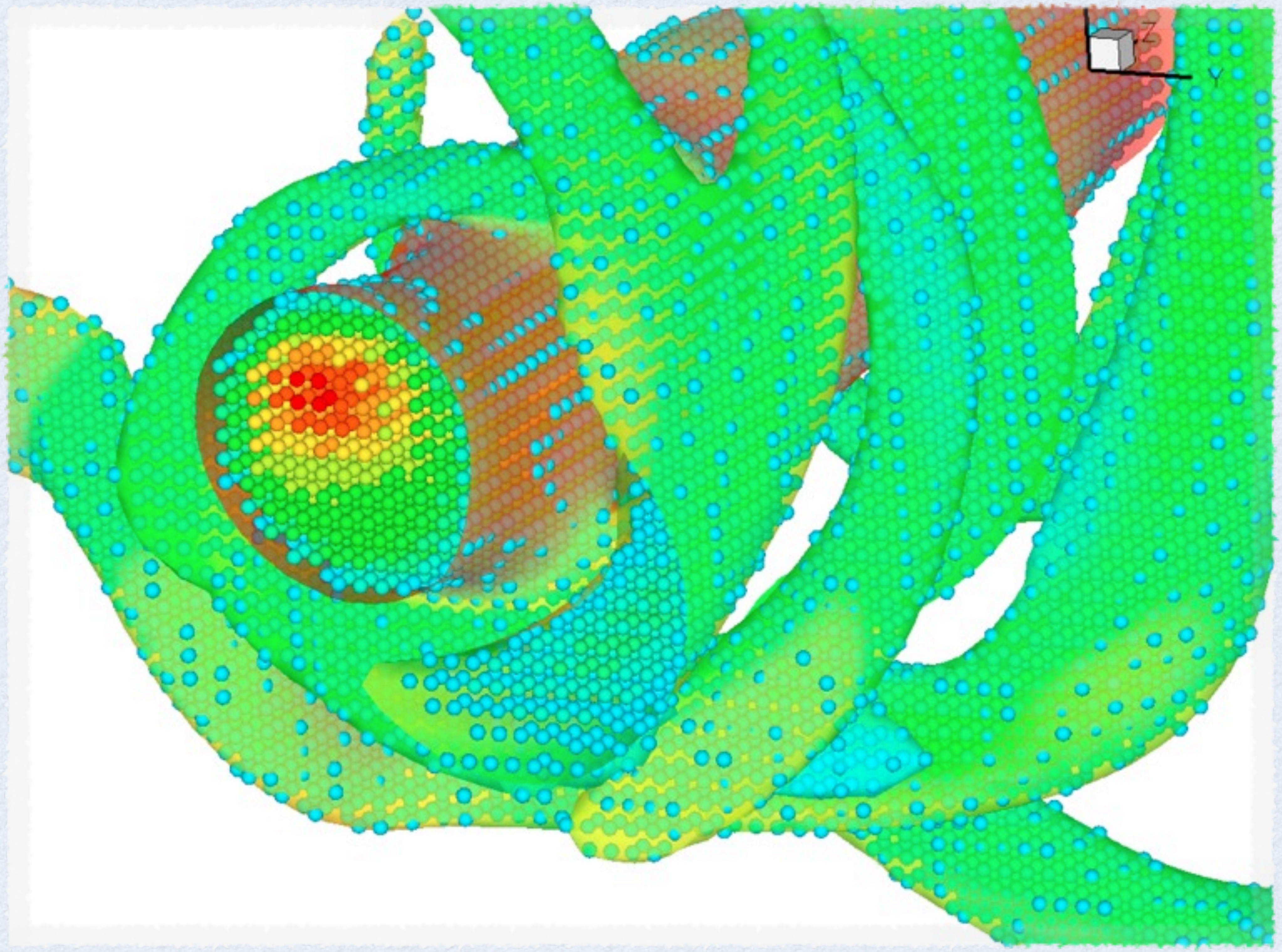
16K Cores – 10 Billion Particles – 60%

Runs at IBM Watson Center - BLue Gene/L



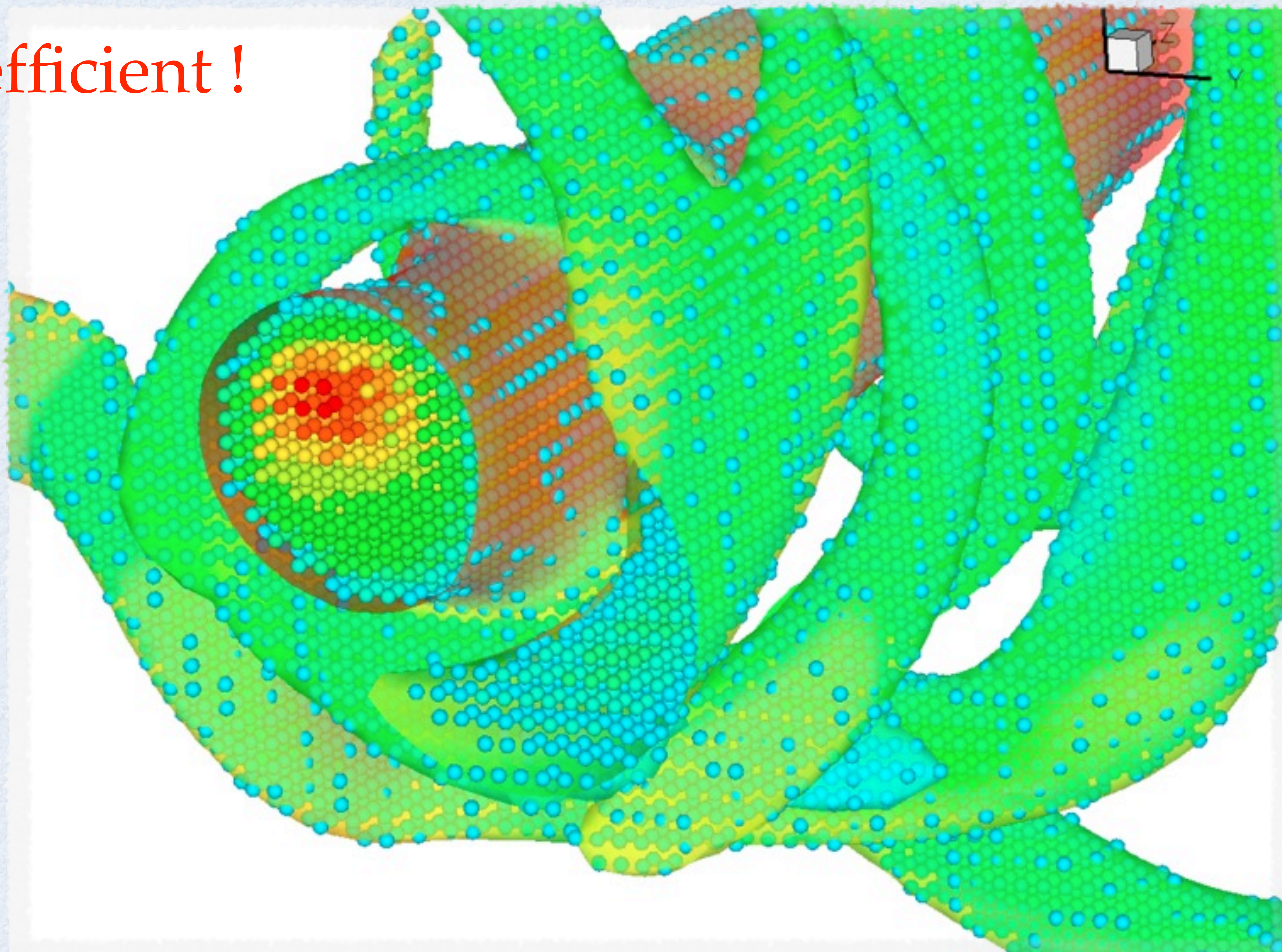
Chatelain P., Curioni A., Bergdorf M., Rossinelli D., Andreoni W., Koumoutsakos P., Billion Vortex Particle Direct Numerical Simulations of Aircraft Wakes, Computer Methods in Applied Mech. and Eng. 197/13-16, 1296-1304, 2008

PARTICLES ARE ADAPTIVE



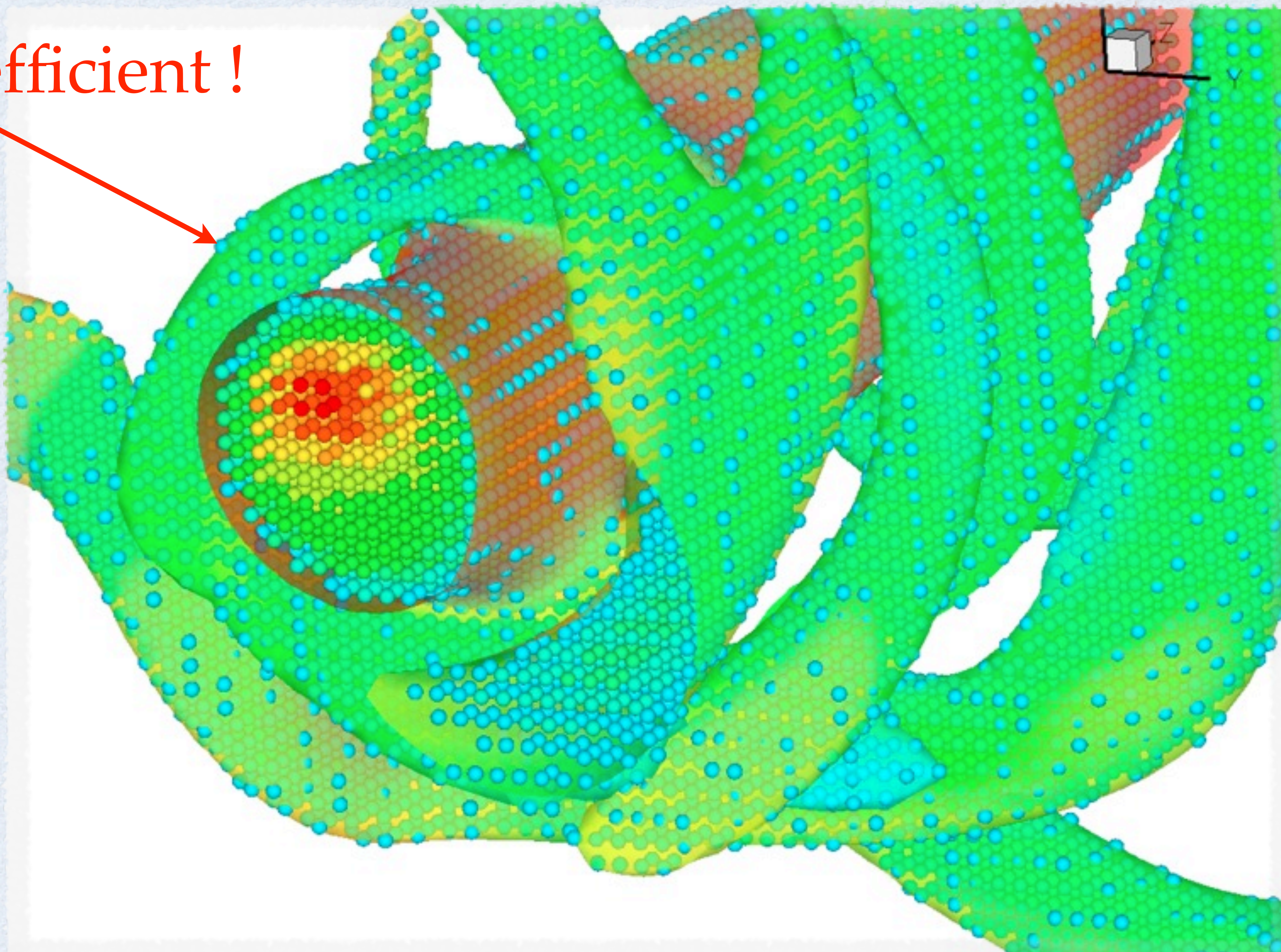
PARTICLES ARE ADAPTIVE

yet inefficient !

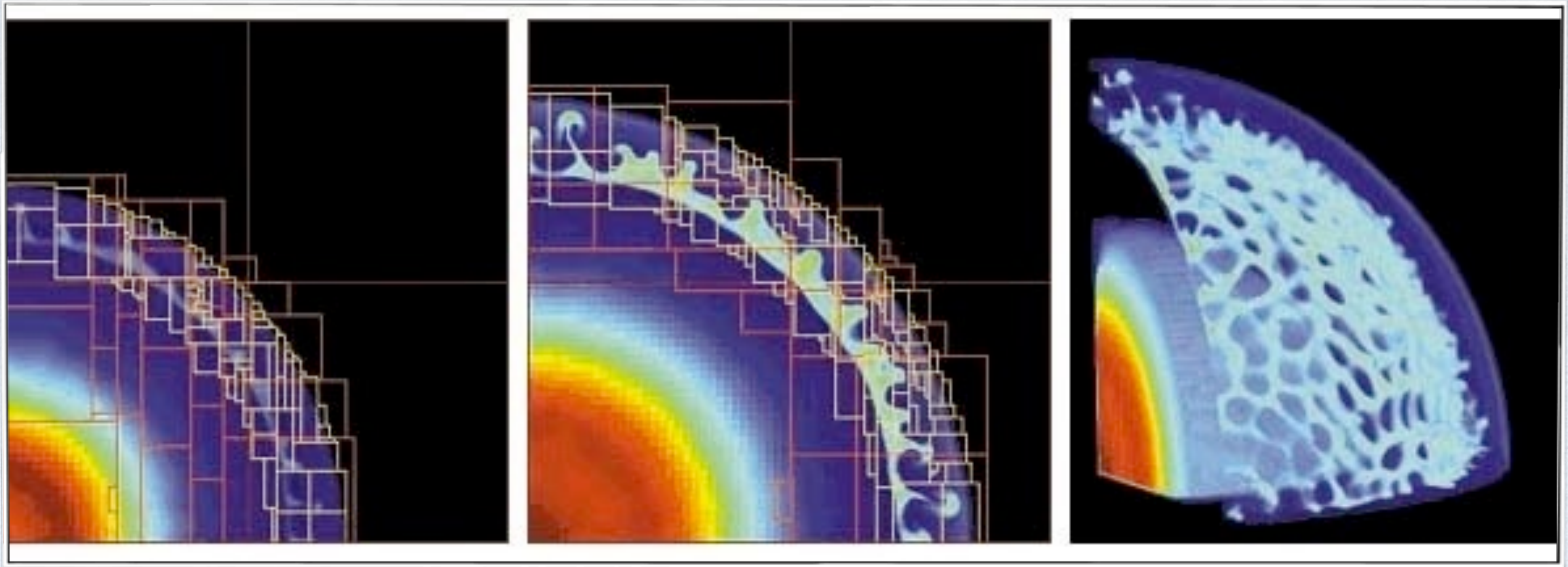


PARTICLES ARE ADAPTIVE

yet inefficient !

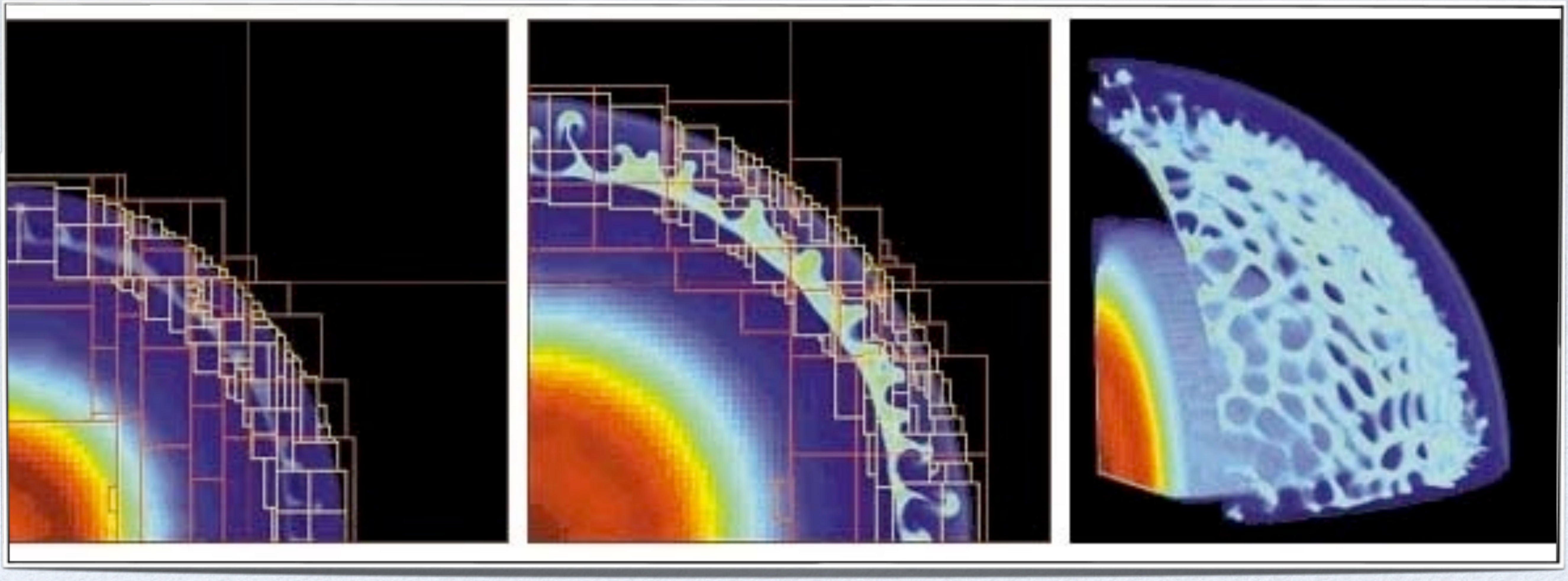


Adaptive Mesh Refinement



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

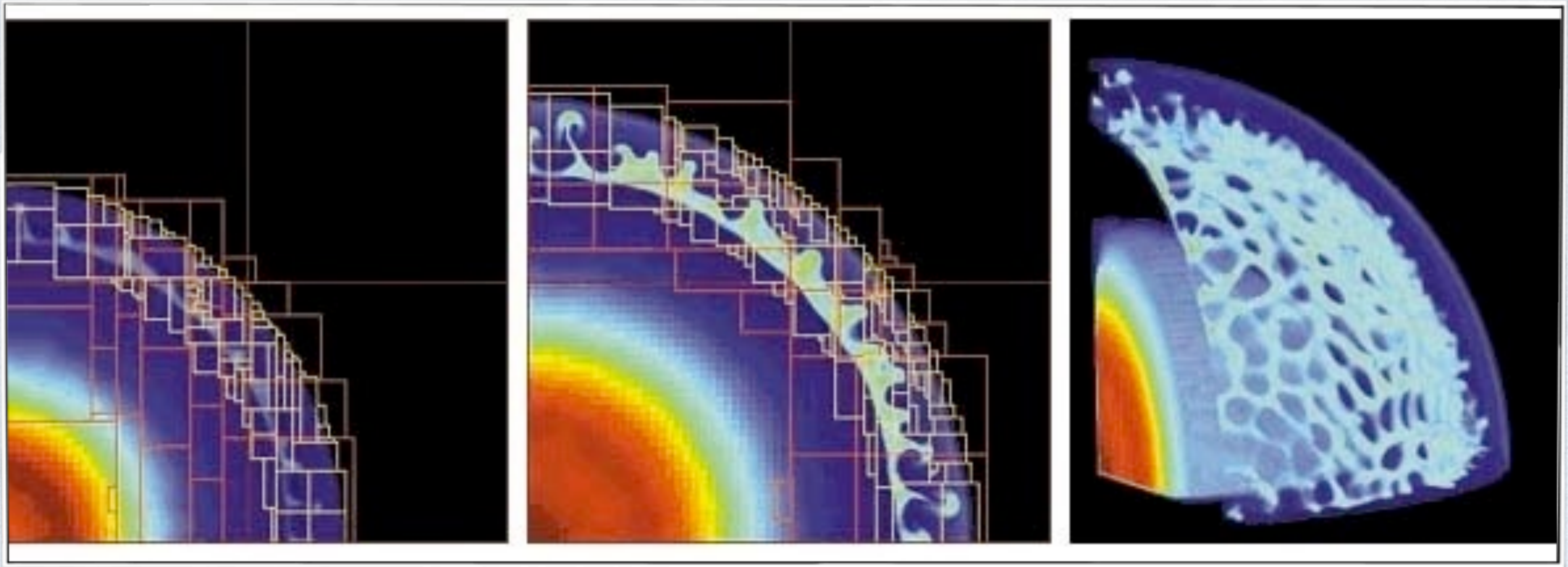
Adaptive Mesh Refinement



- Support of unstructured grids

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

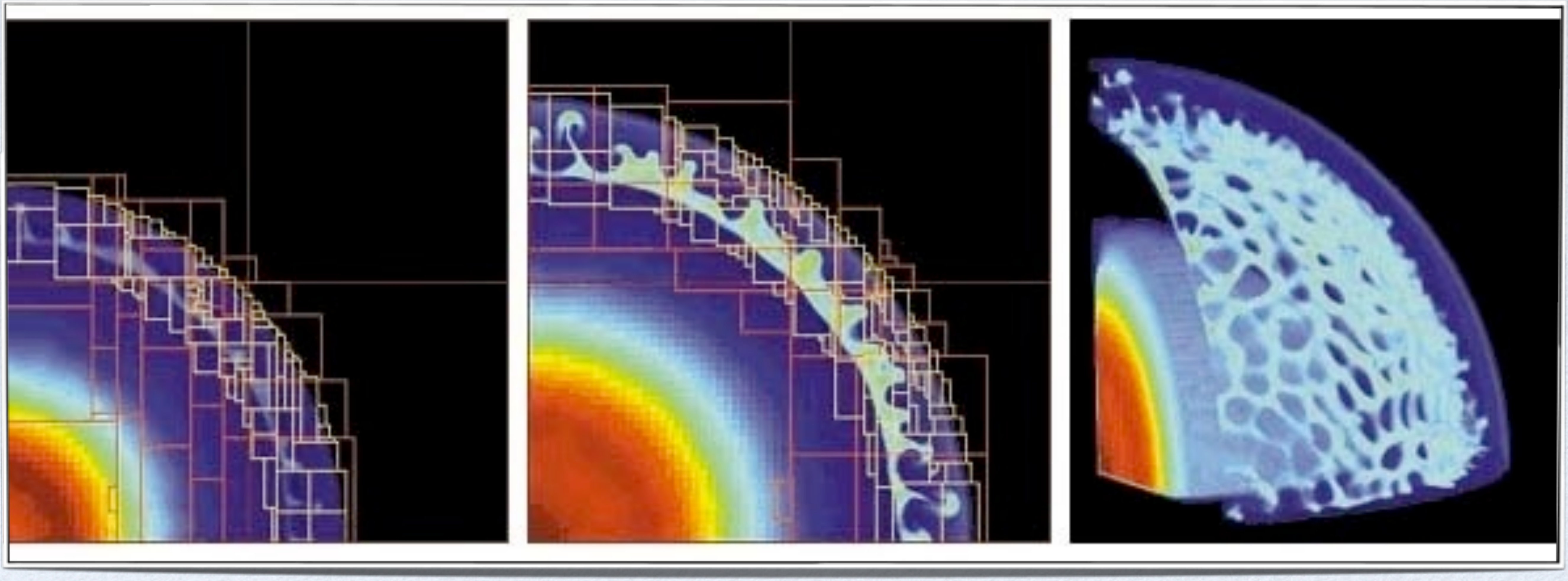
Adaptive Mesh Refinement



- Support of unstructured grids
- Different mesh orientations

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

Adaptive Mesh Refinement

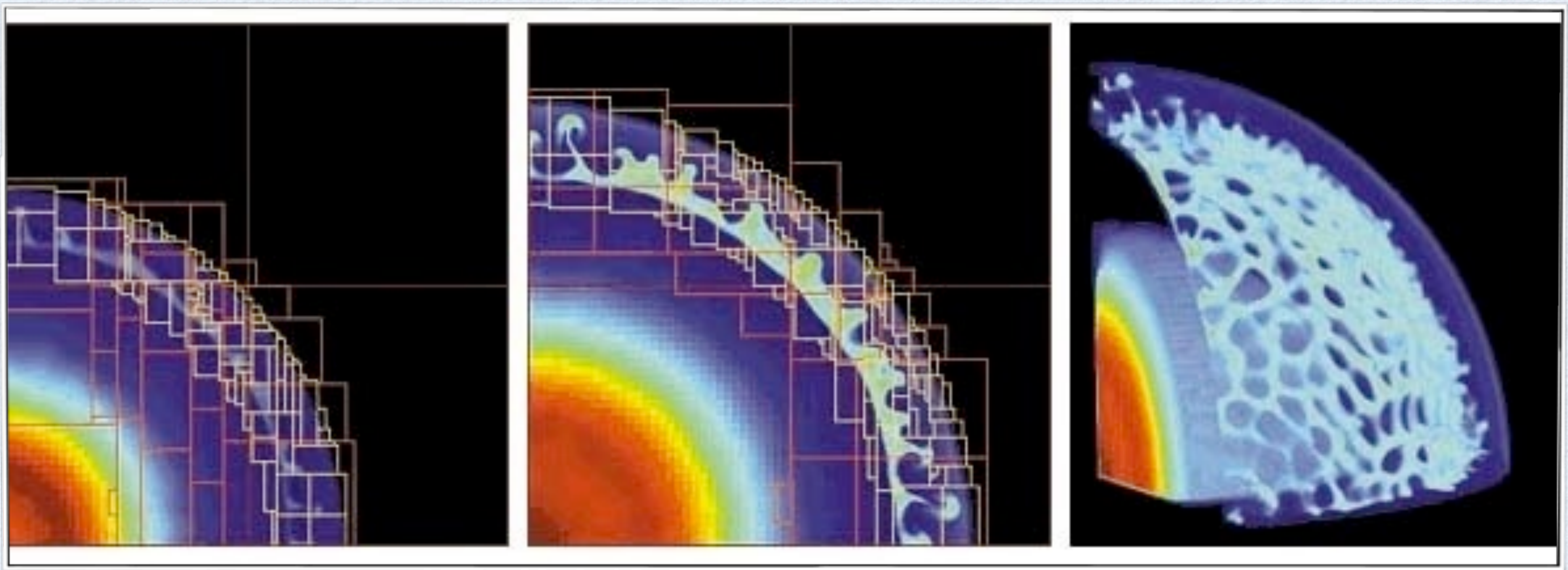


- Support of unstructured grids
- Different mesh orientations

- Low compression rate (Gradient, curvature)

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

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

Wavelet Compression

50:1

Multiresolution function representation:

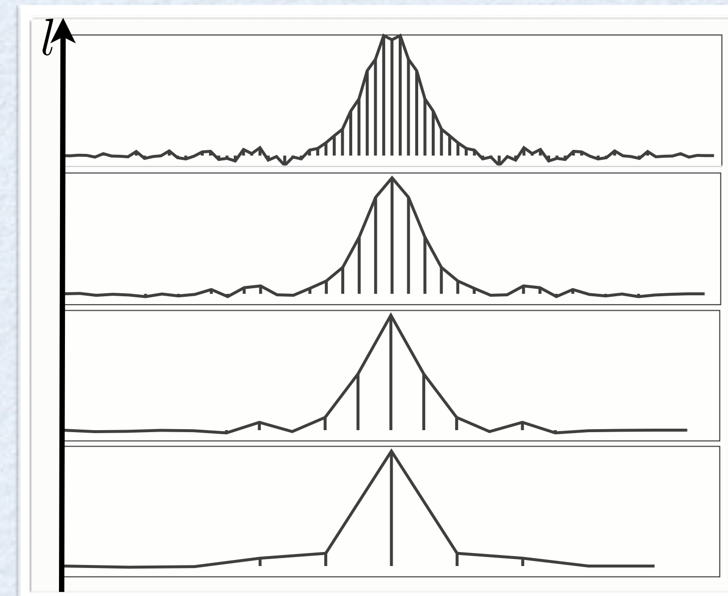
Analysis (collocation): d_k^l ~ | fine - Prediction(coarse) |

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

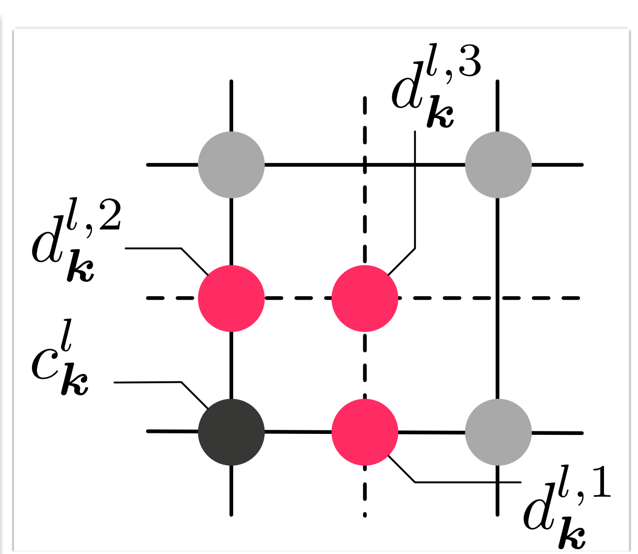
GROUND LEVEL

DETAIL COEFFICIENTS

WAVELETS



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

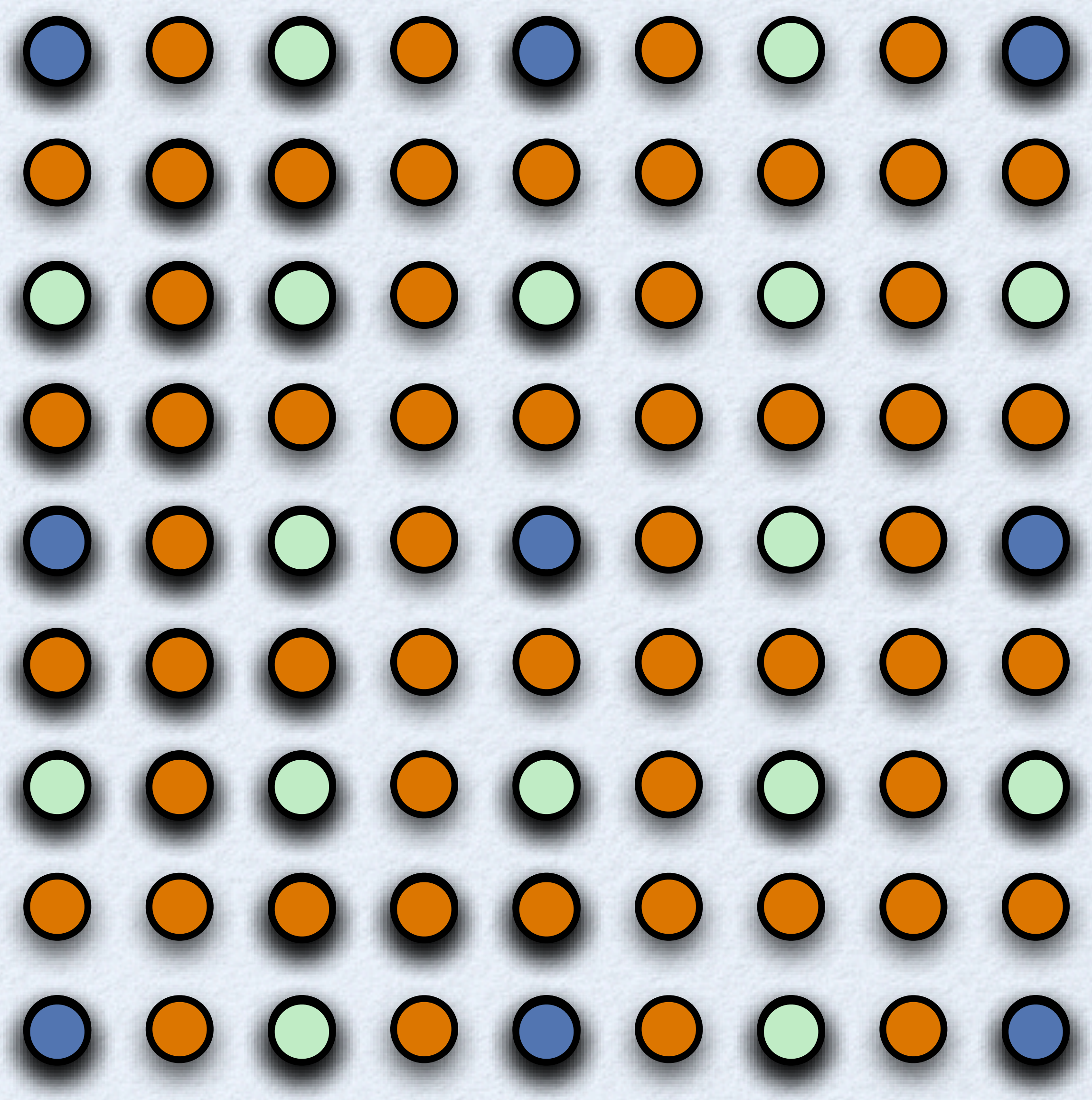
PARTICLETS : REMESHED PARTICLES + WAVELETS

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

"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

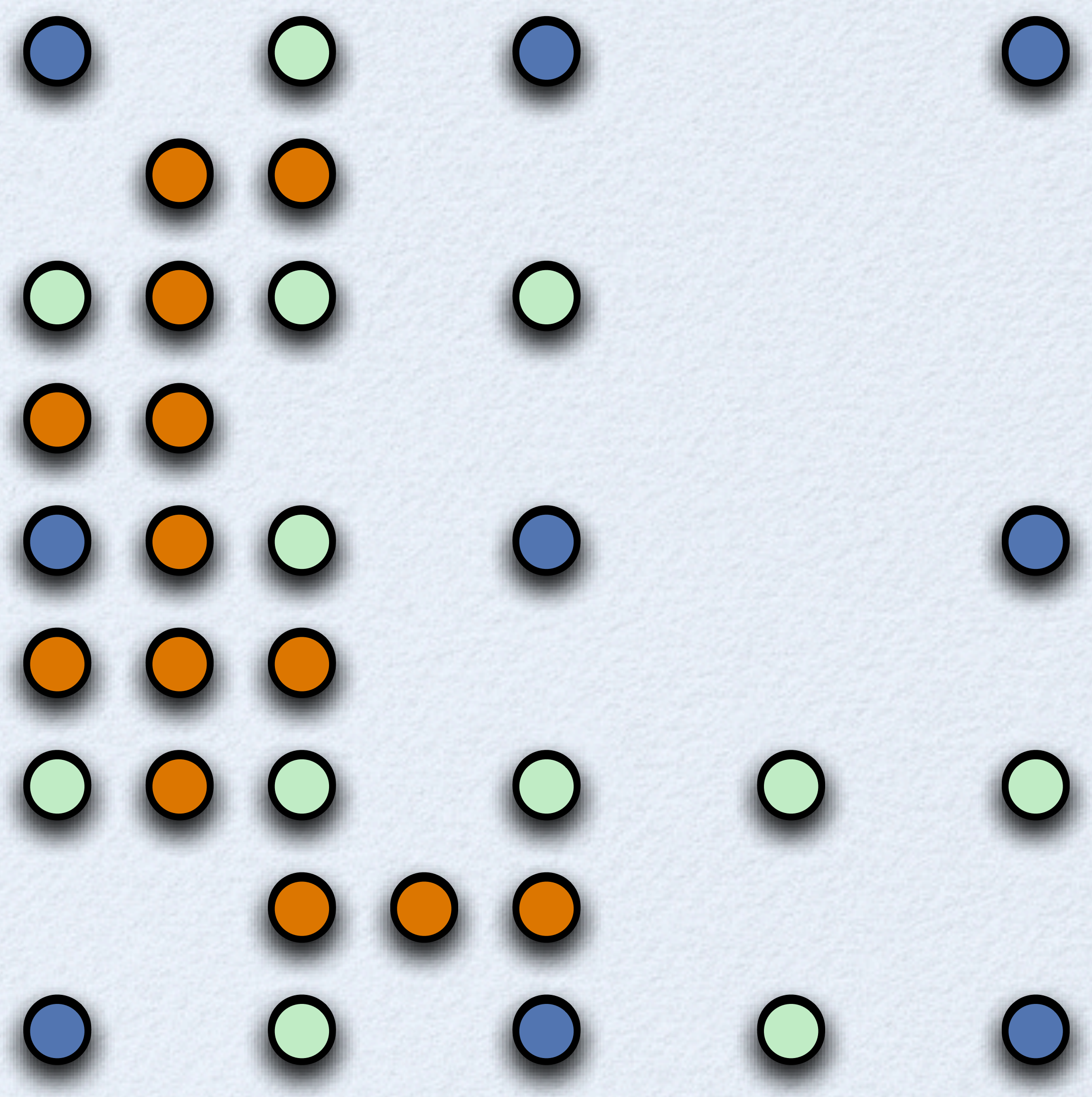
PARTICLETS : REMESHED PARTICLES + WAVELETS

