Developing a hybrid atomistic-continuum simulation method for biological macromolecules

Sean L. Seyler†, Charles E. Seyler‡, Oliver Beckstein†

†Department of Physics, Center for Biological Physics, Arizona State University
‡Department of Electrical and Computer Engineering, Cornell University

Blue Waters Symposium
May 17, 2017

Equilibrium sampling of biological macromolecules

Need a robust, atomistic approach

- Heterogeneous, nanoscale structures w/ partial charges
- (Ionic) solvent-solute interactions
- Full mechanistic picture requires high spatiotemporal resolution

Molecular mechanics (classical MD)

$$\dot{\mathbf{x}}_i = \mathbf{v}_i(t) \qquad m_i\dot{\mathbf{v}}_i = -\boldsymbol{\nabla}_{\mathbf{x}_i} U\left(\mathbf{x}(t)\right) \qquad U \text{ (all-atom force field)}$$

Sampling cost†

$$\sim \frac{\text{cost}}{\text{step}} \times (\#\text{ steps needed})$$

$$\sim \text{ (cost of energy call)} \times \text{(potential landscape roughness)}$$

† D. M. Zuckerman. (2011) *Annu. Rev. Biophys.* **40**

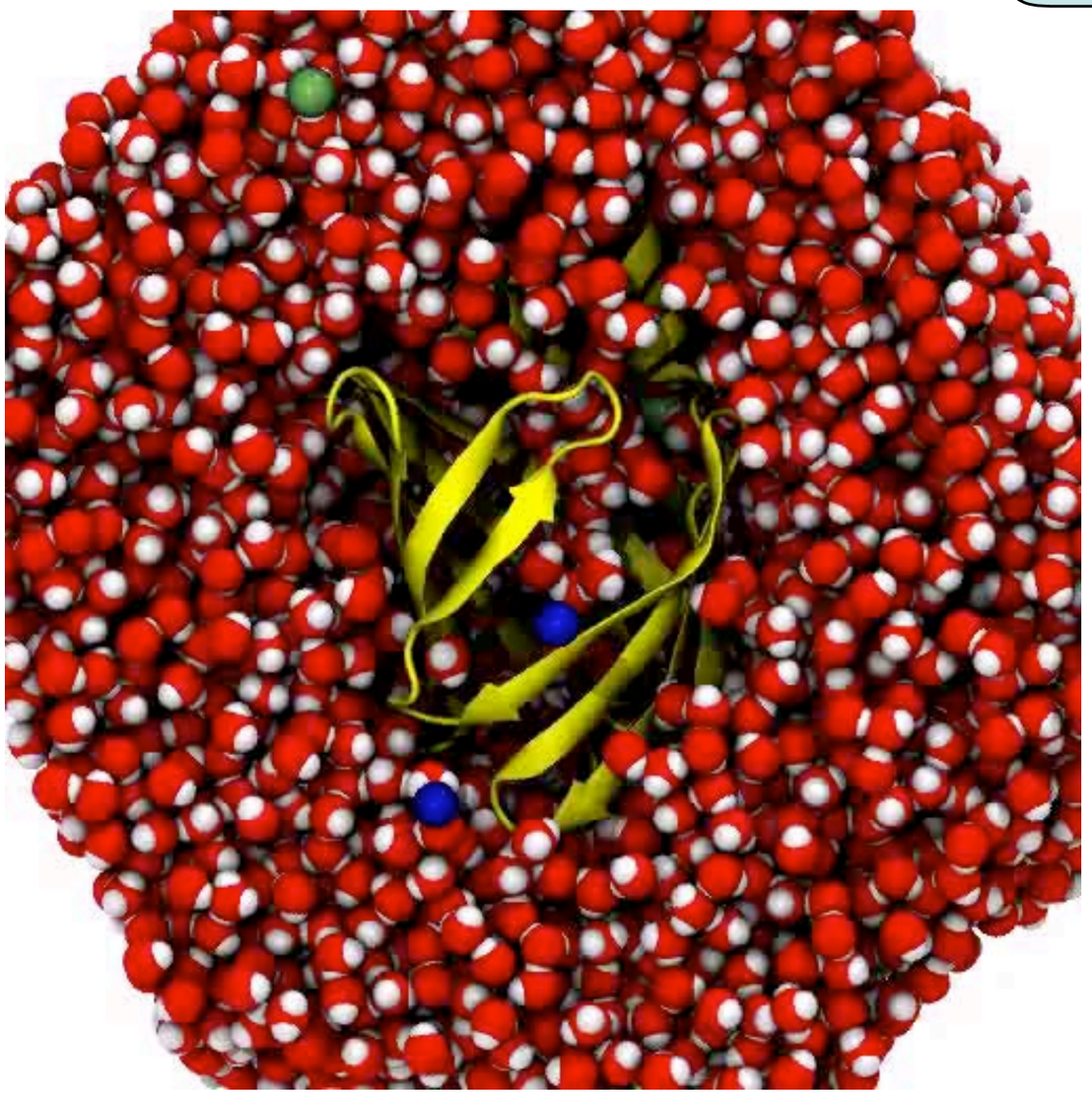
Timesteps:
femtoseconds

Billions to trillions of steps

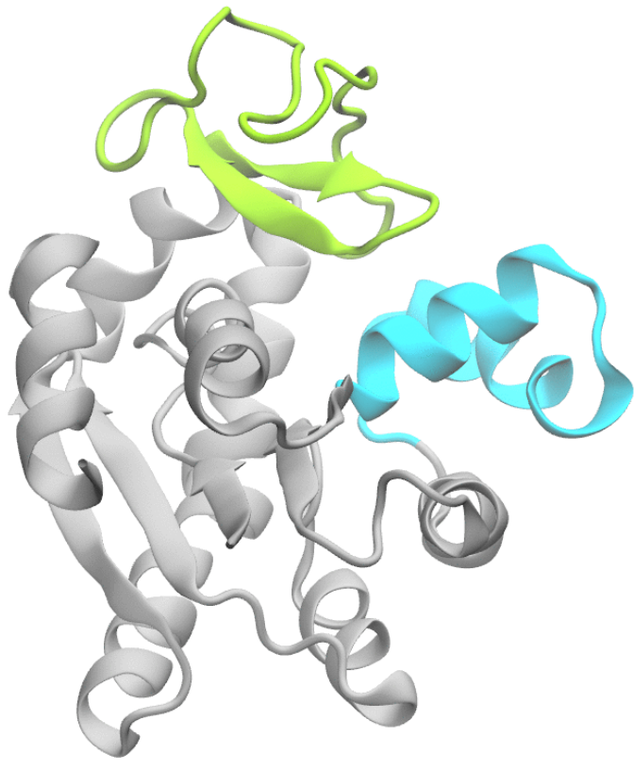
Timescales of interest:
micro- to milliseconds+

- **Explicit solvent** (water)

➡ solvent buffers against electrostatic self-interaction



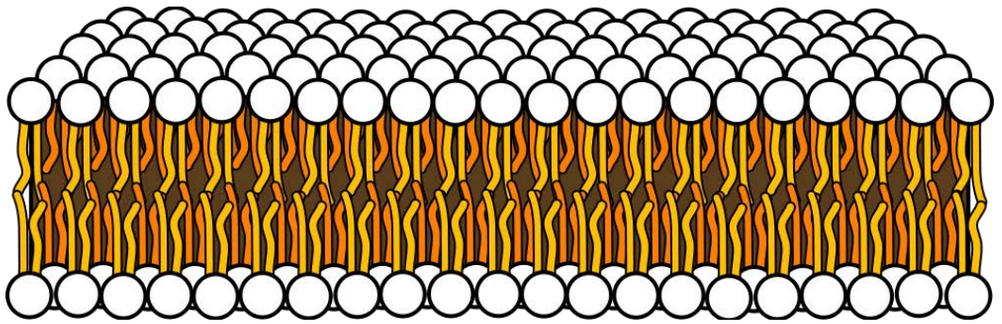
Conformational transitions
- membrane transporters
- ion channel gating
- enzyme/binding kinetics



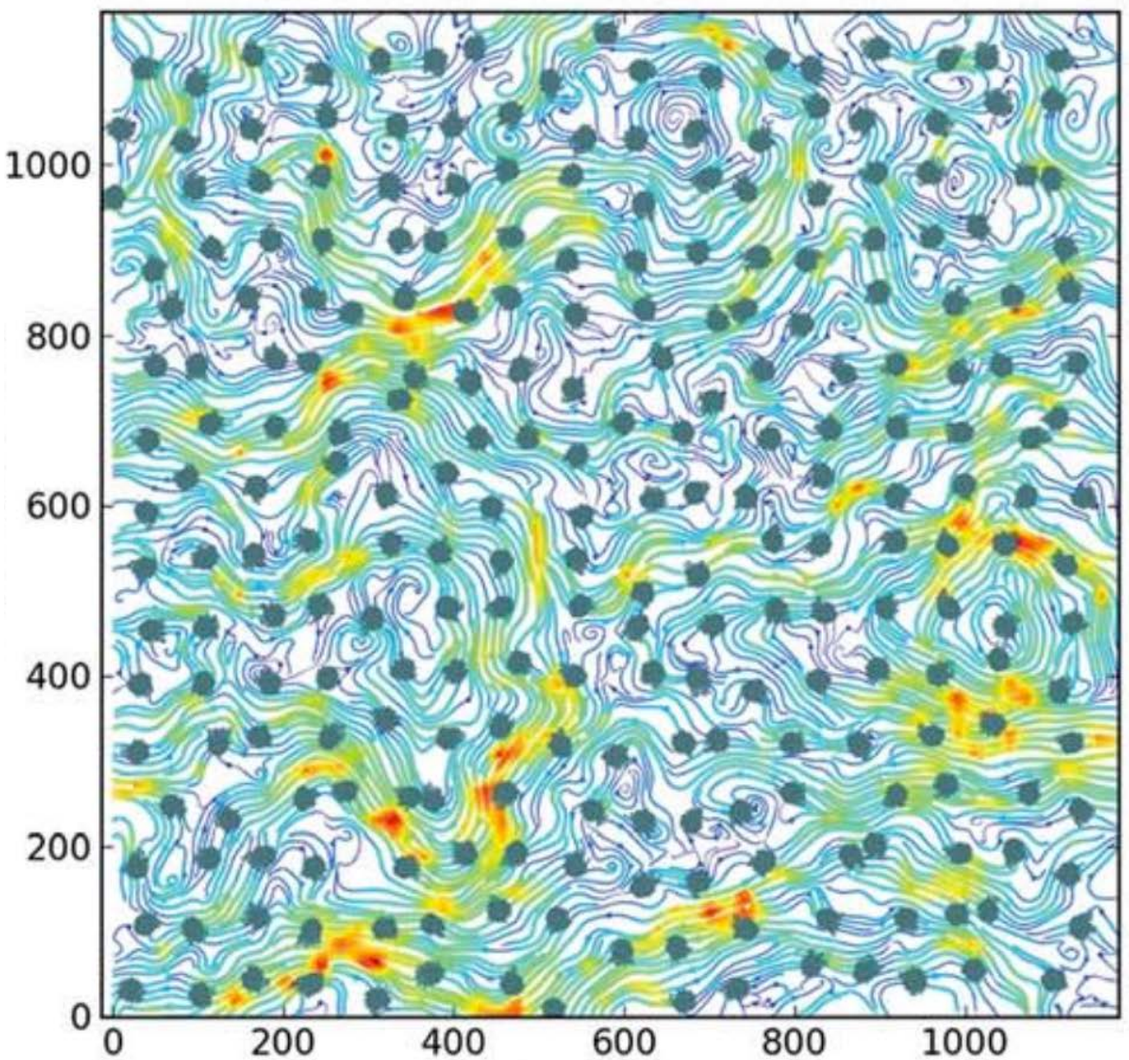
- **Enhanced sampling** (*e.g., targeted MD, metadynamics, replica exchange, …*)

