

Complex fluids in the GPU era

Algorithms and simulations

Raúl P. Peláez

Universidad Autónoma de Madrid

May 9, 2022

Supervisor: Rafael Delgado-Buscalioni



Universidad Autónoma
de Madrid

- My work during these past years has been devoted to exploiting the GPU for the simulation of complex fluids.
- This is a GPU. A powerful hardware with a unique programming model.
- Harvesting its tremendous raw power requires developing algorithms and software specifically tailored for it.
- Now, before we start, let me give you a summary of this talk.

Talk outline

- 1 Introduction
- 2 Elements of a complex fluid simulation
- 3 New doubly periodic solvers
- 4 UAMMD
- 5 Conclusions

2022-05-29

Complex fluids in the GPU era

- └ Introduction
- └ Talk outline

- I will start by introducing what complex fluids are, with an overview of the numerical and mathematical machinery we use to simulate them in a computer.
- Then I will talk about the main algorithmic hardships that arise in this field and I will focus on how to solve some of them with a GPU.
- Part of my contributions consist on the developing of new GPU-enabled algorithms for hydrodynamics and electrostatics in confined geometries, which I will also introduce.
- The main contribution of my work lies in UAMMD, a GPU software framework for complex fluids that I built from scratch during these years and that includes, among others, all the algorithms and solvers that I will mention in this talk.
- I will then end with some conclusions.

Talk outline

- └ Introduction
- └ Elements of a complex fluid simulation
- └ New doubly periodic solvers
- └ UAMMD
- └ Conclusions

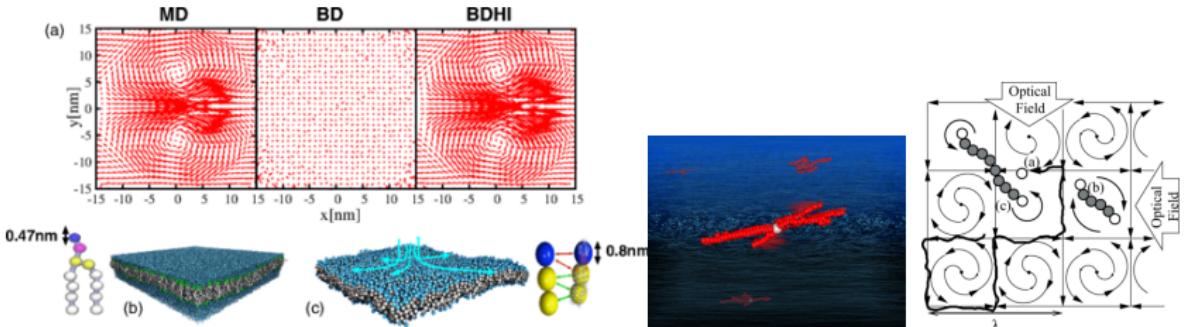
Raúl P. Peláez

Complex fluids in the GPU era

2 / 46

Complex fluids

The coexistence between a liquid and solid phase



[Panzuela, S. and Delgado-Buscalioni, R. PRL 2018.] - [Raul P. Pelaez and Delgado-Buscalioni, R. Macromolecules 2020] - [Meléndez, M. et. al. PRE 2019.]



[Courtesy of Pablo Ibañez]

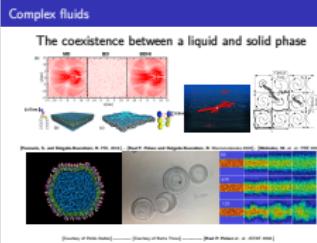
[Courtesy of Berta Tinao]

[Raul P. Pelaez et. al. JSTAT 2018.]

Complex fluids in the GPU era

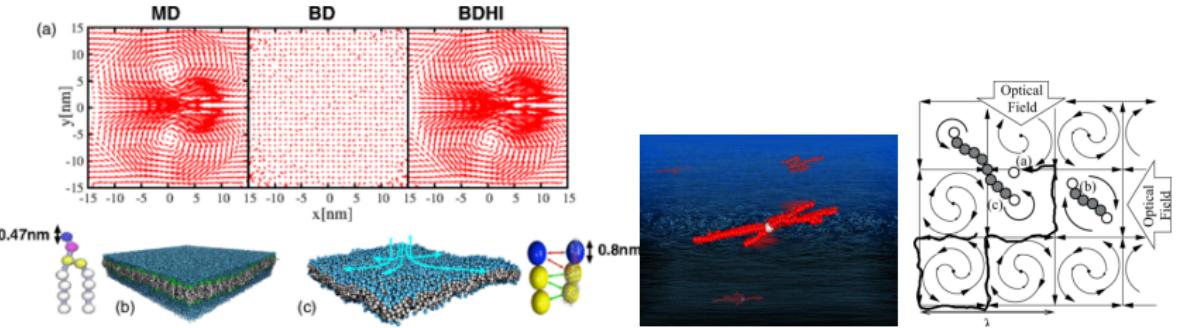
- └ Introduction
- └ Complex fluids
- └ Complex fluids

2022-05-29

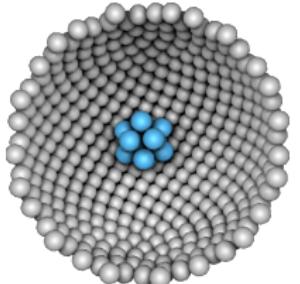
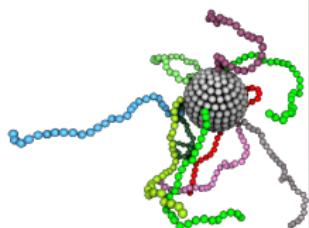


- I showed you what a GPU is at the start. In order to understand the rest of the title you also need to know what a complex fluid is.
- Although its definition is a little more general, we understand a complex fluids as the coexistence between a solid and a liquid phase. And note that with solid, we usually mean soft, such as a colloidal particle or a cell's lipidic membrane.
- This broad definition encloses a wide variety of systems, many times of biological nature. In this slide I show you a bunch of examples, all of them taken from past and ongoing works in our group.
- Starting up we can see coarse-grained representations of a lipid bilayer, a star polymer under shear flow, a nano dumbell swimming and spinning in a fluid under optically-induced vortexes.
- A virus capsid, filled with a particular protein, which is being pressed by an AFM tip. Next we have a group of lipidic vesicles, one of them contains a ferromagnetic particle subjected to a magnetic field, spinning the vesicle. This allows to somewhat control the locomotion of the vesicles. The final picture shows the diffusion of the concentration of a colloidal suspension under strong two-dimensional confinement in different regimes.

The coexistence between a liquid and solid phase



[Panzuela, S. and Delgado-Buscalioni, R. PRL 2018.] - [Raul P. Pelaez and Delgado-Buscalioni, R. Macromolecules 2020] - [Meléndez, M. et. al. PRE 2019.]



[Courtesy of Pablo Palacios]

— [Courtesy of Berta Tinao]

— [Courtesy of Pablo Palacios]

- Here, the figure to the right shows a model of the rotating vesicles a member of group, Pablo, is studying now, in part using the tools and algorithms that I will shortly introduce.
 - Pablo has also studied other kinds of magnetic nanoparticles.
 - These examples showcase enormous span of the spatio-temporal scales associated with complex fluids. Sometimes we need to focus on single polymers, or describe a single protein in detail. Other times we want to model vesicles that are almost visible to the naked eye.

Complex fluids

The spatio-temporal landscape of numerical techniques.

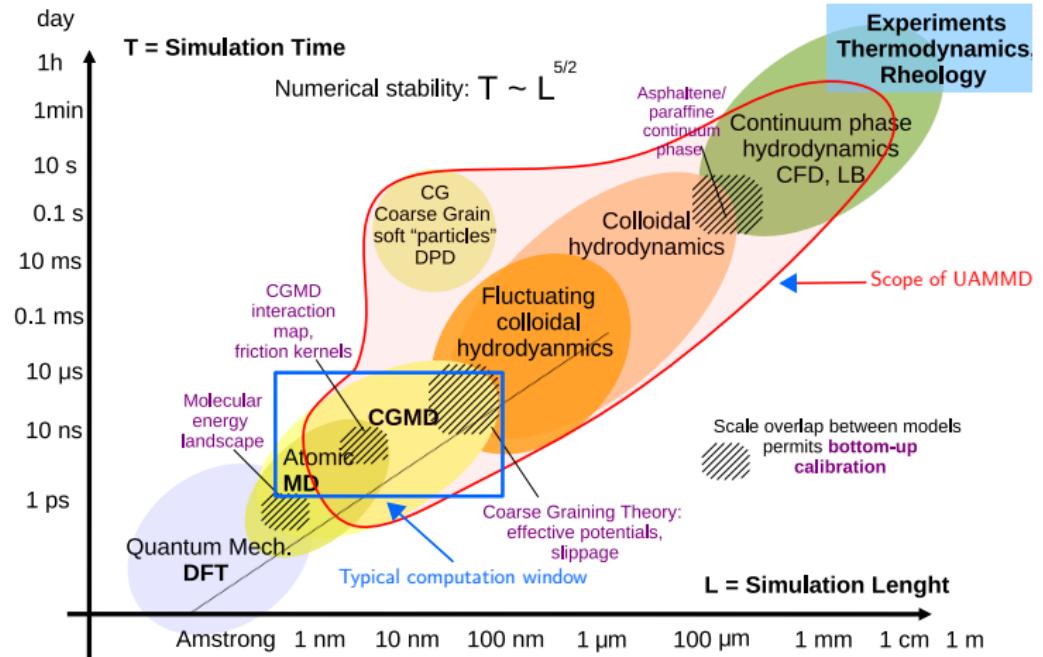


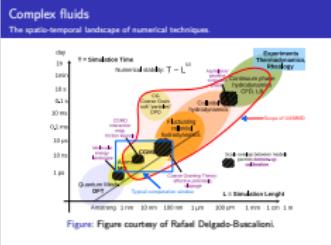
Figure: Figure courtesy of Rafael Delgado-Buscalioni.

Complex fluids in the GPU era

└─ Introduction

Complex fluids

└ Complex fluids



- The modeling and simulation of complex fluids thus greatly exceeds the scope of the typical computation window. Speaking in terms of this numerical landscape we cannot specialize in a single window. If we want to model a complex fluid, we must take into account sizes ranging from nanometers to meters and times going from nanoseconds to seconds. And even then, we are restricted to a narrow-ish area of the landscape, being unable to, for instance, simulate a nanometer-sized system for several hours of physical time.
 - I claim that my software, UAMMD, covers the whole range of the complex-fluid domain, and I hope to convince you during this talk that this is true.
 - Anyway, countless numerical and mathematical techniques exists depending on our window of interest, and their naming scheme can sometimes be a little obtuse.

Complex fluids

The spatio-temporal landscape of numerical techniques.

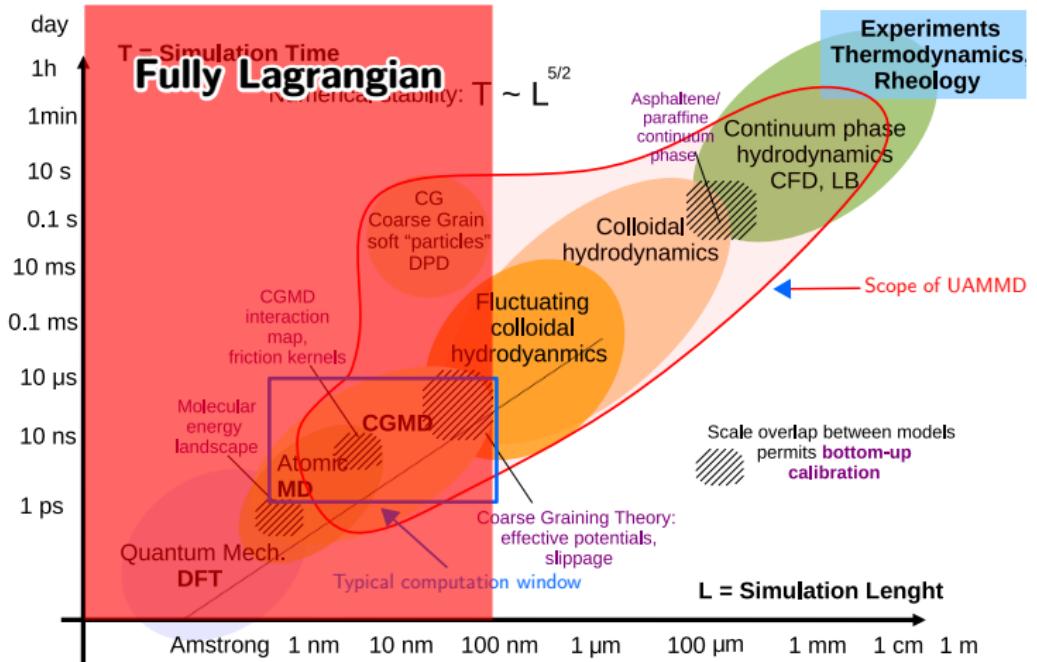
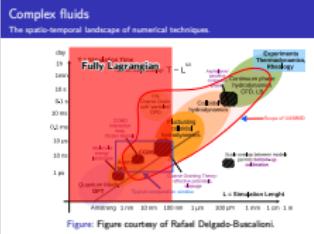


Figure: Figure courtesy of Rafael Delgado-Buscalioni.