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

"ground" level

detail coefficients

wavelets



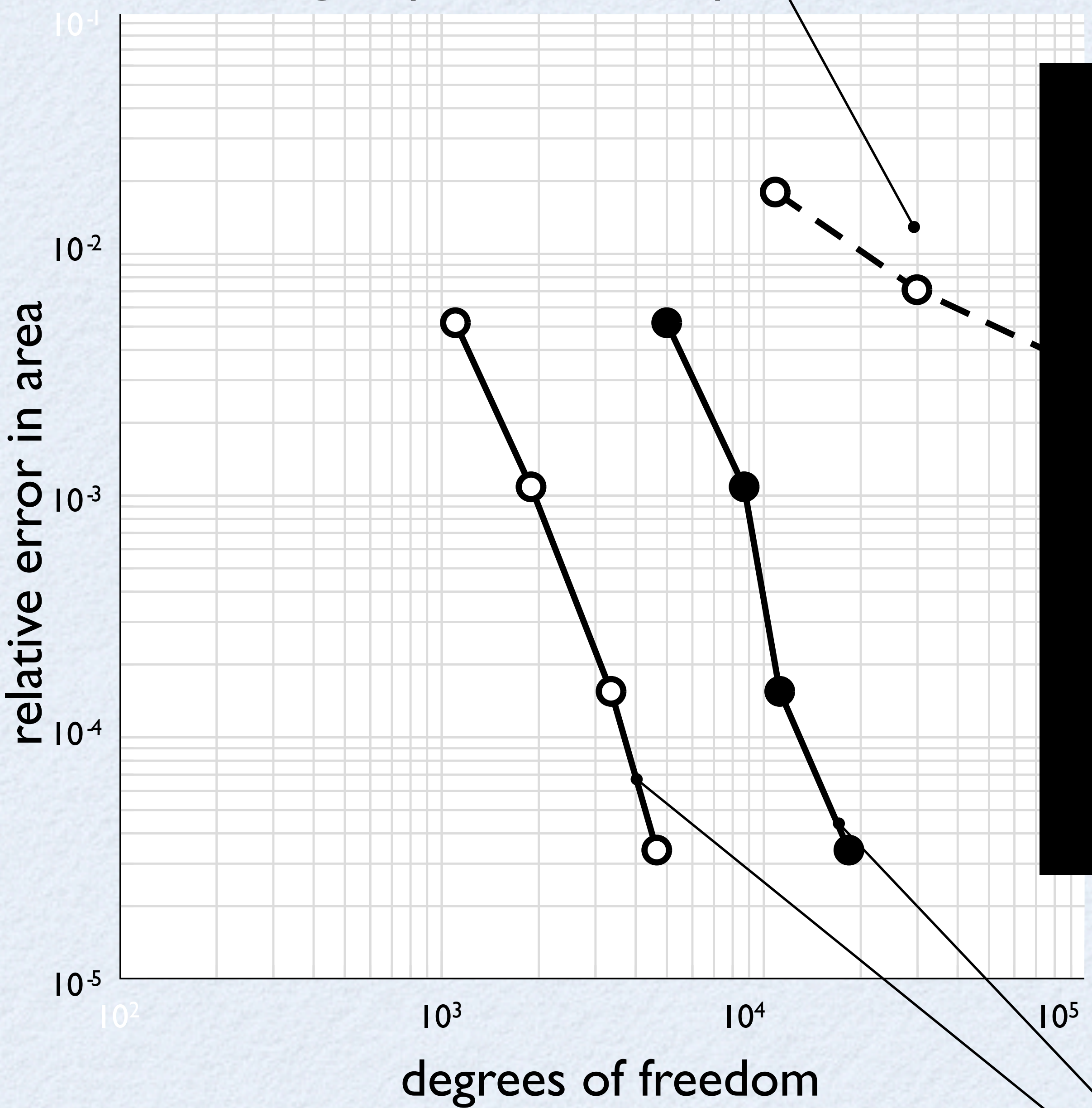
1. Remesh
2. Wavelets- Compress/Adapt
3. Convect
4. Wavelets Reconstruct
5. GOTO 1

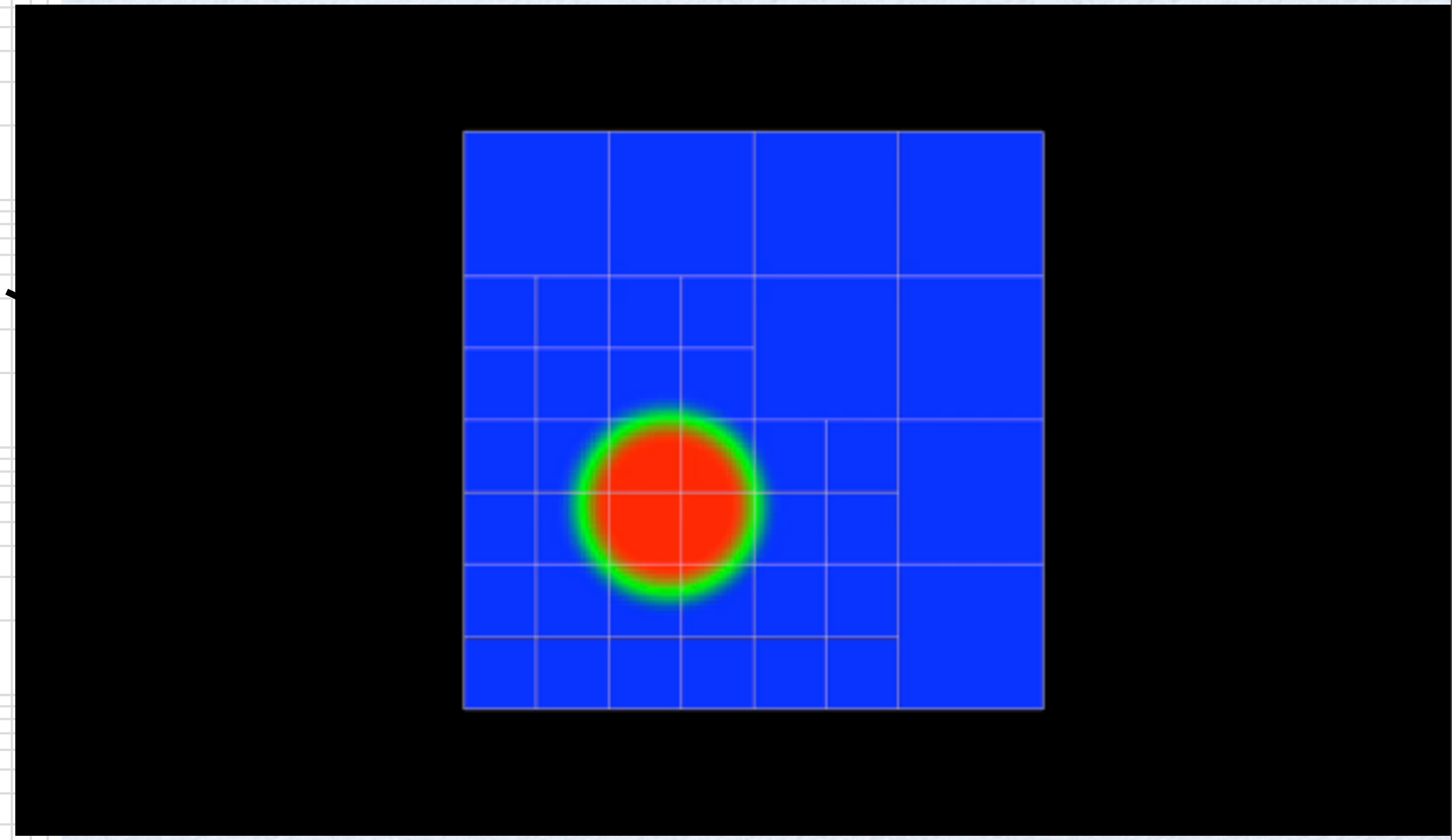
M. Bergdorf, P. Koumoutsakos. A Lagrangian Particle-Wavelet Method, Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal, 5(3), 980-995, 2006

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

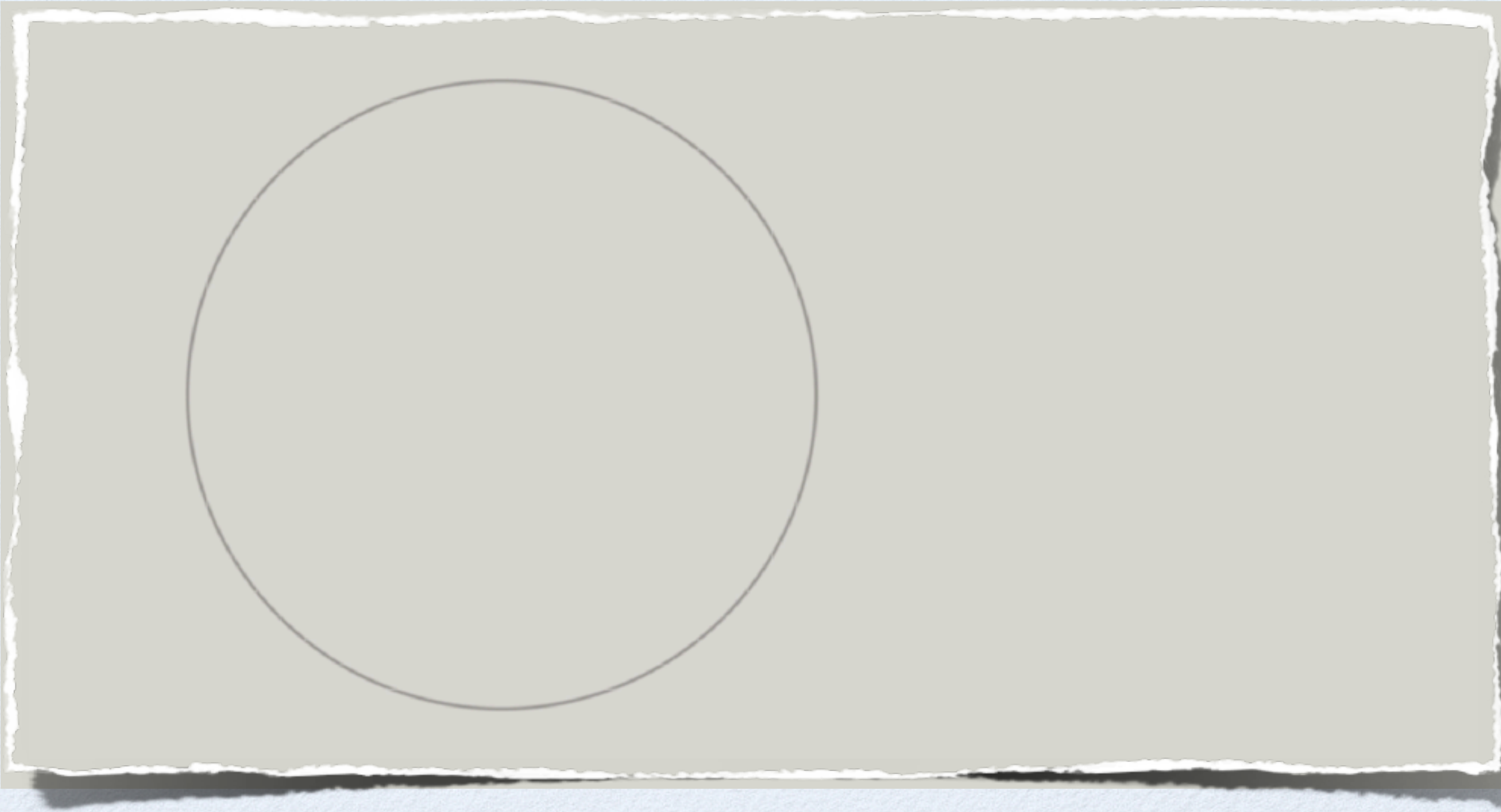
$CFL_{max} \sim 40$

Shock Bubble Interaction

(M=3, At=0.8)

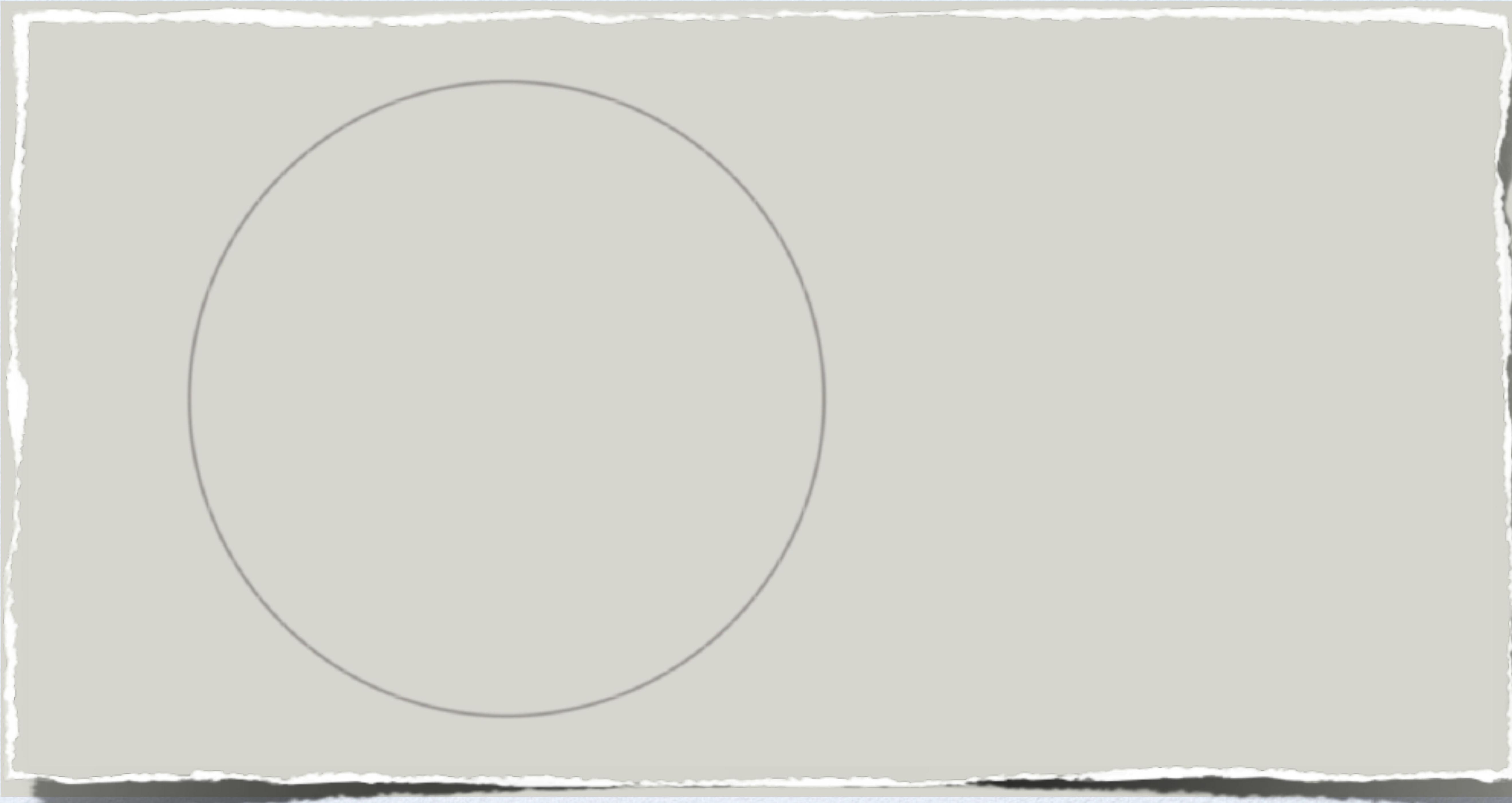
Shock Bubble Interaction

(M=3, At=0.8)



Shock Bubble Interaction

(M=3, At=0.8)



FINEST RESOLUTION EQUIVALENT : 8000 x 8000 uniform grid

~40 times smaller adaptive grid

BOUNDARIES + ALGORITHMS

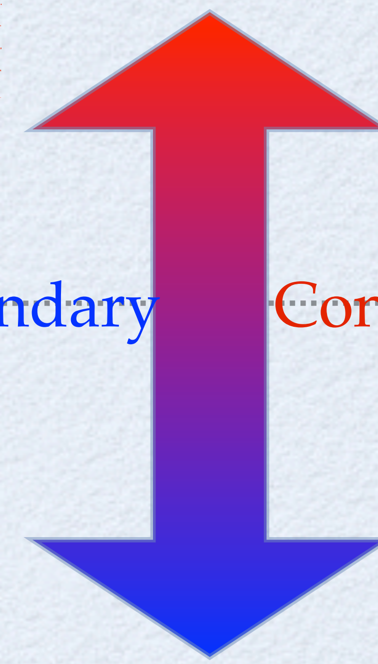
COUPLING AND BOUNDARY CONDITIONS

Atomistic:

Molecular Dynamics

$$m\frac{d^2r}{dt^2} = F$$

+ Fc



Boundary Conditions

+ Fa

m : mass **r** : position **F** : force

Continuum:

Navier- Stokes Eqs.

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2\mathbf{u}$$

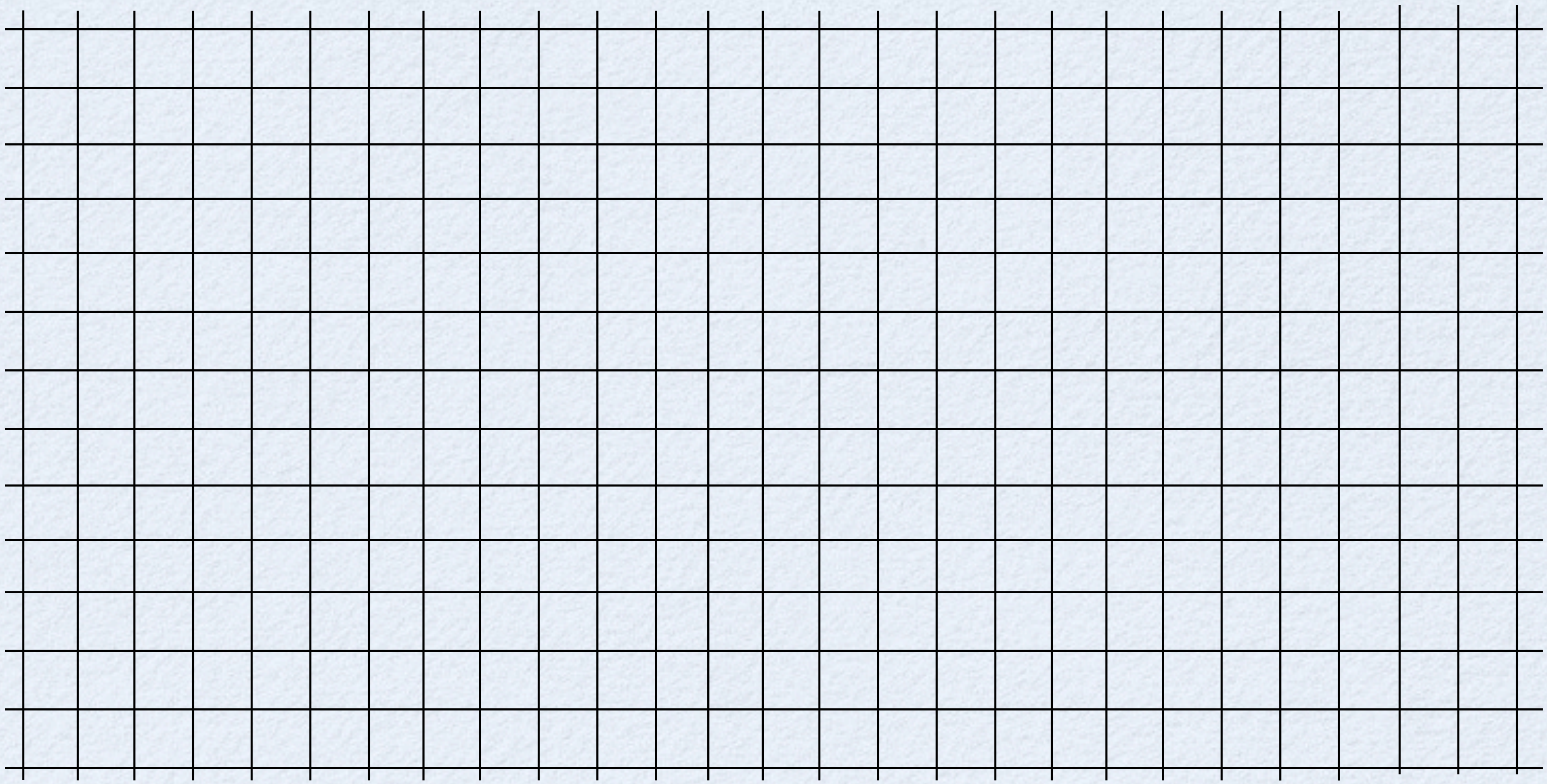
$$\frac{D\rho}{Dt} = \rho\nabla\cdot\mathbf{u}$$

u : velocity, **P** : pressure, ρ : density, ν : viscocity

Boundary Conditions = Coupling

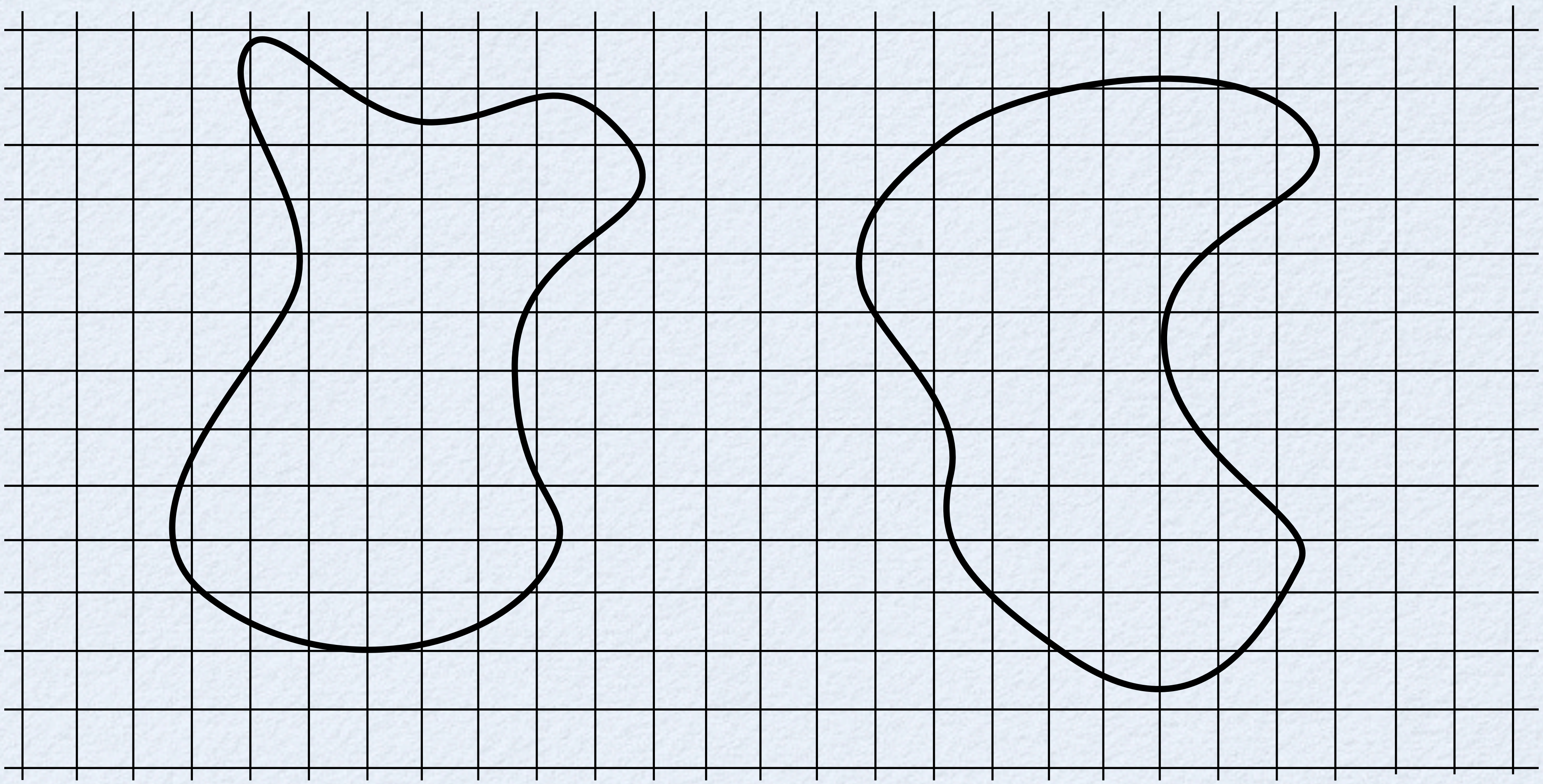
$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + f (\text{enforces b.c.})$$

Boundary Conditions = Coupling



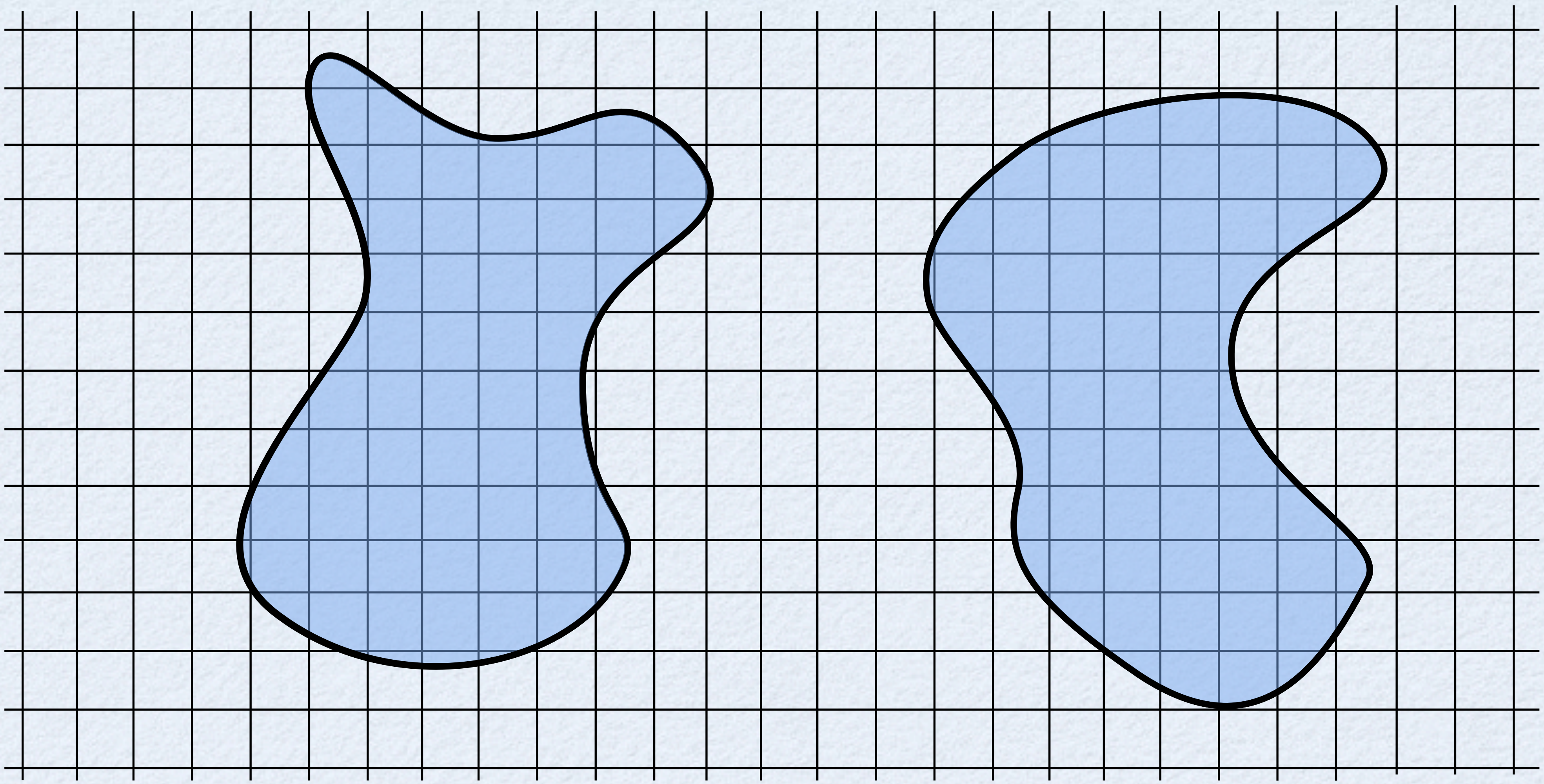
$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} \; + f(\text{enforces b.c.})$$

Boundary Conditions = Coupling



$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} \; + f(\text{enforces b.c.})$$

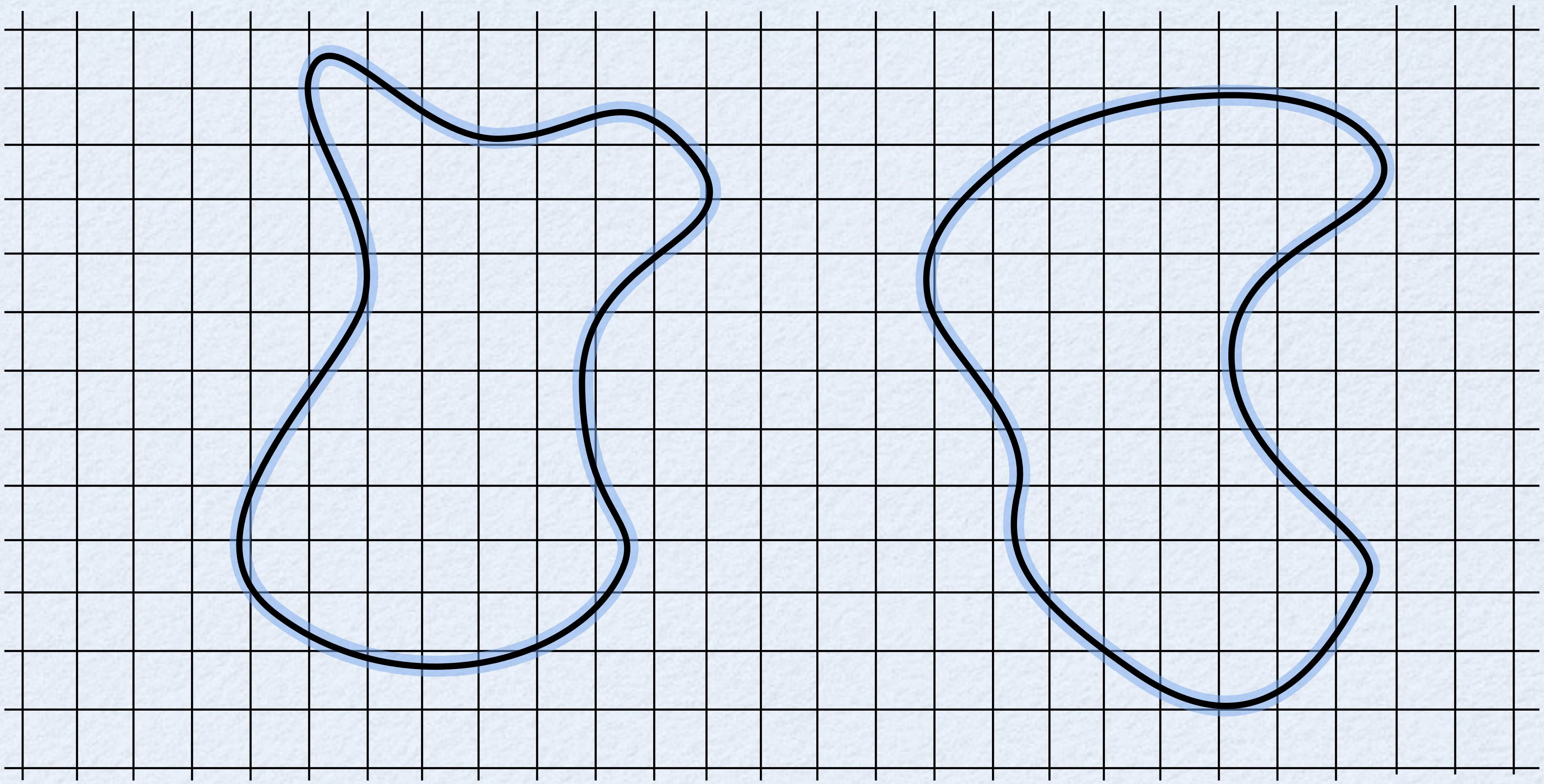
Boundary Conditions = Coupling



$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} \; + f(\text{enforces b.c.})$$

Penalization Method: $f(\mathbf{x}) = \lambda \chi_S (\mathbf{u}_S - \mathbf{u})$

Boundary Conditions = Coupling



$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} \; + f(\text{enforces b.c.})$$

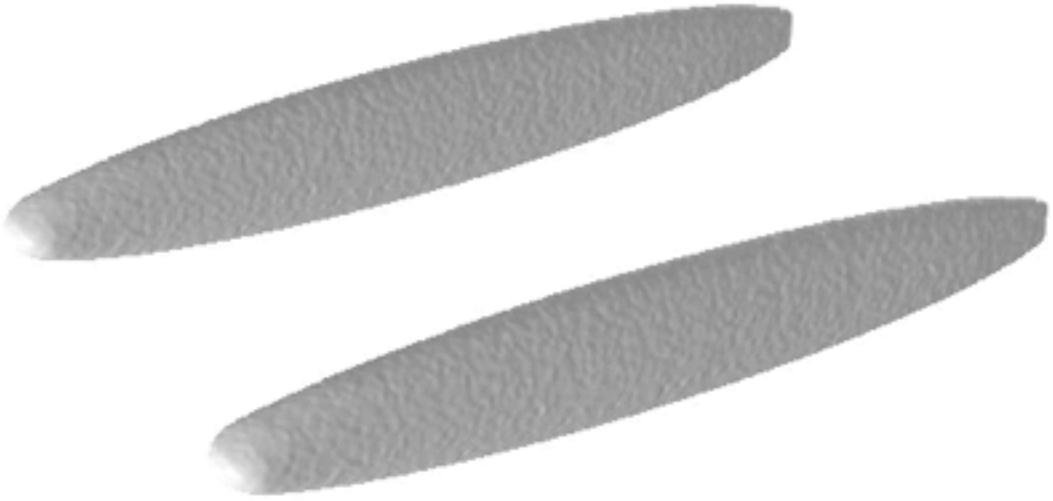
Penalization: $f(\mathbf{x}) = \lambda \chi_S (\mathbf{u}_S - \mathbf{u})$

P. Angot, C. H. Bruneau, and P. Fabrie,, Numer. Math. , 1999

Immersed Boundary: $f(\mathbf{x}) = \kappa \delta_S (\mathbf{x}_S - \mathbf{x})$

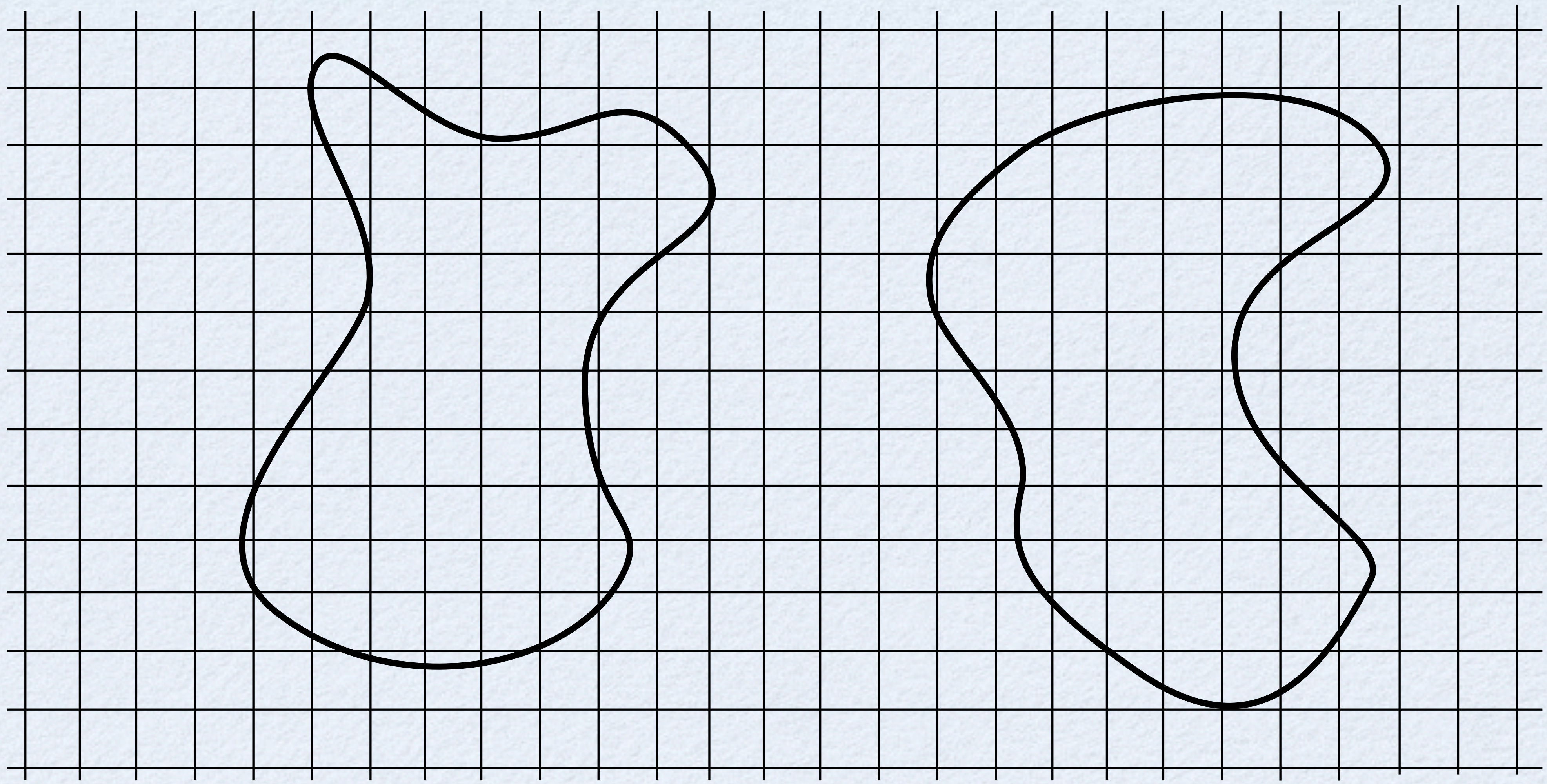
C.S. Peskin, J. Comput. Phys., (1977)