- **Coarse-graining** (*elastic network models, structure-based potentials, …*)
- **Implicit solvation** (*up to 10-100 times cheaper*)

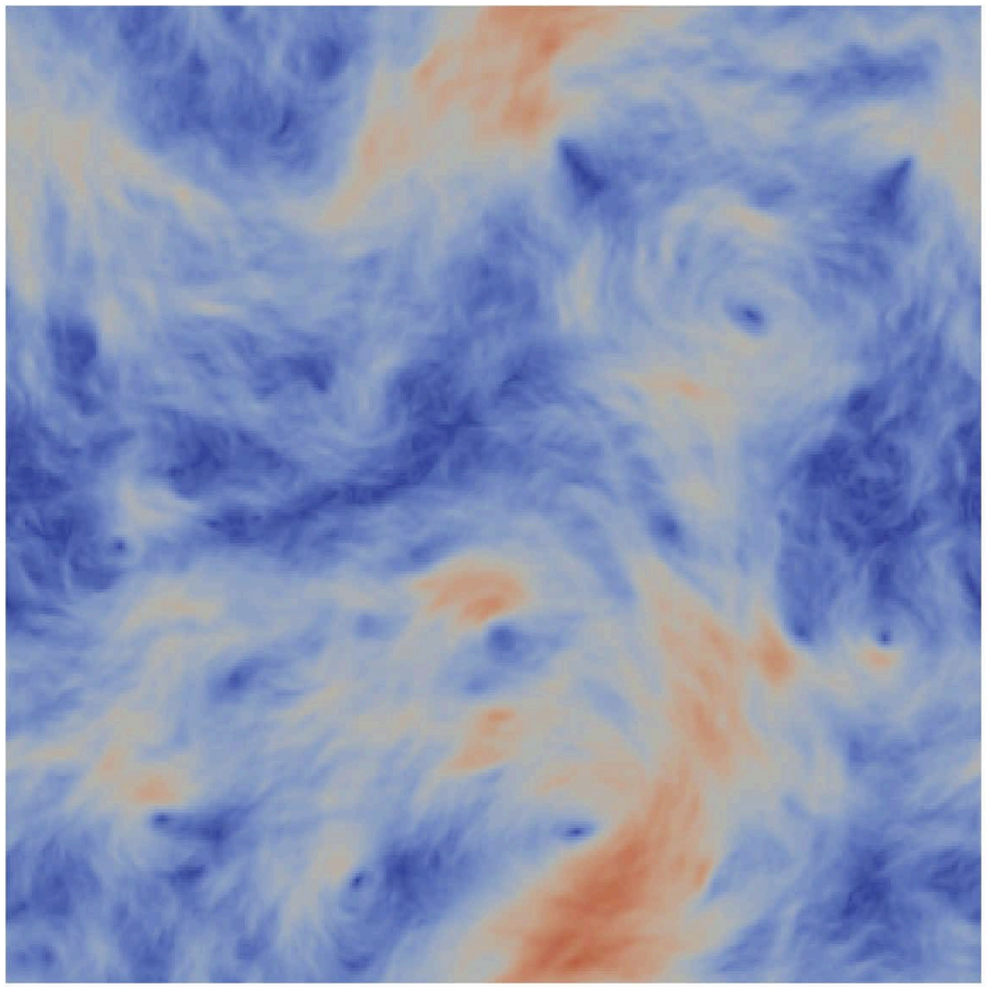
Hydrodynamics in lipid bilayers



Streamlines (velocity field) of a planar lipid



Velocity field (magnitude) for 2D turbulent system



†M. Chavent et al. *Faraday Discussions* **169**, 455 (2014)

Hybrid atomistic-continuum model

Multiscale approach to preserve atomistic detail

- All-atom **MD** in *restricted subdomain*…
- efficient continuum **hydrodynamic** model for *surrounding solvent*