Complex fluids in the GPU era

└ Introduction
└ Complex fluids
└ Complex fluids

2022-05-29



- Broadly speaking, we can categorize the landscape into three families of methods.
- When studying small systems, we usually choose a fully Lagrangian scheme, meaning that we track the evolution of individual particles, or markers.

Complex fluids

The spatio-temporal landscape of numerical techniques.

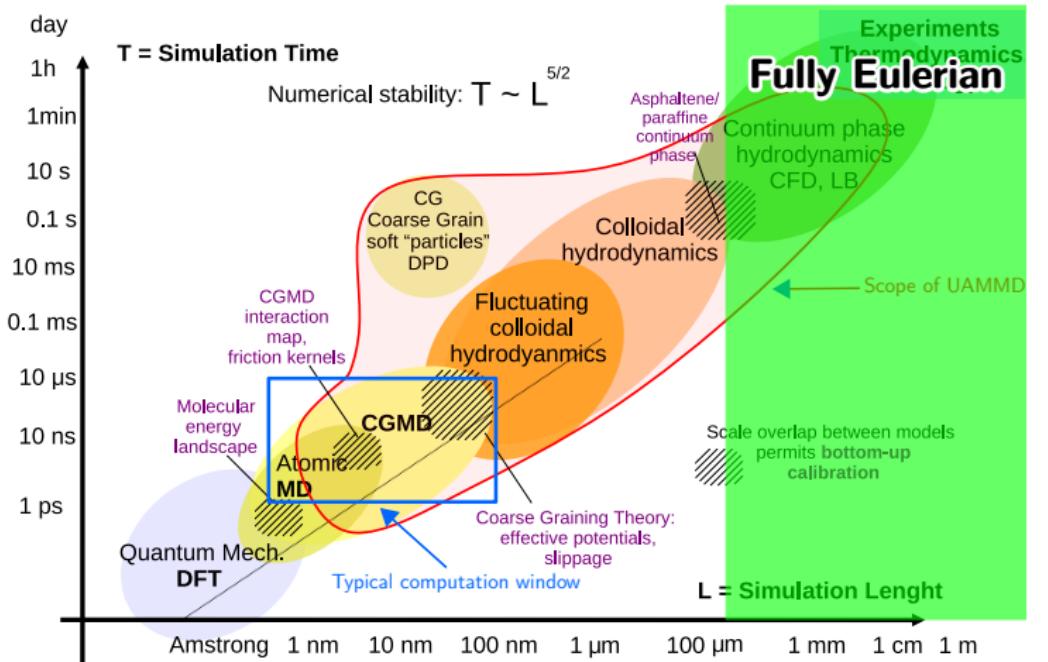


Figure: Figure courtesy of Rafael Delgado-Buscalioni.

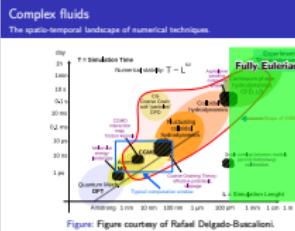
Complex fluids in the GPU era

—Introduction

└ Complex fluids

└ Complex fluids

- On the contrary, when focusing on macroscopic systems we usually choose a fully Eulerian description, where we loose track of individual particles and instead describe continuous fields.



Complex fluids

The spatio-temporal landscape of numerical techniques.

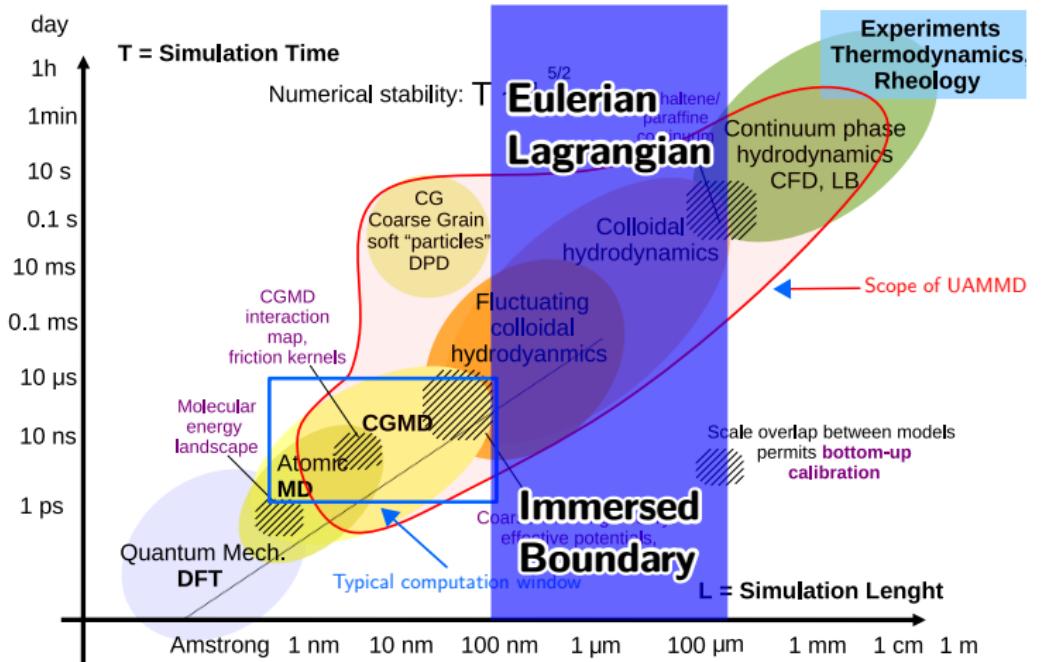
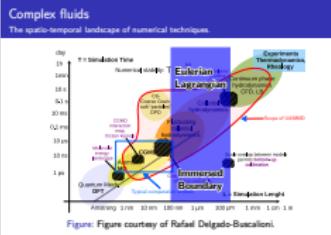


Figure: Figure courtesy of Rafael Delgado-Buscalioni.

Complex fluids in the GPU era

└ Introduction
└ Complex fluids
└ Complex fluids

2022-05-29



- In the middle, in what we call the mesoscale, there is a mixed domain of Eulerian-Lagrangian methods, where we describe both a group of particles and continuous fields that are somehow coupled.
- The Immersed boundary method is the canonical strategy used to describe this coupling.
- This is the realm of hydrodynamics, where most of the examples I showed you take place.
- I would like to point out now that my algorithmic contributions lie in this regime.

Complex fluids

The spatio-temporal landscape of numerical techniques.

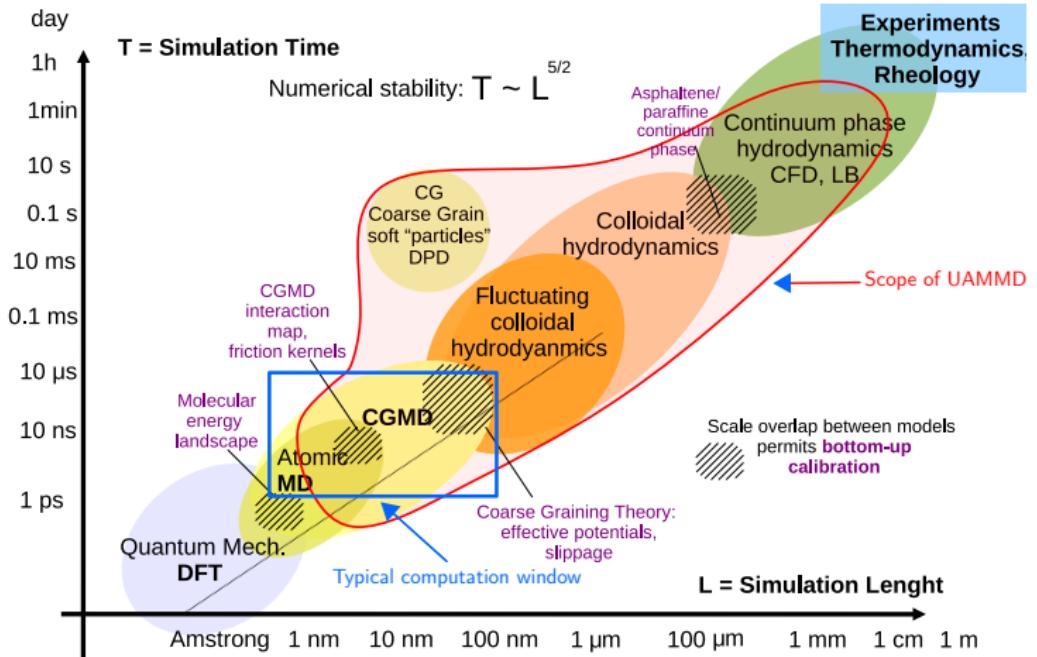
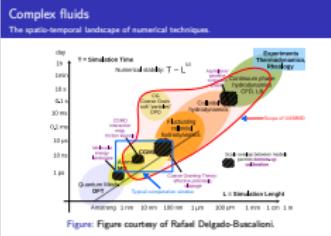


Figure: Figure courtesy of Rafael Delgado-Buscalioni.

Complex fluids in the GPU era

└ Introduction
└ Complex fluids
└ Complex fluids

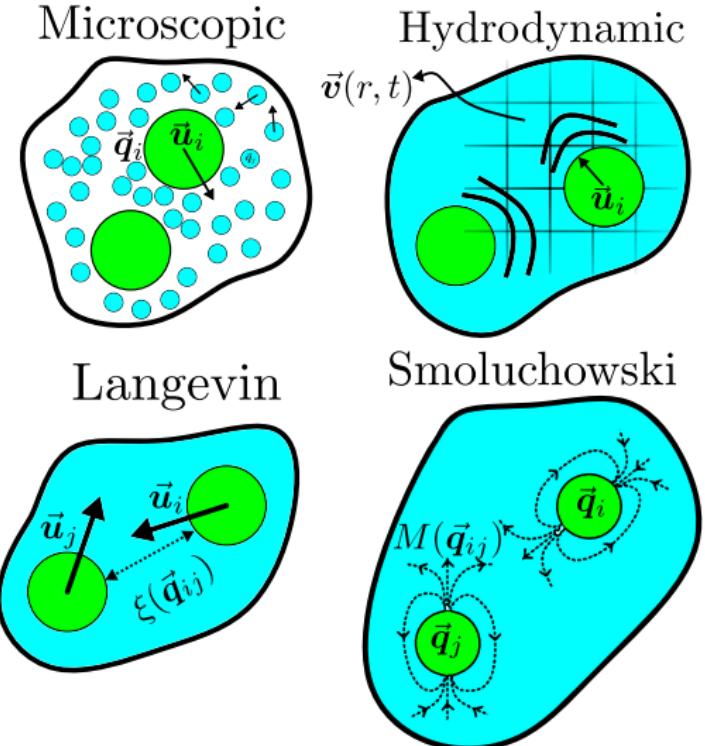
2022-05-29



- Let us focus in the red area, covering an, in principle, disparate range of descriptions all related to complex fluids.
- If we start at the atomic scale and want to climb up the spatio-temporal scale of modeling we can make use of coarse graining techniques, where we want to somehow average out the fast degrees-of-freedom that are of no interest to us. For instance, if we are describing a submarine running under the sea (by all means a complex fluid) we really want to leave the individual water atoms out of our description.