FISH SCHOOLING



2 FISH (OBVIOUSLY)

Boundary Conditions = Coupling

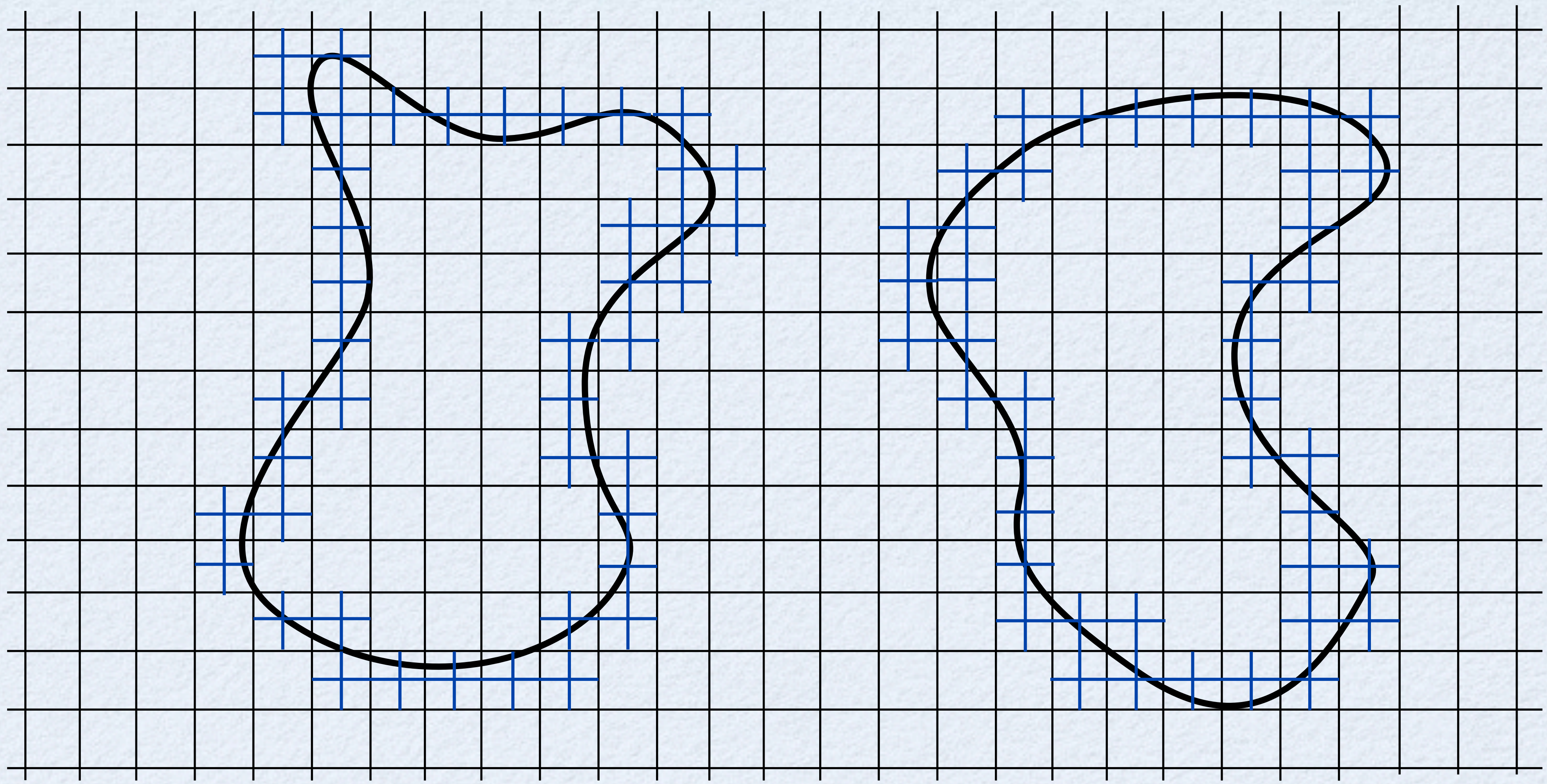


$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + f(\text{enforces b.c.})$$

Penalization Method: $f(\mathbf{x}) = \lambda \chi_S(\mathbf{u}_S - \mathbf{u})$

Immersed Boundary Method: $f(\mathbf{x}) = \kappa \delta_S(\mathbf{x}_S - \mathbf{x})$

Boundary Conditions = Coupling

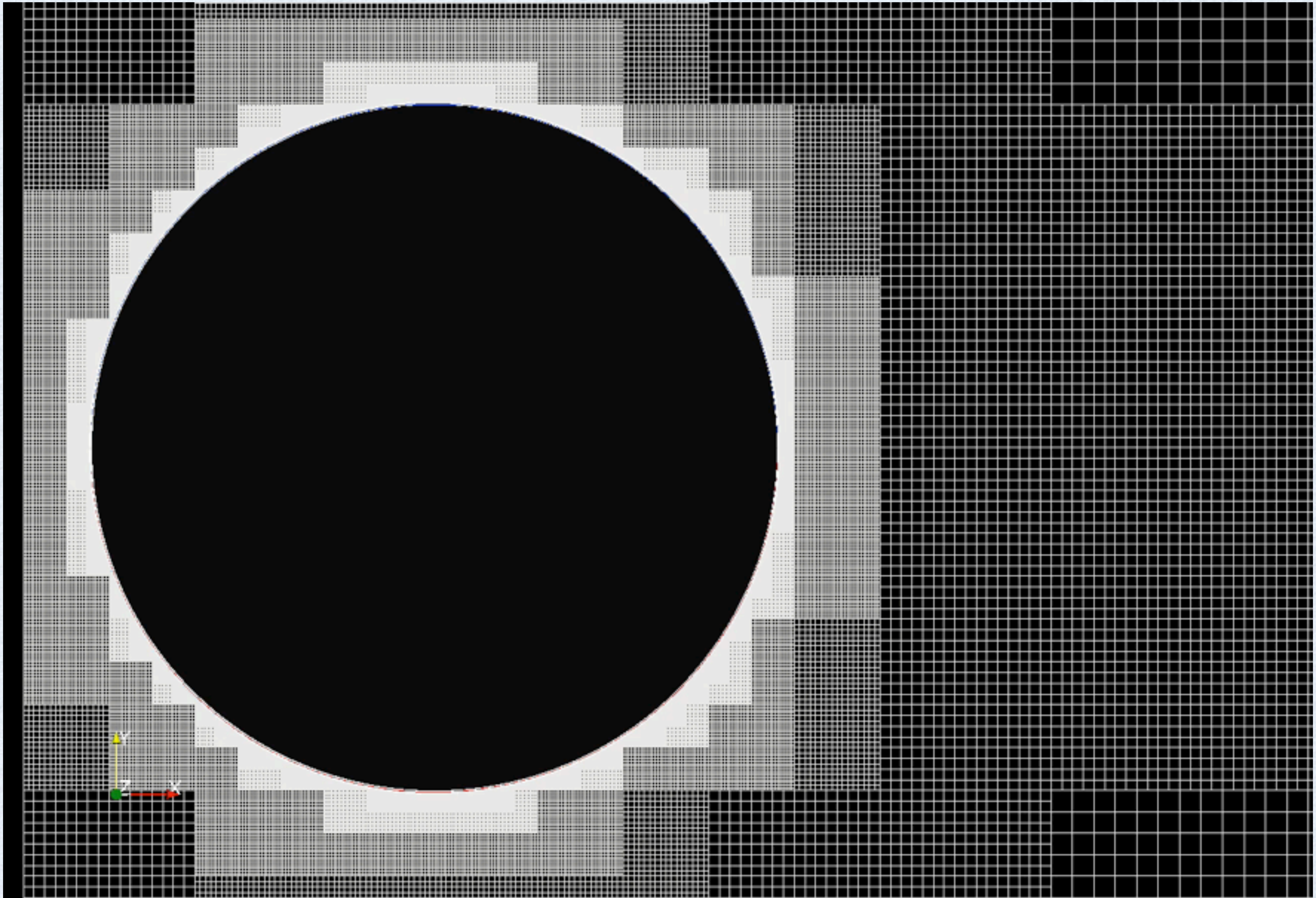


$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} \; + f(\text{enforces b.c.})$$

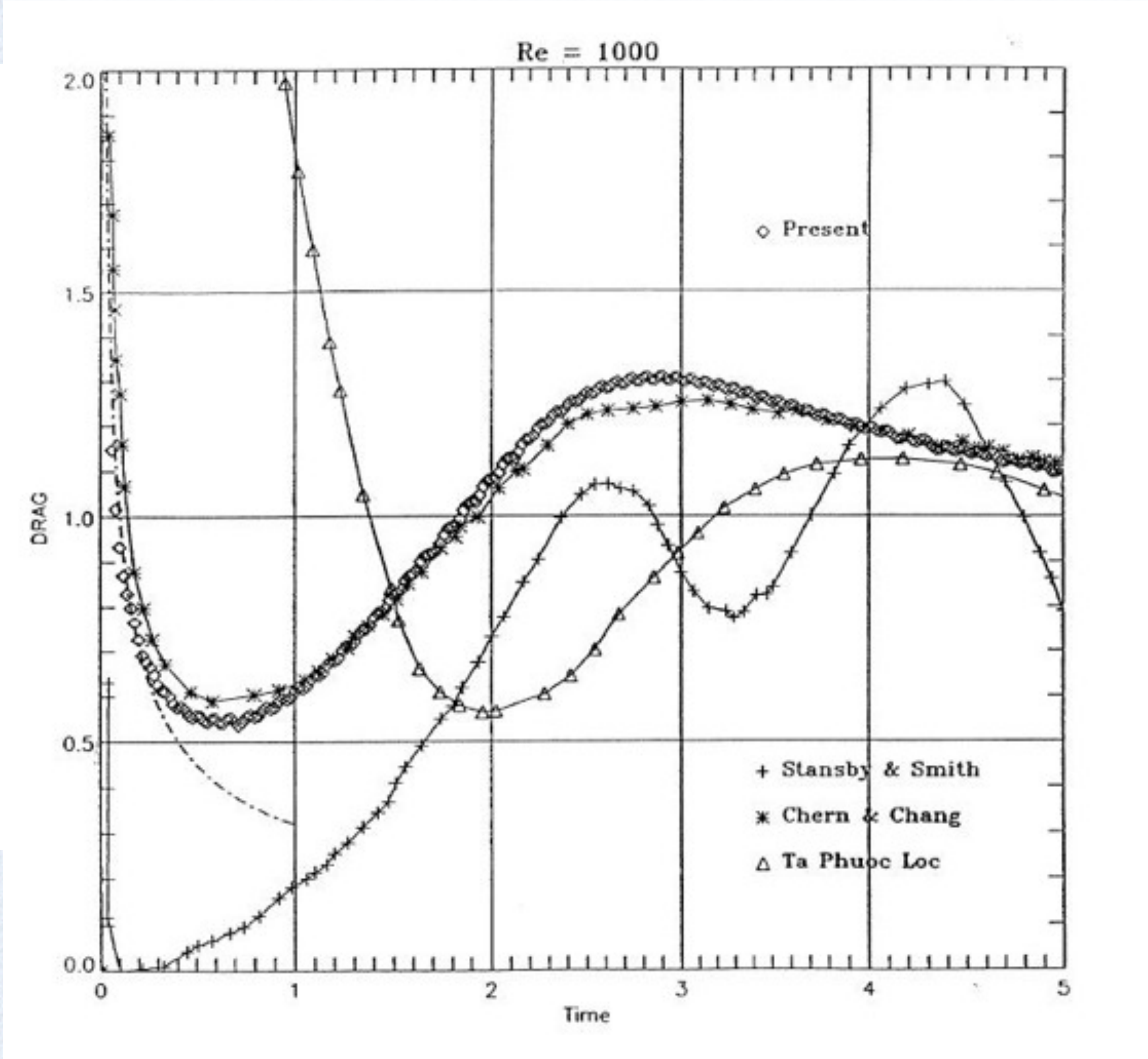
Penalization Method: $f(\mathbf{x}) = \lambda \chi_S (\mathbf{u}_S - \mathbf{u})$

Immersed Boundary Method: $f(\mathbf{x}) = \kappa \delta_S (\mathbf{x}_S - \mathbf{x})$

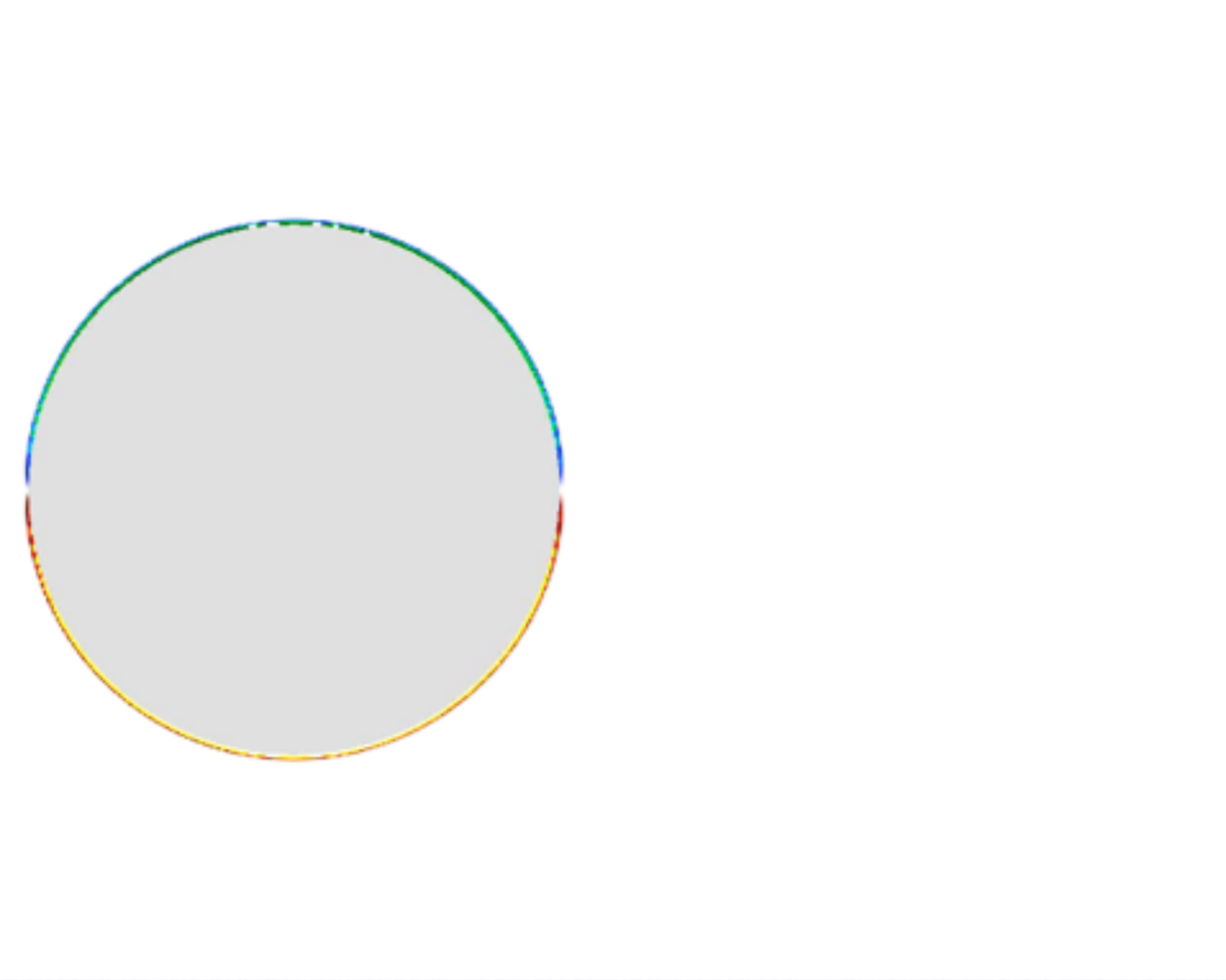
Re 9500 : Multiresolution + Multicores + (multi)GPUs

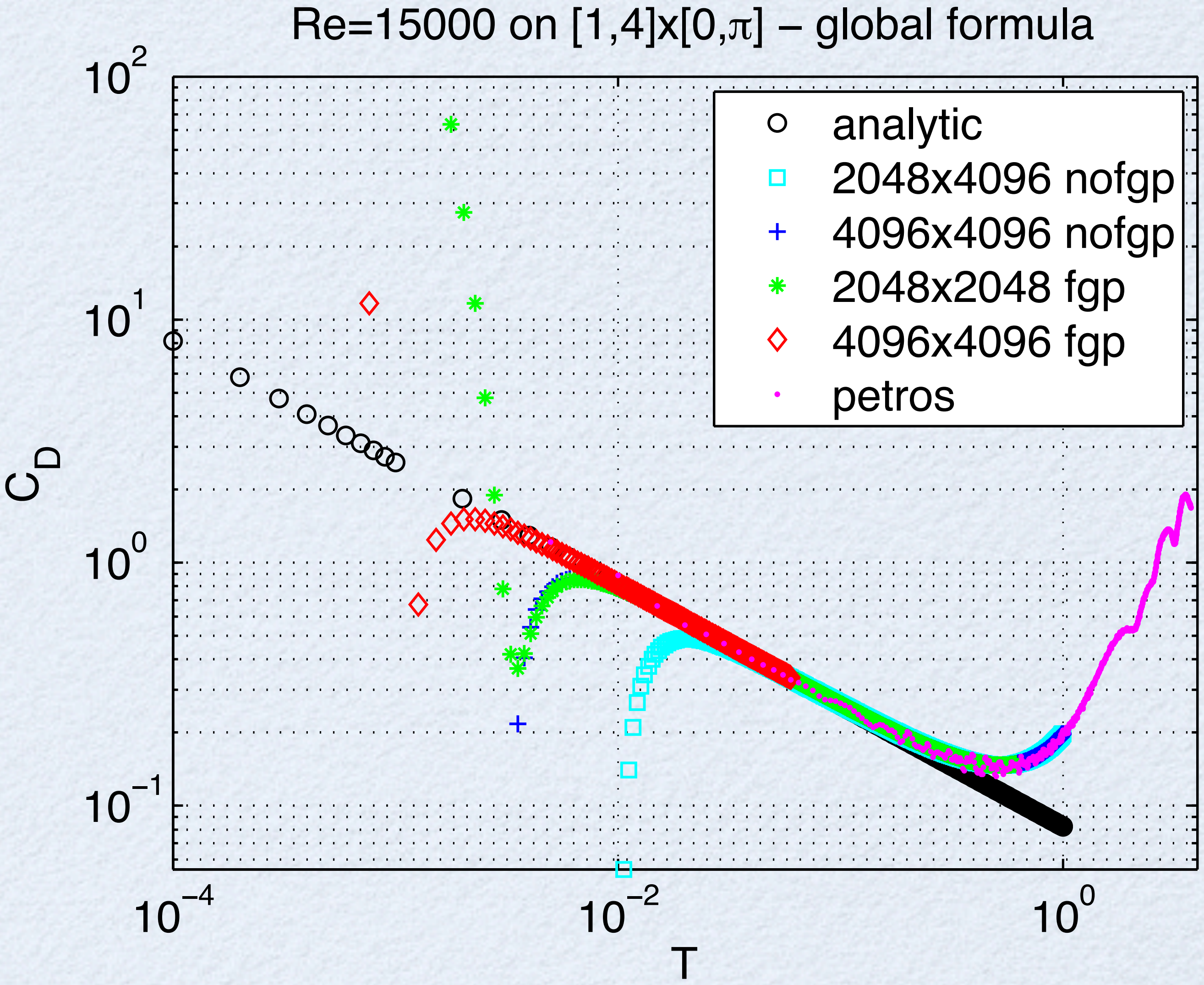


Benchmark : The Impulsively Started Cylinder

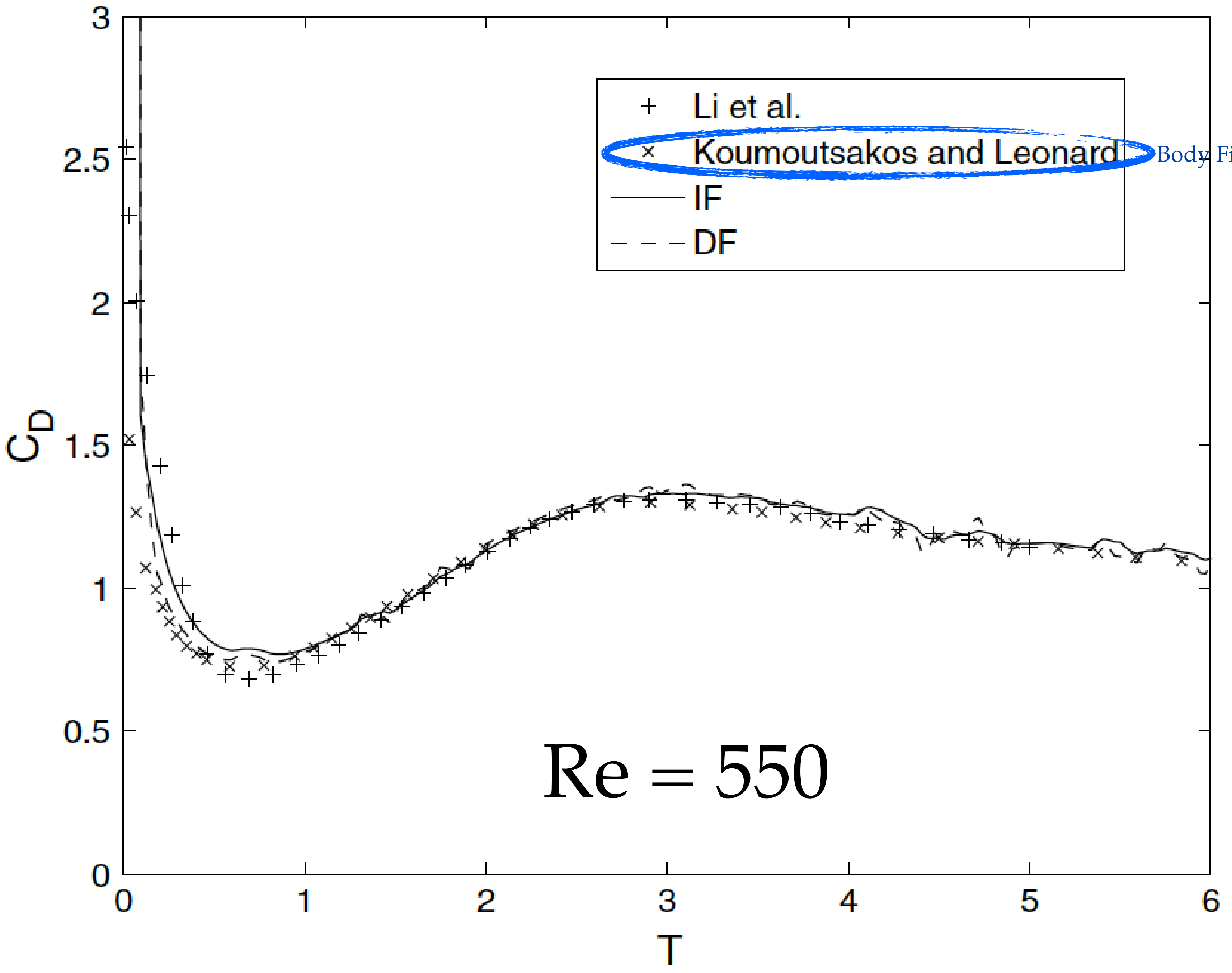


Benchmark : The Impulsively Started Cylinder

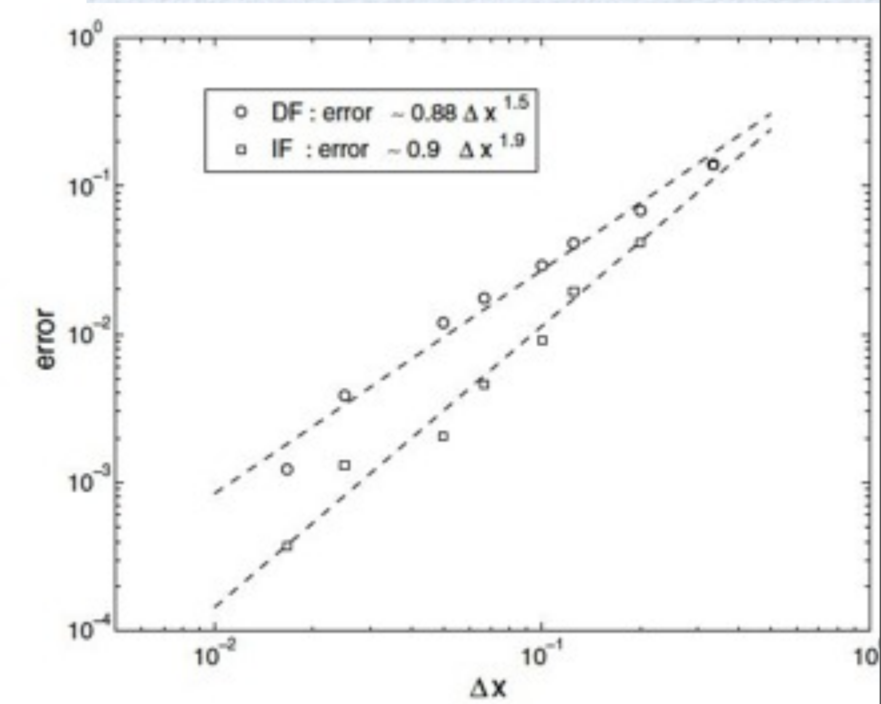




Lattice Boltzmann + Immersed Boundaries



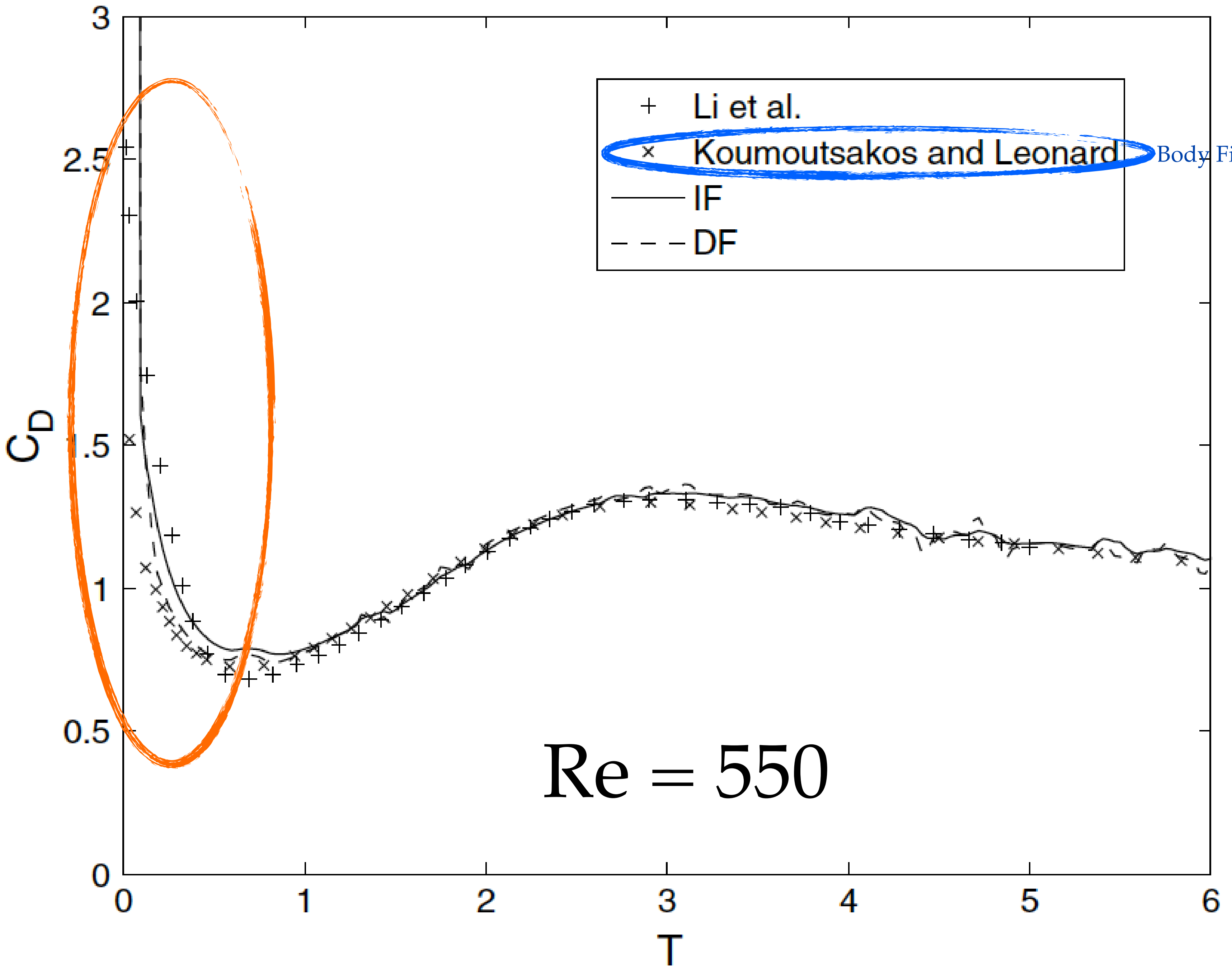
Body Fitted Grids + Vorticity Boundary Conditions

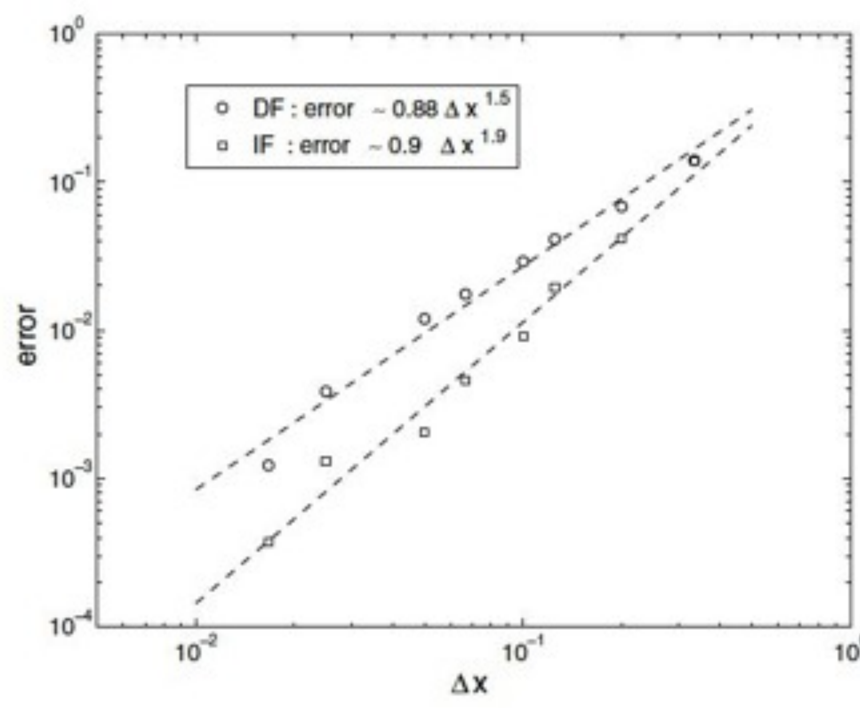


Convergence Rate

Dupuis A., Chatelain P., Koumoutsakos P., An Immersed Boundary-Lattice Boltzmann Method for the Simulation of the Flow Past an Impulsively Started Cylinder, **J. Computational Physics**, 227, 9, 4486-4498, 2008

Lattice Boltzmann + Immersed Boundaries

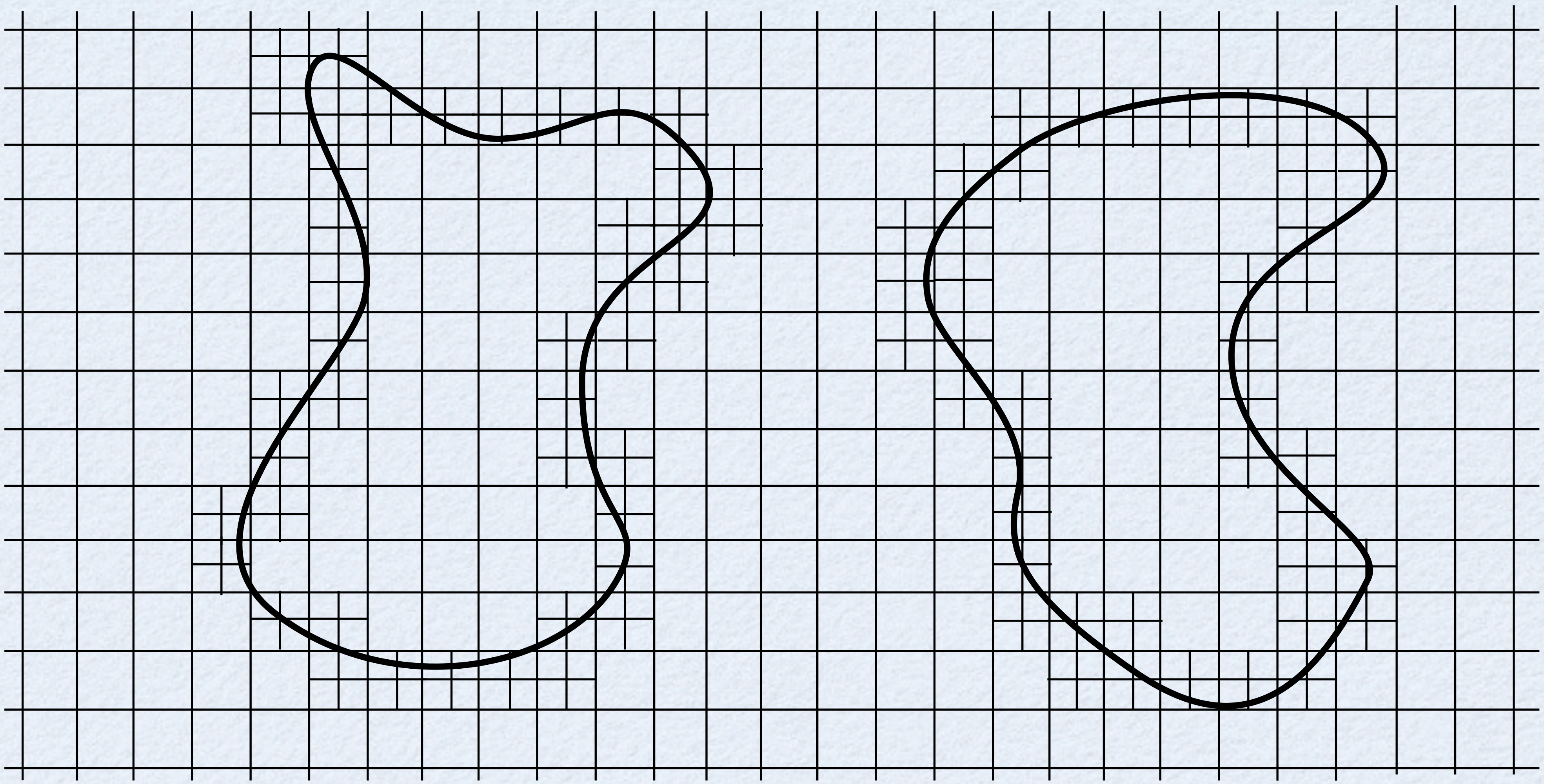




Convergence Rate

Dupuis A., Chatelain P., Koumoutsakos P., An Immersed Boundary-Lattice Boltzmann Method for the Simulation of the Flow Past an Impulsively Started Cylinder, **J. Computational Physics**, 227, 9, 4486-4498, 2008

Boundary Conditions = Coupling

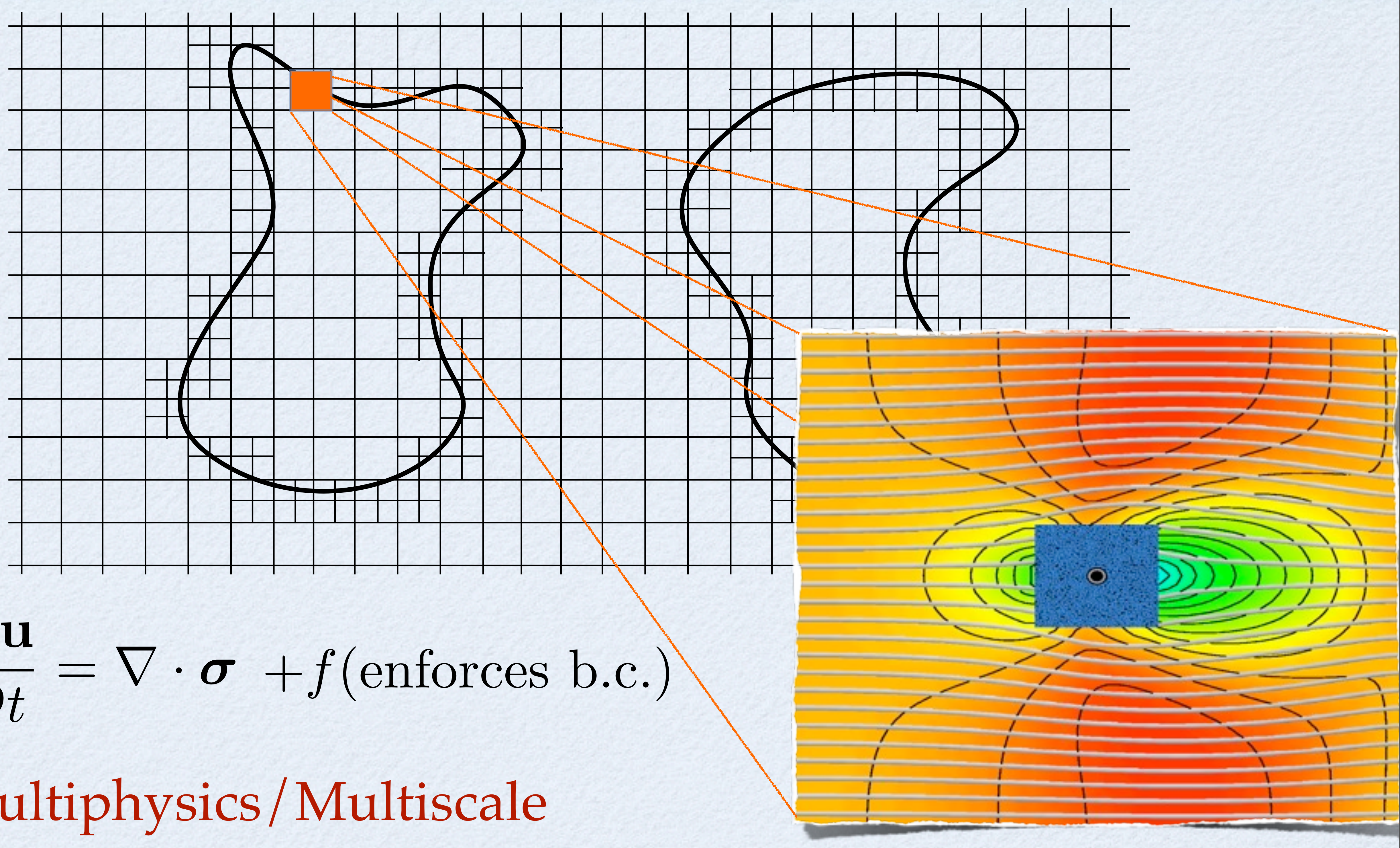


$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} \; + f(\text{enforces b.c.})$$