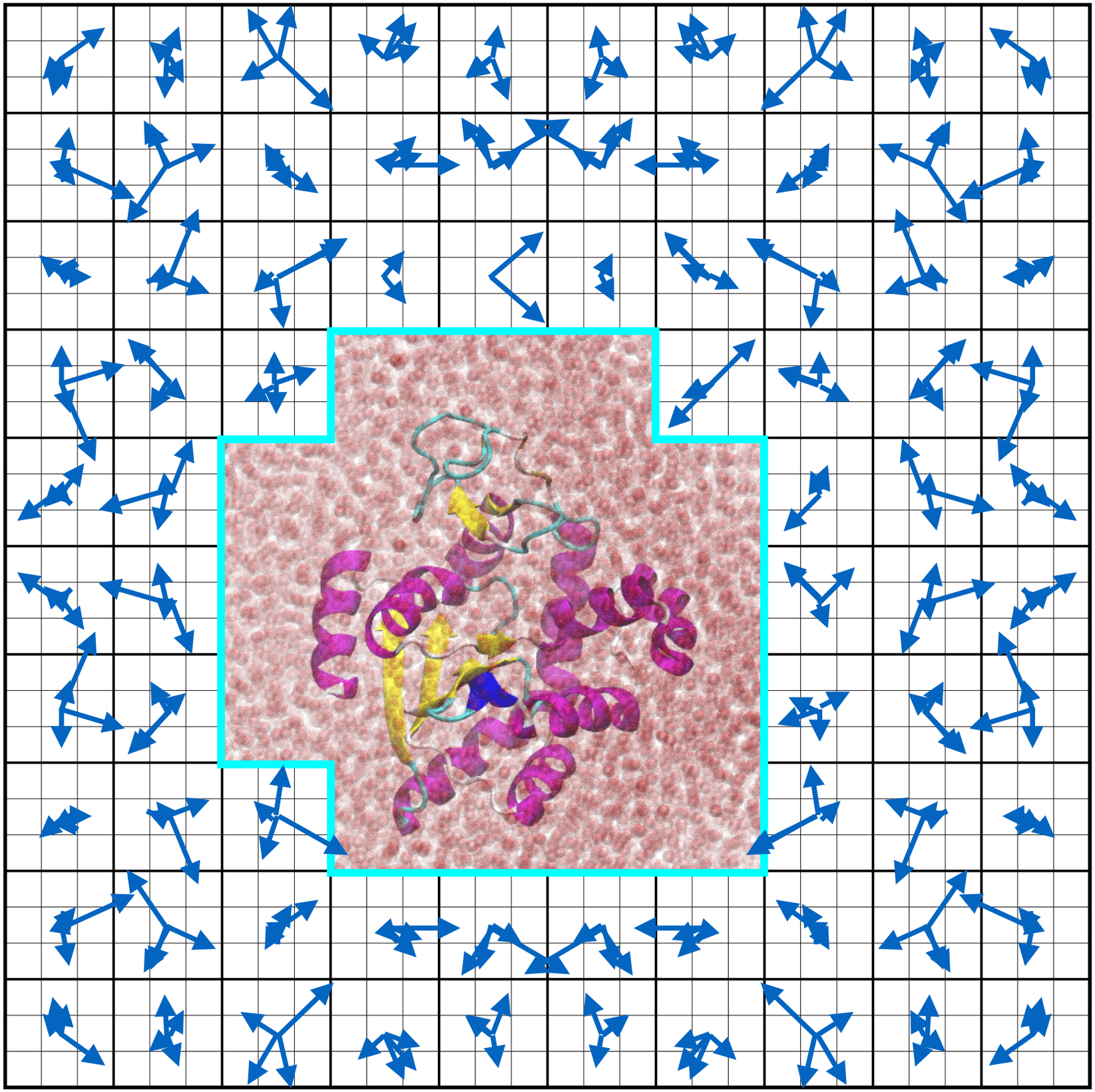
Direct (continuum-solute) coupling

Continuum-solvent coupling

Hybrid approaches are *general*

- E.g.,
  - **DSMC** (vs MD)
  - **Lattice Boltzmann** (vs Navier-Stokes)

The basic idea: less explicit solvent



Basic components

Standard MD codes

- Need: MD engine that supports *biomolecular force fields*
  - open-source
  - interface (**Python**) for communicating w/ external code
- Choice: LAMMPS*‡

3D fluctuating hydrodynamics code

- Need: 3D FVM-like hydro solver for compressible, dense fluids
- Choice: custom discontinuous Galerkin (DG) code†
  - open-source
  - interface (**Python**) for communicating w/ external code

Umbrella (driver) code

- **Python**

† X. Zhao, Y. Yang, & C. E. Seyler. (2014) *J. Comput. Phys.* **278**

* S-H. Ko, et al. (2014) *J. Mech. Sci. Technol.* **28**

‡ F. E. Mackay, et al. (2013) *Comput. Phys. Commun.* **184**

Hydrodynamic equation in conservation form

$$\partial_t \phi + \boldsymbol{\nabla} \cdot (\phi \mathbf{u}) = S(\phi)$$

Mass

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \mathbf{u}) = 0$$

Account for *thermal fluctuations* in **stress** and heat flux (not shown)†

Momentum

$$\partial_t (\rho \mathbf{u}) + \boldsymbol{\nabla} \cdot \left(\rho \mathbf{u}\mathbf{u} + p\vec{\vec{I}} + \vec{\vec{\sigma}} + \cancel{\vec{\vec{S}}} \right) = 0$$