Complex fluids

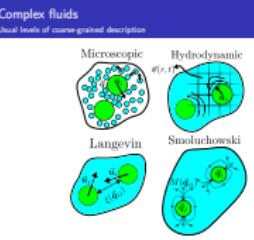
Usual levels of coarse-grained description



Complex fluids in the GPU era

- Introduction
- Complex fluids
 - Complex fluids

2022-05-29

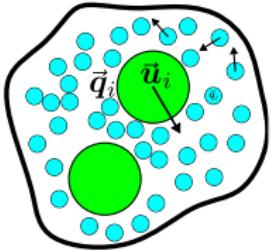


In particular, we can cover the whole range using the following four levels of coarse-grained description. Going through them will help us understand how we can design a framework that covers them all.

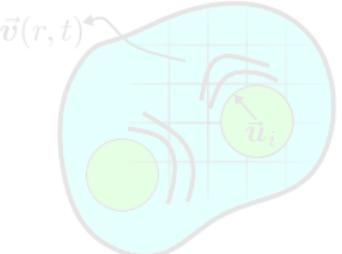
Complex fluids

Usual levels of coarse-grained description

Microscopic



Hydrodynamic



Langevin

Smoluchowski

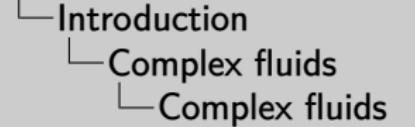
Molecular Dynamics (MD)

$$m\ddot{q} = F$$
$$\dot{u} = \dot{q}$$

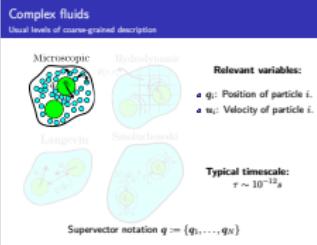
Typical timescale:
 $\tau \sim 10^{-12} s$

Supervector notation $q := \{q_1, \dots, q_N\}$

Complex fluids in the GPU era



2022-05-29



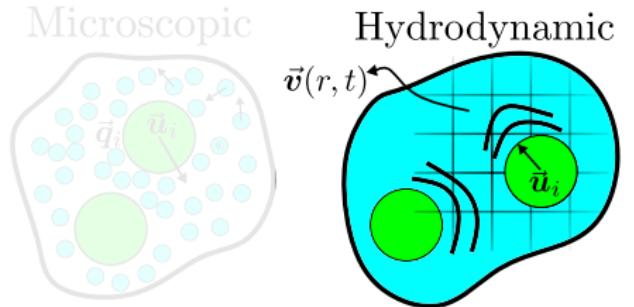
Relevant variables:

- q_i : Position of particle i .
- u_i : Velocity of particle i .

- At the lowest level (the most detailed one) we find the microscopic level, where we need to describe the positions and velocities of every atom (or small group of atoms) involved in the system. This includes every water molecule conforming the fluid in addition to the particles embedded in it. The Newton equations of motion govern this level, where the inter-particle forces are of atomic origin. Notice that this is a fully Lagrangian description.
- The typical timescale is given by the fast-moving fluid particles.
- BTW, note that I am using the so-called supervector notation here, where a symbol without a subindex often refers to the list of all related quantities. As opposed to, for instance, q_i , which refers to a specific particle called "i". I will abuse this notation throughout this talk, so beware!.

Complex fluids

Usual levels of coarse-grained description



Fluctuating Hydrodynamics

$$\rho \partial_t \mathbf{v} = -\boldsymbol{\partial}_r \cdot \boldsymbol{\sigma} + \mathbf{f} + \text{fluct}$$

$$\int_{V_p} \mathbf{f} dr = \mathbf{F}_i \quad \text{and} \quad \mathbf{u}_i = \int_{V_p} \mathbf{v} dr$$

$$\partial_t := \frac{\partial}{\partial t} \rightarrow \boldsymbol{\partial}_r := \nabla := (\partial_x, \partial_y, \partial_z)$$

Relevant variables:

- \mathbf{q}_i : Position of particle i .
- \mathbf{u}_i : Velocity of particle i .
- $\mathbf{v}(r, t)$: Fluid velocity field.

Typical timescale:

$$\tau \sim 10^{-\{9-6\}} s$$

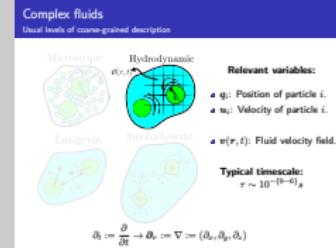
Complex fluids in the GPU era

Introduction

Complex fluids

Complex fluids

2022-05-29



- Going up we have the hydrodynamic level, where the individual degrees of freedom of the fluid particles are lost and we are left, instead, with a continuous velocity field representing them. The math tells us that when we do this, we must reintroduce the lost degrees-of-freedom as thermal fluctuations.
- Thus this is an Eulerian-Lagrangian formalism and is governed by the Navier-Stokes equations describing the fluid. A fluid which exists in a two-way coupling with a group of embedded particles.
- This coupling is far from trivial, but in essence it consists of the particles communicating their forces to the fluid via a force density and, in return, the particles following the fluid by sampling its surrounding velocity. V_p here is the volume of a particle (whatever that means).
- The typical timescale is given by the speed of sound of the fluid as well as its vorticity. In other words, how fast it communicates information.

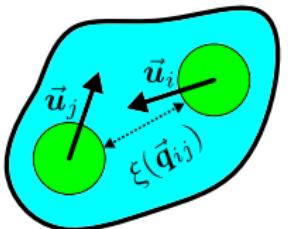
Complex fluids

Usual levels of coarse-grained description

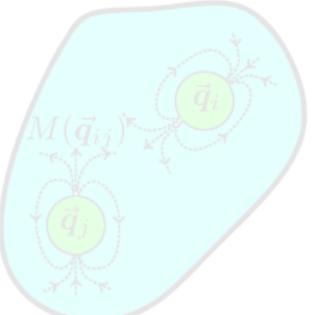
Langevin Dynamics (LD)

$$mdu = \mathbf{F}dt - \xi u dt + \sqrt{2\xi k_B T} \tilde{\mathbf{W}}$$

Langevin



Smoluchowski



$\tilde{\mathbf{W}}$: Wiener increments, $\mathcal{N}(0, dt)$

Relevant variables:

- \mathbf{q}_i : Position of particle i .
- \mathbf{u}_i : Velocity of particle i .
- $\xi(\mathbf{q}_{ij})$: Friction kernel.

Typical timescale:

$\tau \sim 10^{-5} s$

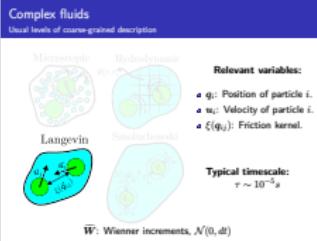
Complex fluids in the GPU era

Introduction

Complex fluids

Complex fluids

2022-05-29



- When particles move slowly compared to the solvent characteristic times, hydrodynamic interactions, which were previously mediated by an explicit velocity field, are effectively instantaneous. We call this the Langevin level.
- We eliminate the fluid velocity, transforming it into a friction kernel (and of course, fluctuations). Typically, this friction kernel will be defined pairwise, although it could simply be a constant, or depend on the entire particle configuration.
- Only the positions and momenta of the particles remain as relevant variables (taking us back to a Lagrangian description once again), which obey a Langevin equation.
- The characteristic decorrelation time of the particle's velocities governs this scale.

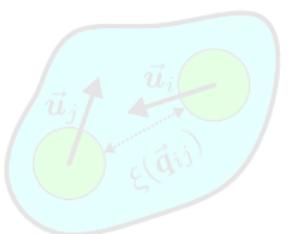
Complex fluids

Usual levels of coarse-grained description

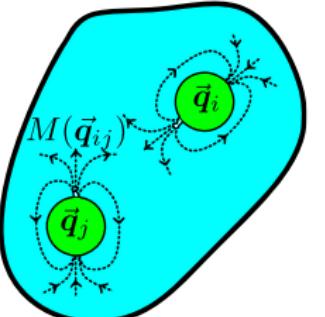
Brownian Dynamics (BD)

$$dq = \mathcal{M}F dt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

Langevin



Smoluchowski



$d\tilde{W}$: Wiener increments, $\mathcal{N}(0, dt)$

Relevant variables:

- q_i : Position of particle i .

- $\mathcal{M}(q_{ij})$: Mobility tensor.

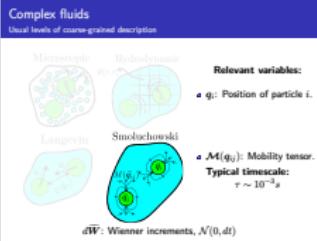
Typical timescale:

$$\tau \sim 10^{-3} s$$

Complex fluids in the GPU era

- └ Introduction
 - └ Complex fluids
 - └ Complex fluids

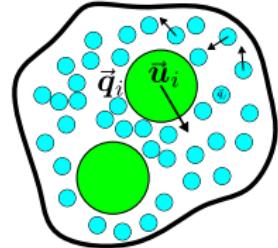
2022-05-29



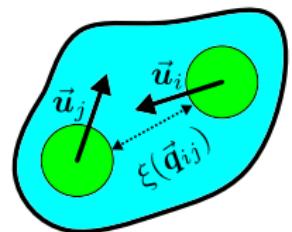
Complex fluids

Usual levels of coarse-grained description

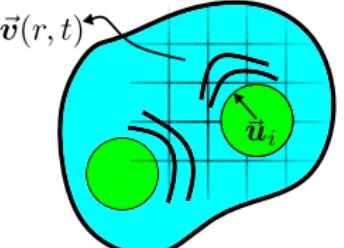
Microscopic



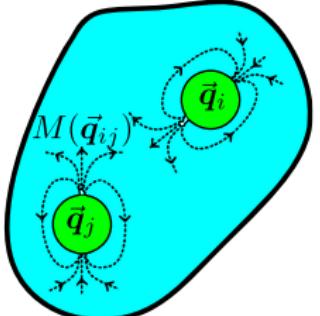
Langevin



Hydrodynamic



Smoluchowski



Relevant variables:

- \mathbf{q}_i : Position of particle i .
- \mathbf{u}_i : Velocity of particle i .
- $\xi(\mathbf{q}_{ij})$: Friction kernel.
- $\mathbf{v}(\mathbf{r}, t)$: Fluid velocity field.
- $\mathcal{M}(\mathbf{q}_{ij})$: Mobility tensor.

Timescale range:

$$\tau \sim [10^{-12}, 10^{-3}]s$$

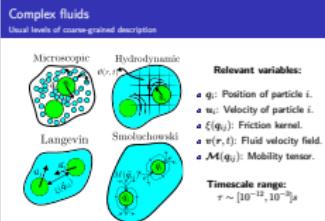
Complex fluids in the GPU era

Introduction

Complex fluids

Complex fluids

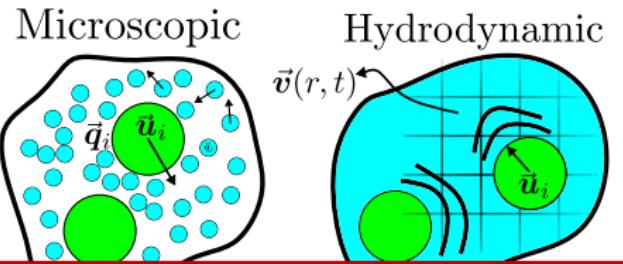
2022-05-29



- Let me show you again all the levels of description, which overall allows us to describe a really wide timescale range.
- Now, there is a not-so-hidden theme here that has allowed me to design a software framework encasing all of this. Have you noticed it?

Complex fluids

Usual levels of coarse-grained description



Relevant variables:

- \vec{q}_i : Position of particle i .
- \vec{u}_i : Velocity of particle i .