Multiphysics/Multiscale

$$f(\mathbf{x}) \approx F(\text{Atomistic Simulations})$$

Boundary Conditions = Coupling



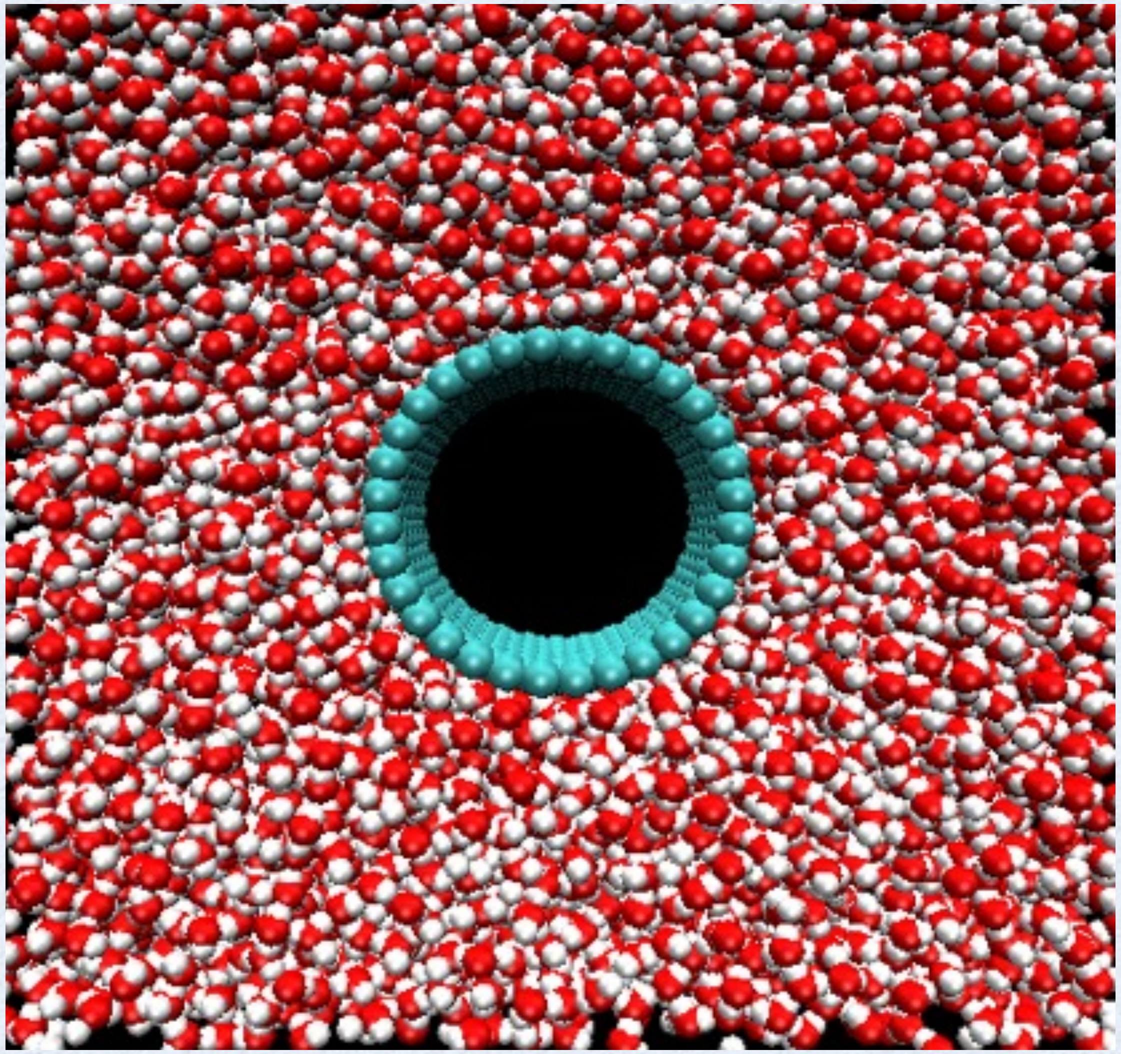
$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} \; + f(\text{enforces b.c.})$$

Multiphysics / Multiscale

$$f(\mathbf{x}) \approx F(\text{Atomistic Simulations})$$

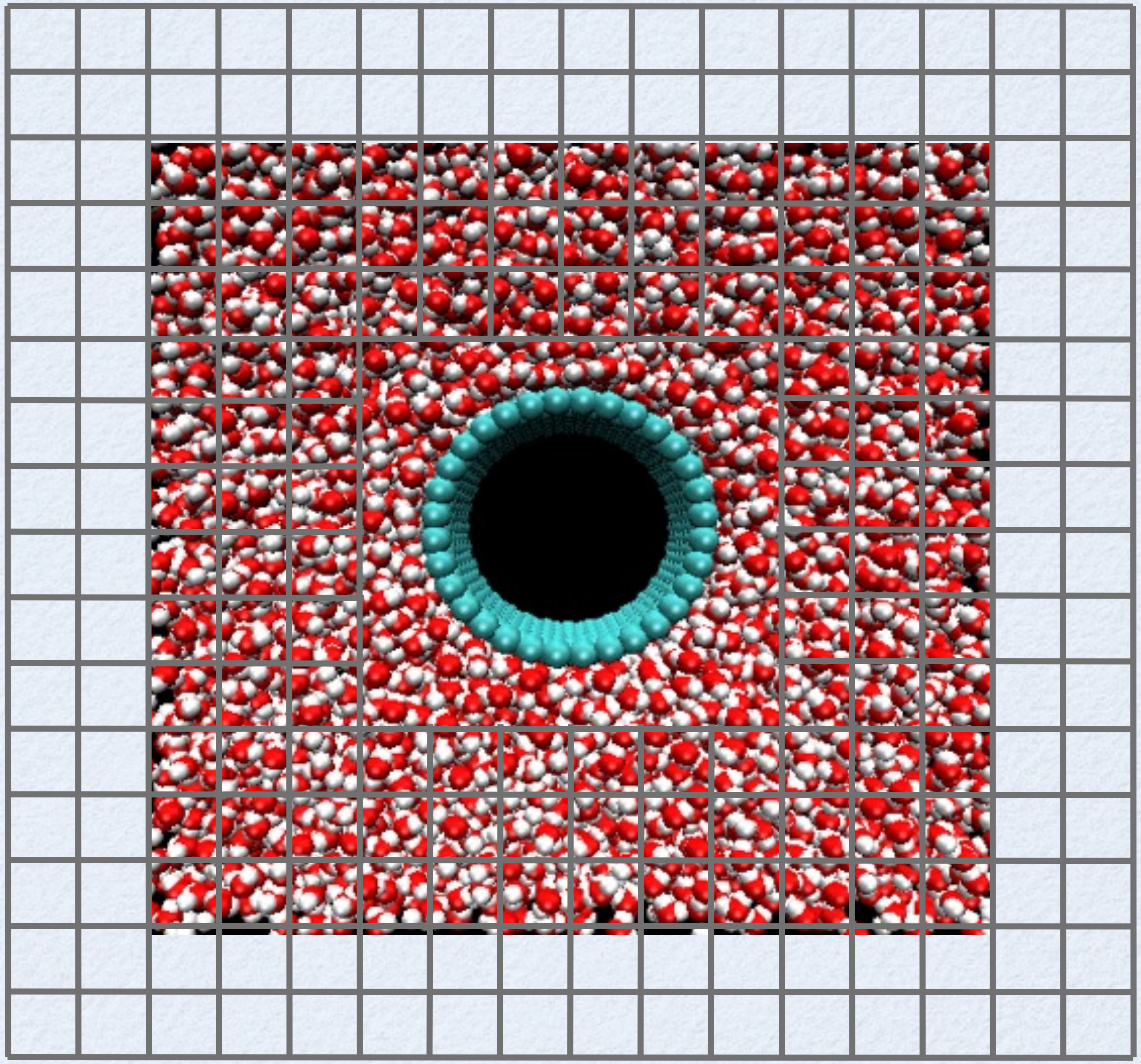
Schwarz DD for Liquids

T. Werder, J. H. Walther, P. Koumoutsakos, Hybrid atomistic-continuum method for the simulation of dense fluid flow, **J. Computational Physics**, 205, 373 - 390, 2005



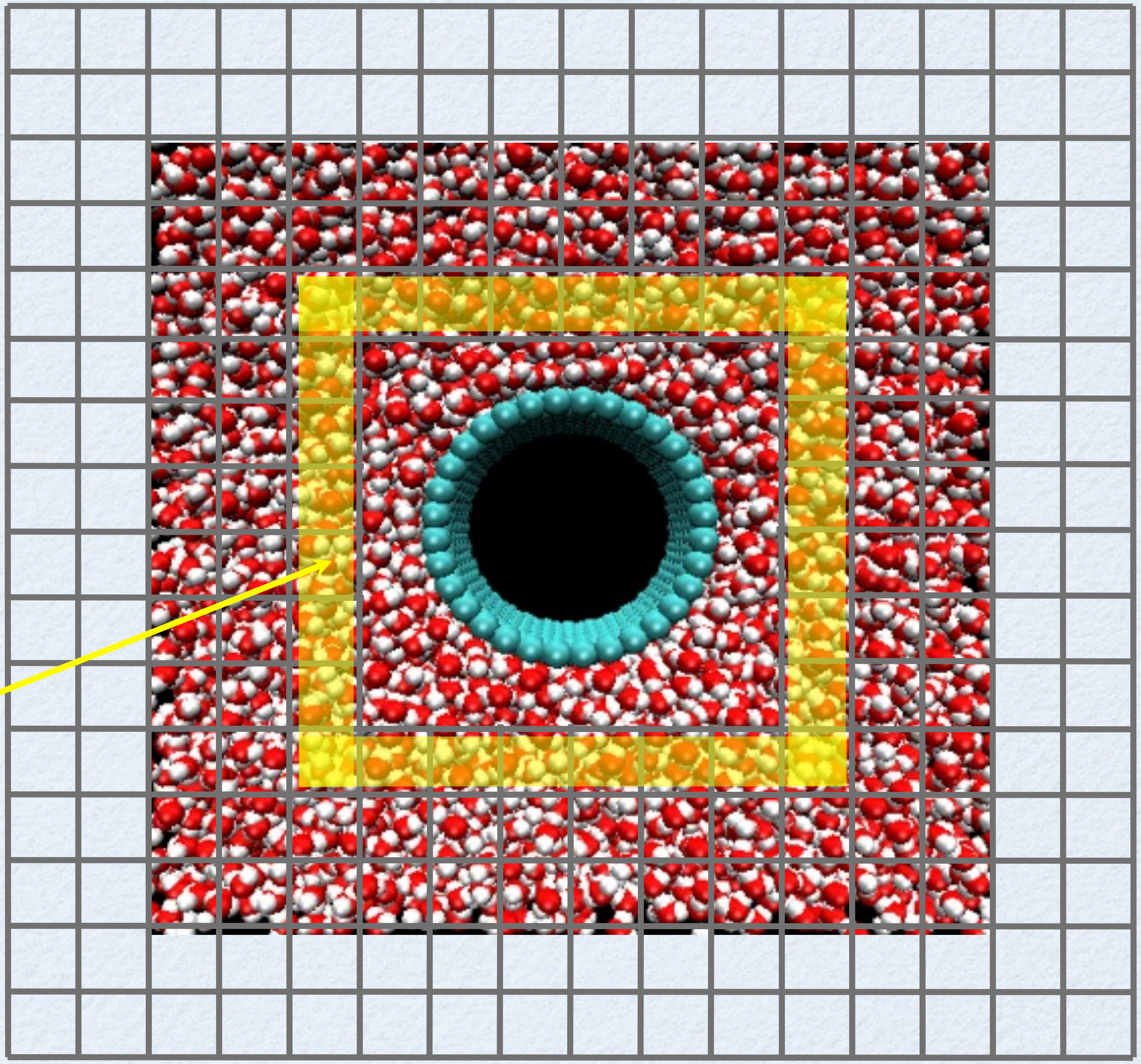
Schwarz DD for Liquids

T. Werder, J. H. Walther, P. Koumoutsakos, Hybrid atomistic-continuum method for the simulation of dense fluid flow, **J. Computational Physics**, 205, 373 - 390, 2005



Schwarz DD for Liquids

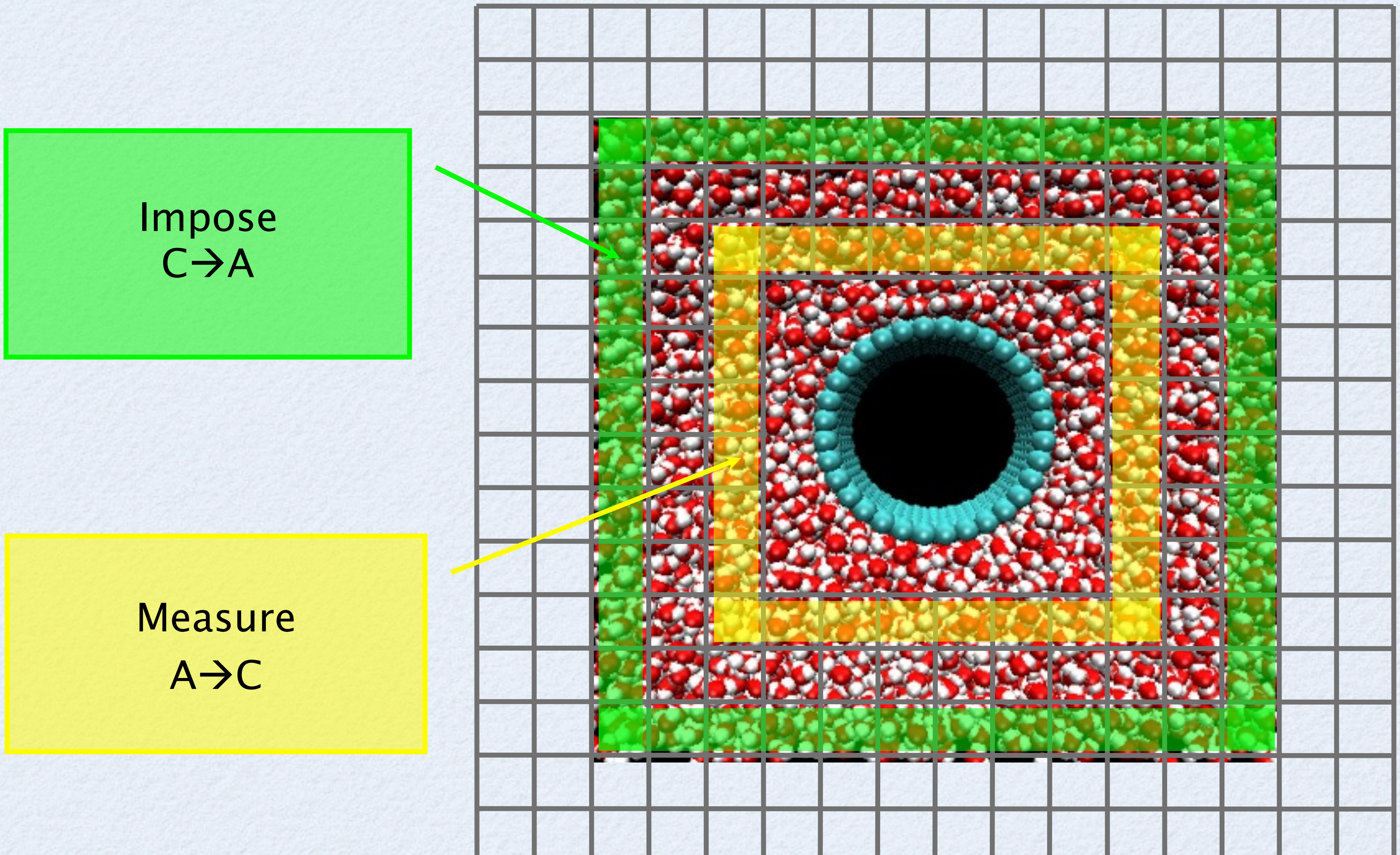
T. Werder, J. H. Walther, P. Koumoutsakos, Hybrid atomistic-continuum method for the simulation of dense fluid flow, **J. Computational Physics**, 205, 373 - 390, 2005



Measure
A→C

Schwarz DD for Liquids

T. Werder, J. H. Walther, P. Koumoutsakos, Hybrid atomistic-continuum method for the simulation of dense fluid flow, **J. Computational Physics**, 205, 373 - 390, 2005



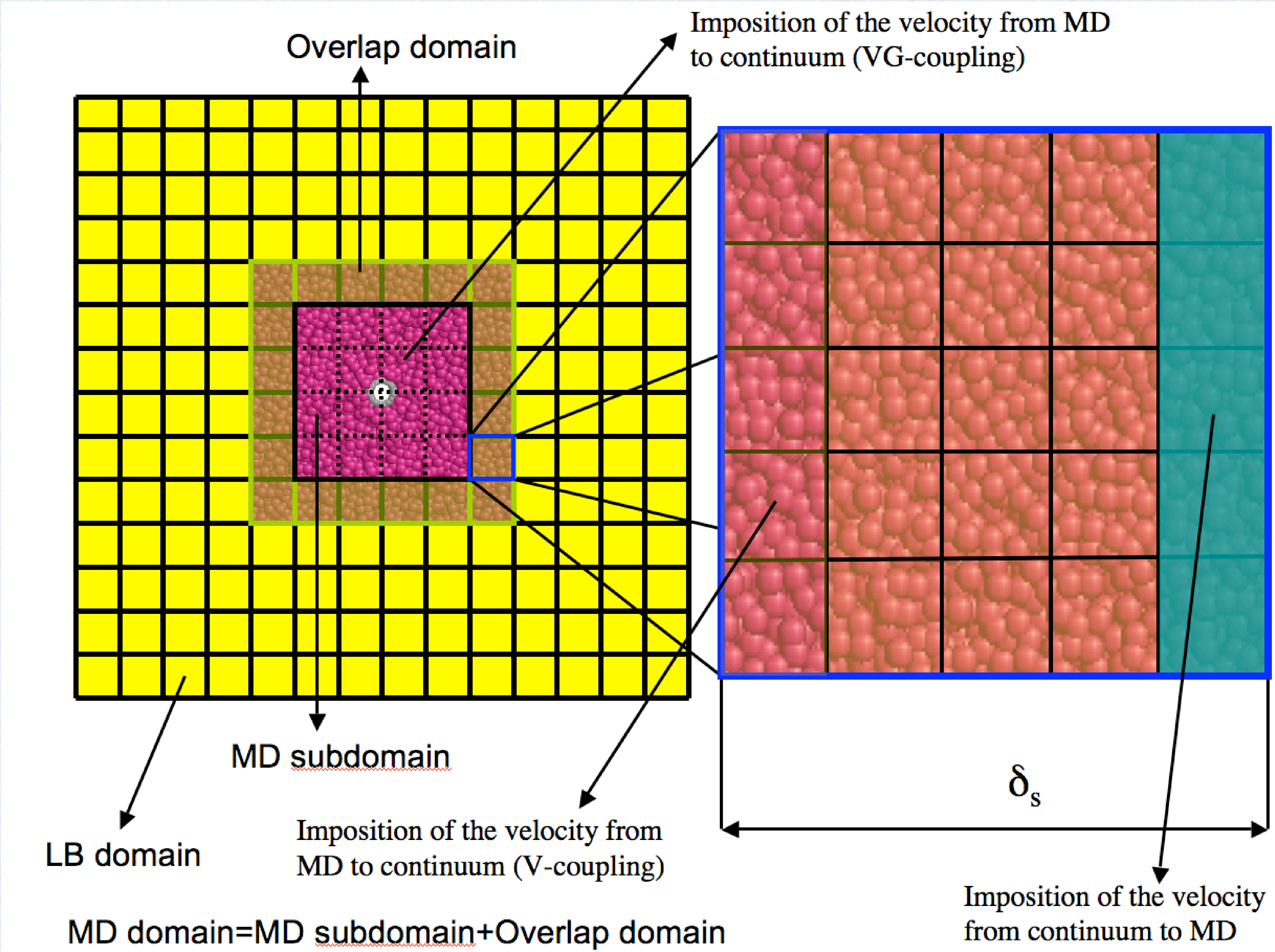
- Iterate, until the solution in the overlap region converges.
- Conservative scheme – transport coefficients in A and

Bridging FLUX & SCHWARZ DD Algorithms

Dupuis A., Kotsalis E.M, Koumoutsakos P., Coupling Lattice Boltzmann and Molecular Dynamics Models for Dense Fluids, **Physical Review E**, 75, 046704, 2007

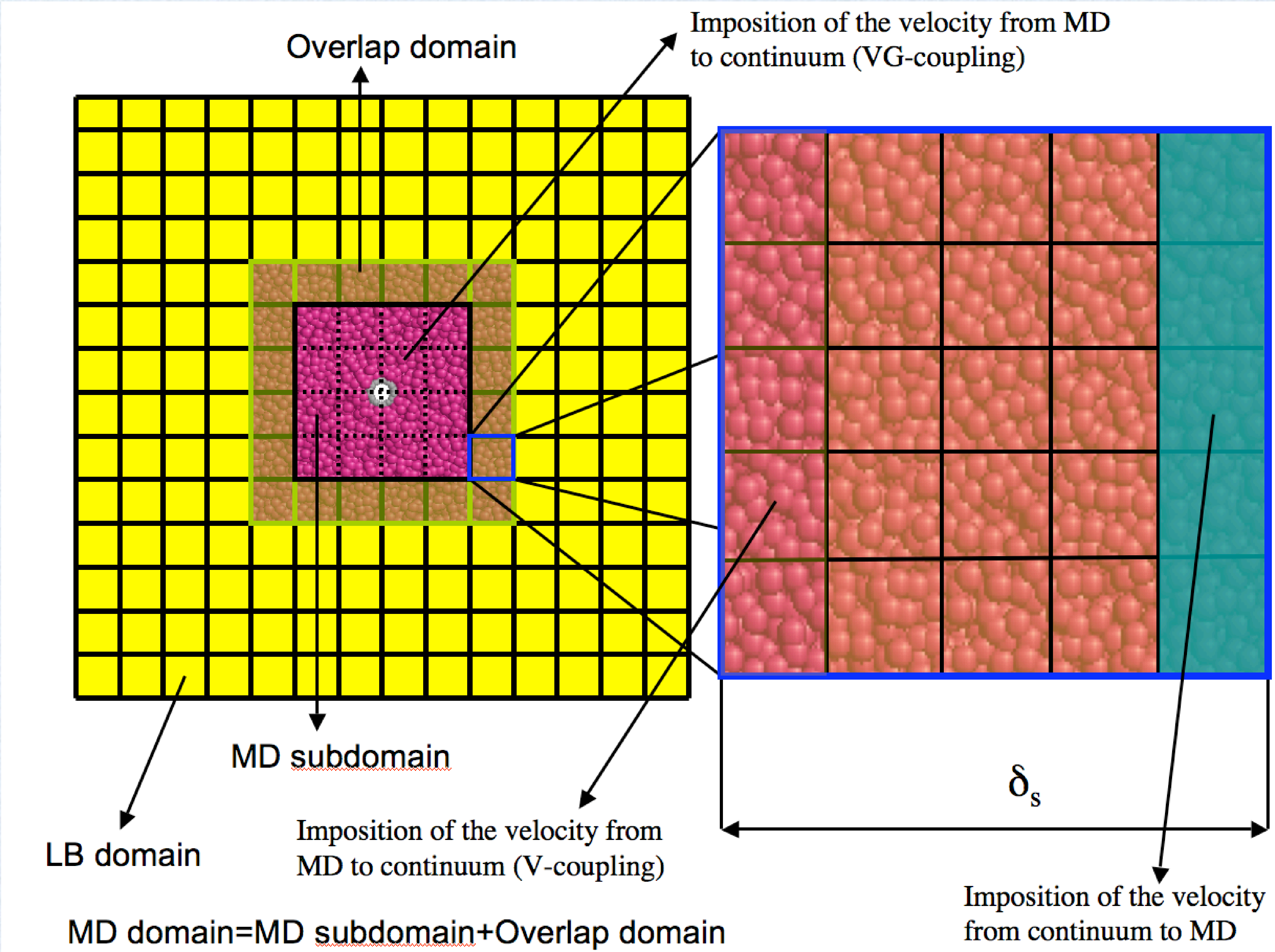
Bridging FLUX & SCHWARZ DD Algorithms

Dupuis A., Kotsalis E.M, Koumoutsakos P., Coupling Lattice Boltzmann and Molecular Dynamics Models for Dense Fluids, **Physical Review E**, 75, 046704, 2007



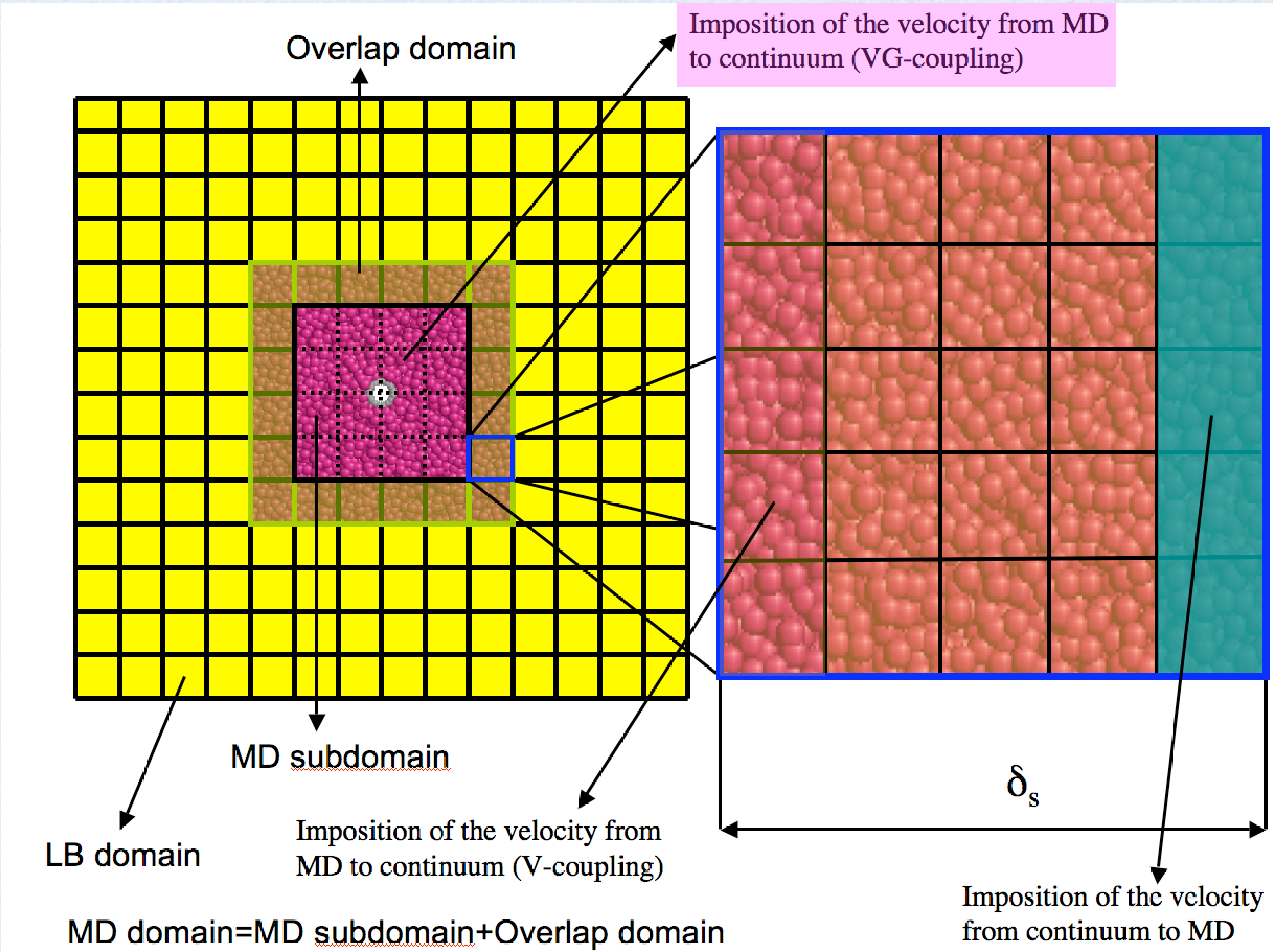
Bridging FLUX & SCHWARZ DD Algorithms

Dupuis A., Kotsalis E.M, Koumoutsakos P., Coupling Lattice Boltzmann and Molecular Dynamics Models for Dense Fluids, **Physical Review E**, 75, 046704, 2007



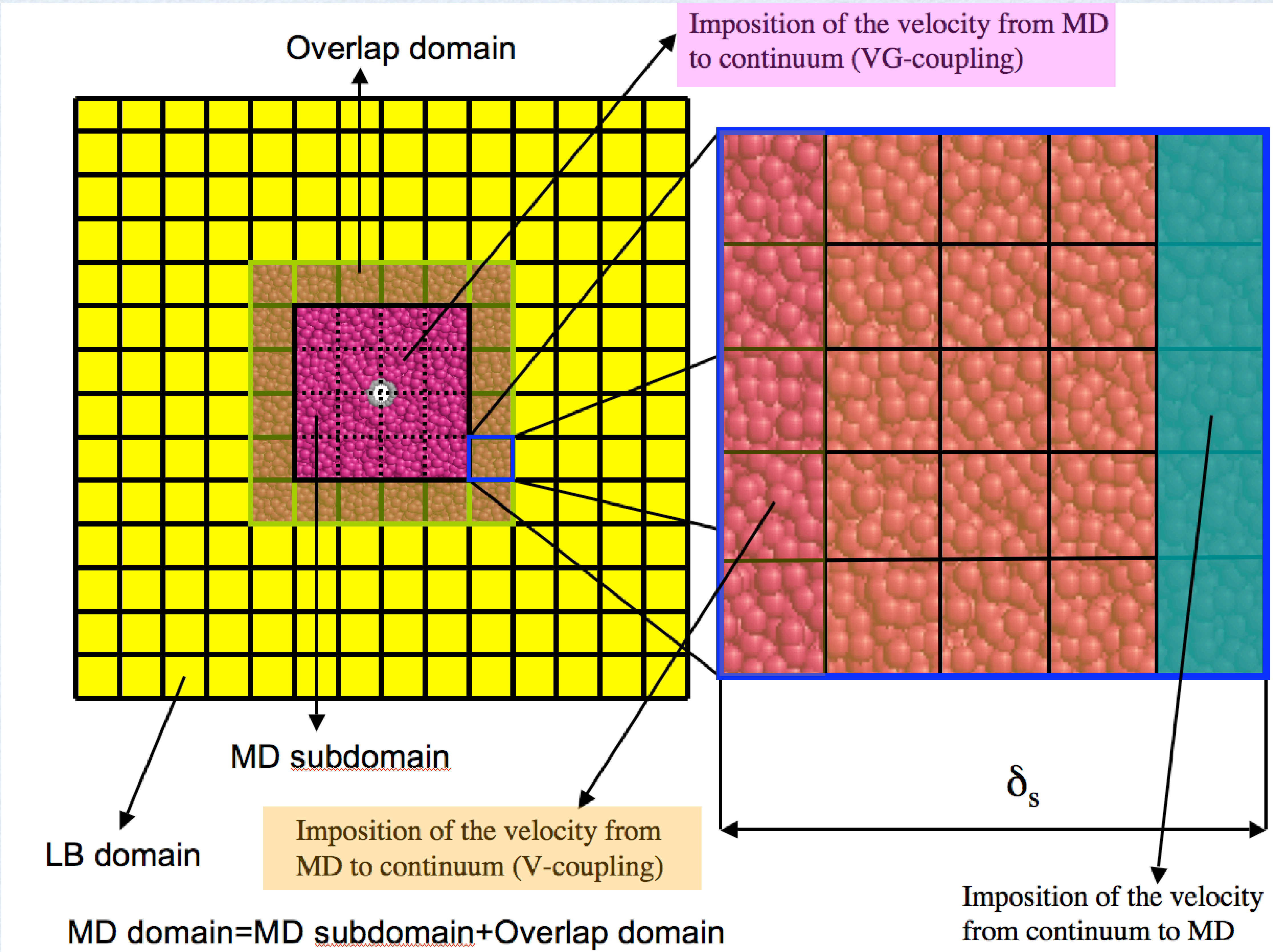
Bridging FLUX & SCHWARZ DD Algorithms

Dupuis A., Kotsalis E.M, Koumoutsakos P., Coupling Lattice Boltzmann and Molecular Dynamics Models for Dense Fluids, **Physical Review E**, 75, 046704, 2007

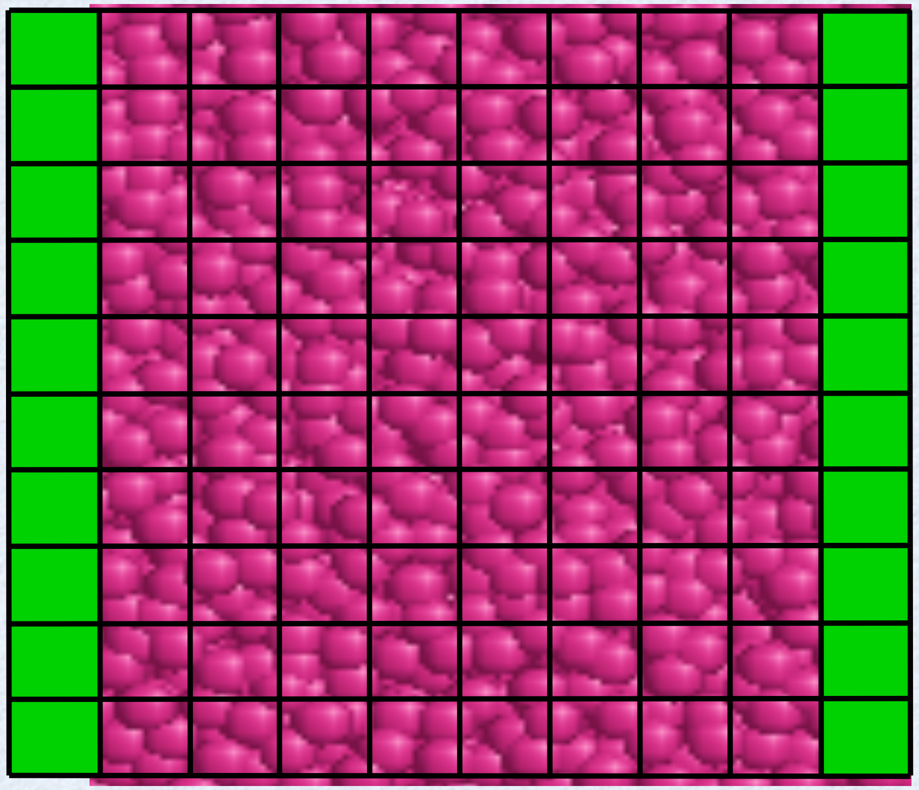


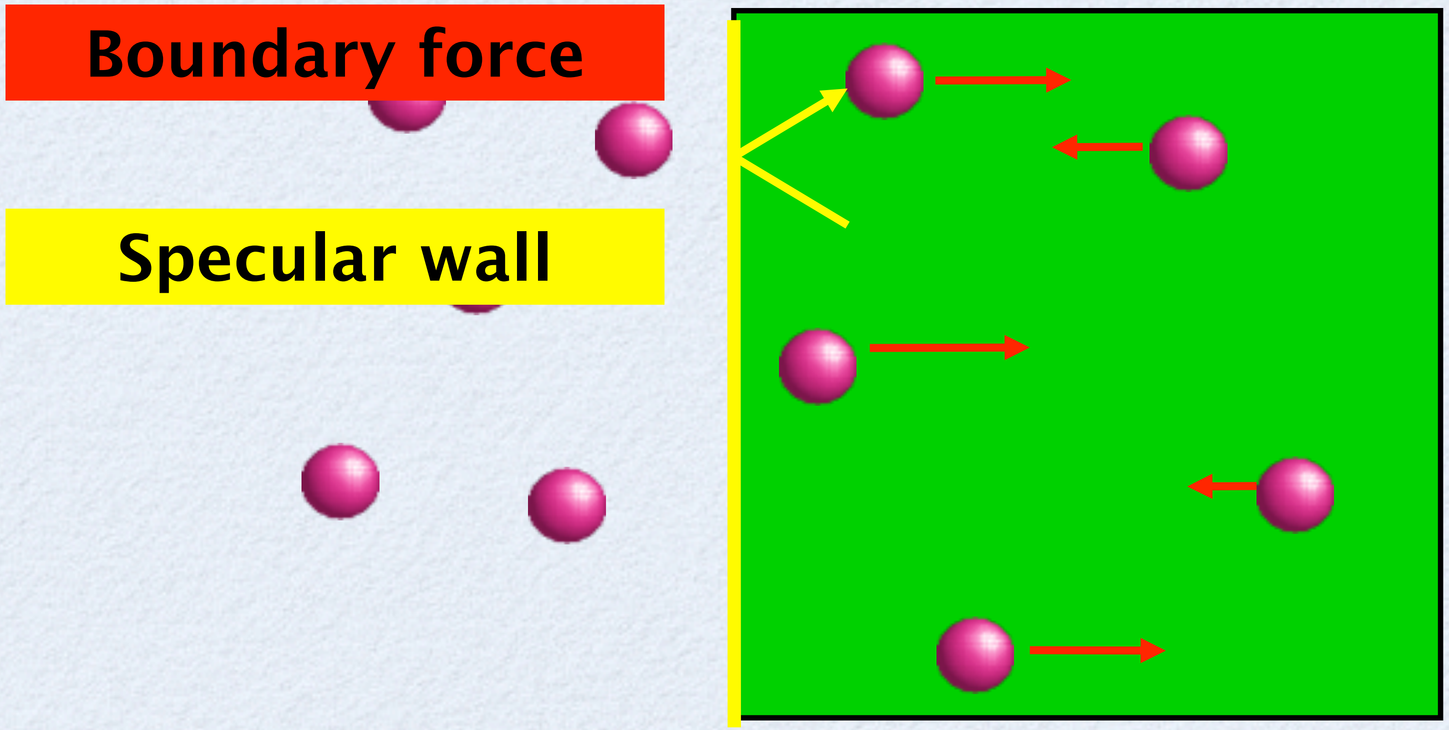
Bridging FLUX & SCHWARZ DD Algorithms

Dupuis A., Kotsalis E.M, Koumoutsakos P., Coupling Lattice Boltzmann and Molecular Dynamics Models for Dense Fluids, **Physical Review E**, 75, 046704, 2007