Energy

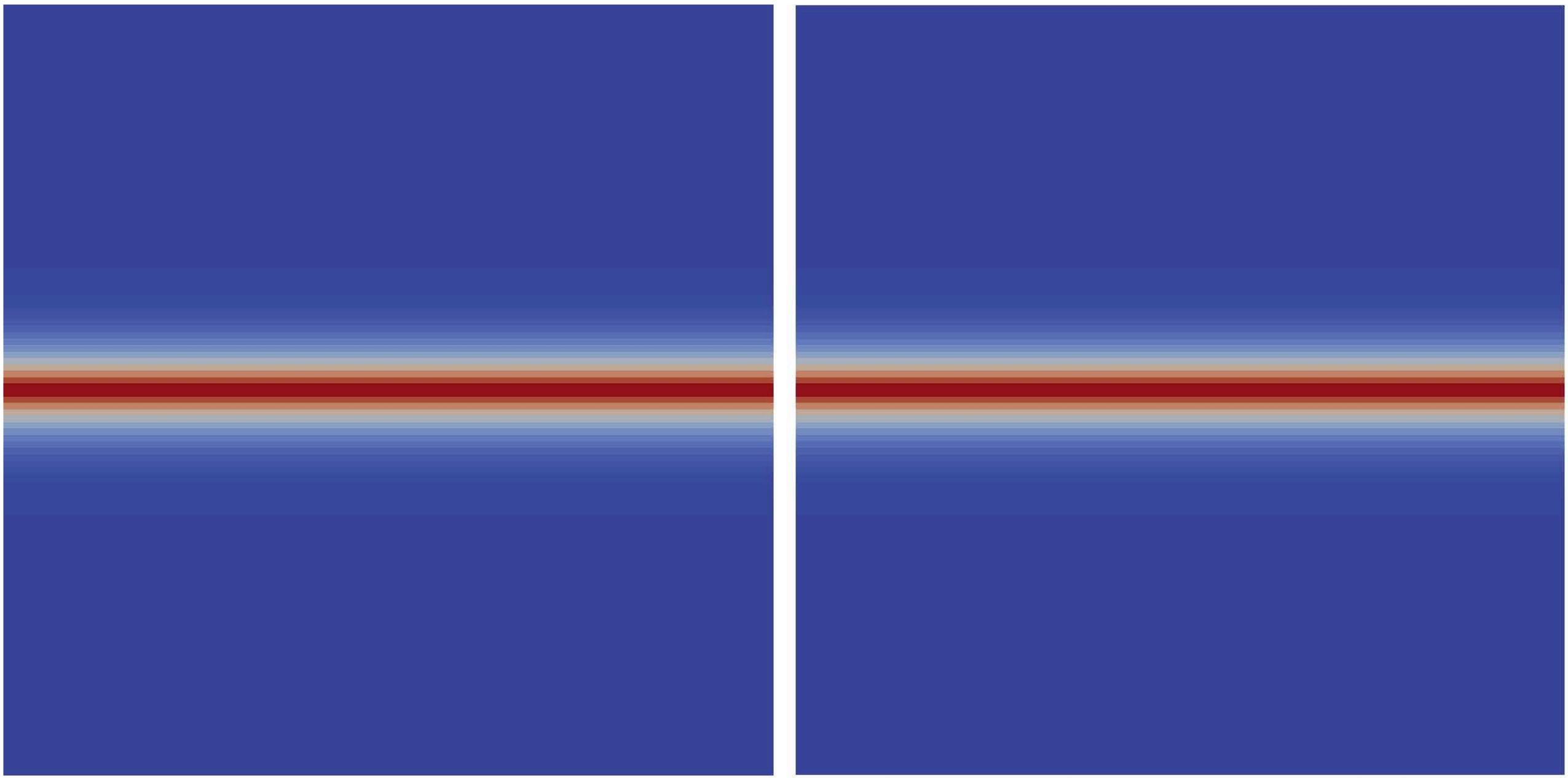
$$\partial_t \mathcal{E} + \boldsymbol{\nabla} \cdot \left[\mathbf{u}(\mathcal{E} + p) + \mathbf{u} \cdot \left(\vec{\vec{\sigma}} + \cancel{\vec{\mathcal{S}}} \right) \right] = 0$$

~~Stress tensor = Navier-Stokes~~ Linearized 10-moment equations

$$\partial_t \vec{\vec{\sigma}} + p_0 \left(\left[\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^\mathsf{T} - \frac{2}{3}(\boldsymbol{\nabla} \cdot \mathbf{u})\vec{\vec{I}} \right] \right) = -\nu \left(\vec{\vec{\sigma}} + \vec{\mathcal{S}} \right)$$

† L. D. Landau & E. M. Lifschitz. (1959) *Fluid Mechanics.* **6**
‡ H. Grad. (1949) *Comm. Pure Appl. Math.* **2**, 331–407