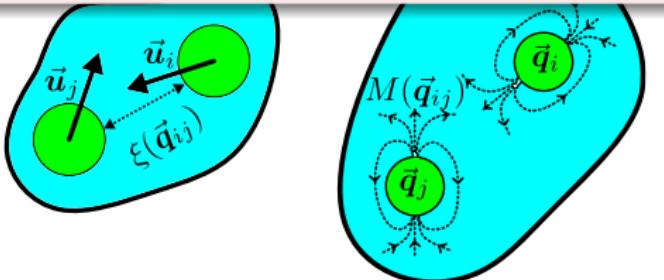
We always have

Interacting *particles* with a *state* that evolves.

- $\mathcal{M}(\vec{q}_{ij})$: Mobility tensor.

Timescale range:

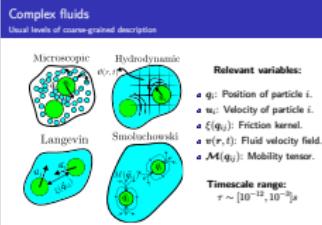
$$\tau \sim [10^{-12}, 10^{-3}]s$$



Complex fluids in the GPU era

└ Introduction
└ Complex fluids
└ Complex fluids

2022-05-29



Talk outline

1 Introduction

2 Elements of a complex fluid simulation

- Short range
- Long range
- Particle-grid coupling

3 New doubly periodic solvers

4 UAMMD

5 Conclusions

Complex fluids in the GPU era

Elements of a complex fluid simulation

Talk outline

2022-05-29

Talk outline

Introduction

- Elements of a complex fluid simulation
 - Short range
 - Long range
 - Particle-grid coupling

New doubly periodic solvers

UAMMD

Conclusions

- Later, I will show you how I leveraged this sentence to create UAMMD, but first I want to discuss the other side of a complex fluid simulation. We have seen about the kind of solvers that we can use to unravel the dynamics of particles for simulations in different spatio-temporal windows, but there is a long way between having equations for the solvers and writing a GPU code for them.
- Besides the mathematical hardships we may face along the way (and we will see some of them later on), we have to tame the algorithmic complexity of the interactions, them being either those arising from the dynamics (such as hydrodynamics) or from particle-particle couplings (such as sterics or electrostatics).
- In this regard, we basically encounter the same set of three problems again and again, which I would like to go through in this next section.

- 1 Short range interactions
- 2 Long range interactions
- 3 Particle-grid coupling

Complex fluids in the GPU era

└ Elements of a complex fluid simulation

└ Computational challenges

- We have three types of problems to solve. Short- and long-range interactions and Eulerian-Lagrangian coupling.
- The first one arises, for instance, with steric interactions, which are cut-off after a certain distance.
- And the problem here lies in finding the particles inside the radius of action of the red particle checking the least amount of black particles possible.

Computational challenges

- Short range interactions
- Long range interactions
- Particle-grid coupling

2022-05-29

Computational challenges

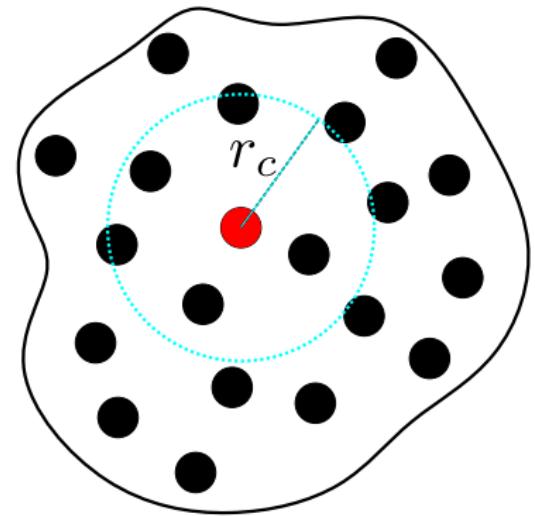
1 Short range interactions

2 Long range interactions

3 Particle-grid coupling

Example: Lennard-Jones potential

$$U_{LJ}(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] & r < r_c \\ 0 & r \geq r_c \end{cases}$$



Complex fluids in the GPU era

Elements of a complex fluid simulation

Computational challenges

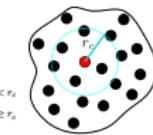
2022-05-29

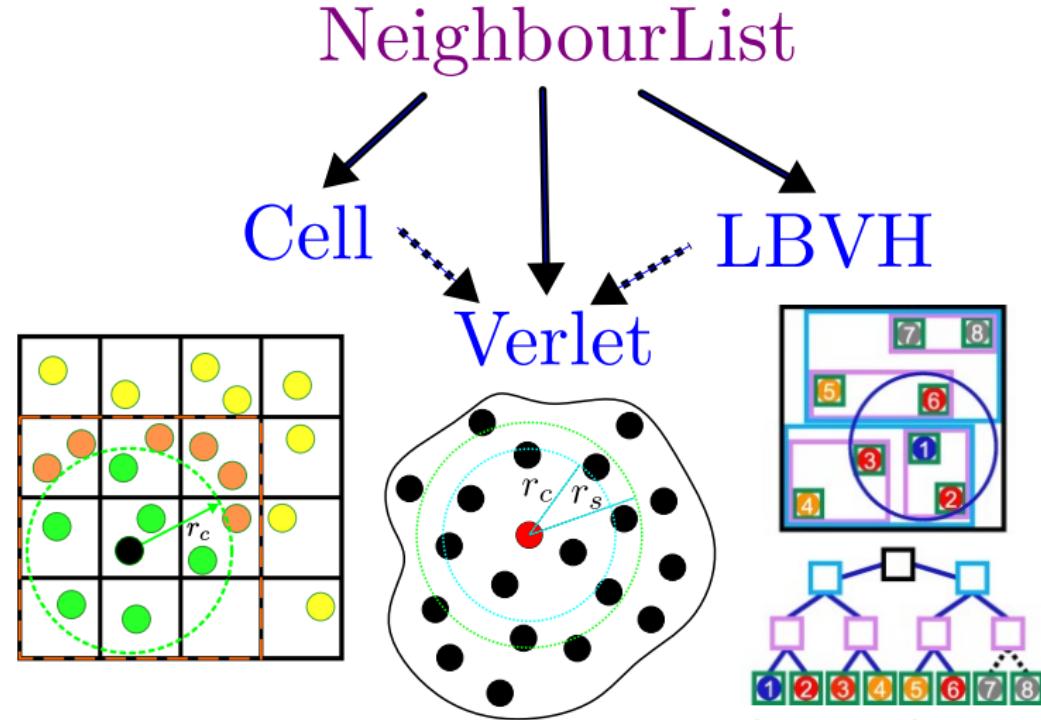
Computational challenges

- Short range interactions
- Long range interactions
- Particle-grid coupling

Example: Lennard-Jones potential

$$U_{LJ}(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] & r < r_c \\ 0 & r \geq r_c \end{cases}$$

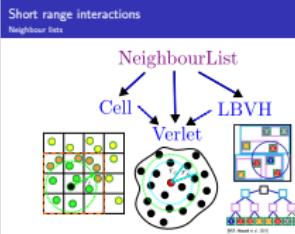




Complex fluids in the GPU era

- Elements of a complex fluid simulation
 - Short range
 - Short range interactions

2022-05-29



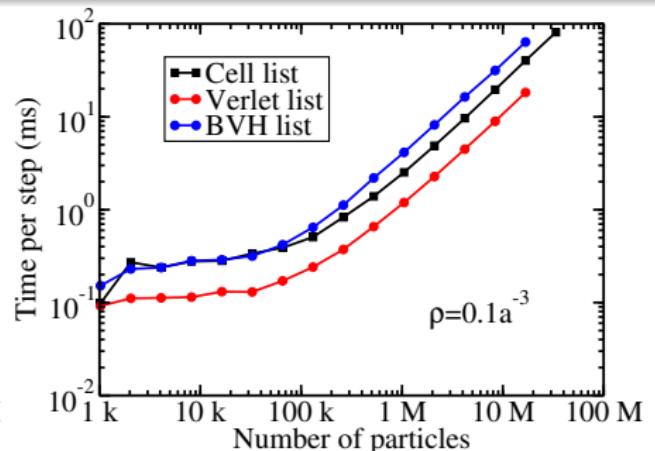
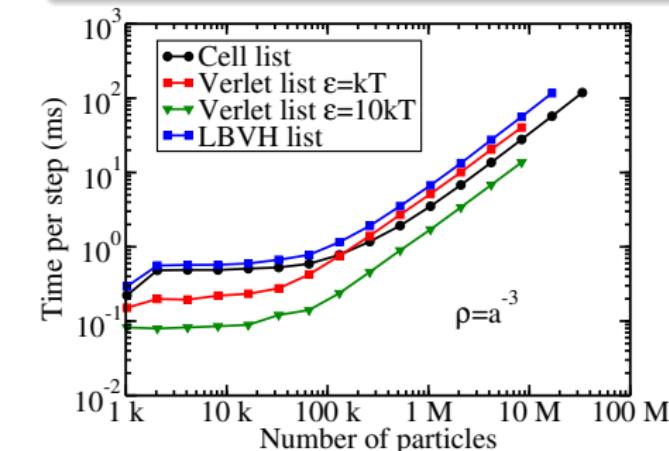
- The canonical solution for this is using a neighbour list.
- In the GPU we find three main strategies for it. Each of them typically excelling in different situations.
 - The cell list works by binning the domain and identifying the bin of each particle. Then, the neighbours of the black particle are found by visiting the neighbouring bins. The orange particles are false positives.
 - When the cell list fails, for instance when there are large concentration disparities, we can use tree-based lists. In particular, UAMMD implements the so-called Linear Bounding Volume Hierarchy algorithm, which partitions the domain in a hierarchical fashion, with boxes encasing increasingly large boxes.
 - In any case, we can leverage the Verlet strategy, which consists of using any of the two to elaborate an interaction list with an artificially enlarged cut of radius. This way, the list can be reused for several steps, as long as the particles do not diffuse a lot.

Short range interactions

Neighbour lists: Performance

Example: Lennard-Jones potential, $r_c = 2.5\sigma$ and $\varepsilon = kT$.

$$U_{LJ}(r) = \begin{cases} 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] & r < r_c \\ 0 & r \geq r_c \end{cases}$$

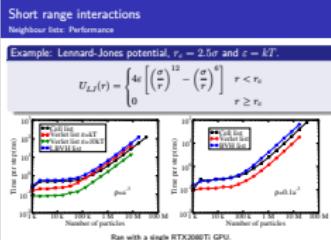


Ran with a single RTX2080Ti GPU.

Complex fluids in the GPU era

- └ Elements of a complex fluid simulation
 - └ Short range
 - └ Short range interactions

2022-05-29



- Let me show you how these compare in UAMMD's implementations.
- In this example I am showing timings for the computation of the forces for a Lennard Jonnes liquid at two different densities. We are seeing time per step versus the number of particles as I make the system bigger. There is a lot to unpack in this figure but sadly not a lot of time, so the take away message here is that they are all somewhat similar and they all present linear scaling. It is also worth noting that we can update 10 million particles at around 100 times a second.

Computational challenges

- 1 Short range interactions
- 2 Long range interactions
- 3 Particle-grid coupling

Complex fluids in the GPU era
└ Elements of a complex fluid simulation
 └ Long range
 └ Computational challenges

2022-05-29

- Short range interactions
- Long range interactions
- Particle-grid coupling

The next challenge is long, or infinitely, ranged interactions.