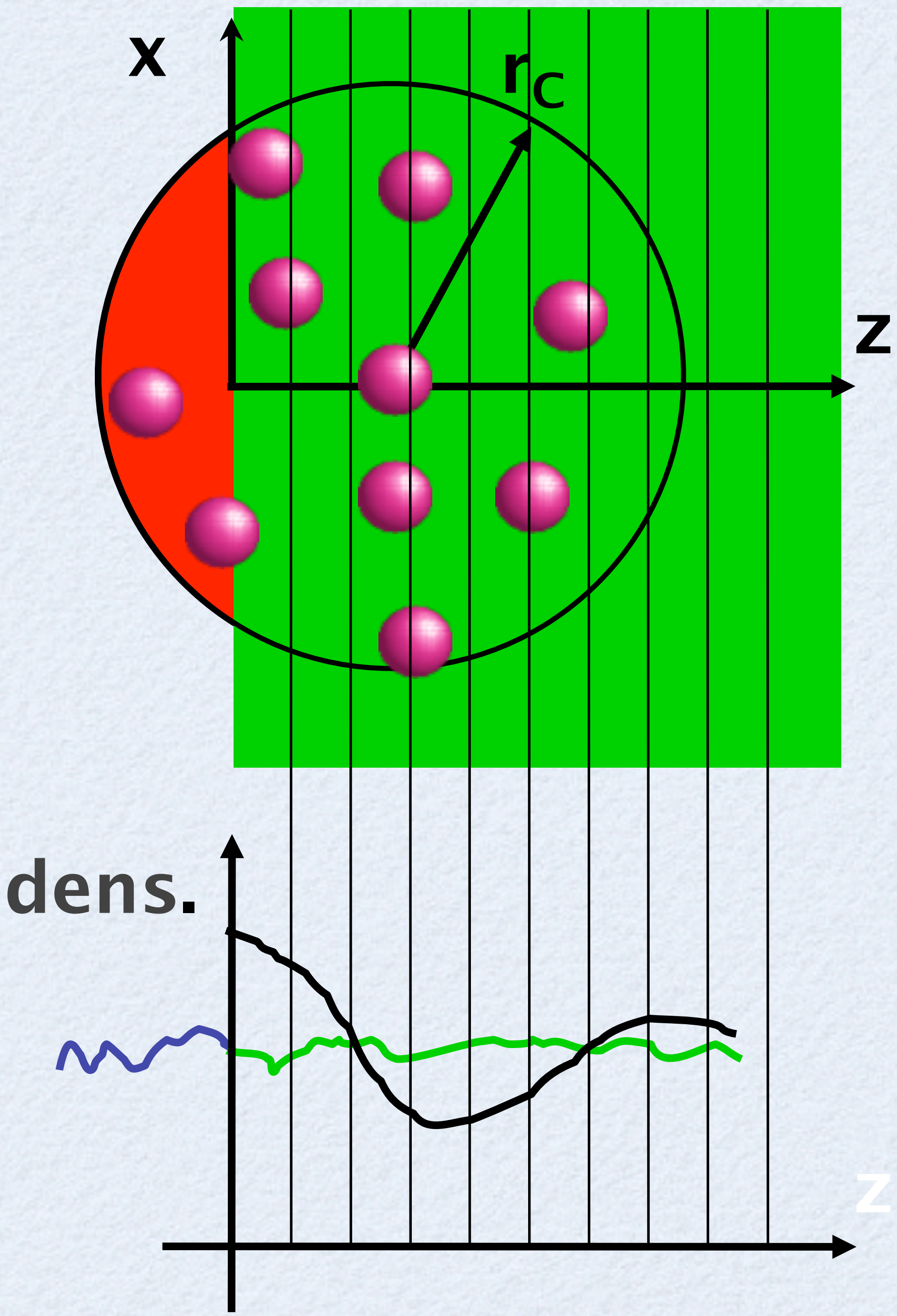
TEST 1 : EQUILIBRIUM   Non-Periodic MD





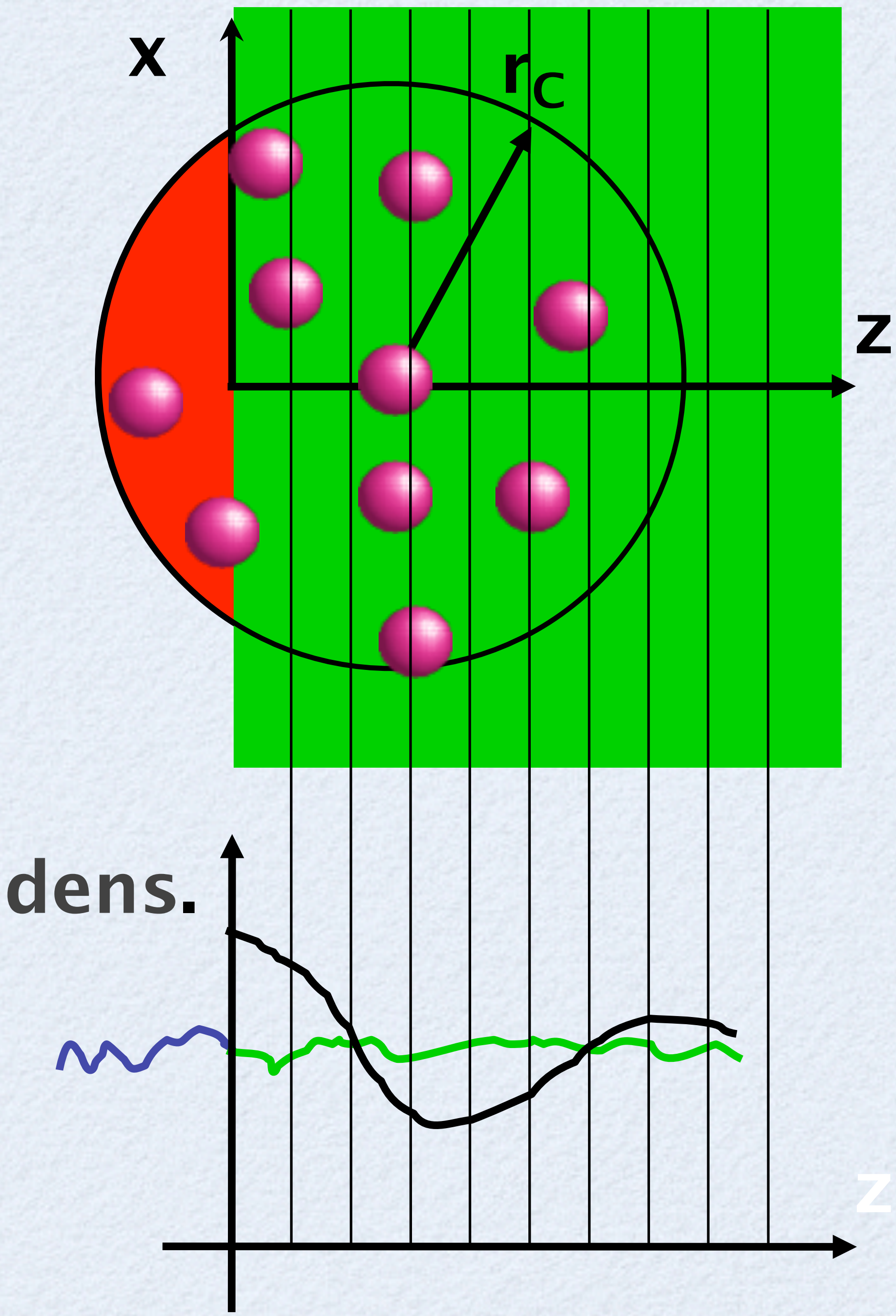
NON_PERIODICITY & BOUNDARY FORCE

How can we account for the particles in the red domain?



NON_PERIODICITY & BOUNDARY FORCE

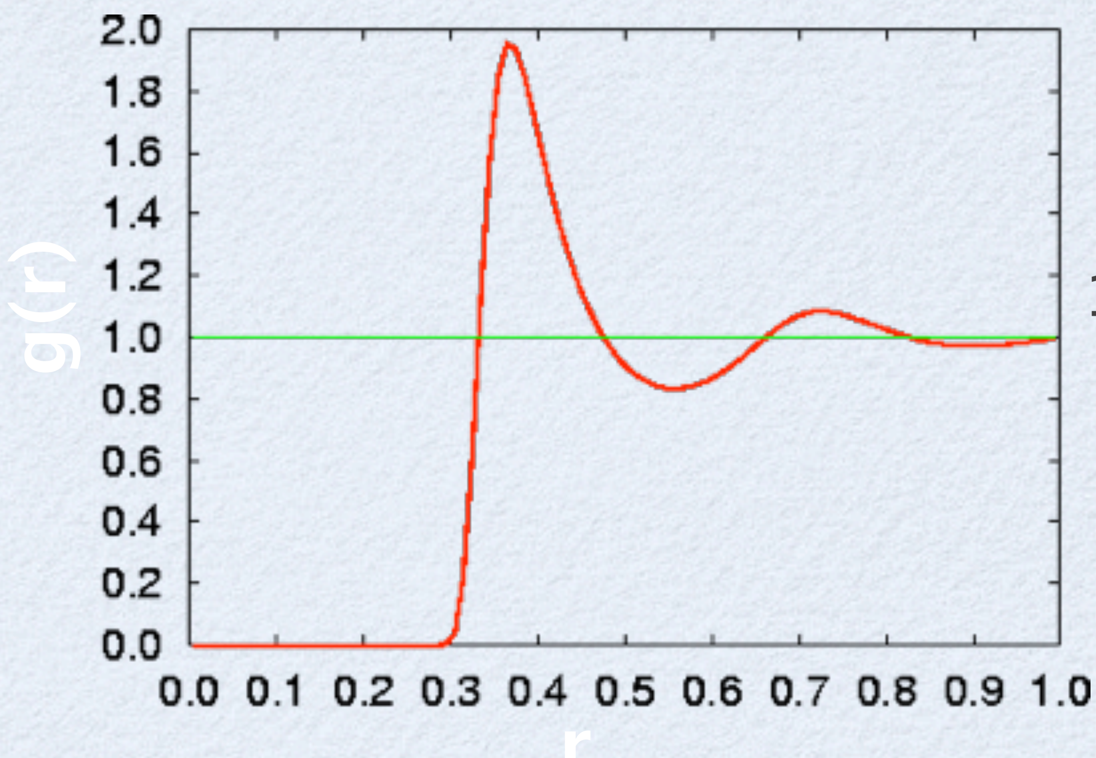
How can we account for the particles in the red domain?



Take fluid structure into

$$\rho(r) = \int_0^r 4\pi r'^2 \rho g(r') dr'$$

$$F_m(z) = -2\pi\rho \int_{\text{red}} g(r) \frac{\partial U(r)}{\partial r} \frac{z}{r} x dx dz$$

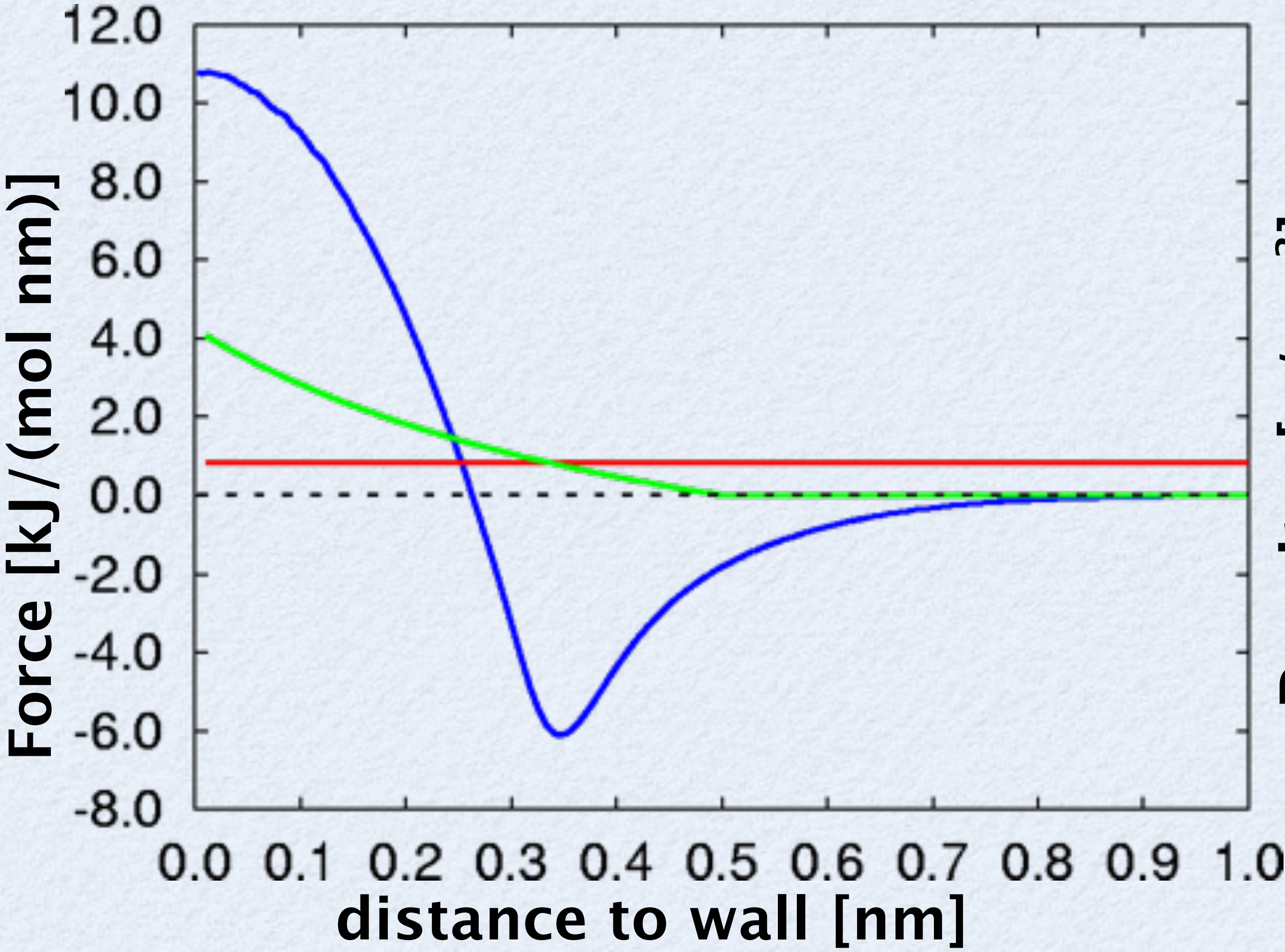


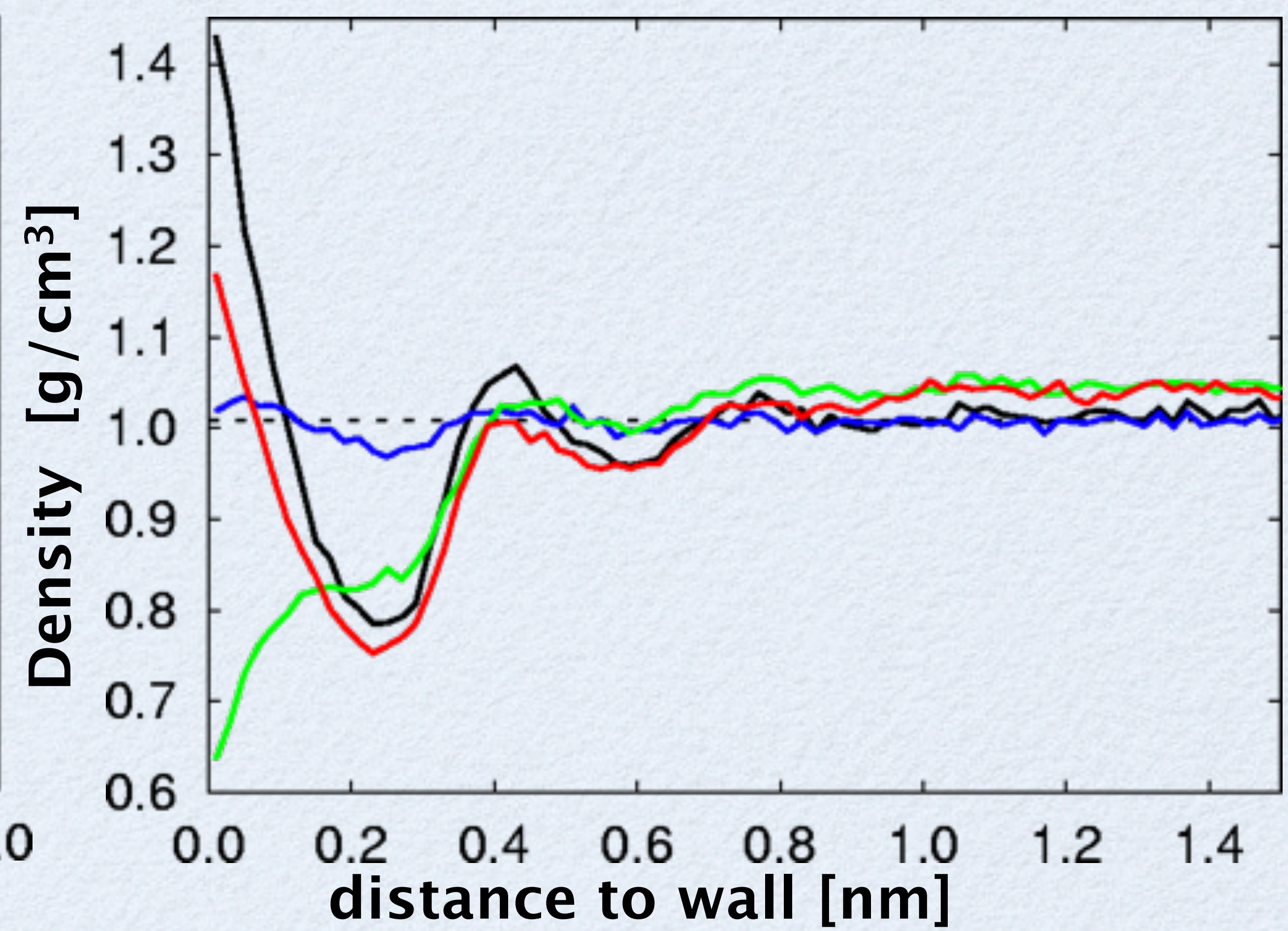
215 K
1.0 gcm^{-3}

A comparison of Forces

No force

Uniform distribution (O'Connell 1995[A])





Repulsive (Nie et al. 2004)

Use fluid structure (Werder et al. 2005)

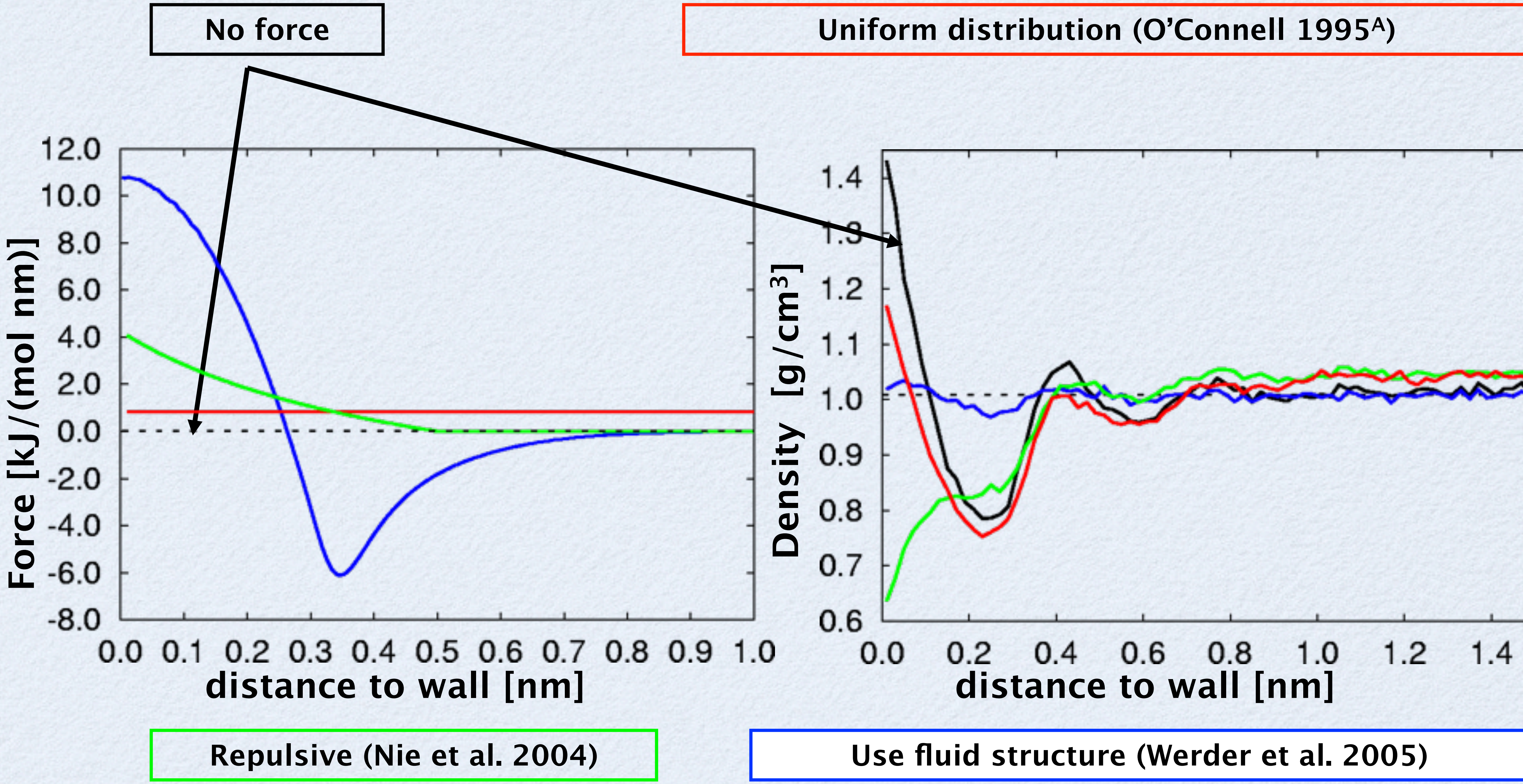
S. T. O'Connell and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flow. Phys. Rev. E, 52(6):R5792-R5795, 1995

X. B. Nie, S. Y. Chen, W. N. E and M. O. Robbins. A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow. J. Fluid Mech., 500:55-64, 2004

.

T. Werder, J. H. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flow. J. Comput. Phys., 205: 373-390, 2005.

A comparison of Forces



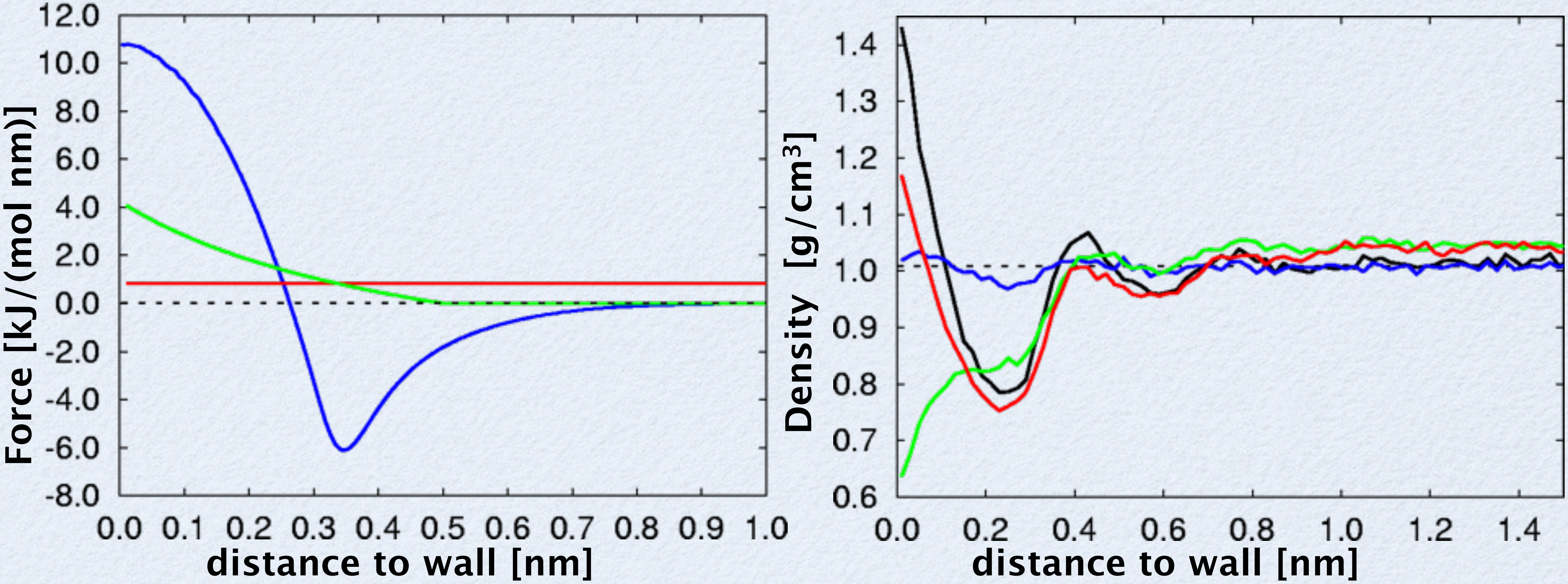
S. T. O'Connell and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flow. Phys. Rev. E, 52(6):R5792-R5795, 1995

X. B. Nie, S. Y. Chen, W. N. E and M. O. Robbins. A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow. J. Fluid Mech., 500:55-64, 2004
.
T. Werder, J. H. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flow. J. Comput. Phys., 205: 373-390, 2005.

A comparison of Forces

No force

Uniform distribution (O'Connell 1995[A])



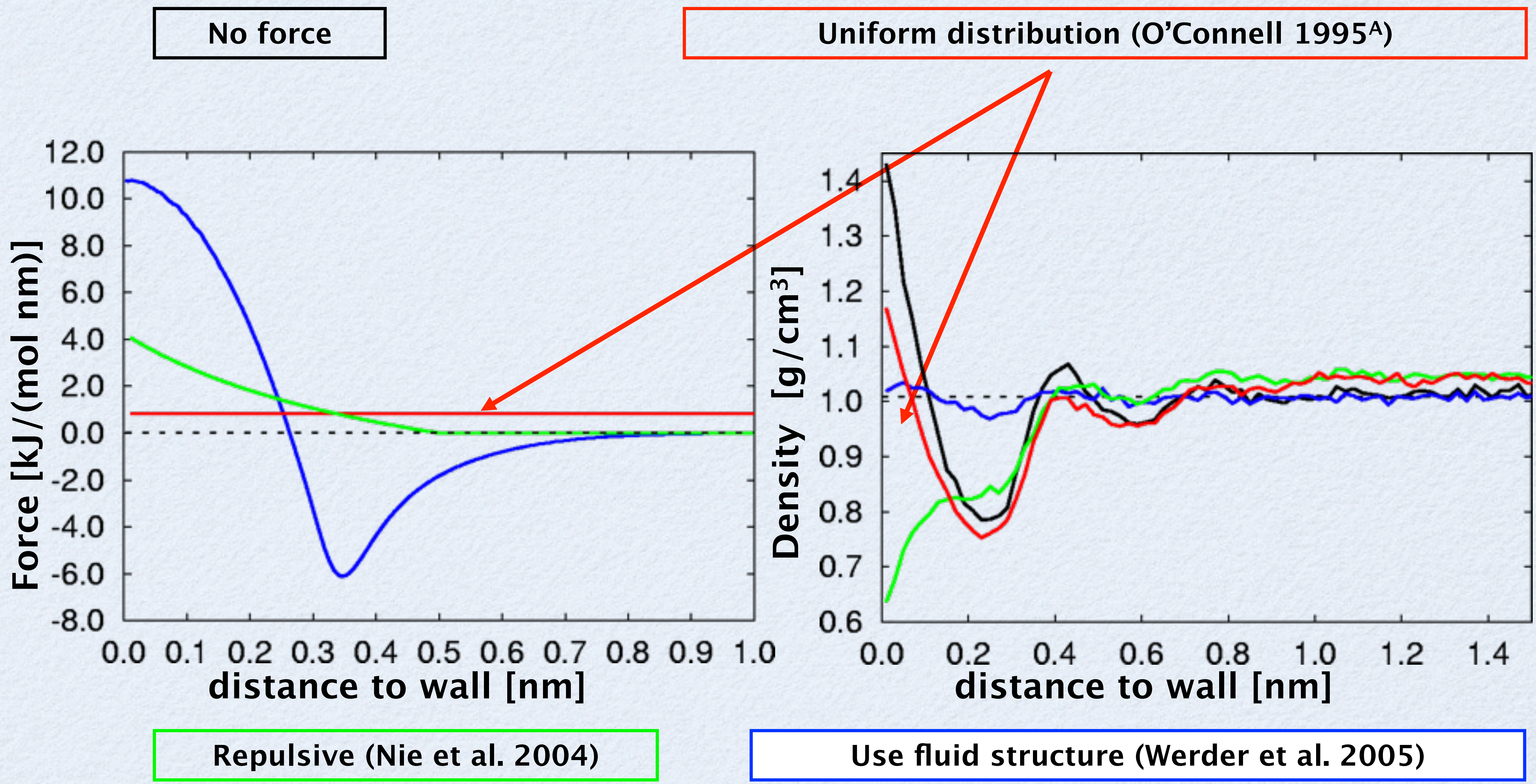
Repulsive (Nie et al. 2004)

Use fluid structure (Werder et al. 2005)

S. T. O'Connell and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flow. Phys. Rev. E, 52(6):R5792-R5795, 1995

X. B. Nie, S. Y. Chen, W. N. E and M. O. Robbins. A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow. J. Fluid Mech., 500:55-64, 2004
.
T. Werder, J. H. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flow. J. Comput. Phys., 205: 373-390, 2005.

A comparison of Forces



S. T. O'Connell and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flow. Phys. Rev. E, 52(6):R5792-R5795, 1995

X. B. Nie, S. Y. Chen, W. N. E and M. O. Robbins. A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow. J. Fluid Mech., 500:55-64, 2004

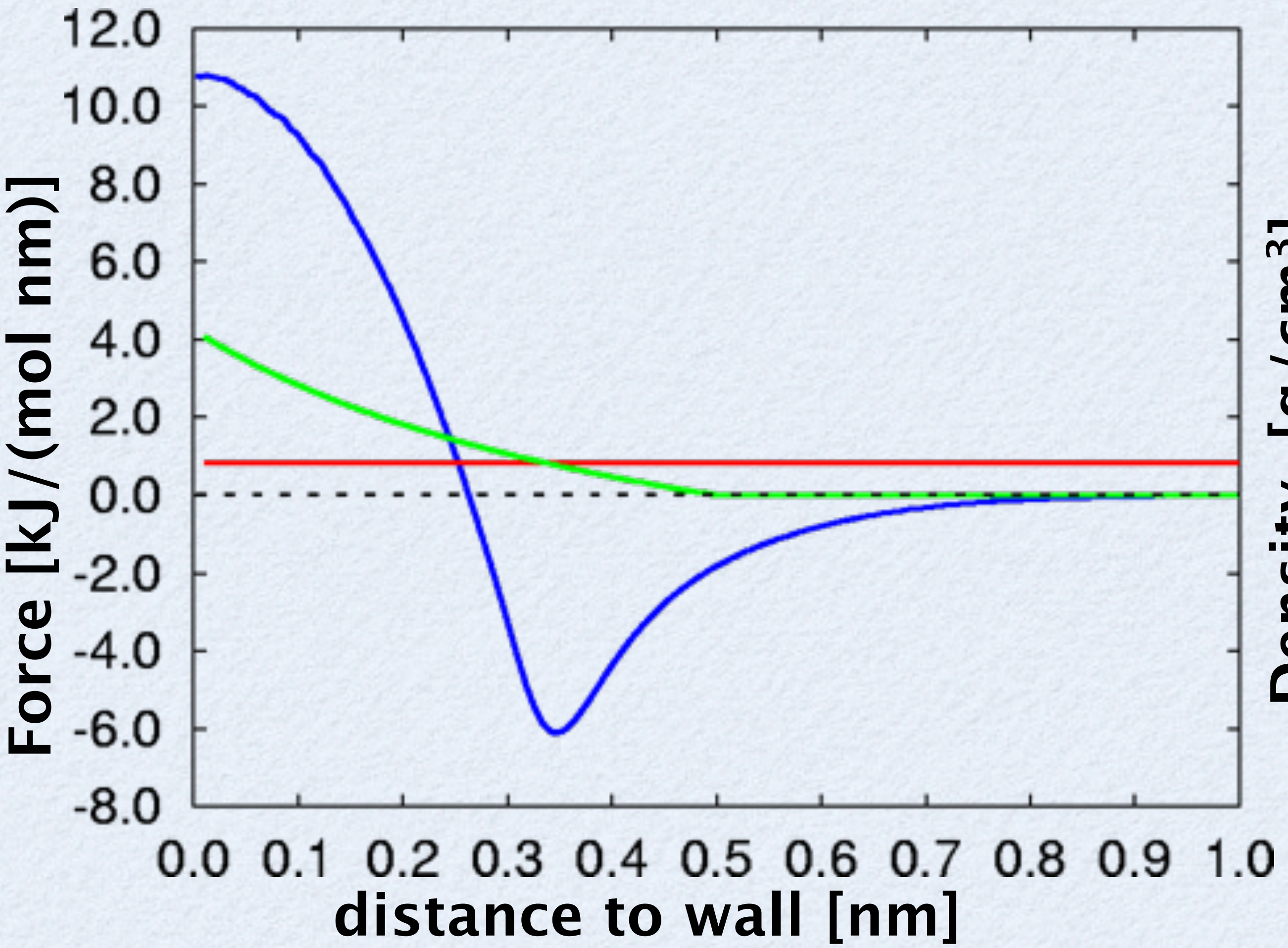
.

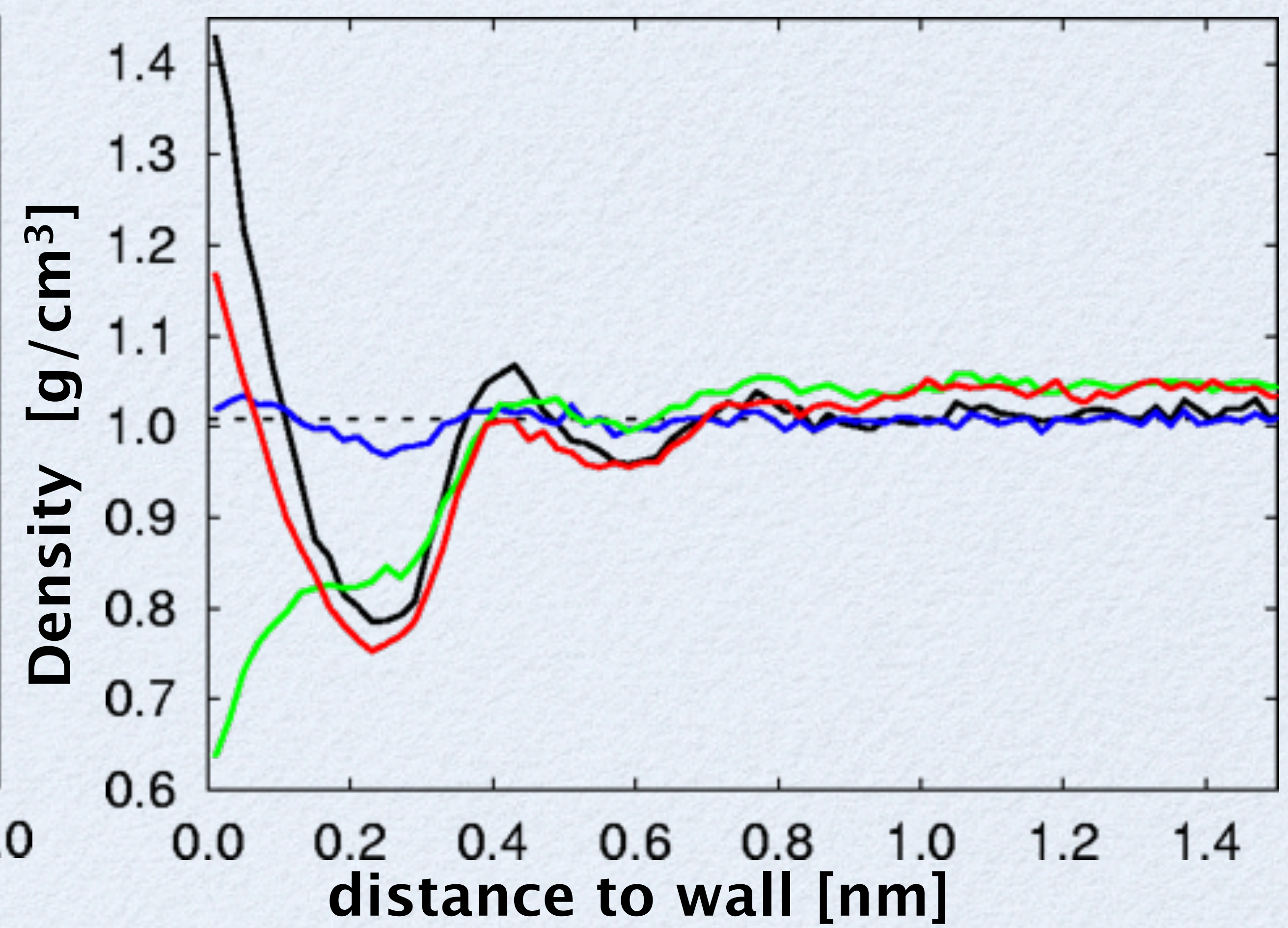
T. Werder, J. H. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flow. J. Comput. Phys., 205: 373-390, 2005.