2D nanojet: with and without fluctuations

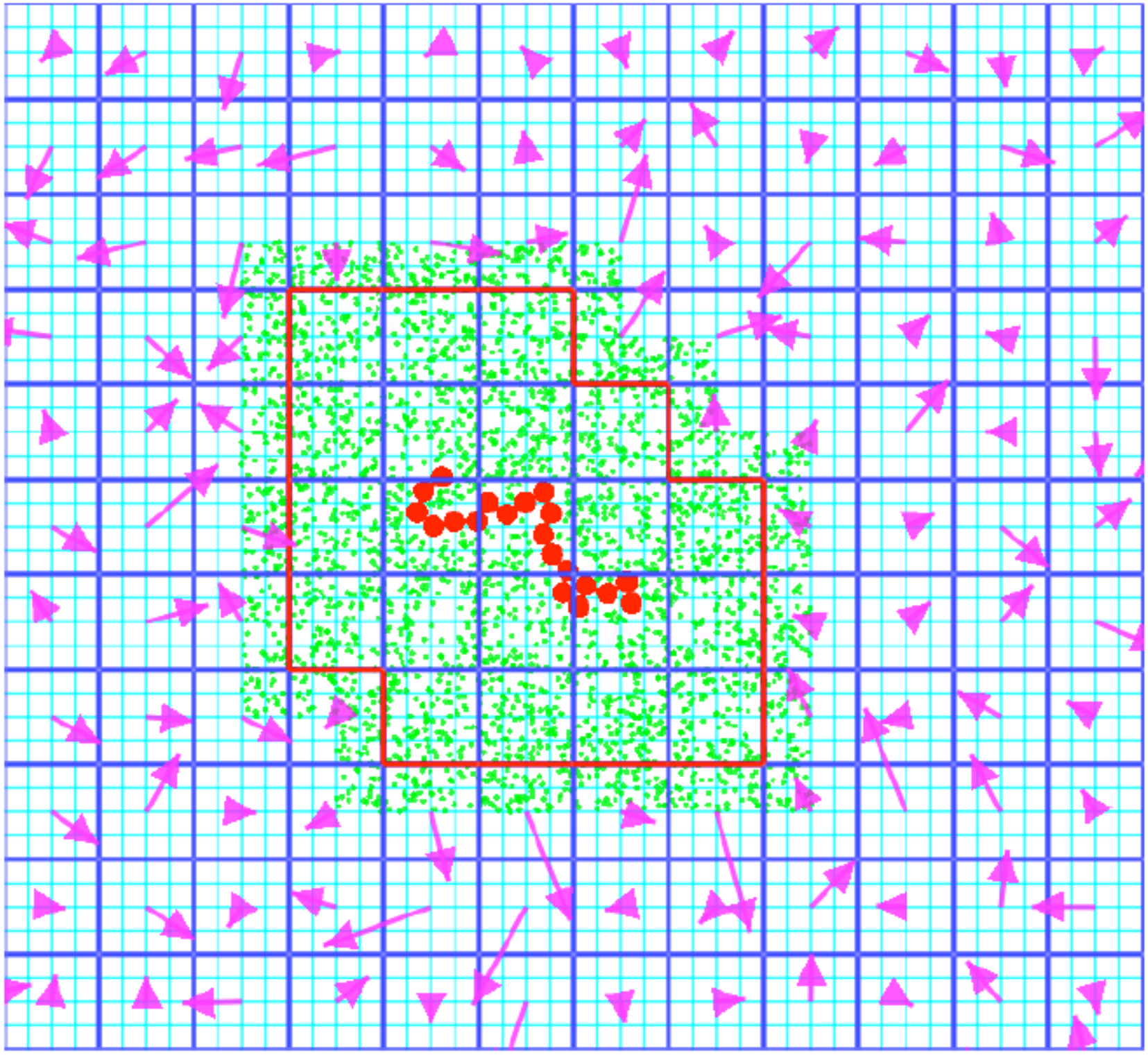


Navier-Stokes

fluctuating hydrodynamics

jet velocity

Coupling via reservoir/buffer region



A. Donev et al. *J. Multiscale Model. Simul.* **8** (2010)

Putting the pieces together

Dynamical coupling

- Communication via “reservoir” particles in overlap region

State variables

$$\rho/\mathbf{x}_i \quad \mathbf{u}/\mathbf{p}_i \quad \mathcal{E}/P/T$$

Fluxes

$$\rho\mathbf{u} \qquad \rho\mathbf{u}\mathbf{u} + p\vec{\vec{I}} + \vec{\vec{\sigma}}$$