Computational challenges

1 Short range interactions

2 Long range interactions

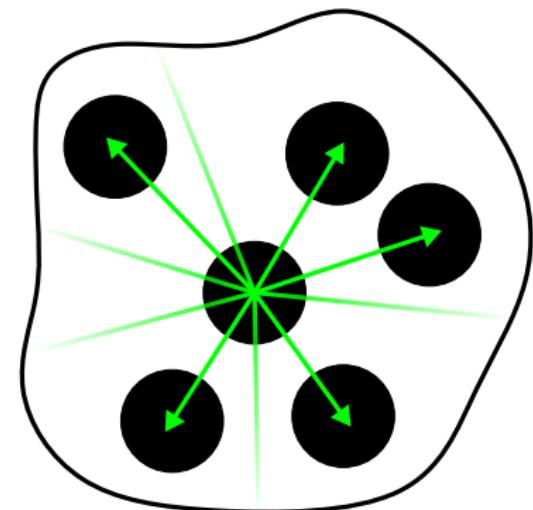
3 Particle-grid coupling

Electrostatics:

$$U_{\text{Coulomb}}(r) = \frac{1}{4\pi\epsilon_0} \frac{Q_i Q_j}{r}$$

Hydrodynamics:

$$\mathcal{O}(r) = \frac{1}{8\pi\eta r} \left(\mathbb{I} - \frac{\mathbf{r} \otimes \mathbf{r}}{r^2} \right)$$

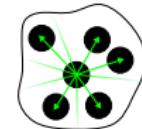


Complex fluids in the GPU era
└ Elements of a complex fluid simulation
 └ Long range
 └ Computational challenges

2022-05-29

Computational challenges

- Short range interactions
- Long range interactions
- Particle-grid coupling



$$U_{\text{Coulomb}}(r) = \frac{1}{4\pi\epsilon_0} \frac{Q_i Q_j}{r}$$

Hydrodynamics:

$$\mathcal{O}(r) = \frac{1}{8\pi\eta r} \left(\mathbb{I} - \frac{\mathbf{r} \otimes \mathbf{r}}{r^2} \right)$$

- Such is the case, for instance, of electrostatics and hydrodynamics, interactions decaying just as the inverse of distance.
- We could go ahead and check every pair of particles in the system. A quite GPU-friendly algorithm, alas with a high operation count.

Computational challenges

1 Short range interactions

2 Long range interactions

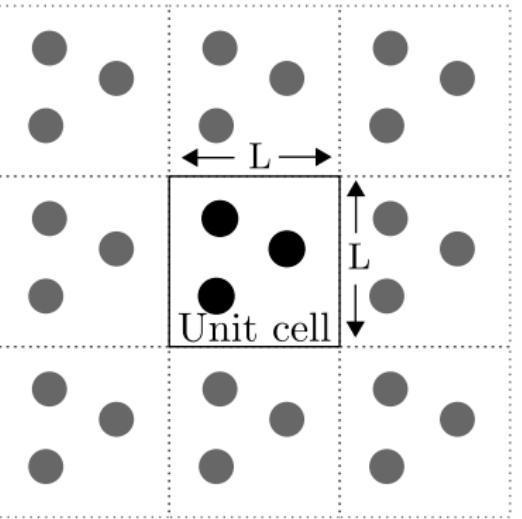
3 Particle-grid coupling

Electrostatics:

$$U_{\text{Coulomb}}(r) = \frac{1}{4\pi\epsilon_0} \frac{Q_i Q_j}{r}$$

Hydrodynamics:

$$\mathcal{O}(r) = \frac{1}{8\pi\eta r} \left(\mathbb{I} - \frac{\mathbf{r} \otimes \mathbf{r}}{r^2} \right)$$



Complex fluids in the GPU era
└ Elements of a complex fluid simulation
 └ Long range
 └ Computational challenges

2022-05-29

Computational challenges

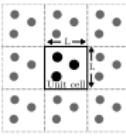
- Short range interactions
- Long range interactions
- Particle-grid coupling

Electrostatics:

$$U_{\text{Coulomb}}(r) = \frac{1}{4\pi\epsilon_0} \frac{Q_i Q_j}{r}$$

Hydrodynamics:

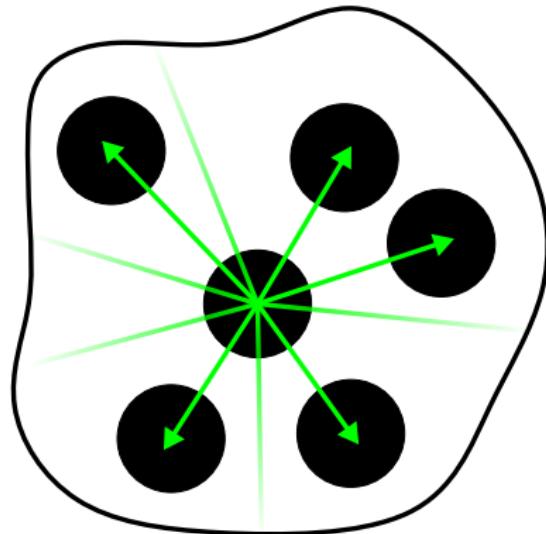
$$\mathcal{O}(r) = \frac{1}{8\pi\eta r} \left(\mathbb{I} - \frac{\mathbf{r} \otimes \mathbf{r}}{r^2} \right)$$



And that is without taking into account periodic boundary conditions, which would take the operation count of this naive approach to infinity.

Long range interactions

- Brute force
- Spectral methods
- Hierarchical/multigrid methods

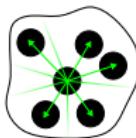


Complex fluids in the GPU era
└ Elements of a complex fluid simulation
 └ Long range
 └ Long range interactions

2022-05-29

Long range interactions

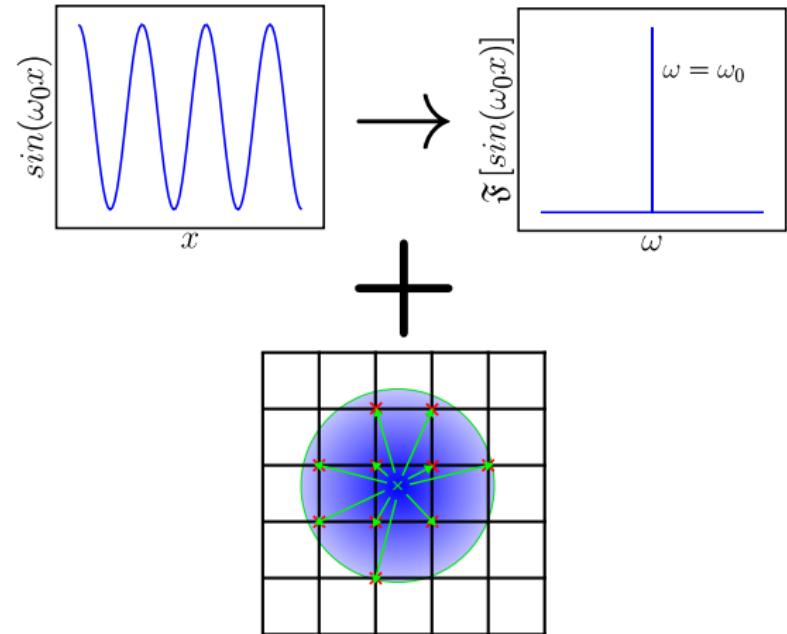
- Brute force
- Spectral methods
- Hierarchical/multigrid methods



- Luckily, we have some alternatives beside brute-forcing it that scale linearly with the number of particles, even with periodic boundary conditions.
- Mainly, we can distinguish between spectral and hierarchical or multigrid methods. I have left the second type out of this talk, since I have not focused on this family of algorithms during my phd.

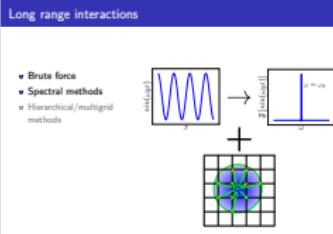
Long range interactions

- Brute force
- Spectral methods
- Hierarchical/multigrid methods



2022-05-29

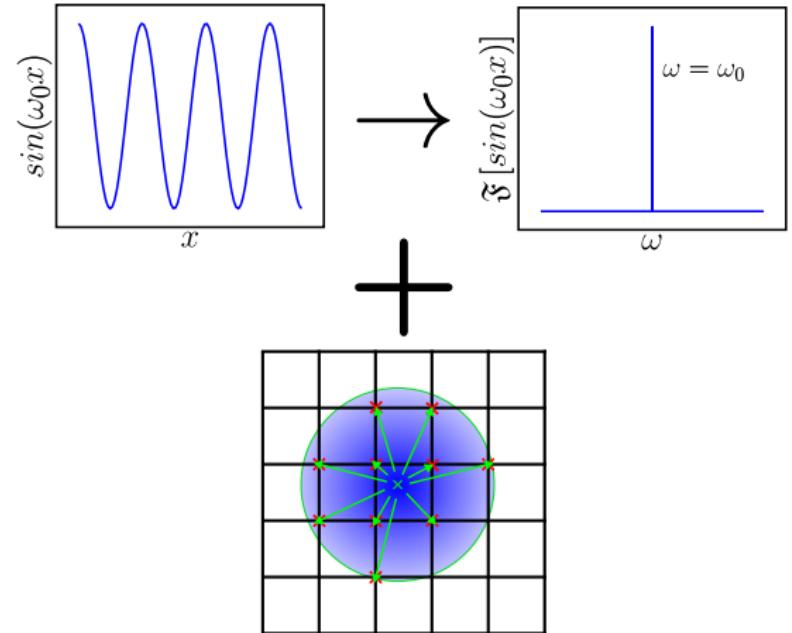
Complex fluids in the GPU era
└ Elements of a complex fluid simulation
 └ Long range
 └ Long range interactions



- Spectral methods are based on the Fast Fourier transform, which happens to be incredibly efficient on the GPU.
- We are dealing with particles, though, so we need a Non Uniform Fourier transform. Which involves transforming our Lagrangian markers into an Eulerian field.

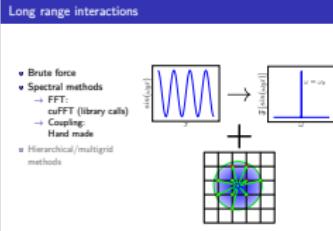
Long range interactions

- Brute force
- Spectral methods
 - FFT:
cuFFT (library calls)
 - Coupling:
Hand made
- Hierarchical/multigrid methods



Complex fluids in the GPU era
└ Elements of a complex fluid simulation
 └ Long range
 └ Long range interactions

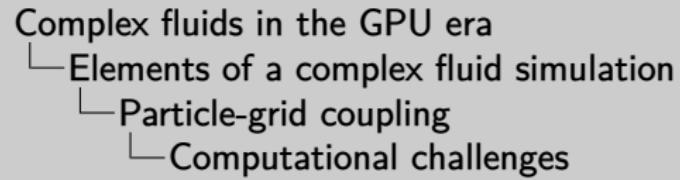
2022-05-29



- Spectral methods are based on the Fast Fourier transform, which happens to be incredibly efficient on the GPU.
- We are dealing with particles, though, so we need a Non Uniform Fourier transform. Which involves transforming our Lagrangian markers into an Eulerian field.

Computational challenges

- 1 Short range interactions
- 2 Long range interactions
- 3 Particle-grid coupling



2022-05-29

- And this takes us to the third challenge, which is the communication between particles and a grid.

Computational challenges

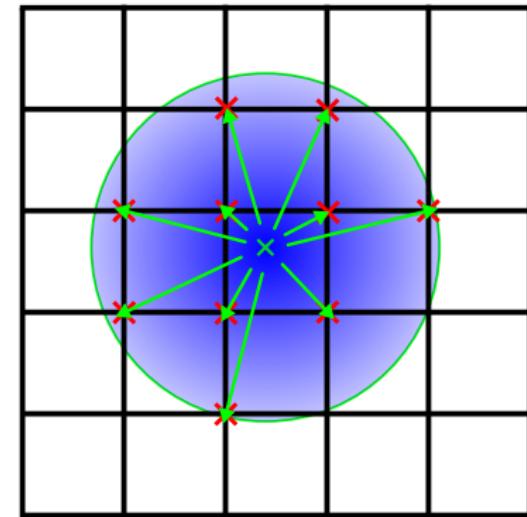
1 Short range interactions

2 Long range interactions

3 Particle-grid coupling

Spreading (\mathcal{S}):

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r})\mathbf{F} := \sum_i \mathbf{F}_i \delta_a(\mathbf{r} - \mathbf{q}_i)$$



Supervector notation $\mathbf{F} := \{\mathbf{F}_1, \dots, \mathbf{F}_N\}$

Complex fluids in the GPU era

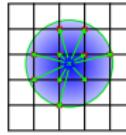
└ Elements of a complex fluid simulation
 └ Particle-grid coupling
 └ Computational challenges

2022-05-29

Computational challenges

- Short range interactions
- Long range interactions
- Particle-grid coupling

Spreading (\mathcal{S}):
$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r})\mathbf{F} := \sum_i \mathbf{F}_i \delta_a(\mathbf{r} - \mathbf{q}_i)$$



Supervector notation $\mathbf{F} := \{\mathbf{F}_1, \dots, \mathbf{F}_N\}$

- We can use this coupling for a number of things. For instance, to transform a series of forces, \mathbf{F} , acting on a group of particles into a force density discretized on a grid. We call this operation spreading.
- Note that, for a given location, we must sum over all particles, since there might be overlap, i.e, several markers contributing to the same position.