A comparison of Forces

No force

Uniform distribution (O'Connell 1995[A])





Repulsive (Nie et al. 2004)

Use fluid structure (Werder et al. 2005)

S. T. O'Connell and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flow. Phys. Rev. E, 52(6):R5792-R5795, 1995

X. B. Nie, S. Y. Chen, W. N. E and M. O. Robbins. A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow. J. Fluid Mech., 500:55-64, 2004

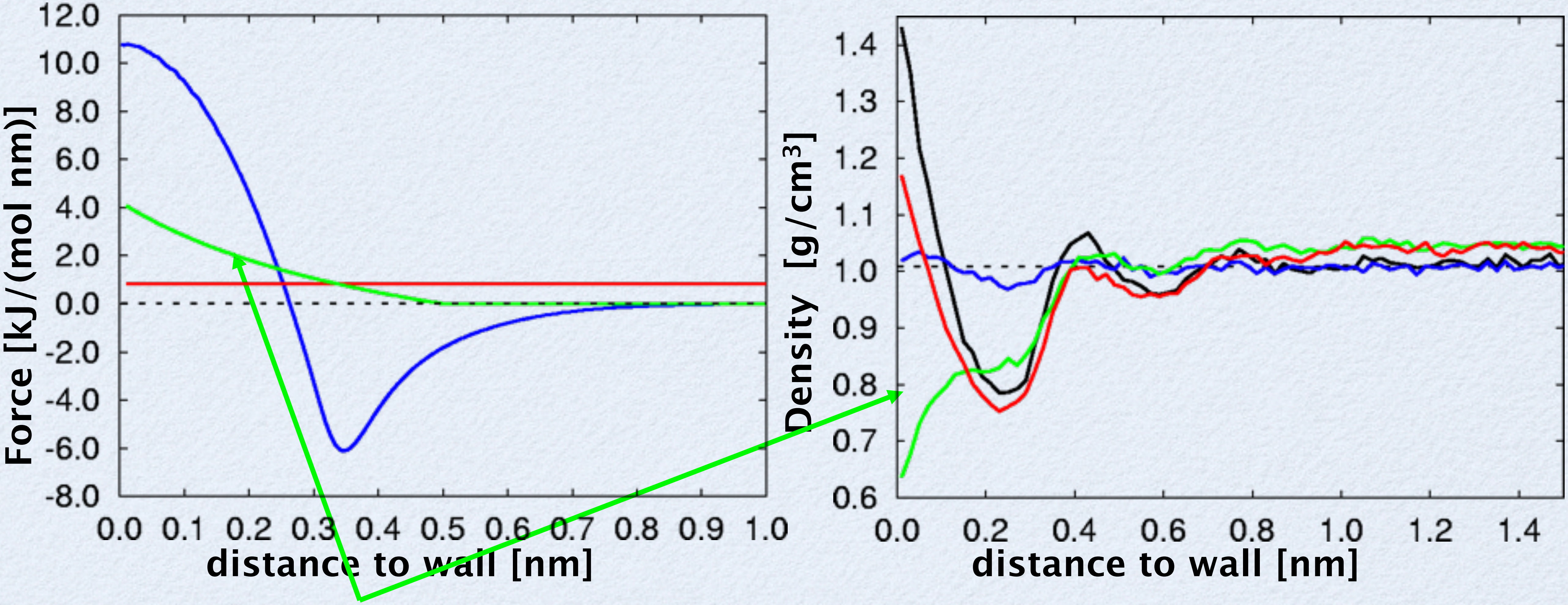
.

T. Werder, J. H. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flow. J. Comput. Phys., 205: 373-390, 2005.

A comparison of Forces

No force

Uniform distribution (O'Connell 1995[A])



Repulsive (Nie et al. 2004)

Use fluid structure (Werder et al. 2005)

S. T. O'Connell and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flow. Phys. Rev. E, 52(6):R5792-R5795, 1995

X. B. Nie, S. Y. Chen, W. N. E and M. O. Robbins. A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow. J. Fluid Mech., 500:55-64, 2004

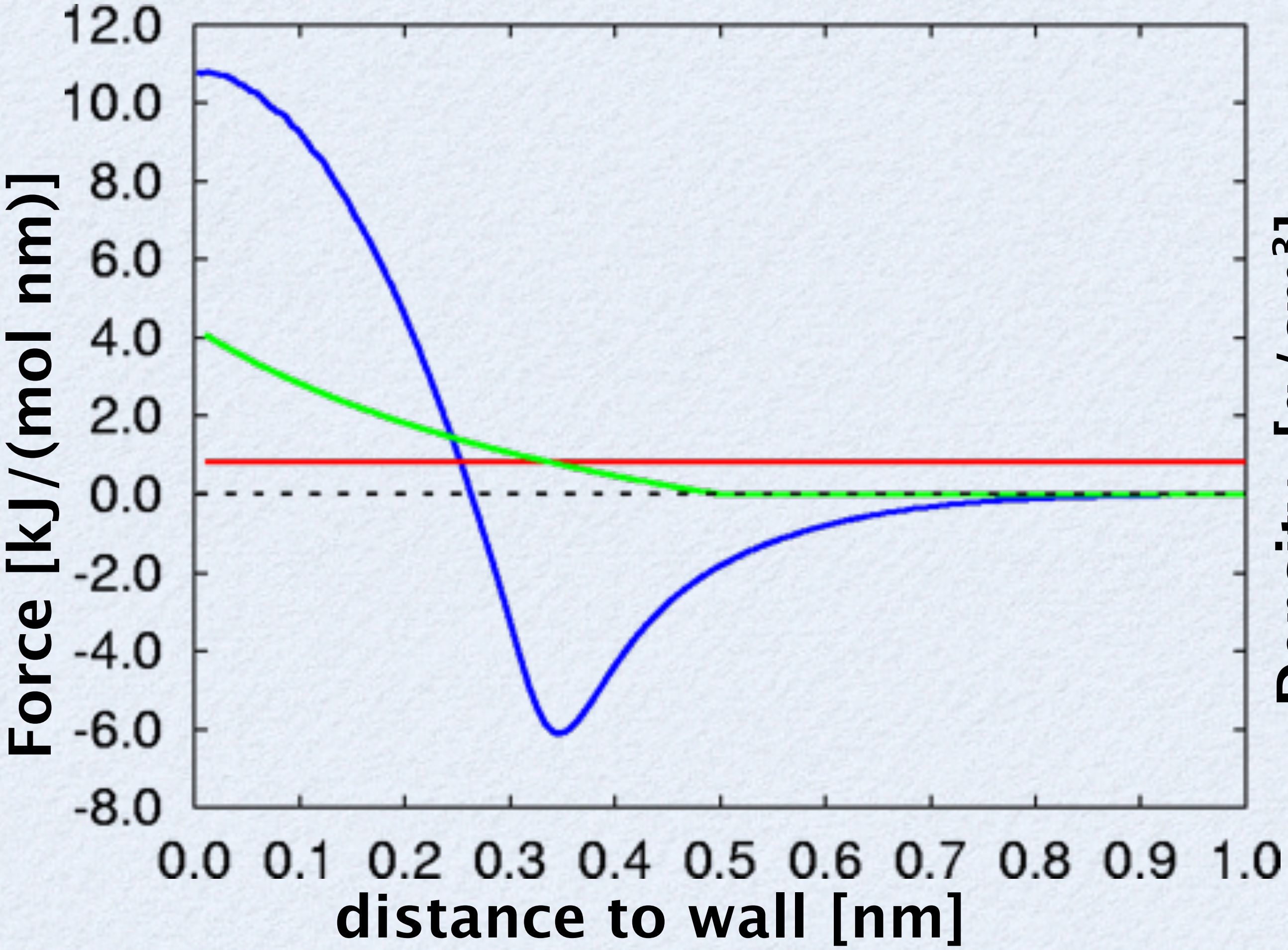
.

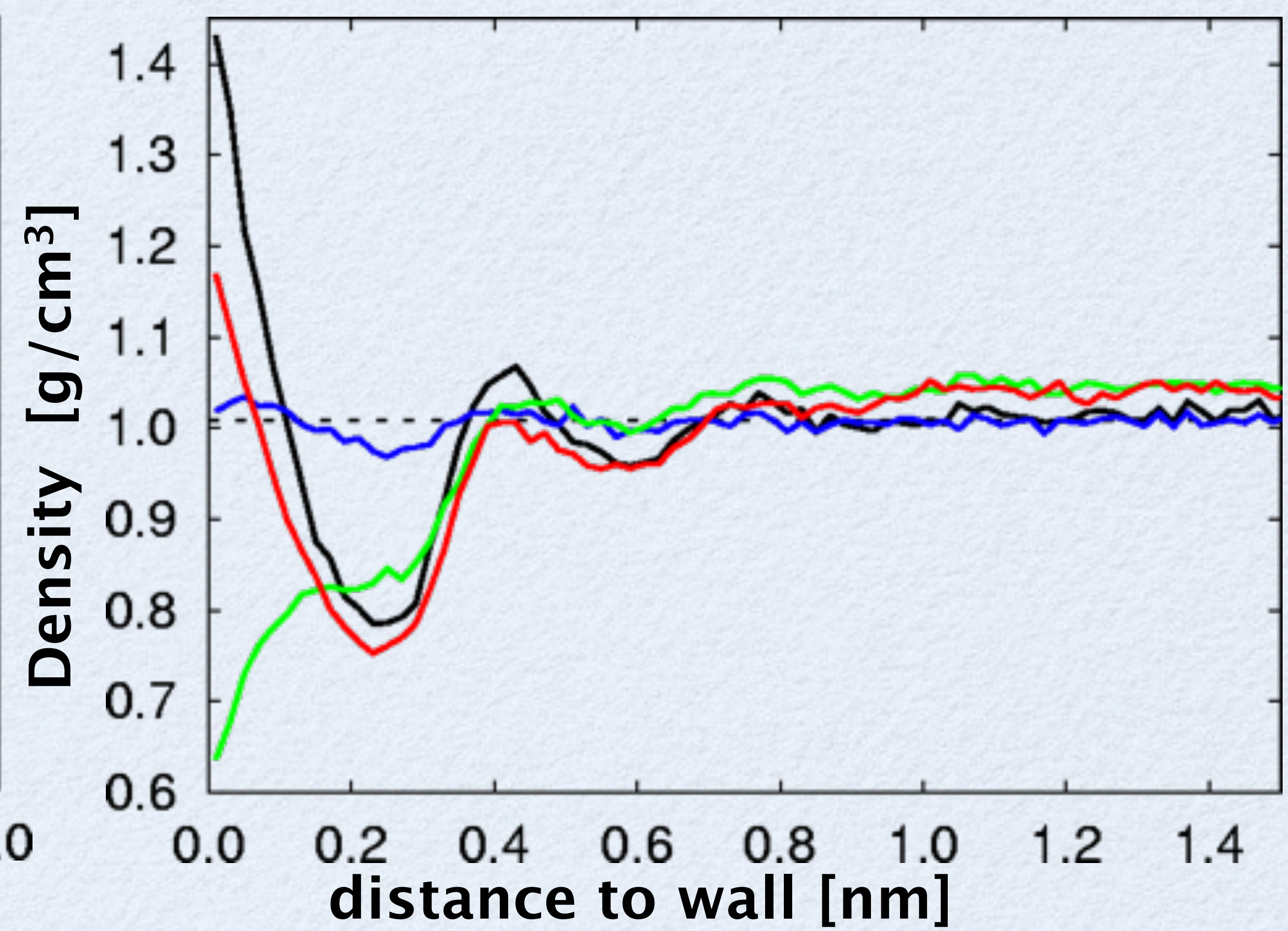
T. Werder, J. H. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flow. J. Comput. Phys., 205: 373-390, 2005.

A comparison of Forces

No force

Uniform distribution (O'Connell 1995[A])





Repulsive (Nie et al. 2004)

Use fluid structure (Werder et al. 2005)

S. T. O'Connell and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flow. Phys. Rev. E, 52(6):R5792-R5795, 1995

X. B. Nie, S. Y. Chen, W. N. E and M. O. Robbins. A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow. J. Fluid Mech., 500:55-64, 2004

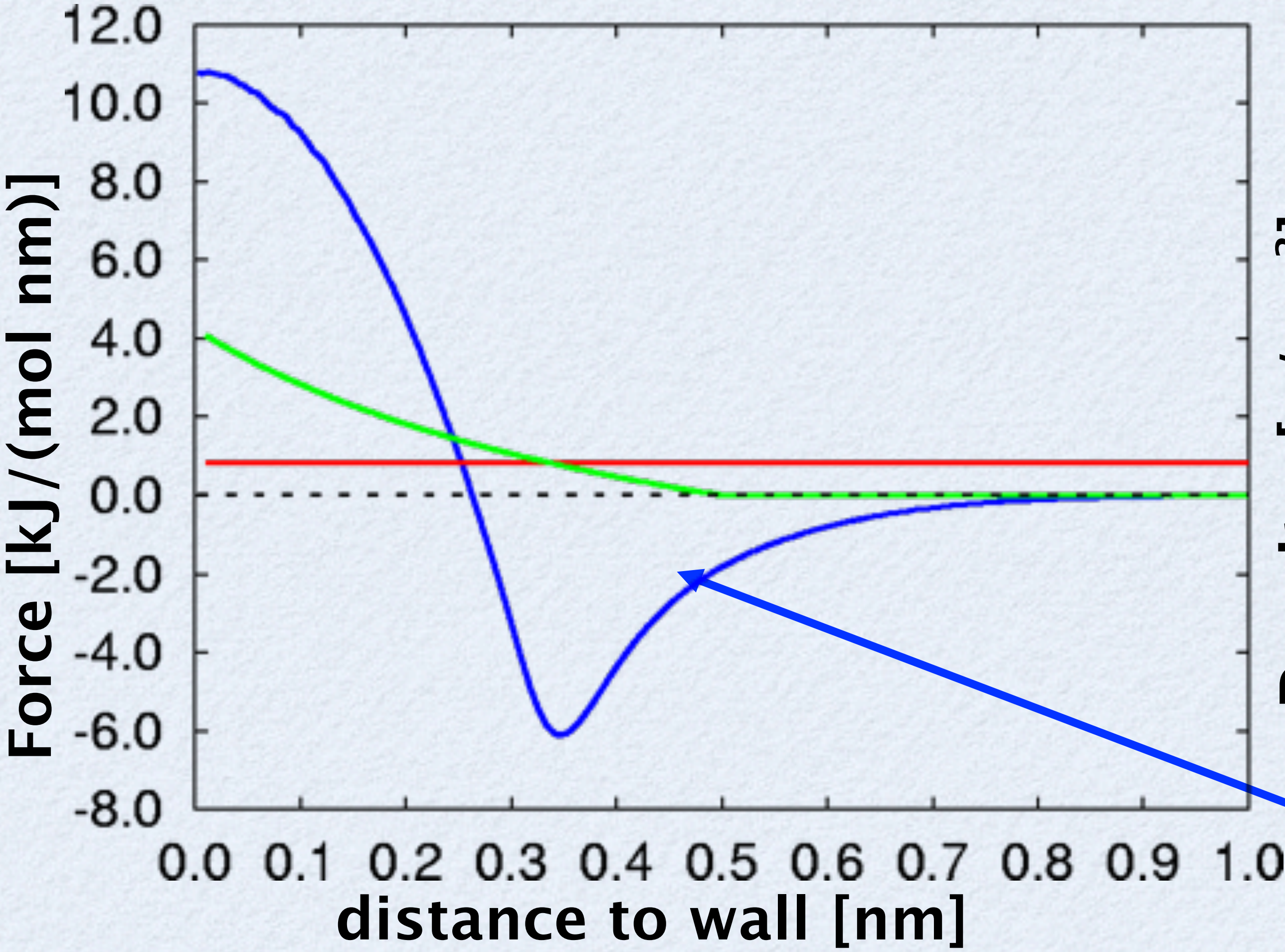
.

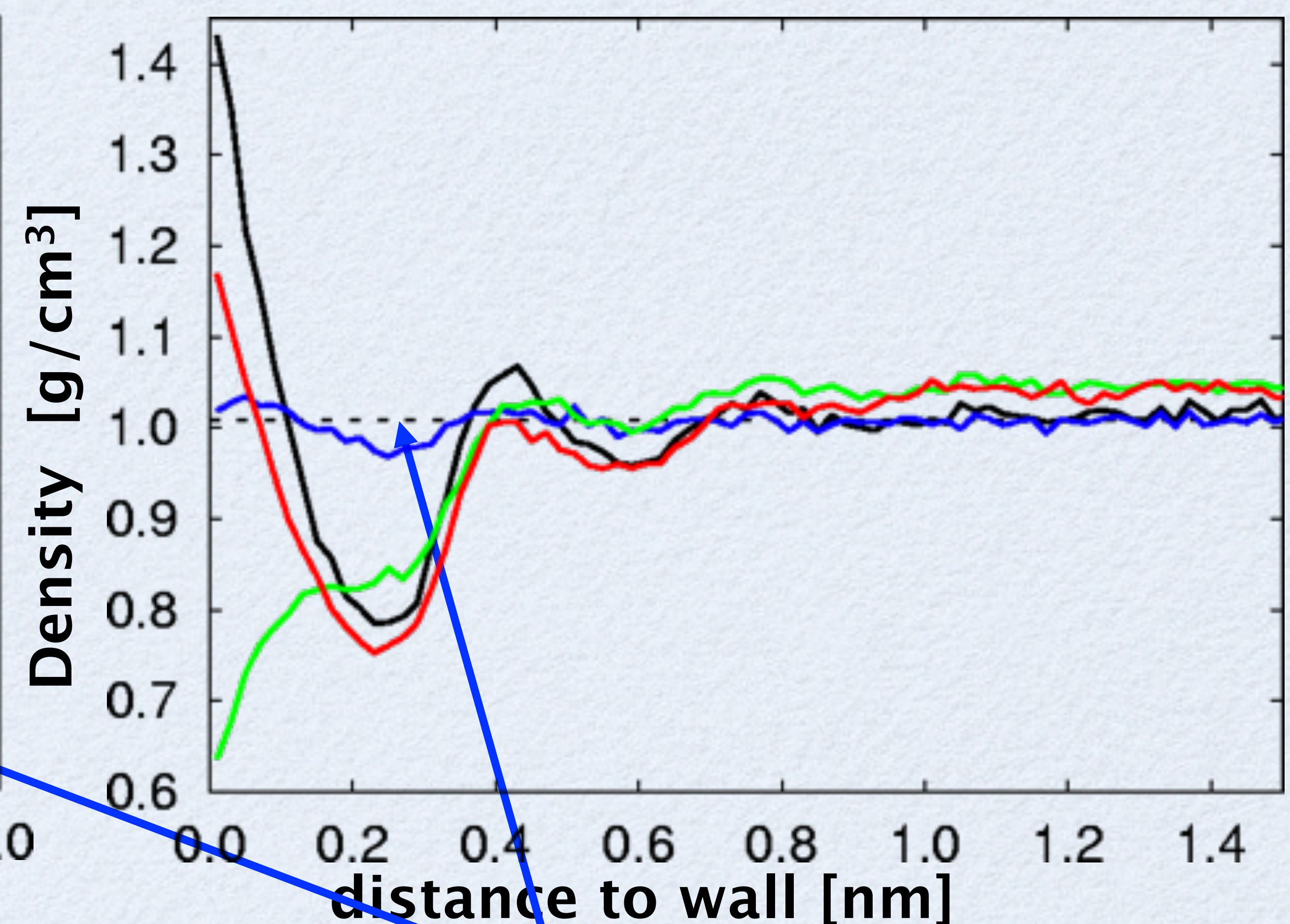
T. Werder, J. H. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flow. J. Comput. Phys., 205: 373-390, 2005.

A comparison of Forces

No force

Uniform distribution (O'Connell 1995[A])





Repulsive (Nie et al. 2004)

Use fluid structure (Werder et al. 2005)

S. T. O'Connell and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flow. Phys. Rev. E, 52(6):R5792-R5795, 1995

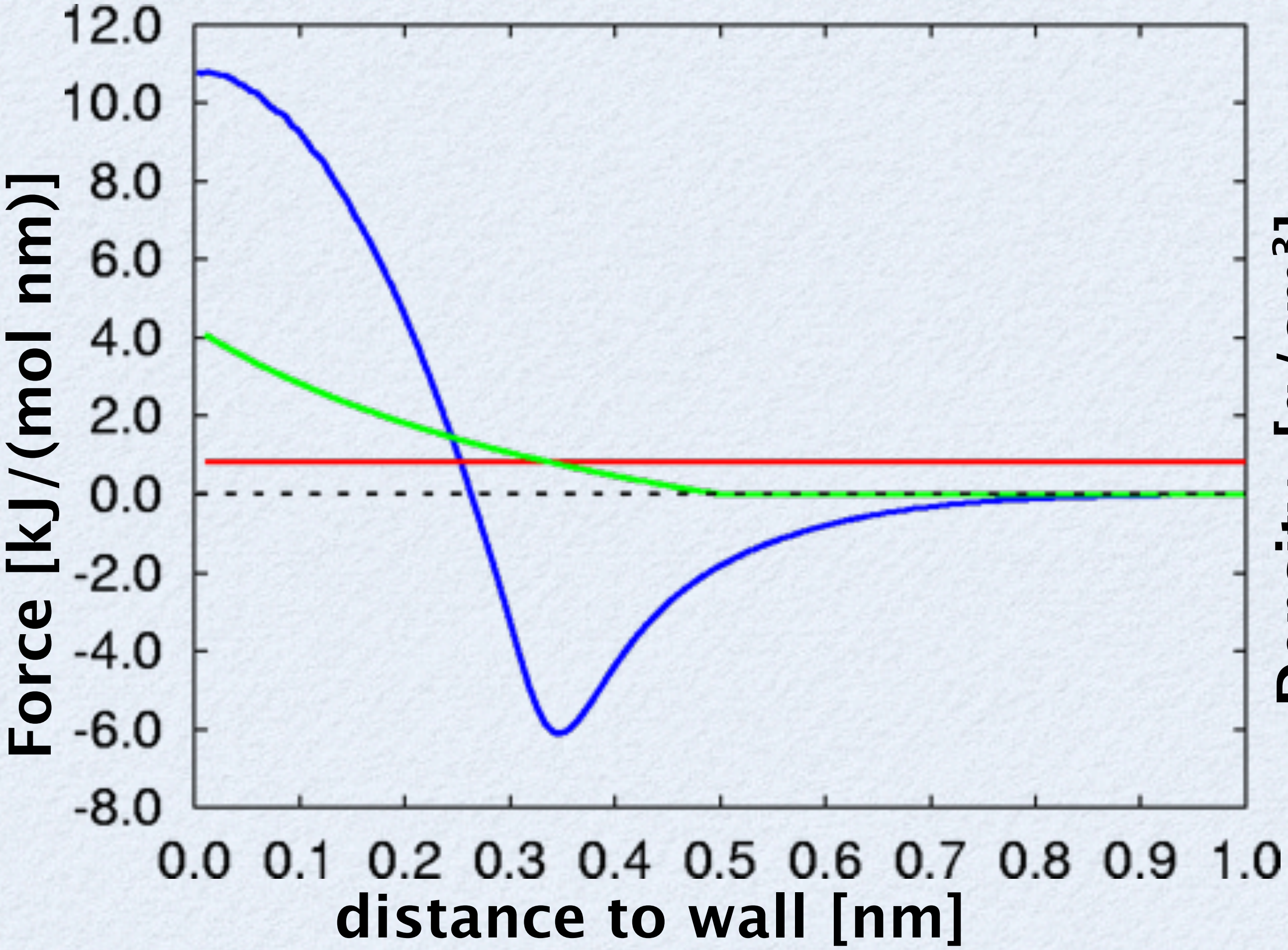
X. B. Nie, S. Y. Chen, W. N. E and M. O. Robbins. A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow. J. Fluid Mech., 500:55-64, 2004
.

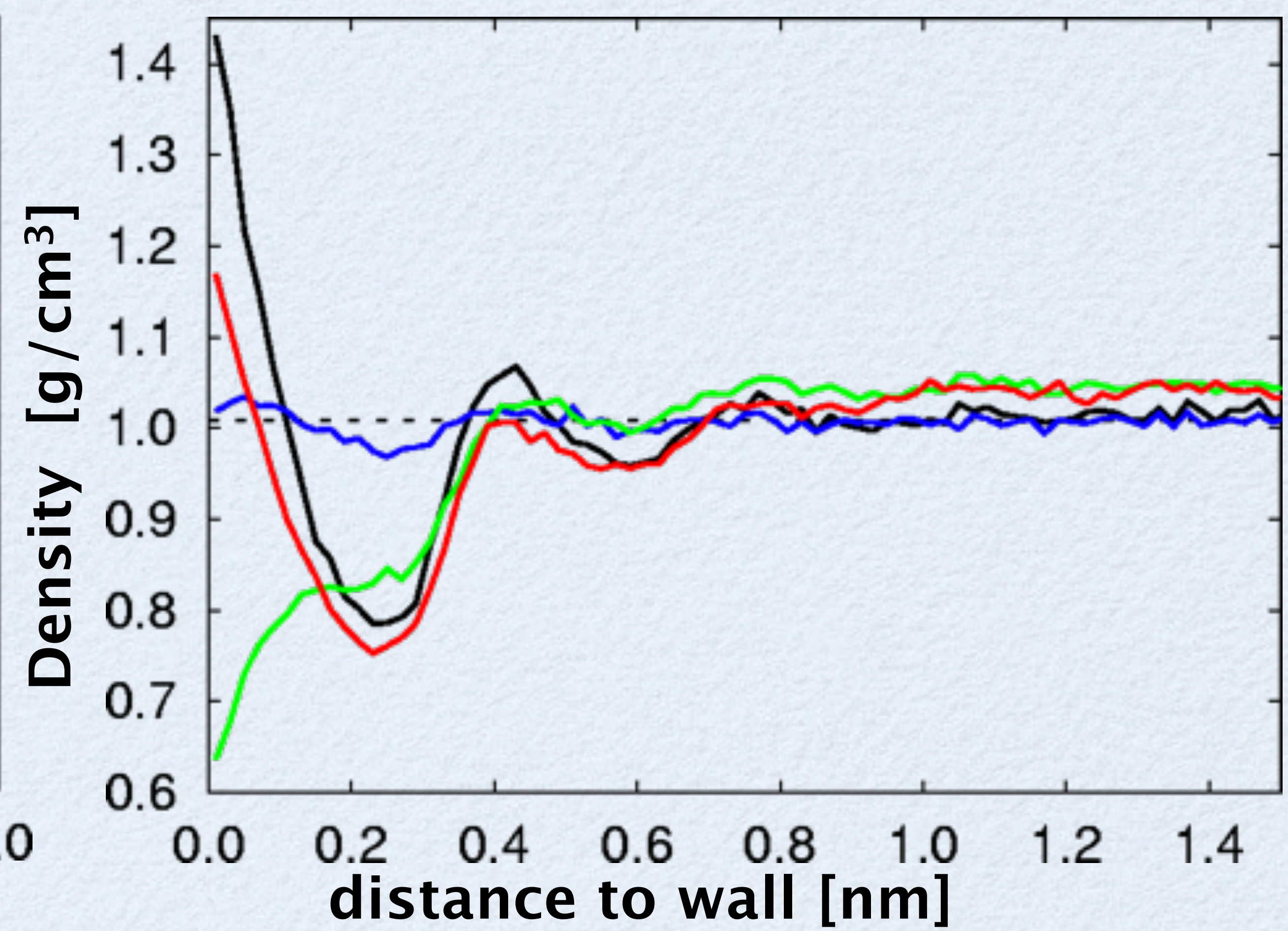
T. Werder, J. H. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flow. J. Comput. Phys., 205: 373-390, 2005.

A comparison of Forces

No force

Uniform distribution (O'Connell 1995[A])





Repulsive (Nie et al. 2004)

Use fluid structure (Werder et al. 2005)

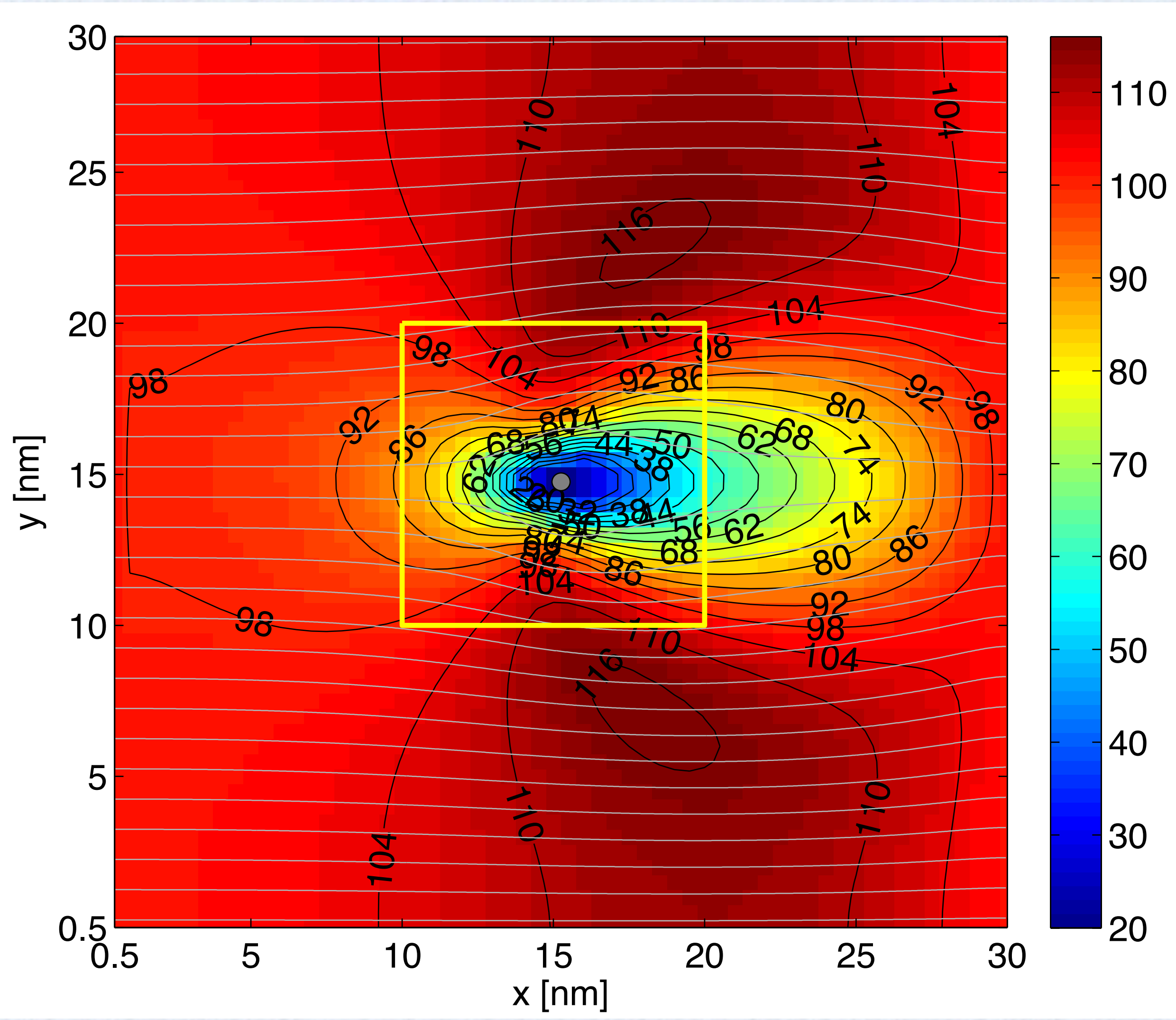
S. T. O'Connell and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flow. Phys. Rev. E, 52(6):R5792-R5795, 1995

X. B. Nie, S. Y. Chen, W. N. E and M. O. Robbins. A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow. J. Fluid Mech., 500:55-64, 2004

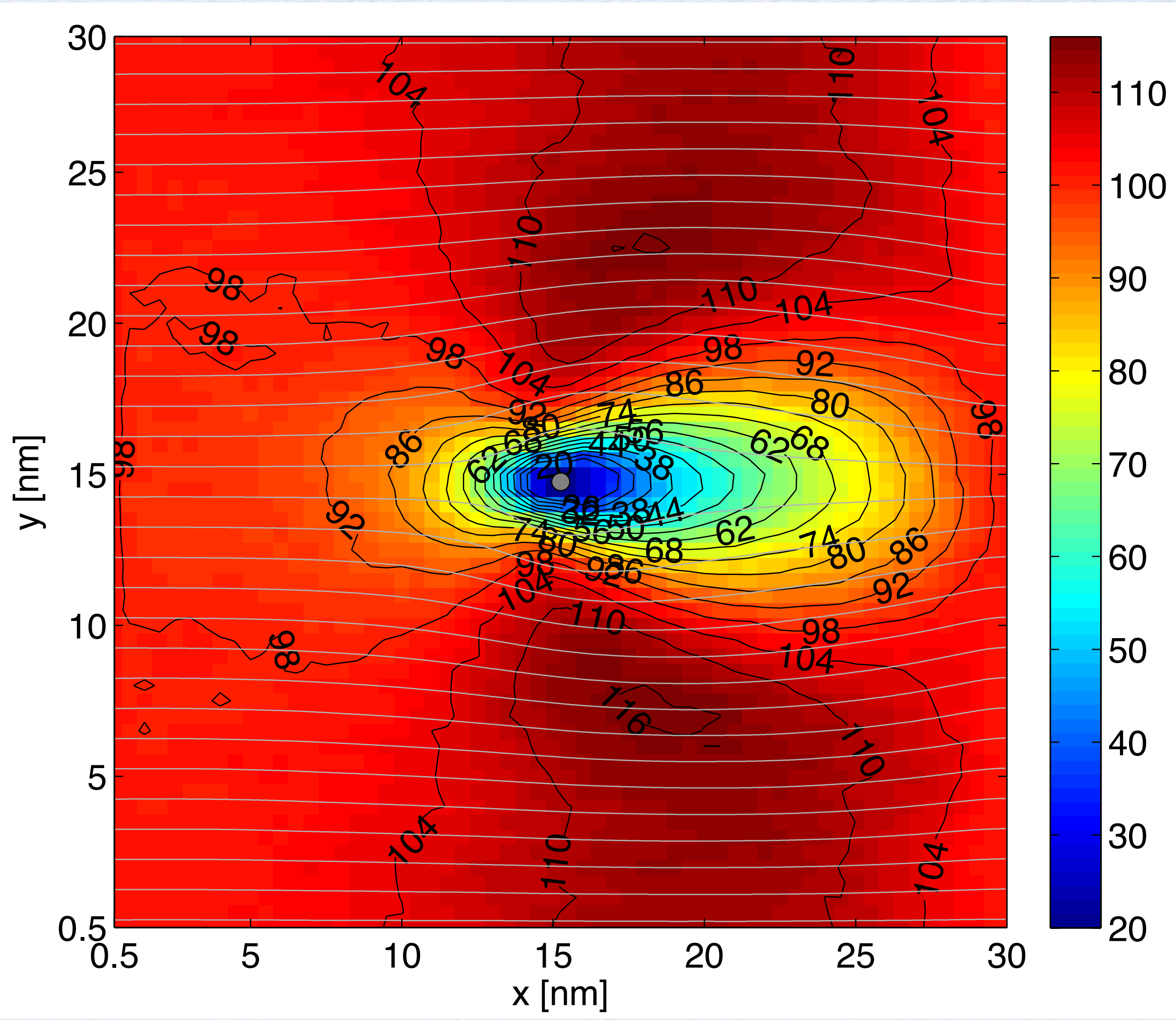
.