$$\mathbf{u}(\mathcal{E} + p) + \mathbf{u} \cdot \vec{\vec{\sigma}}$$

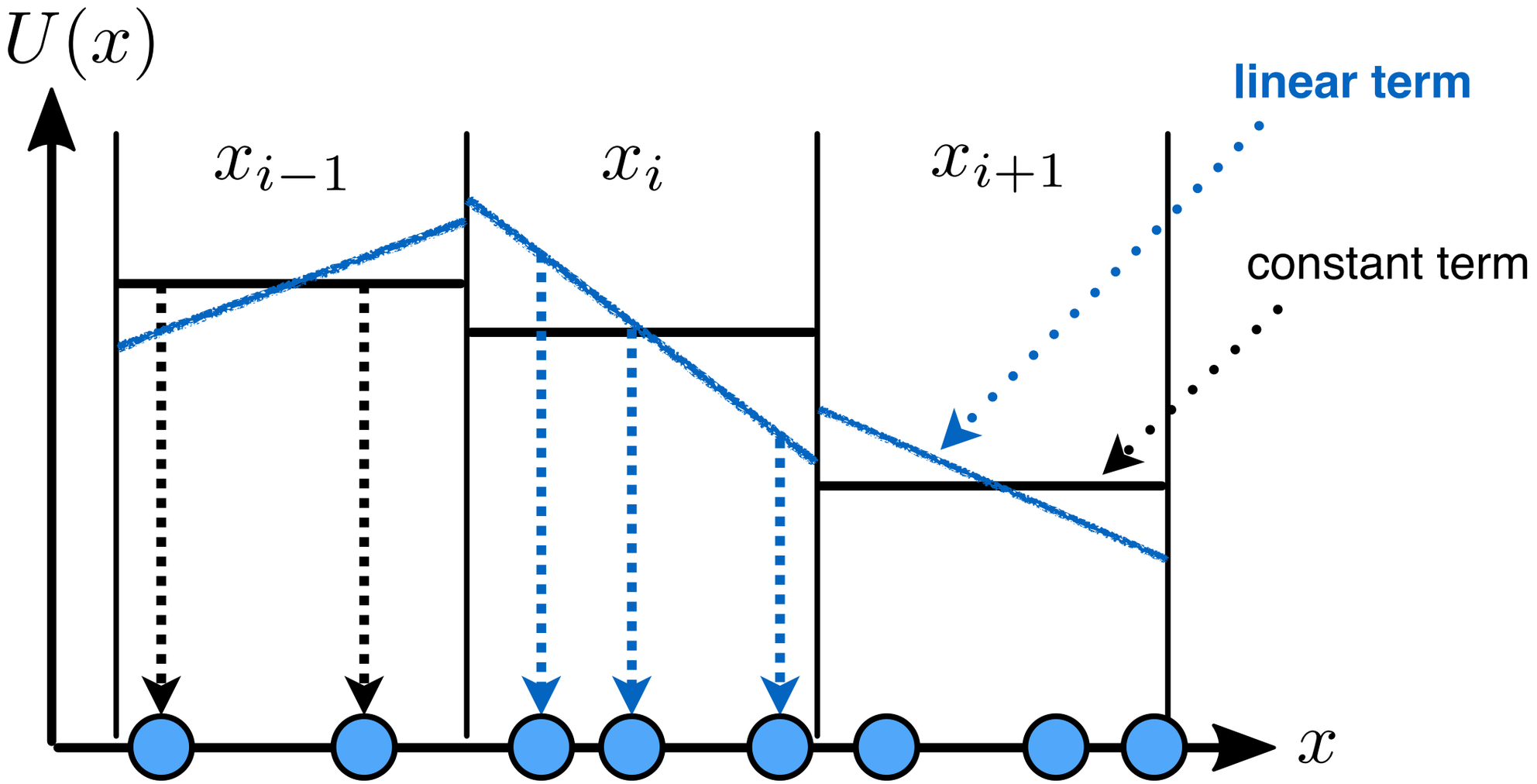
Inter-domain boundary conditions

$\Omega_A \rightarrow \Omega_C$ Simple. Take averages over microscopic variables

$\Omega_C \rightarrow \Omega_A$ Complex. Need mapping from fields variables to atomic coordinates

K. M. Mohamed & A. A. Mohamad. (2010) *Nanofluidics* **8**

Hybrid simulation with DG FHD



- With DG:
  - ➡ natural interpolation from fields to particles
  - ➡ spatial resolution decoupled from fluctuations
- In numerical FHD, grid cell size choice:
  - ➡ sets effective observation scale
  - ➡ determines size of fluctuations

MD

Build topology
Set ICs
Equilibrate

Start HAC simulation

Build mesh
Set fluid properties
Set ICs

FHD

Compute total energies/forces on atoms

Integrate by dt_{MD}

Apply BCs

No

$n = N_{MD}$?

Yes

Get atom data from *reservoir*: (e.g., **x**, **v**, **F**)

End

Yes

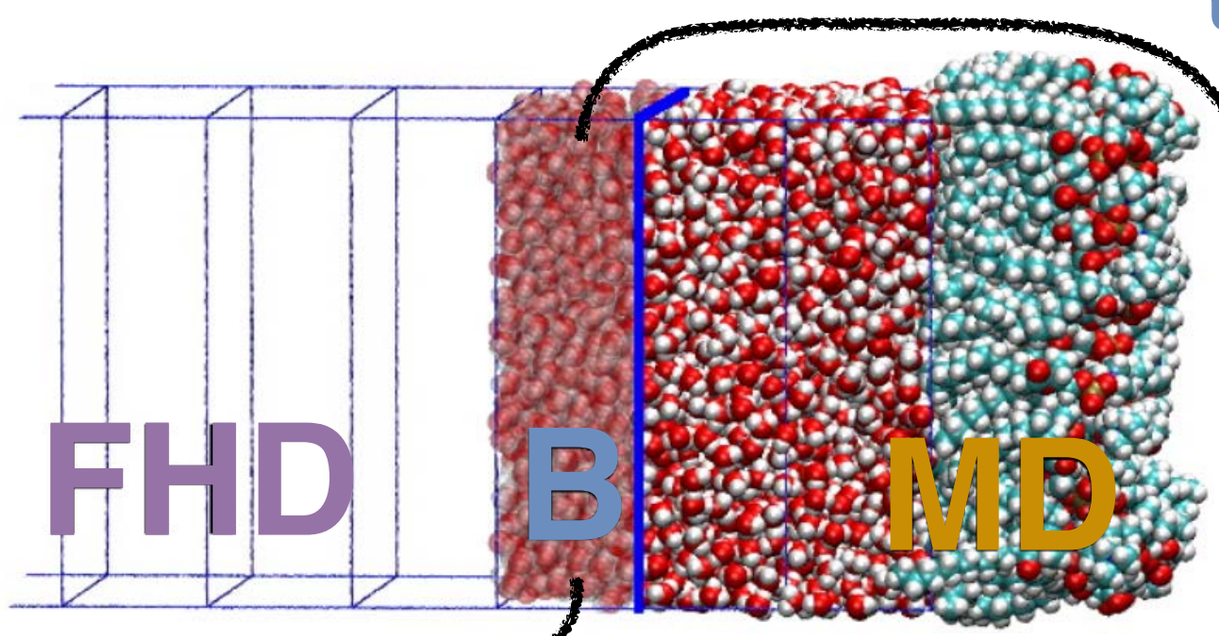
$t > t_{final}$?