Computational challenges

1 Short range interactions

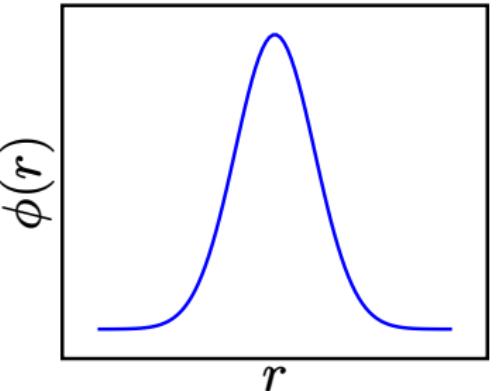
2 Long range interactions

3 Particle-grid coupling

Spreading (\mathcal{S}):

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r})\mathbf{F} := \sum_i \mathbf{F}_i \delta_a(\mathbf{r} - \mathbf{q}_i)$$

$\delta_a(\mathbf{r}) := \phi(r_x)\phi(r_y)\phi(r_z) \rightarrow$ Smeared delta



Example: $\phi(r) \propto e^{-\frac{r^2}{2\sigma^2}}$

Complex fluids in the GPU era

Elements of a complex fluid simulation

Particle-grid coupling

Computational challenges

2022-05-29

Computational challenges

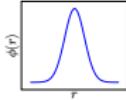
- Short range interactions
- Long range interactions

Particle-grid coupling

Spreading (\mathcal{S}):

$$f(\mathbf{r}) = \mathcal{S}(\mathbf{r})\mathbf{F} := \sum_i \mathbf{F}_i \delta_a(\mathbf{r} - \mathbf{q}_i)$$

$$\delta_a(\mathbf{r}) := \phi(r_x)\phi(r_y)\phi(r_z) \rightarrow$$
 Smeared delta



Example: $\phi(r) \propto e^{-\frac{r^2}{2\sigma^2}}$

- This communication is mediated via a compactly-supported bell-shaped kernel that is separable in the three directions. "a" here denotes the typical width of the kernel, related to the physical size of the particles.

Computational challenges

1 Short range interactions

2 Long range interactions

3 Particle-grid coupling

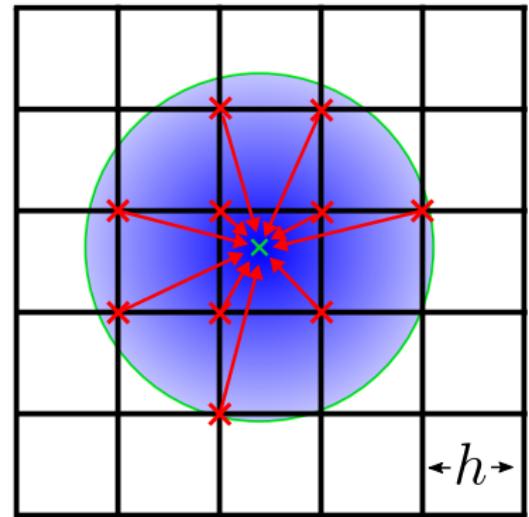
Spreading (\mathcal{S}):

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r})\mathbf{F} := \sum_i \mathbf{F}_i \delta_a(\mathbf{r} - \mathbf{q}_i)$$

$\delta_a(\mathbf{r}) := \phi(r_x)\phi(r_y)\phi(r_z) \rightarrow$ Smeared delta

Interpolation (\mathcal{J}):

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathbf{v}(\mathbf{r}) = \int \mathbf{v}(\mathbf{r}) \delta_a(\mathbf{r} - \mathbf{q}_i) d\mathbf{r}$$



Complex fluids in the GPU era

└ Elements of a complex fluid simulation
 └ Particle-grid coupling
 └ Computational challenges

2022-05-29

Computational challenges

■ Short range interactions

■ Long range interactions

■ Particle-grid coupling

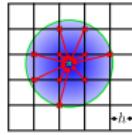
 Spreading (\mathcal{S}):

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r})\mathbf{F} := \sum_i \mathbf{F}_i \delta_a(\mathbf{r} - \mathbf{q}_i)$$

$$\delta_a(\mathbf{r}) := \phi(r_x)\phi(r_y)\phi(r_z) \rightarrow$$
 Smeared delta

 Interpolation (\mathcal{J}):

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathbf{v}(\mathbf{r}) = \int \mathbf{v}(\mathbf{r}) \delta_a(\mathbf{r} - \mathbf{q}_i) d\mathbf{r}$$



Computational challenges

1 Short range interactions

2 Long range interactions

3 Particle-grid coupling

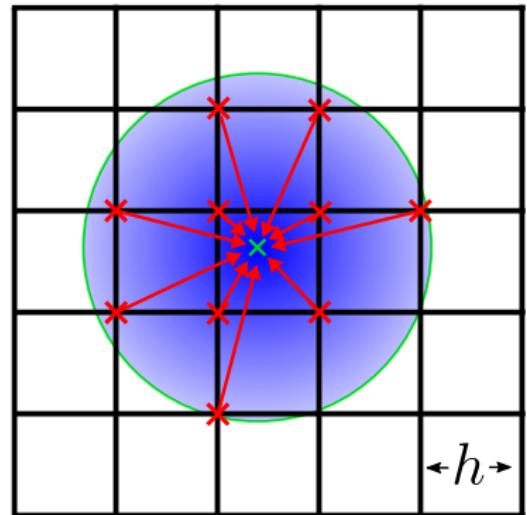
Spreading (\mathcal{S}):

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r})\mathbf{F} := \sum_i \mathbf{F}_i \delta_a(\mathbf{r} - \mathbf{q}_i)$$

$\delta_a(\mathbf{r}) := \phi(r_x)\phi(r_y)\phi(r_z) \rightarrow$ Smeared delta

Interpolation (\mathcal{J}):

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i}\mathbf{v}(\mathbf{r}) \approx \sum_n \mathbf{v}_n \delta_a(\mathbf{r}_n - \mathbf{q}_i) h^3$$



Complex fluids in the GPU era
└ Elements of a complex fluid simulation
 └ Particle-grid coupling
 └ Computational challenges

2022-05-29

Computational challenges

- Short range interactions
- Long range interactions

Particle-grid coupling

Spreading (\mathcal{S}):

$$f(\mathbf{r}) = \mathcal{S}(\mathbf{r})\mathbf{F} := \sum_i \mathbf{F}_i \delta_a(\mathbf{r} - \mathbf{q}_i)$$

$$\delta_a(\mathbf{r}) := \phi(r_x)\phi(r_y)\phi(r_z) \rightarrow$$
 Smeared delta

Interpolation (\mathcal{J}):

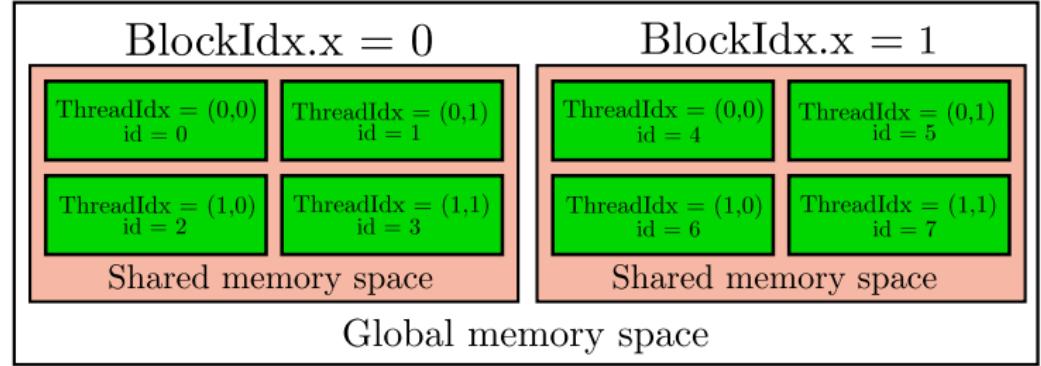
$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i}\mathbf{v}(\mathbf{r}) \approx \sum_n \mathbf{v}_n \delta_a(\mathbf{r}_n - \mathbf{q}_i) h^3$$

- For instance, to compute the velocity of a certain particle based on a velocity field defined on a grid.
- The non-uniform Fourier transform is the basis all the Eulerian-Lagrangian algorithms presented in my work, and thus it is worth describing in more detail how we can efficiently perform the spreading and interpolation operations in a GPU.

Particle-grid coupling

GPU: Basic principles

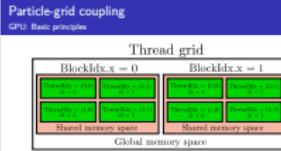
Thread grid



Complex fluids in the GPU era

- └ Elements of a complex fluid simulation
 - └ Particle-grid coupling
 - └ Particle-grid coupling

2022-05-29

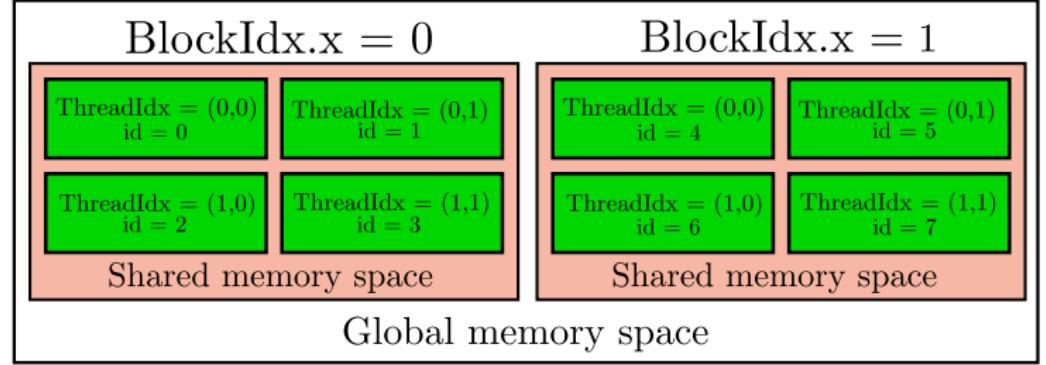


- First, we need to understand a little bit more about how GPUs work.
- Broadly speaking, in a CPU, we typically have dozens of powerful workers, or threads, that, while sharing the same underlying memory pool, largely function as individual processors.

Particle-grid coupling

GPU: Basic principles

Thread grid



Special considerations:

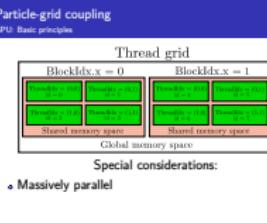
- Massively parallel

Complex fluids in the GPU era

- Elements of a complex fluid simulation
 - Particle-grid coupling
 - Particle-grid coupling

2022-05-29

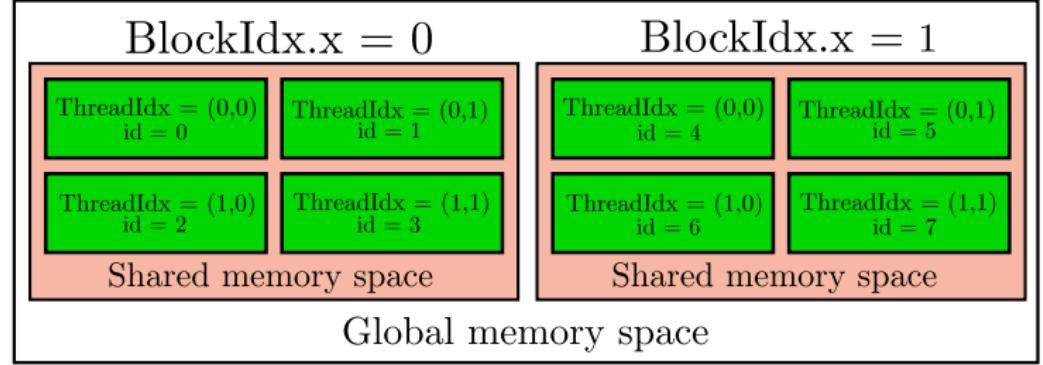
On the contrary, a GPU offers thousands of threads (albeit slower ones), which are grouped into blocks of a few hundreds.