T. Werder, J. H. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flow. J. Comput. Phys., 205: 373-390, 2005.

MD vs Hybrid scheme



Hybrid solution



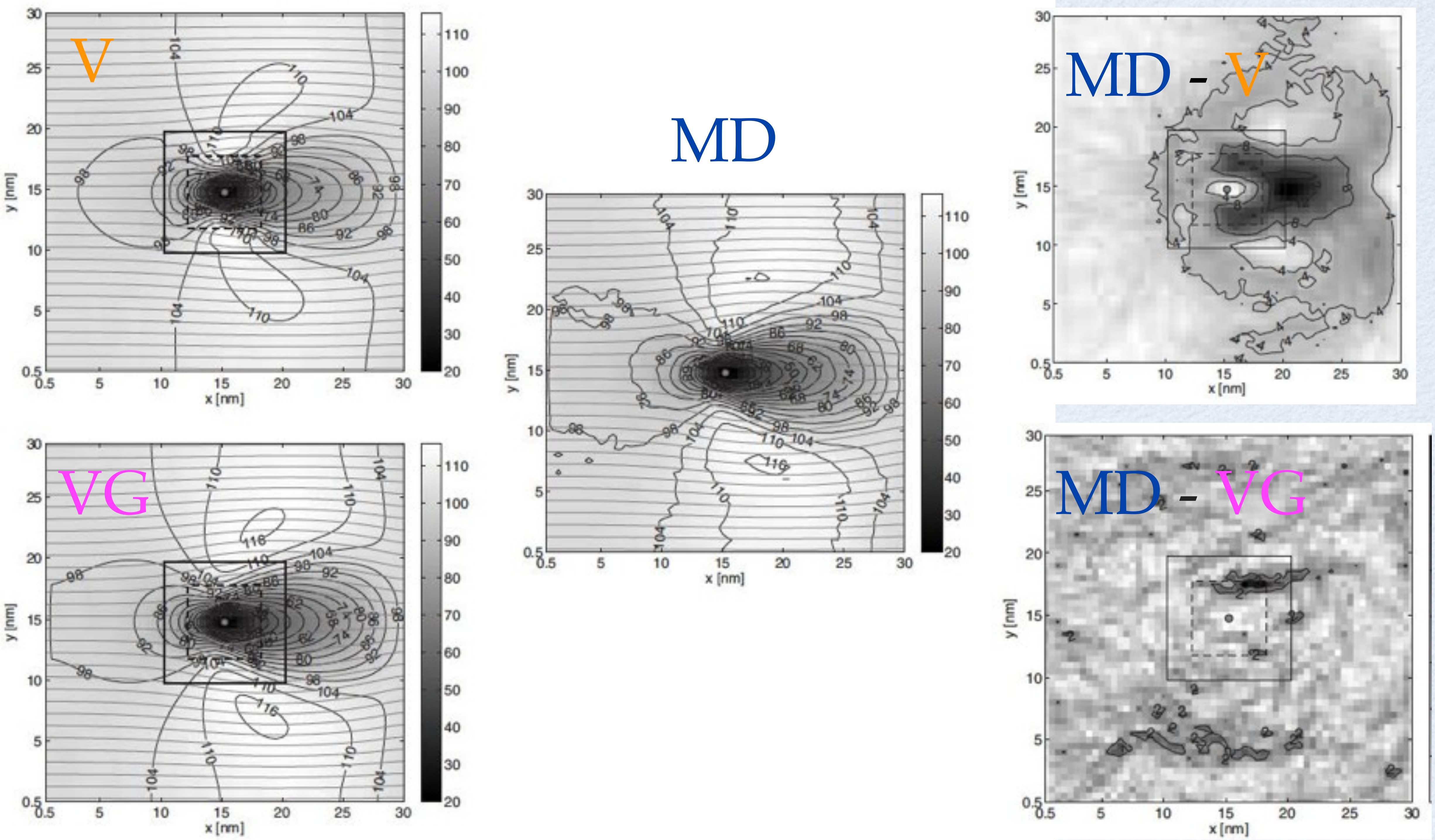
Reference MD solution

Relative Error ~ 1.3%

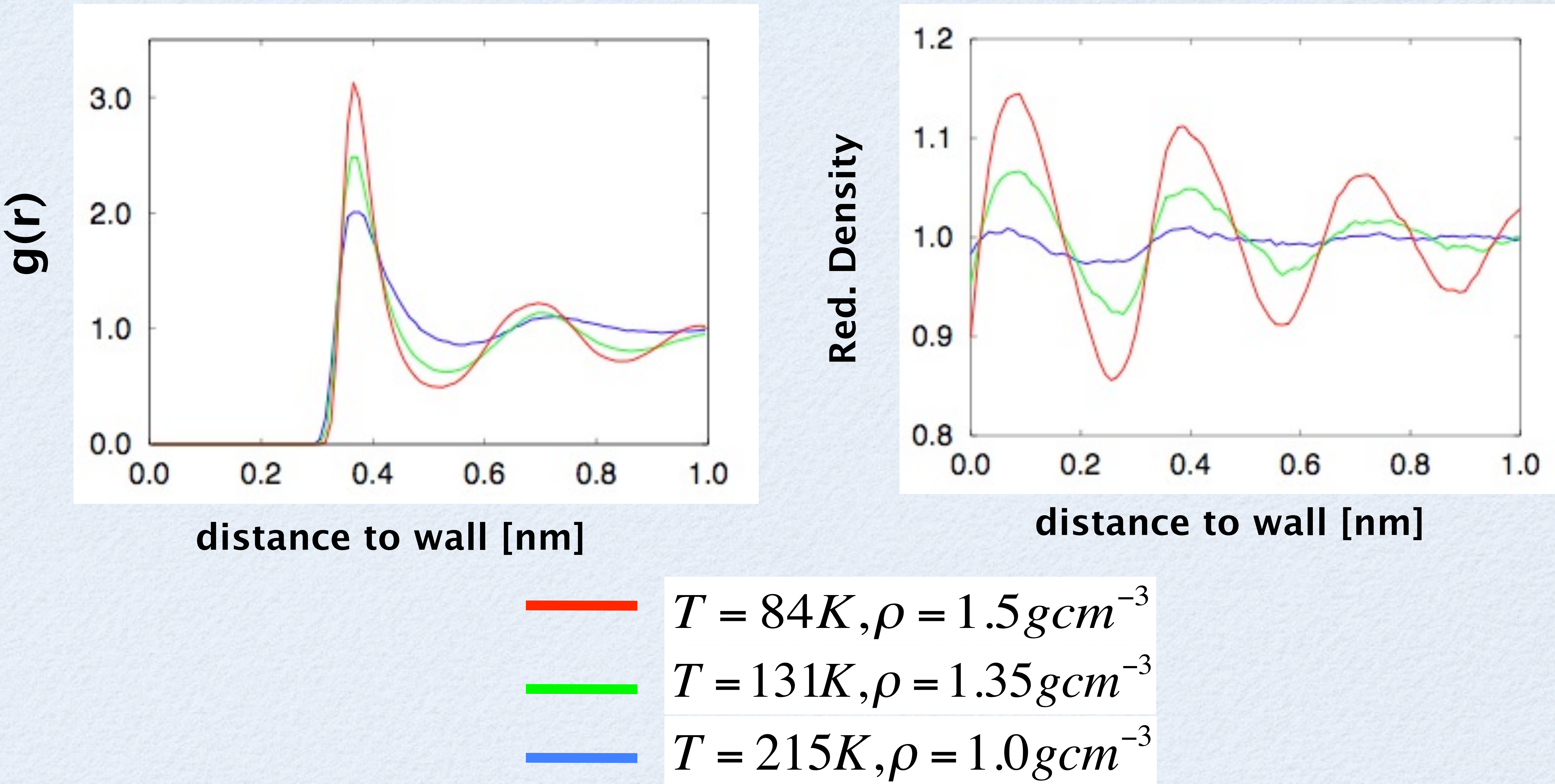
The hybrid scheme is ~ (L/R)3 times faster for a computational domain of size L and a MD subdomain of size R.**

A. Dupuis, E.M. Kotsalis, and PK., Coupling lattice Boltzmann and molecular dynamics for dense fluids. Phys. Rev. E, 75: 046704, 2007

Errors for Different Couplings



The problem with density variations

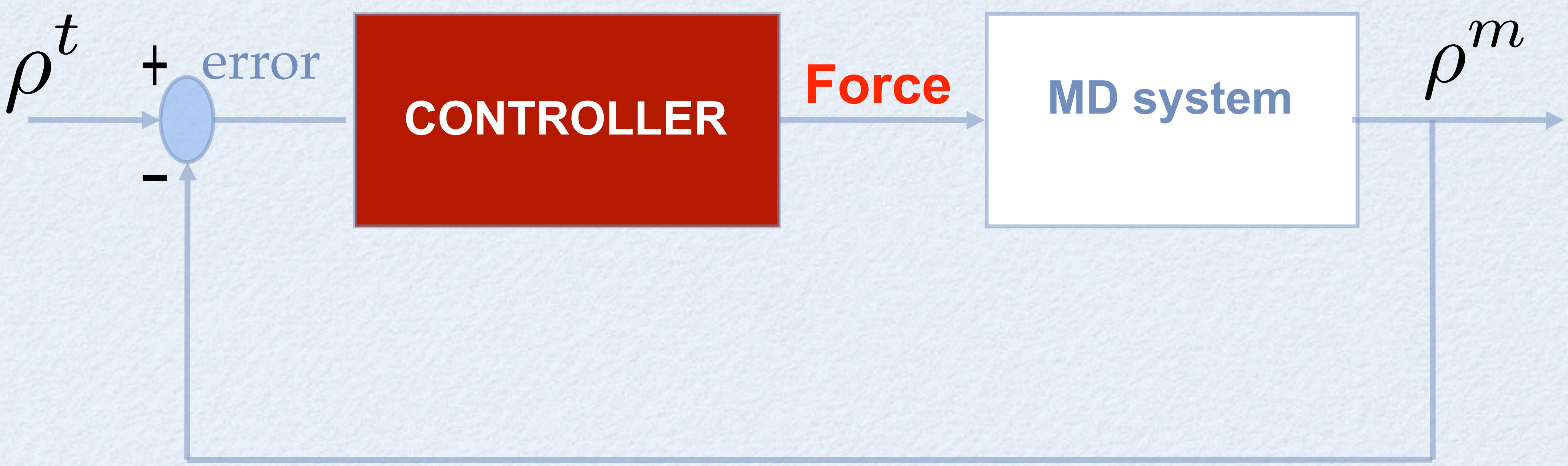


- Density variations depend on liquid state
- **Amplitude** proportional to **structural correlations** in the liquid

a simple **Control approach to Coupling**

E.M. Kotsalis, J.H. Walther, and P. Koumoutsakos., Phys. Rev. E, 2007.

- Controlling of the external boundary force
- measured density ρ^m => target density ρ^t

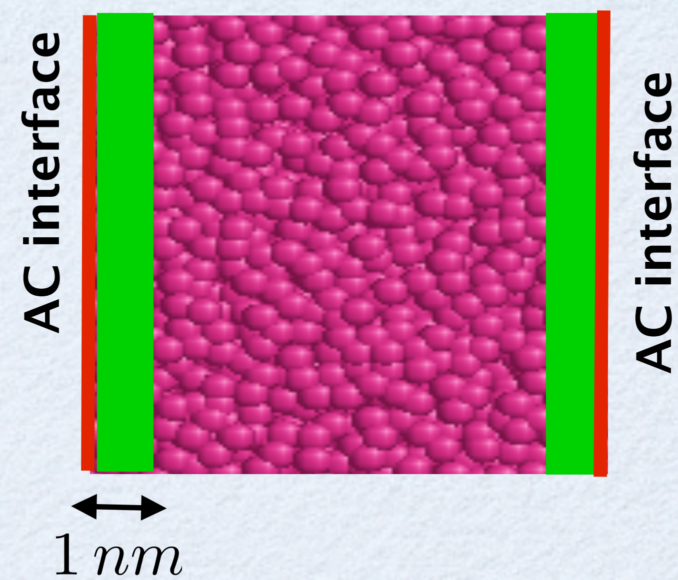


$$e(r) = \rho^t(r) - \rho^m(r)$$

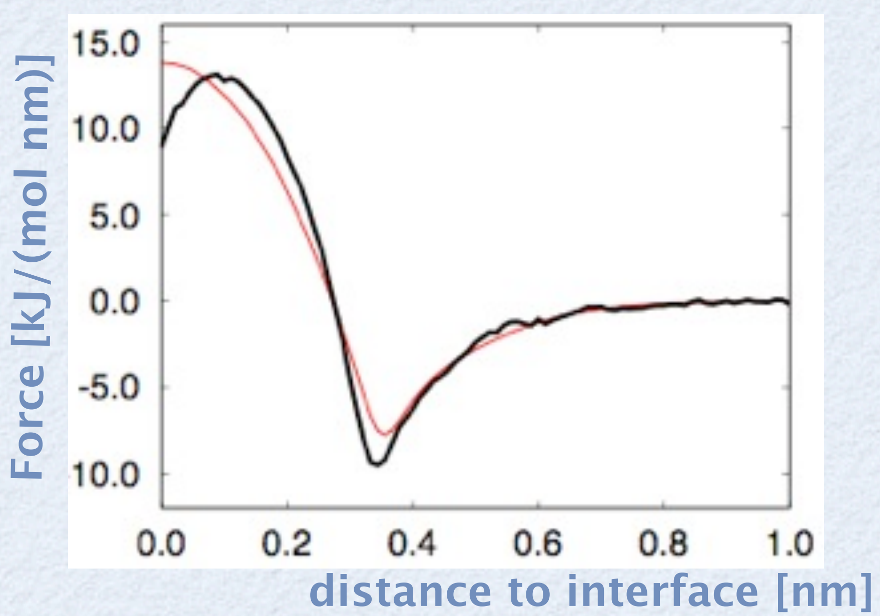
$$\textbf{Force} = k\nabla e(r)$$

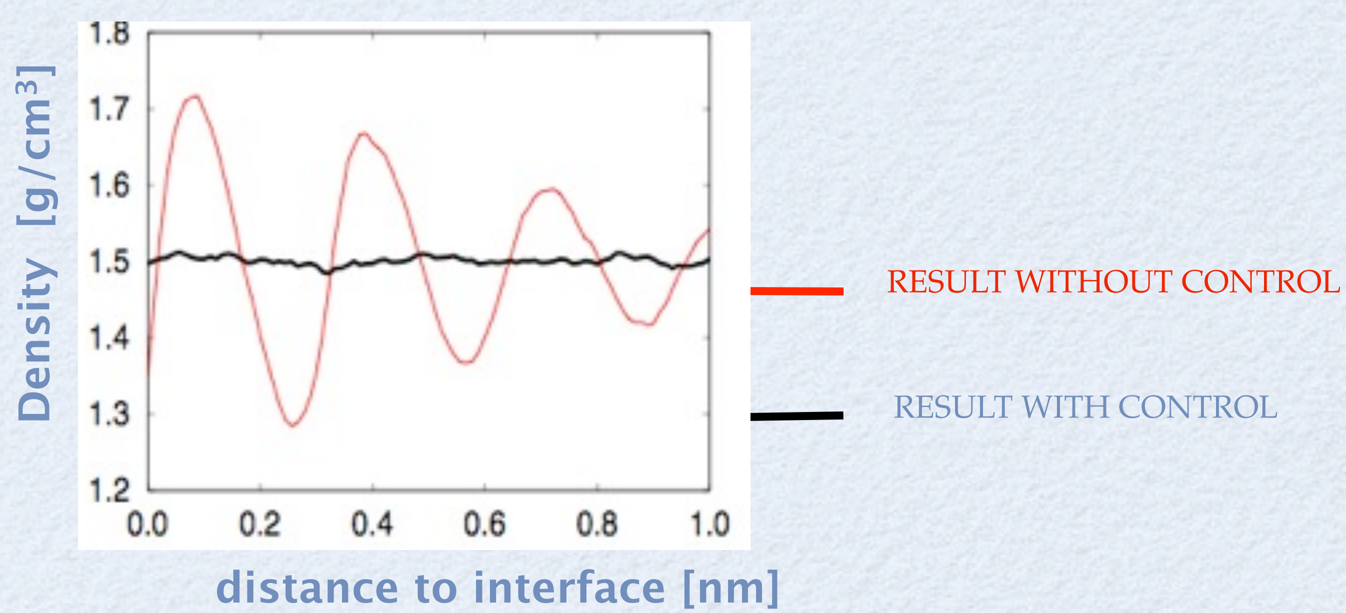
Results with Control Approach I

- at equilibrium (no flow)
- $T = 84K, \rho = 1.5gcm^{-3}$



AC interface = Elastic Boundary+External Force

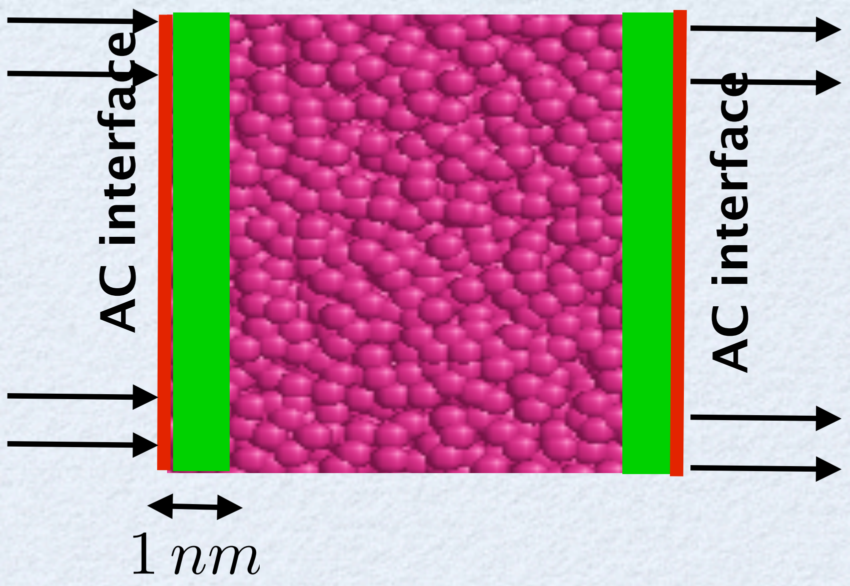




Controller deduces the boundary force

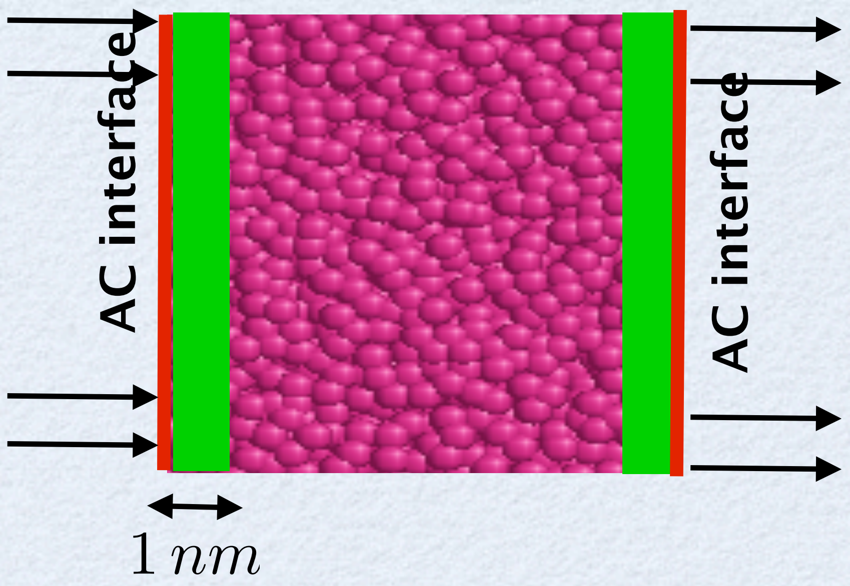
Results with Control Approach II

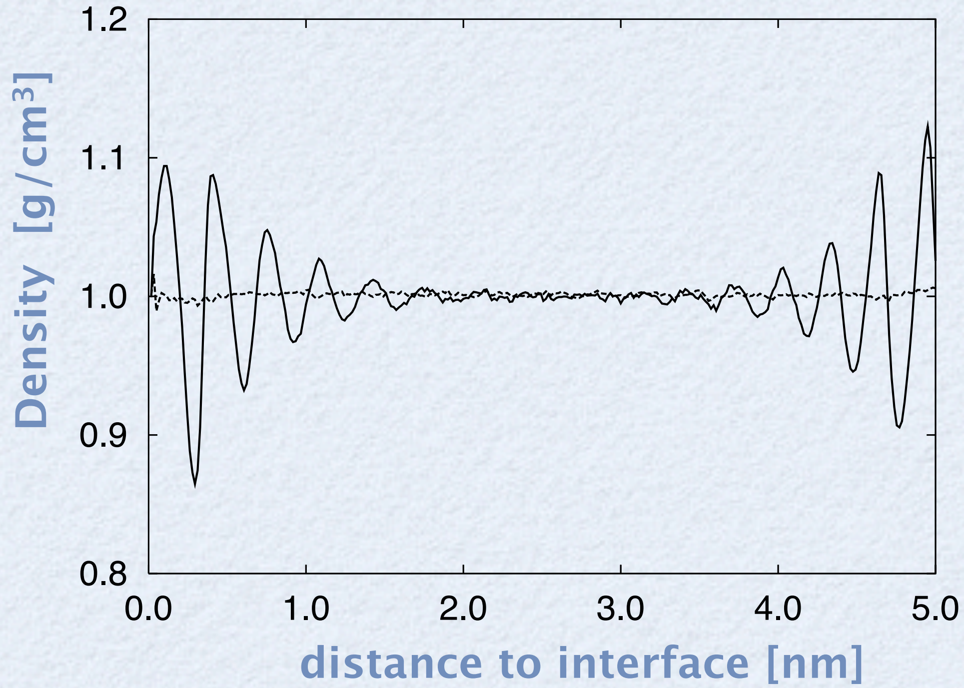
- uniform flow
- $T = 131K, \rho = 1.35gcm^{-3}$