No

Apply BCs

Compute fluxes/sources

Integrate by dt_{FHD}



Get fluid data from *reservoir*: (e.g., ρ, **u**, T, p/σ)

To MD

Compute forces on *fluid* due to *atoms*

Compute forces on *atoms* due to *fluid*

To FHD

G. De Fabritiis, et al. (2006) *PRL* **97**

Project status

HERMESHD

- Hyperbolic Equations and Relaxation Model for Extended Systems of HydroDynamics
- Modular code, callable as library via **Python**
- Open source on GitHub (soon)!

Hybrid simulation

- LAMMPS: called as a library using a Python wrapper
- Driver **calls** HERMES/LAMMPS via **Python** interface, **manages** communication

Why Blue Waters?

- HAC simulation test problems $\Longrightarrow$ comparison with pure MD/FHD
- Get physics of coupling right (e.g., EOS from MD for various water models[‡])

Acknowledgements

Oliver Beckstein

Julio Candanedo

Mahzad Khoshlessan

Anton Kononenko

David L. Dotson

Taylor Colburn

Ian Kenney

Kacey Clark

Charles E. Seyler (Cornell University)

```python
from mpi4py import MPI
import numpy as np
from hermeshd import hermeshd

nx, ny, nz = 80, 80, 1  # 80 x 80 x 1 grid
nq, nB = 11, 8          # 11 eqns (10-moment), 8 basis funcs
t  = np.array(0.00, dtype=float)    # current time
dt = np.array(0.01, dtype=float)    # time step
tf = np.array(10.0, dtype=float)    # final time
dtout = np.array(0.0, dtype=float)  # output frequency

# data arrays for fluid variables: Qio is primary array
Qio = np.empty((nx,ny,nz,nQ,nB), order='F', dtype=np.float32)
Q1  = np.empty_like(Qio)  # temp array
Q2  = np.empty_like(Qio)  # temp array

hermeshd.setup(Qio, t, dt, dtout, MPI.COMM_WORLD.py2f())
while ( t < tf ):
    hermeshd.step(Qio, Q1, Q2, t, dt)
    hermeshd.output(Qio, t, dt, dtout)
hermeshd.cleanup()
```

Run from command line: `mpirun -n 16 python test.py`

f90wrap. http://www.jrkermode.co.uk/f90wrap
P. Peterson. (2009) *Int. J. Comput. Sci. Eng.* **4**, 296–305.

The Ten Moment Equations (Grad's Approach)

$$\partial_t \rho + \partial_{x_k}(\rho u_k) = 0$$

Continuity equation

$$\partial_t(\rho u_i) + \partial_{x_k}\mathcal{E}_{ik} = 0$$

Momentum equation

$$\partial_t \mathcal{E}_{ij} + \partial_{x_k}(\rho u_i u_j u_k + u_i P_{jk} + u_j P_{ki} + u_k P_{ij}) = -\nu S_{ij}$$

Energy equation

$$\mathcal{E}_{ij} = \rho u_i u_j + P_{ij}$$

Total energy tensor

$$P_{ij} = \delta_{ij} p + S_{ij}$$

Pressure tensor

scalar pressure

deviatoric stress tensor

Collision frequency → $\nu = p/\mu$ ← dynamic viscosity

Ten Moment Equations

$$\partial_t \rho + \partial_{x_k}(\rho u_k) = 0 \qquad \partial_t(\rho u_i) + \partial_{x_k}\mathcal{E}_{ik} = 0$$

$$\partial_t \mathcal{E}_{ij} + \partial_{x_k}(\rho u_i u_j u_k + u_i P_{jk} + u_j P_{ki} + u_k P_{ij}) = -\nu S_{ij}$$

$$\mathcal{E}_{ij} = \rho u_i u_j + P_{ij} \qquad P_{ij} = \delta_{ij} p + S_{ij}$$

Component form

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0 \qquad \partial_t(\rho \mathbf{u}) + \nabla \cdot \vec{\vec{\mathcal{E}}} = 0$$

$$\partial_t \vec{\vec{\mathcal{E}}} + \nabla \cdot \left(\rho \mathbf{uuu} + 3\,\mathbf{Sym}\{\mathbf{u}\vec{\vec{P}}\} \right) = -\nu \vec{\vec{S}}$$

$$\vec{\vec{\mathcal{E}}} = \rho \mathbf{uu} + \vec{\vec{P}} \qquad \vec{\vec{P}} = \vec{\vec{I}}\, p + \vec{\vec{S}}$$

Vector Form

Navier-Stokes equations (neglecting heat flow)

$$\partial_t \rho + \partial_{x_k}(\rho u_k) = 0$$

$$\partial_t(\rho u_i) + \partial_{x_k}\left(\rho u_i u_k + \delta_{ik} p + S_{ik}\right) = 0$$

$$\partial_t \mathcal{E} + \partial_{x_k}\left[u_k\left(\mathcal{E} + p\right)\right] = 0$$

$$S_{ij} = -\mu\left(\partial_{x_i} u_j + \partial_{x_j} u_i - \frac{2}{3}\delta_{ij}\partial_{x_k} u_k\right)$$

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\partial_t(\rho \mathbf{u}) + \nabla \cdot \left(\rho \mathbf{u}\mathbf{u} + \vec{\vec{I}} p + \vec{\vec{S}}\right) = 0$$

$$\partial_t \mathcal{E} + \nabla \cdot \left[\mathbf{u}\left(\mathcal{E} + p\right)\right] = 0$$

$$\vec{\vec{S}} = -\mu\left(2\,\mathbf{Sym}\{\nabla \mathbf{u}\} - \frac{2}{3}\vec{\vec{I}}\nabla \cdot \mathbf{u}\right)$$

Component form

Vector form

$$\mathcal{E} = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2$$

Equation for pressure