Particle-grid coupling

GPU: Basic principles

Thread grid



Special considerations:

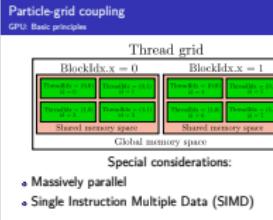
- Massively parallel
- Single Instruction Multiple Data (SIMD)

Complex fluids in the GPU era

- Elements of a complex fluid simulation
 - Particle-grid coupling
 - Particle-grid coupling

2022-05-29

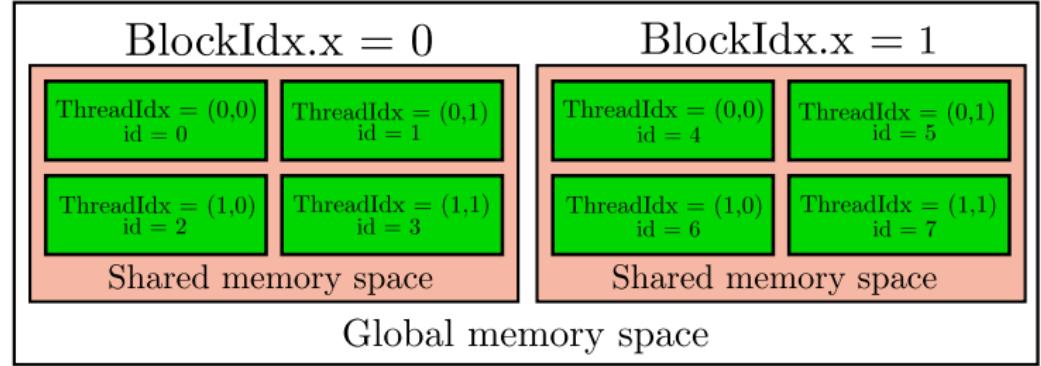
And there is a caveat, all threads in a block must all perform the same operation at the same time.



Particle-grid coupling

GPU: Basic principles

Thread grid



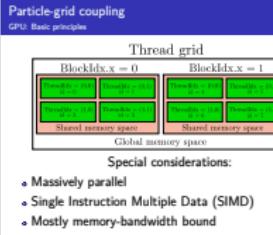
Special considerations:

- Massively parallel
- Single Instruction Multiple Data (SIMD)
- Mostly memory-bandwidth bound

Complex fluids in the GPU era

- └ Elements of a complex fluid simulation
 - └ Particle-grid coupling
 - └ Particle-grid coupling

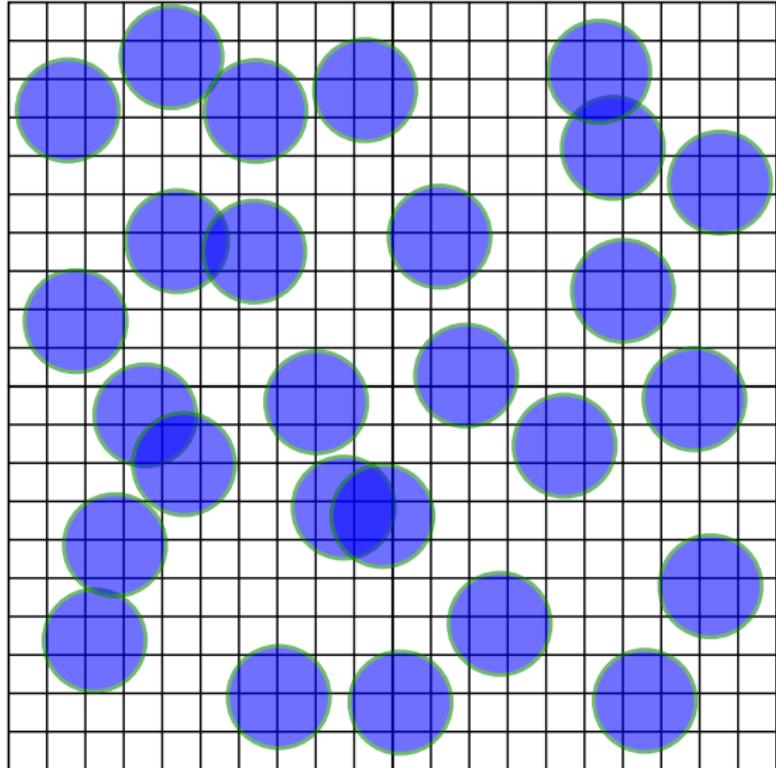
2022-05-29



- This operation can be applied to different data, but GPU payloads tend to be bottlenecked by memory operations.
- To aid with this, we have several memory spaces available. The slow access global memory is accessible by threads in every block, while shared memory is fast but only available to the threads inside the same block and only when the GPU is running a job.

Particle-grid coupling

GPU Common concerns: Cache memory

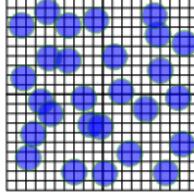


Complex fluids in the GPU era

- └ Elements of a complex fluid simulation
 - └ Particle-grid coupling
 - └ Particle-grid coupling

2022-05-29

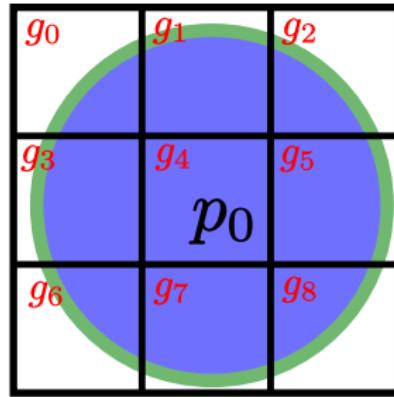
Particle-grid coupling
GPU Common concern: Cache memory



- For spreading and interpolation, we need to take into account a couple of GPU quirks.
- Typically we need to deal with a situation like this one. A distribution of particles, which might overlap, that must communicate with an underlying grid. Each particle needs to do something with the grid points it overlaps. Normally, a thread or group of threads is assigned to each particle.
- The first consideration is cache memory. And this is not particular to the GPU, but its effect is more important due to its memory-bandwidth limitations.

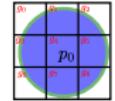
Particle-grid coupling

GPU Common concerns: Cache memory



2022-05-29

- Complex fluids in the GPU era
 - Elements of a complex fluid simulation
 - Particle-grid coupling
 - Particle-grid coupling

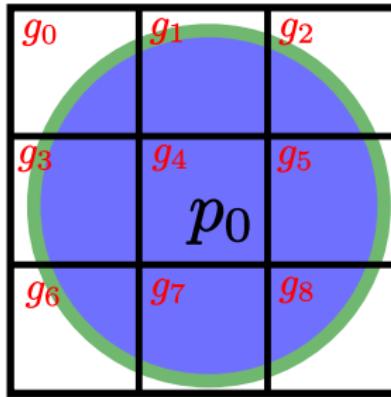


- Zooming in to some arbitrary section of the domain, I have denoted the grid points with "g". Where each thread will be assigned to some particle that may or may not be the same.

Particle-grid coupling

GPU Common concerns: Cache memory

Threads: $t_0 \boxed{t_1} t_2 t_3 t_4 \dots$



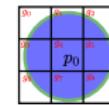
Complex fluids in the GPU era

- └ Elements of a complex fluid simulation
 - └ Particle-grid coupling
 - └ Particle-grid coupling

2022-05-29

Particle-grid coupling
GPU Common concern: Cache memory

Threads: $t_0 \boxed{t_1} t_2 t_3 t_4 \dots$

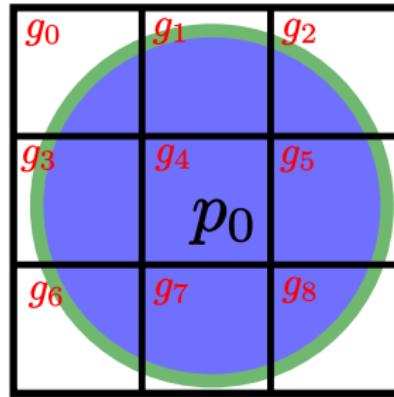


- The thing is that when spreading, threads, "t", will need to write to grid points.

Particle-grid coupling

GPU Common concerns: Cache memory

Threads: $t_0 \boxed{t_1} t_2 t_3 t_4 \dots$

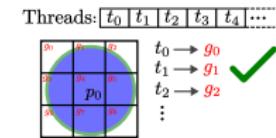


$t_0 \rightarrow g_0$
 $t_1 \rightarrow g_1$
 $t_2 \rightarrow g_2$
⋮



Complex fluids in the GPU era

- Elements of a complex fluid simulation
 - Particle-grid coupling
 - Particle-grid coupling

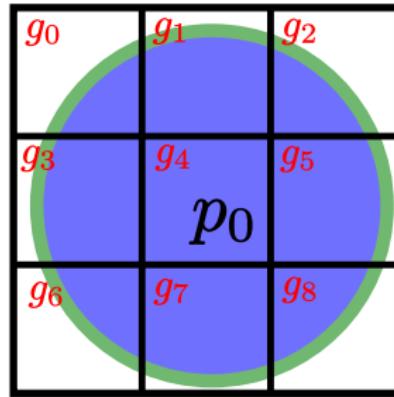


- The GPU greatly benefits from contiguous threads dealing with data that is stored in close memory locations.

Particle-grid coupling

GPU Common concerns: Cache memory

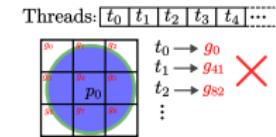
Threads: $t_0 \boxed{t_1} t_2 t_3 t_4 \dots$



$t_0 \rightarrow g_0$
 $t_1 \rightarrow g_{41}$ 
 $t_2 \rightarrow g_{82}$
 \vdots

Complex fluids in the GPU era

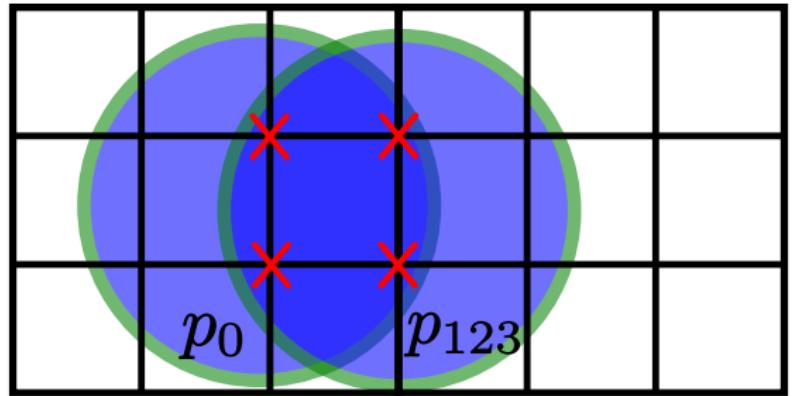
- └ Elements of a complex fluid simulation
 - └ Particle-grid coupling
 - └ Particle-grid coupling



- On the contrary, performance degrades when contiguous threads deal with memory locations that are far away.
- The take away message here is that we really want operations that happen close in physical space to also reside close in GPU memory.

Particle-grid coupling

GPU Common concerns: Atomic operations

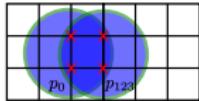


Complex fluids in the GPU era

- └ Elements of a complex fluid simulation
 - └ Particle-grid coupling
 - └ Particle-grid coupling

2022-05-29

Particle-grid coupling
GPU Common concern: Atomic operations

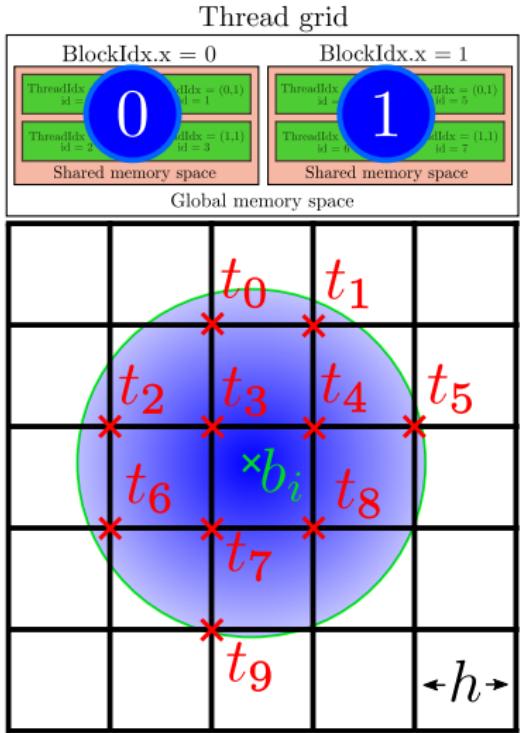


- The other concern is atomic operations. Since different threads may deal with different particles concurrently it can happen that two threads might try to write at the same time to the same place. This is called an atomic operation, and traditionally this collision incurs a large overhead.