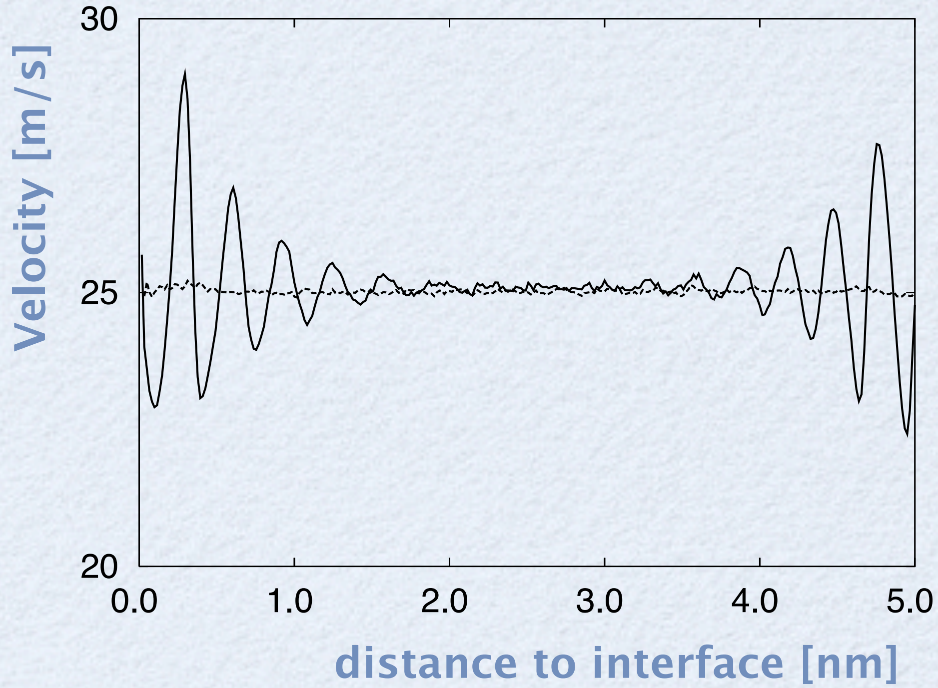


Results with Control Approach II

- uniform flow
- $T = 131K, \rho = 1.35gcm^{-3}$



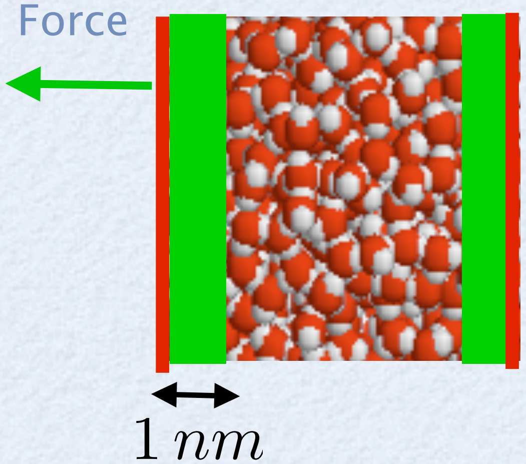




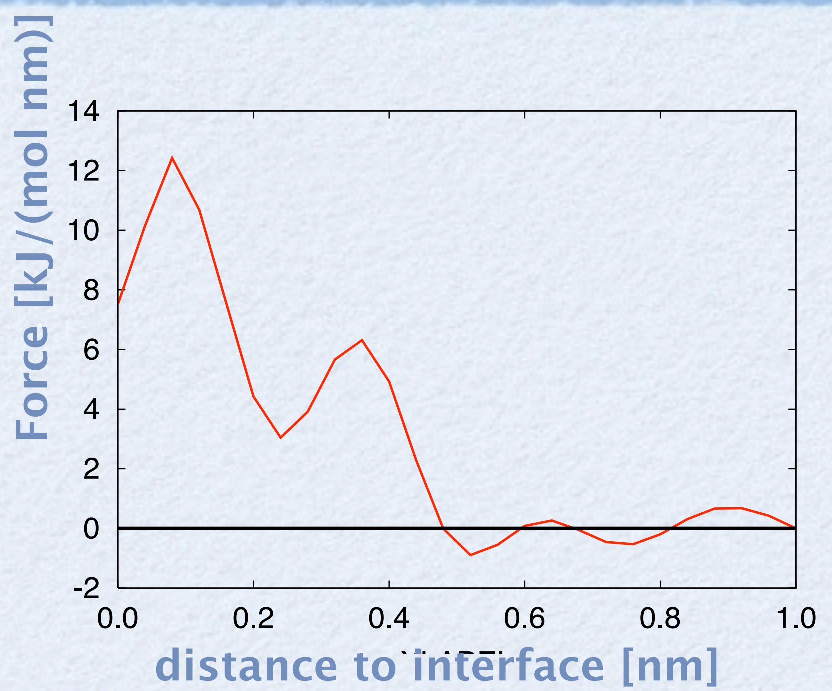
RESULT WITH CONTROL

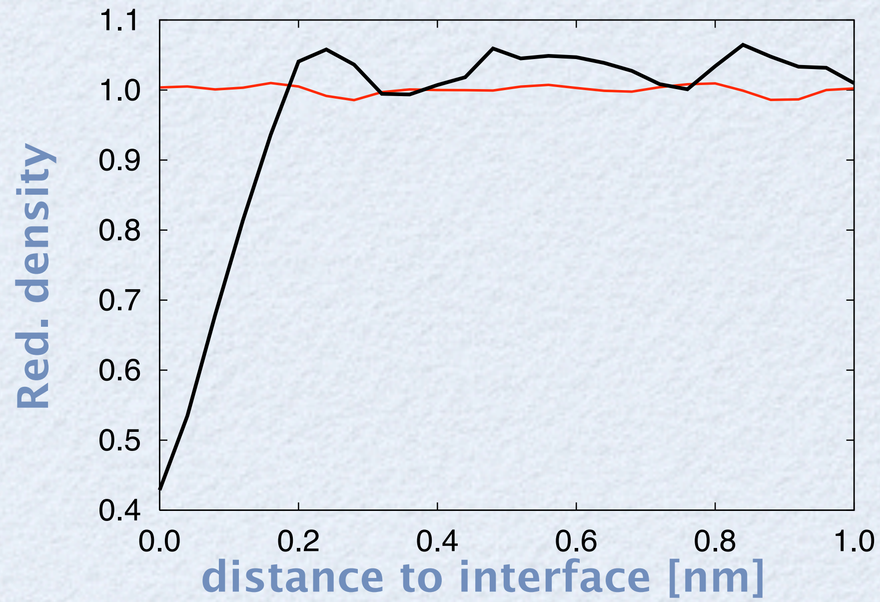
RESULT WITHOUT CONTROL

AC Interface for Water at Equilibrium



1D embedding

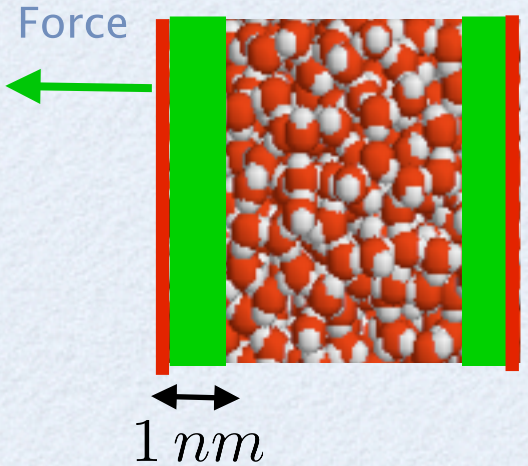




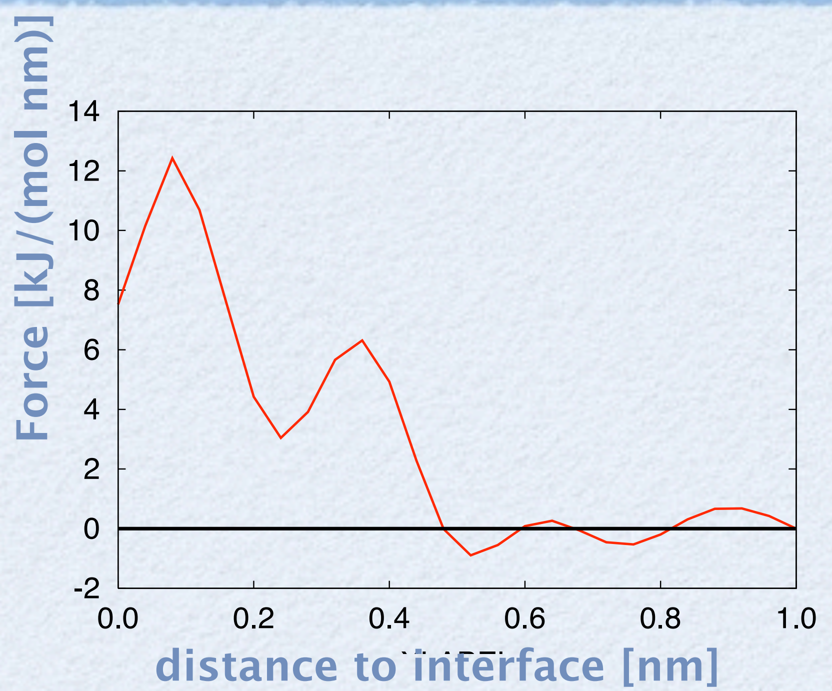
RESULT WITH CONTROL

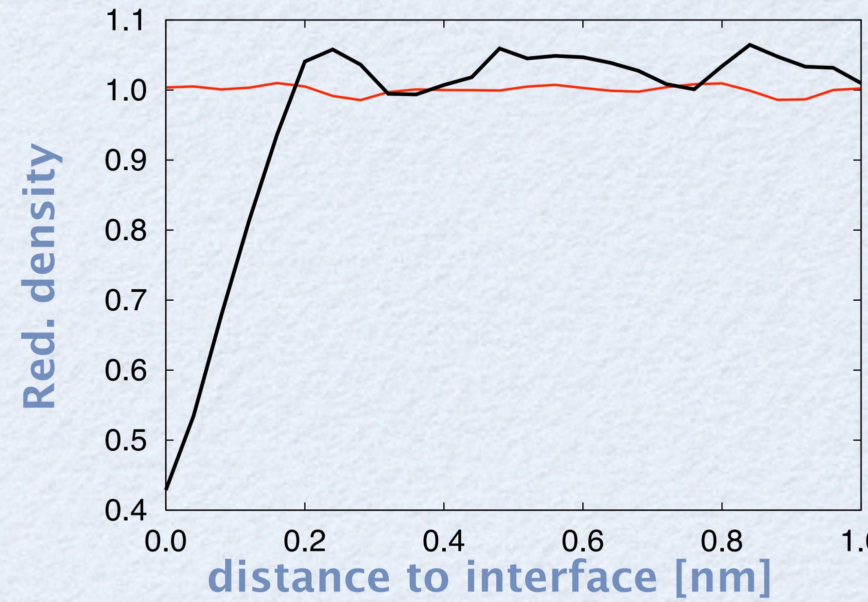
RESULT WITHOUT CONTROL

AC Interface for Water at Equilibrium



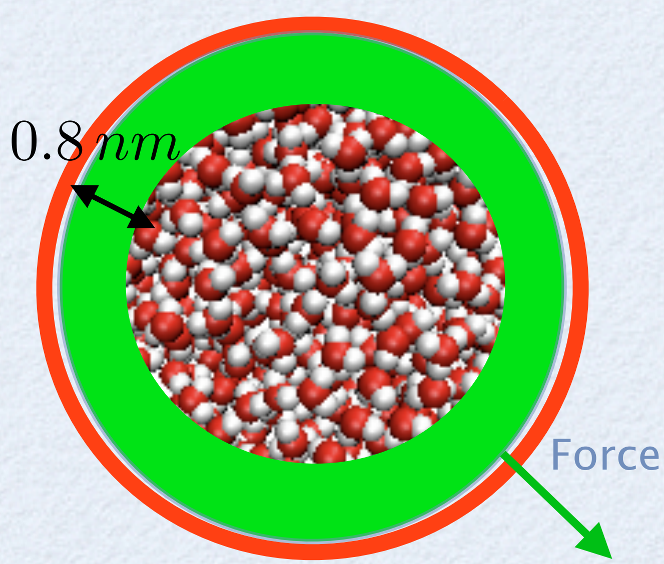
1D embedding



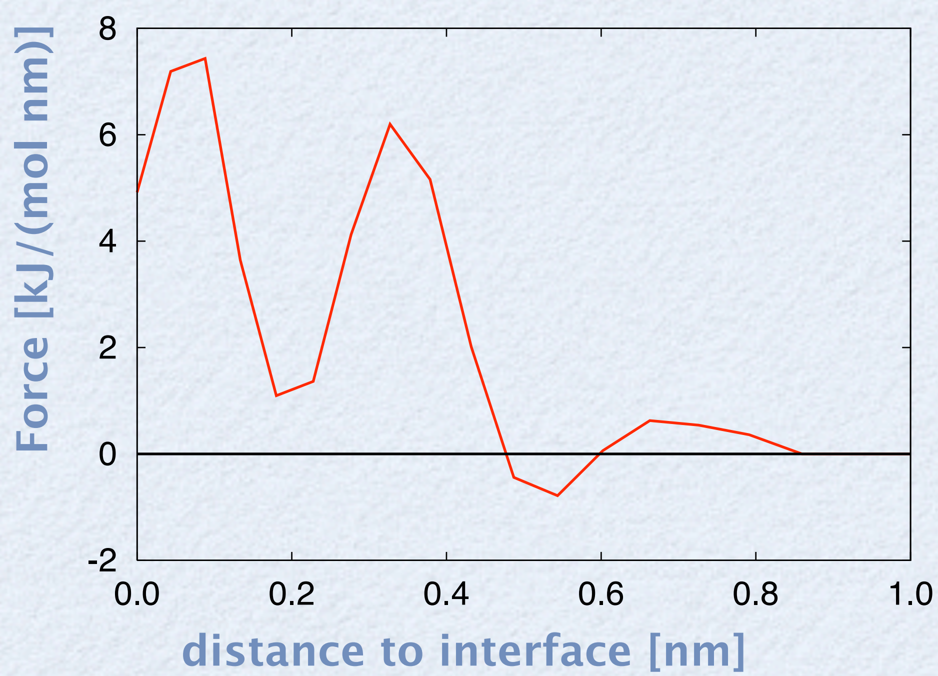


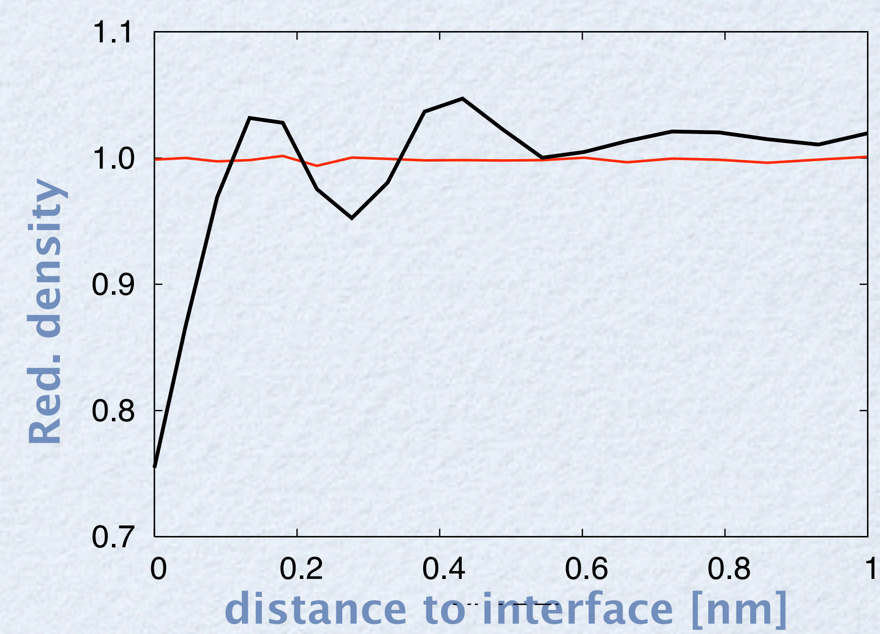
RESULT WITH CONTROL

RESULT WITHOUT CONTROL

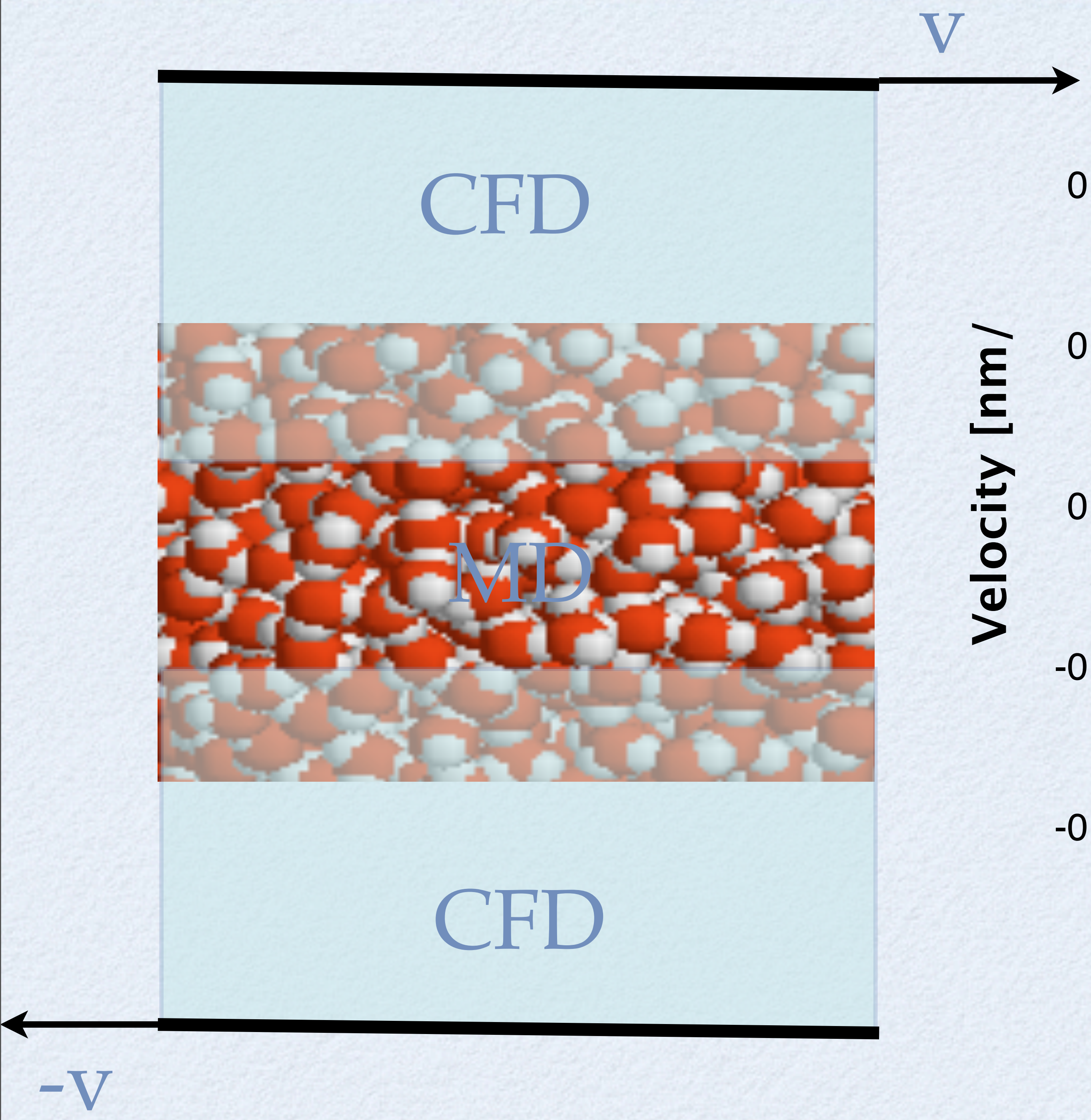


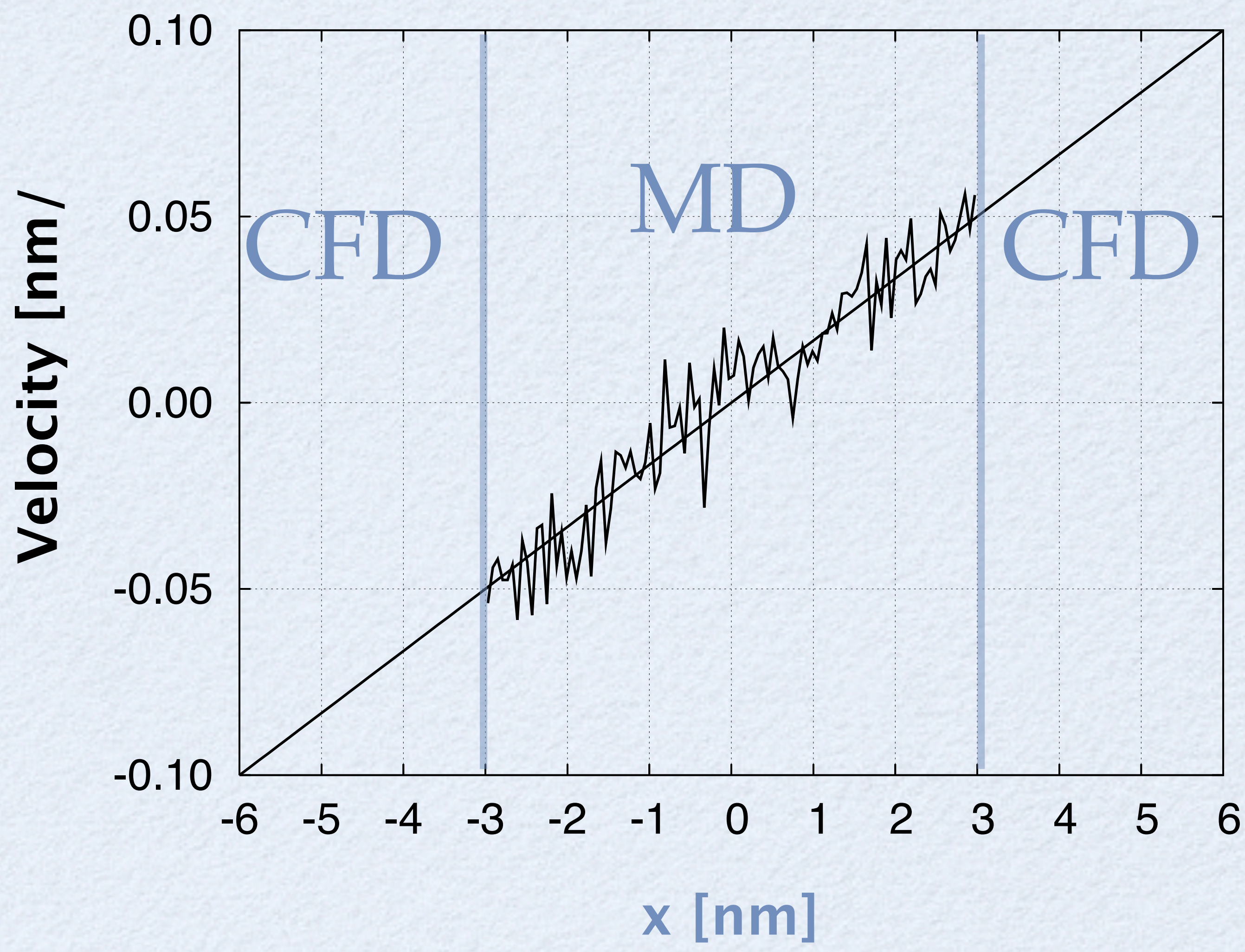
Spherical embedding



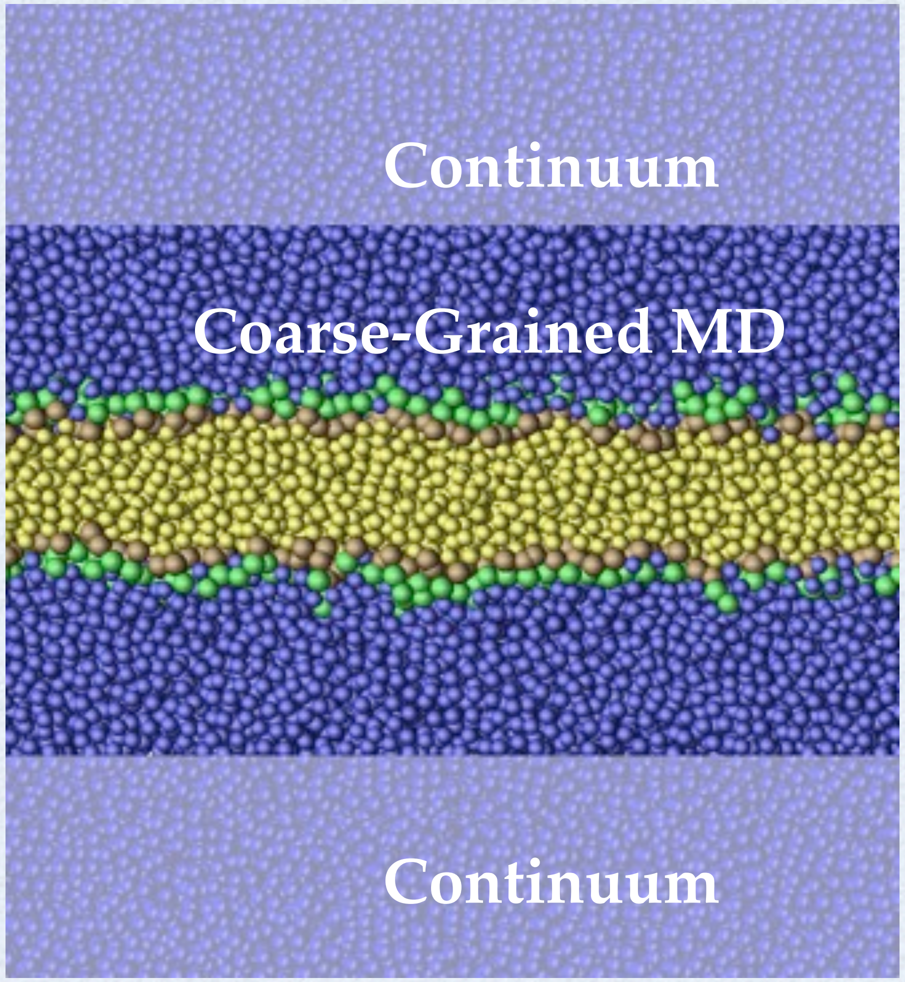


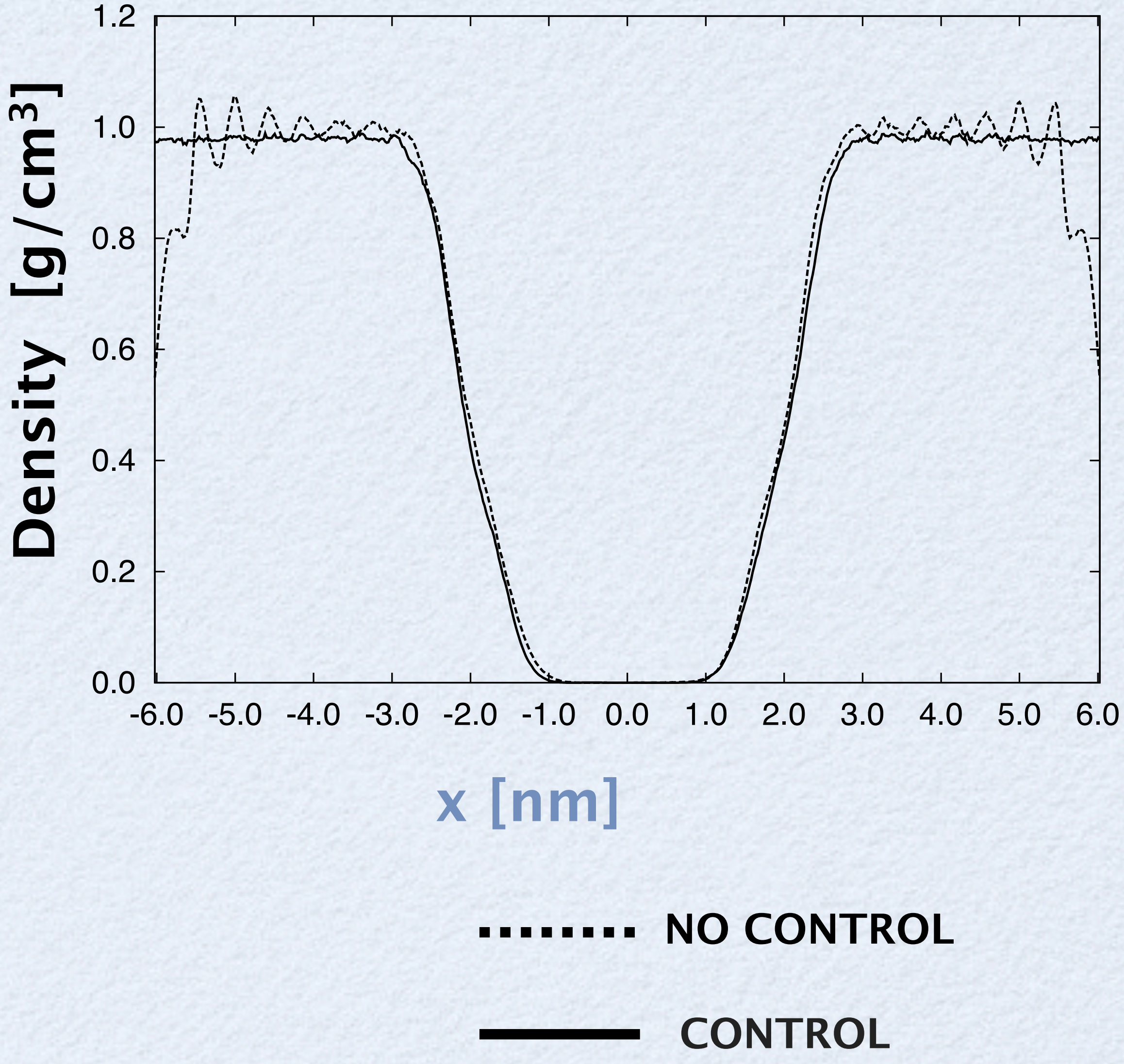
Water Couette Flow





Multiscale Membrane Simulation

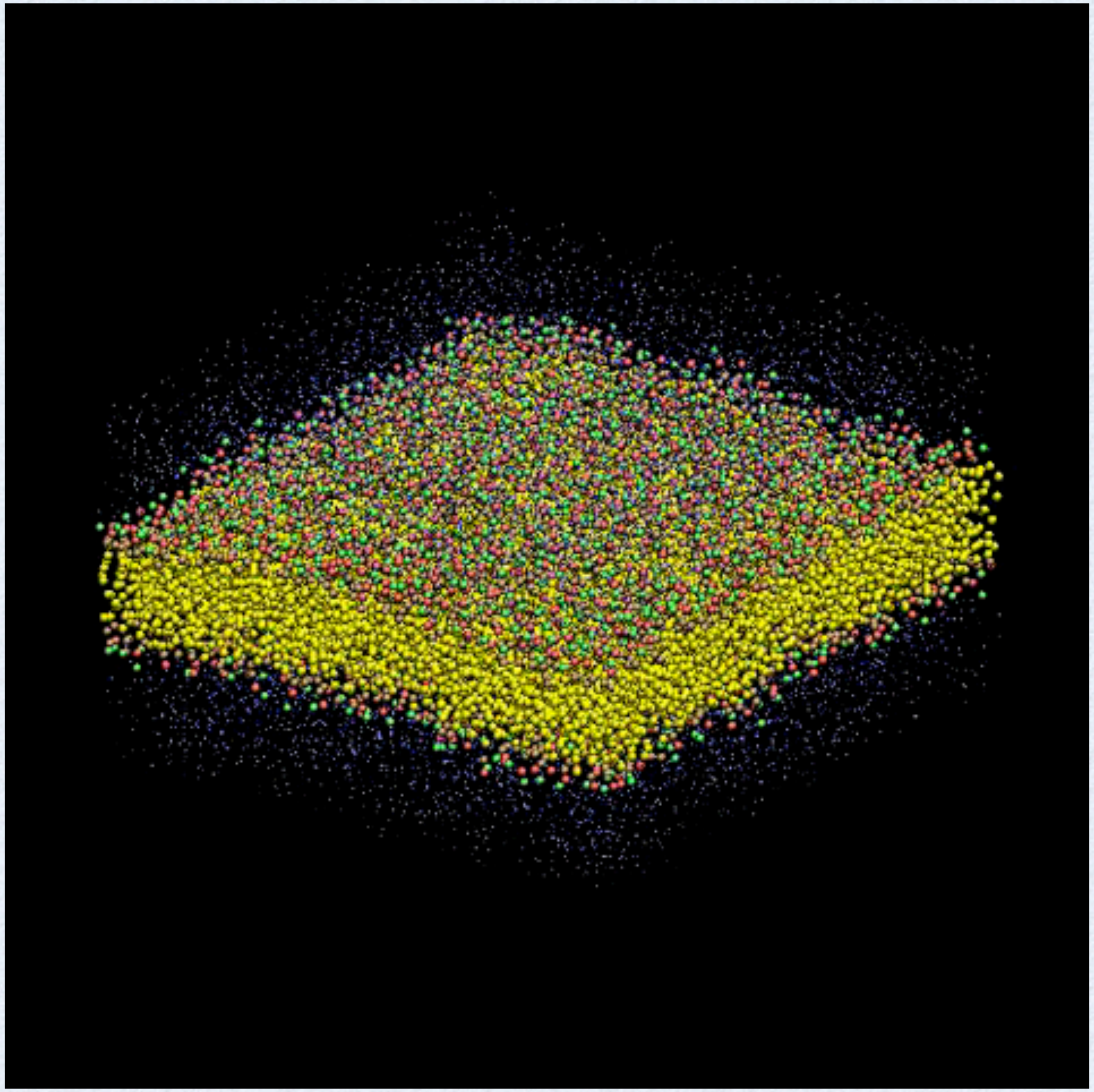


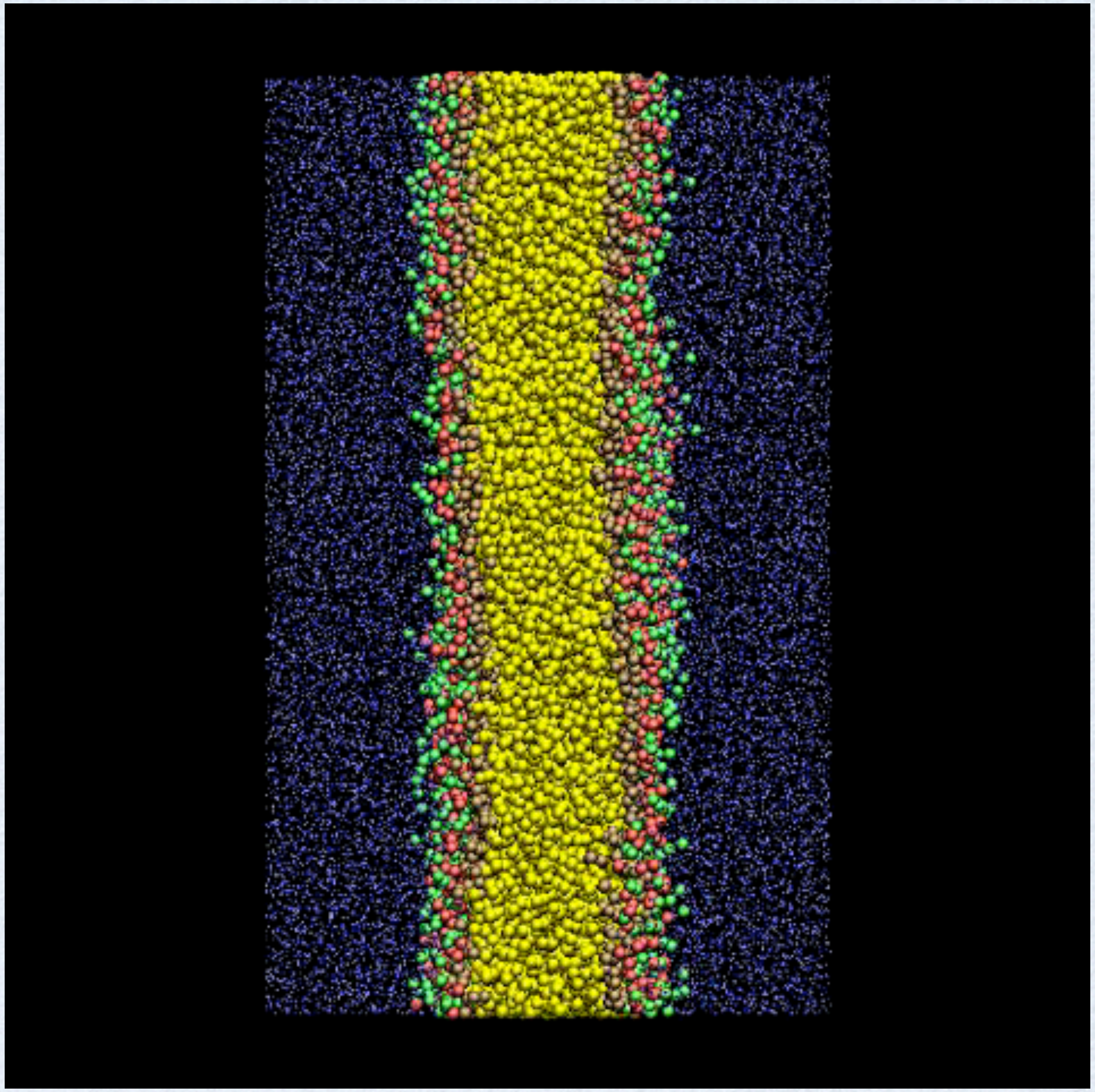


CG DPPC lipid surrounded by CG water.
Size: 12 nm x 20 nm x 20 nm

- 26250 CG water molecules
- 1250 CG lipid molecules

Multiscale Membrane Simulation



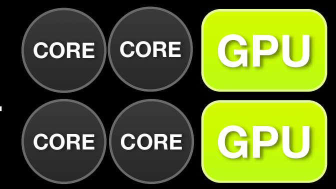


CG DPPC lipid surrounded by CG water.
Size: 12 nm x 20 nm x 20 nm

- 26250 CG water molecules
- 1250 CG lipid molecules

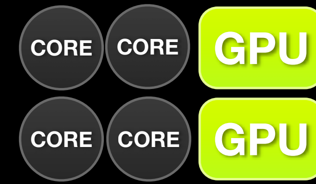
MULTISCALE METHODS

MULTISCALE METHODS +

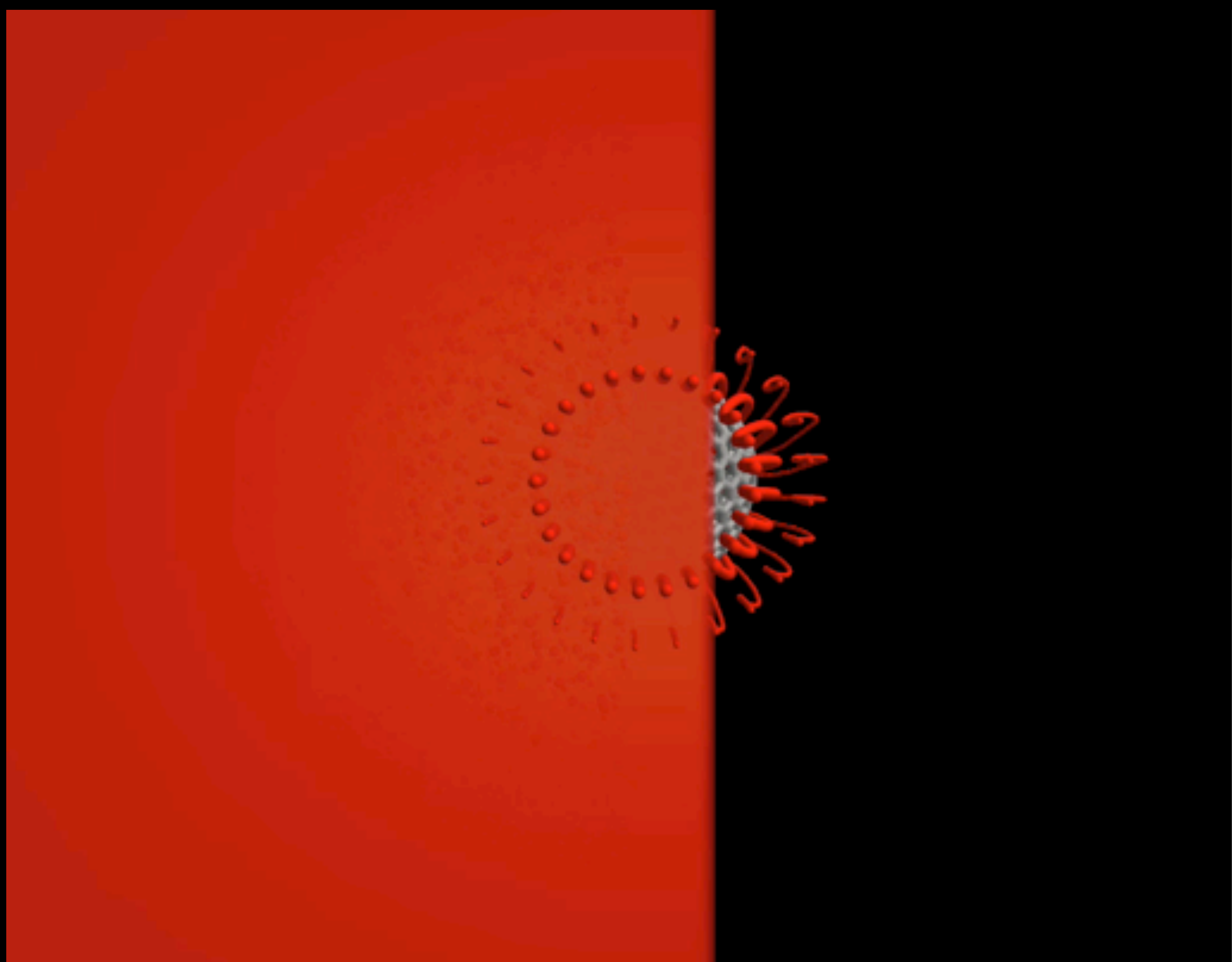


MD - Lattice-Boltzmann

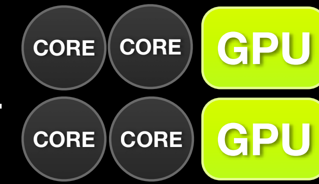
MULTISCALE METHODS +



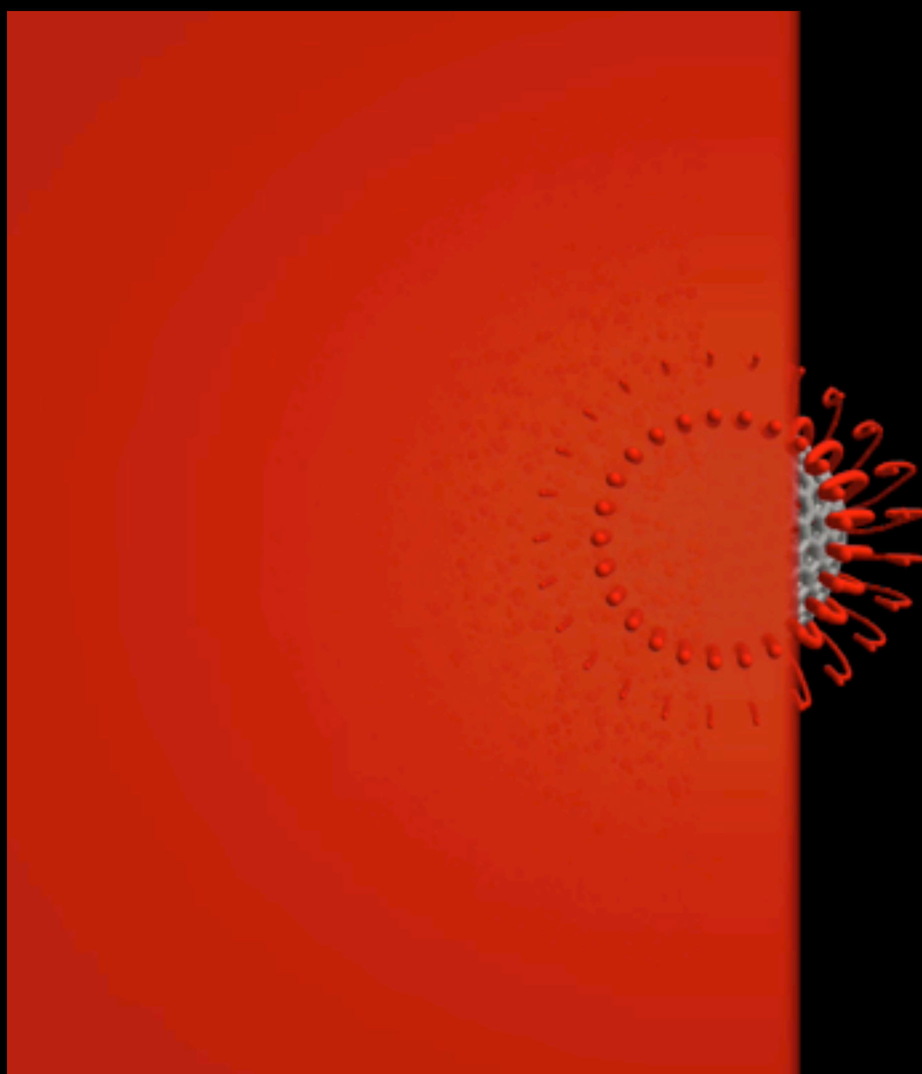
MD - Lattice-Boltzmann

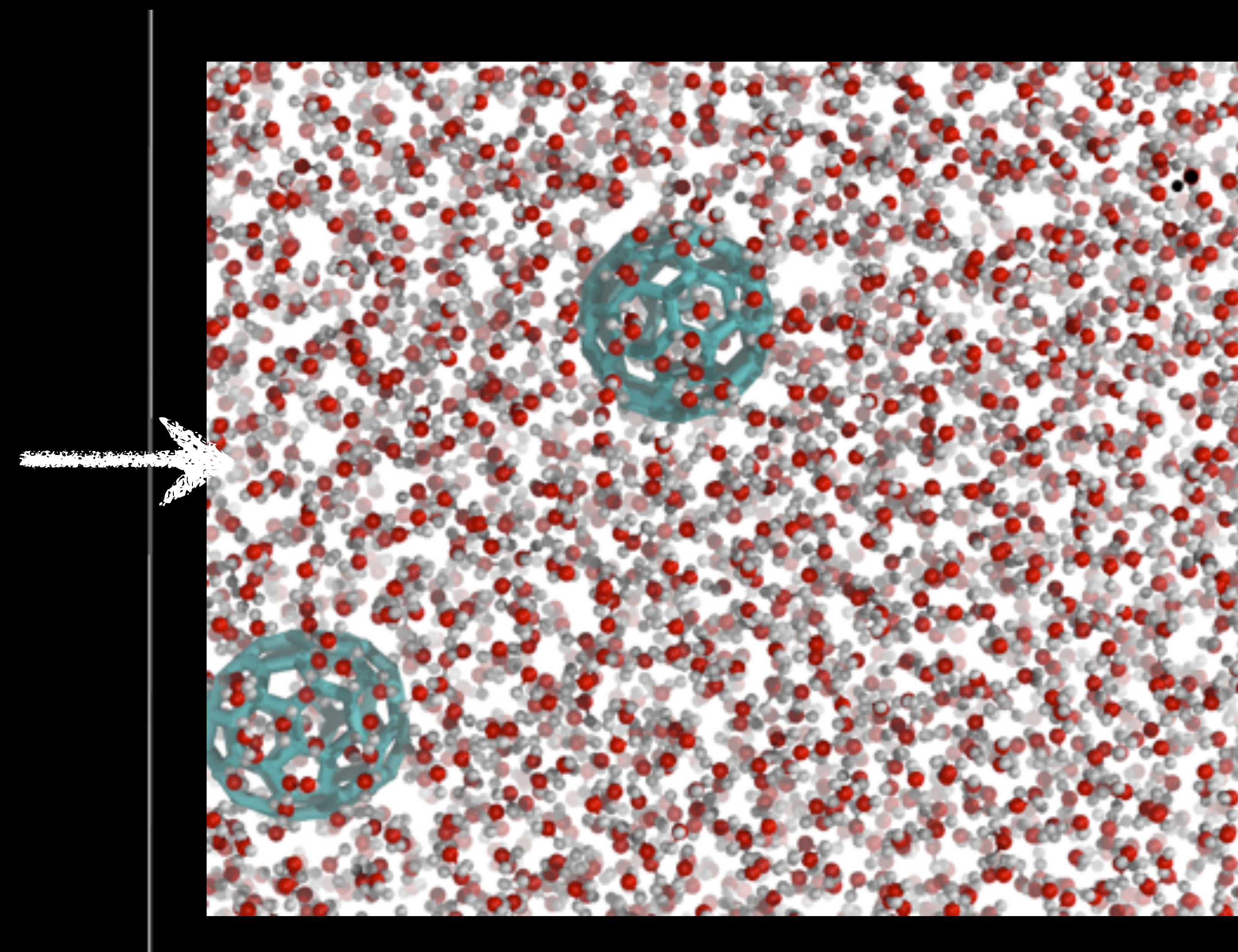


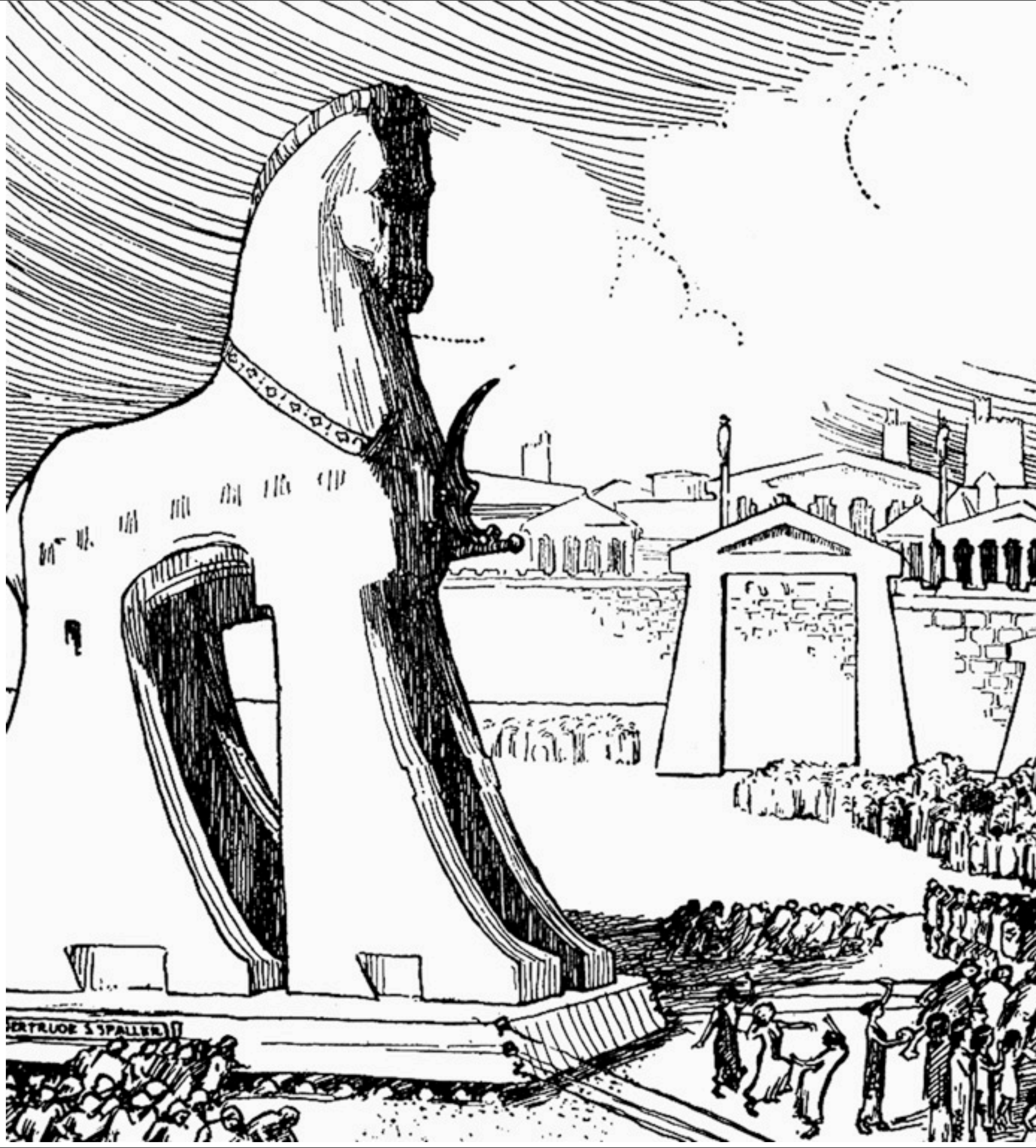
MULTISCALE METHODS +



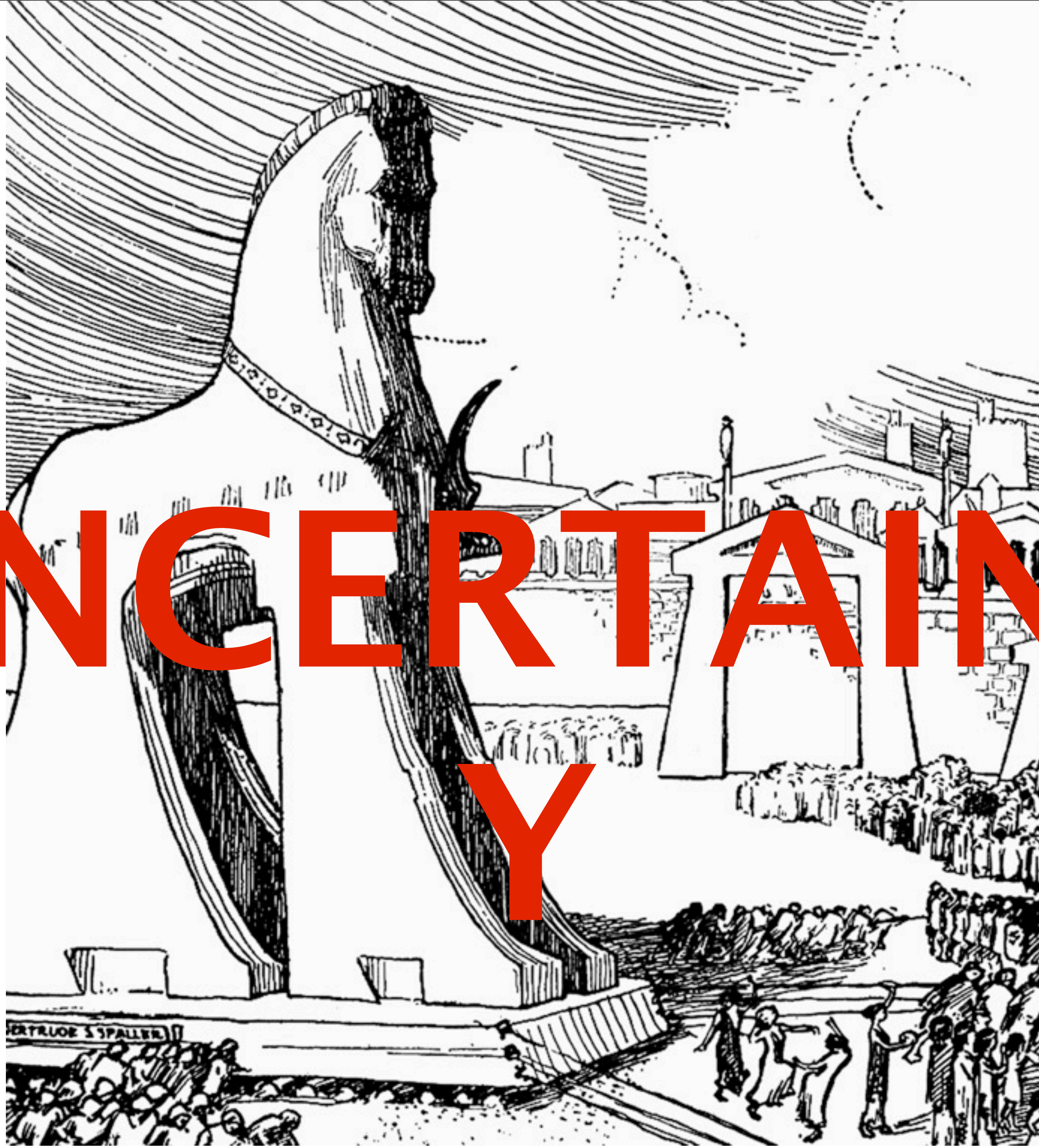
MD - Lattice-Boltzmann





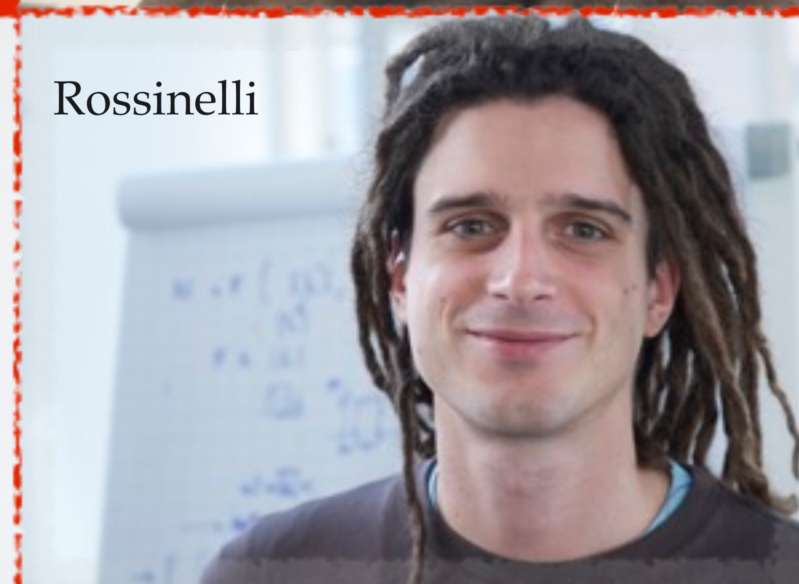

ERTRUDE S SPALLER

UNCERTAINTY




Bergdorf


Walther


Rossinelli


Chatelain


Hedjazialhosseini