Particle-grid coupling

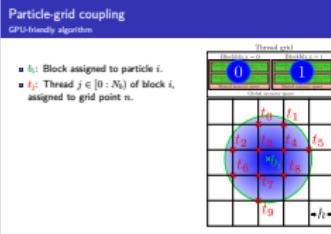
GPU-friendly algorithm

- b_i : Block assigned to particle i .
- t_j : Thread $j \in [0 : N_b)$ of block i , assigned to grid point n .



Complex fluids in the GPU era
└ Elements of a complex fluid simulation
 └ Particle-grid coupling
 └ Particle-grid coupling

2022-05-29



- So remember, we must juggle cache memory and atomics. Let me tell you how my strategy for spreading and interpolation deals with them.
- We assign a block of threads to each particle, so that each thread in the block takes care of a different nearby grid point. This automatically improves memory access, since by construction nearby threads read nearby locations in memory. On the other hand we can take advantage of several threads working on the same particle. Putting common information in shared memory.
- Furthermore, all threads in the same block are guaranteed to be free of atomic operations.

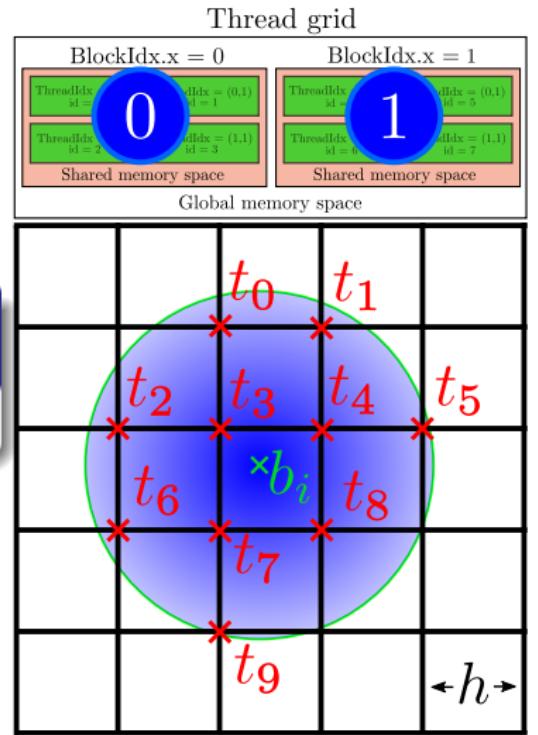
Particle-grid coupling

GPU-friendly algorithm

- b_i : Block assigned to particle i .
- t_j : Thread $j \in [0 : N_b]$ of block i , assigned to grid point n .

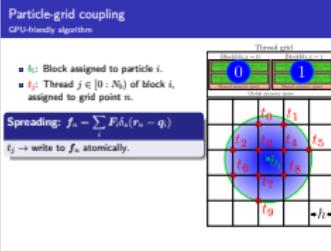
Spreading: $f_n = \sum_i F_i \delta_a(r_n - q_i)$

$t_j \rightarrow$ write to f_n atomically.



- Complex fluids in the GPU era
- Elements of a complex fluid simulation
 - Particle-grid coupling
 - Particle-grid coupling

2022-05-29



- Still, other blocks, assigned to other particles, can cause atomic collisions when spreading. Imagine another particle overlapping with this one.
- In this case, each thread in the block writes the contribution of the particle to a different grid point.

Particle-grid coupling

GPU-friendly algorithm

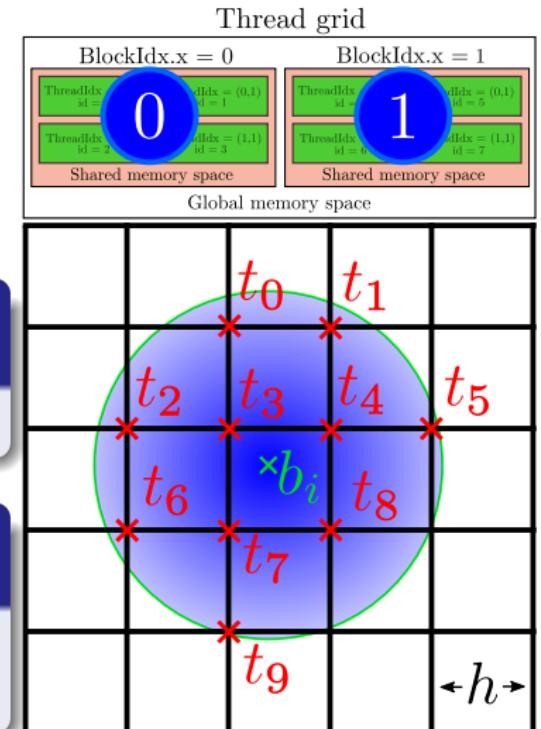
- b_i : Block assigned to particle i .
- t_j : Thread $j \in [0 : N_b)$ of block i , assigned to grid point n .

Spreading: $f_n = \sum_i F_i \delta_a(\mathbf{r}_n - \mathbf{q}_i)$

$t_j \rightarrow$ write to f_n atomically.

Interpolation: $u_i = \sum_n v_n \delta_a(\mathbf{r}_n - \mathbf{q}_i) h^3$

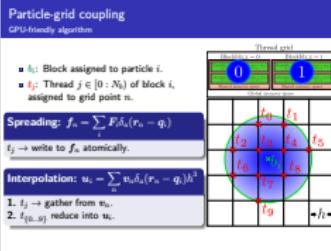
1. $t_j \rightarrow$ gather from v_n .
2. $t_{\{0\dots9\}}$ reduce into u_i .



Complex fluids in the GPU era

- Elements of a complex fluid simulation
 - Particle-grid coupling
 - Particle-grid coupling

2022-05-29



- Interpolation is inherently more efficient, since we only have to read the grid information. Each thread computes the contribution of a different nearby grid point. After summing all of them only one thread needs to write the final result for the particle.

Particle-grid coupling

GPU-friendly algorithm

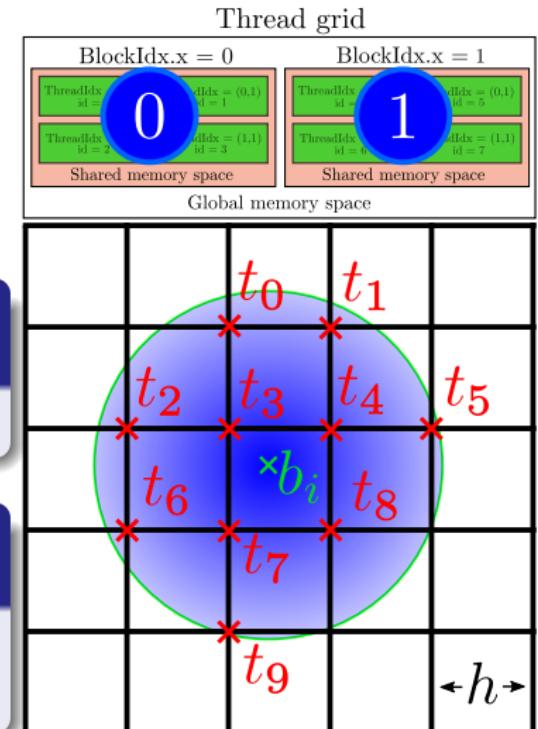
- b_i : Block assigned to particle i .
- t_j : Thread $j \in [0 : N_b)$ of block i , assigned to grid point n .

Spreading: $f_n = \sum_i F_i \delta_a(\mathbf{r}_n - \mathbf{q}_i)$

$t_j \rightarrow$ write to f_n atomically.

Interpolation: $u_i = \sum_n \mathbf{v}_n \delta_a(\mathbf{r}_n - \mathbf{q}_i) h^3$

1. $t_j \rightarrow$ gather from \mathbf{v}_n .
2. $t_{\{0\dots9\}}$ reduce into u_i .

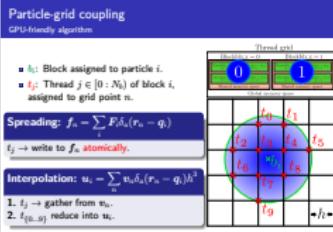


Complex fluids in the GPU era

- Elements of a complex fluid simulation
 - Particle-grid coupling
 - Particle-grid coupling

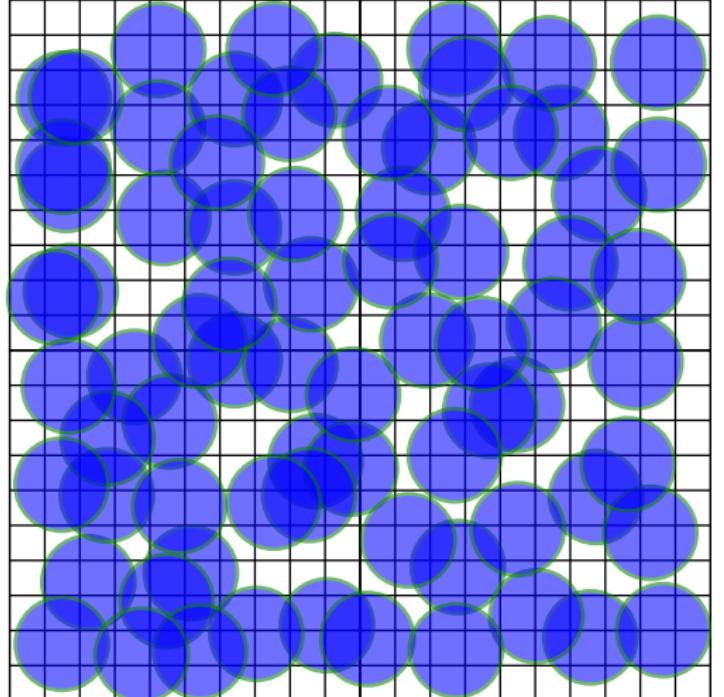
2022-05-29

Let me focus then on spreading, the problematic operation.



Particle-grid coupling

Spreading: Performance of a random distribution



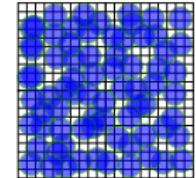
Random 3D distribution with ~ 1 particle per grid point

Complex fluids in the GPU era

- └ Elements of a complex fluid simulation
 - └ Particle-grid coupling
 - └ Particle-grid coupling

2022-05-29

Particle-grid coupling
Spreading: Performance of a random distribution

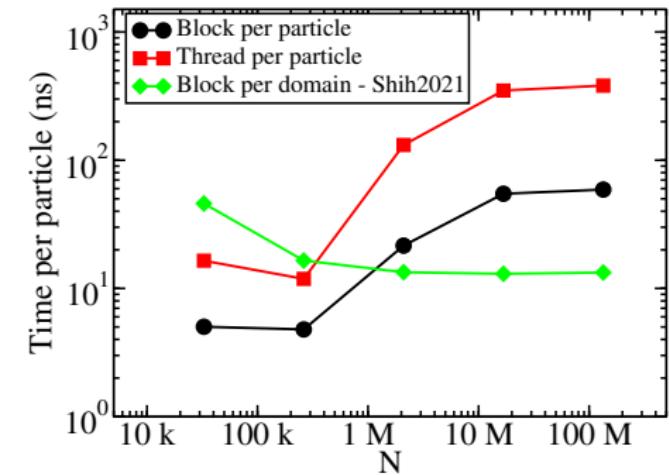


Random 3D distribution with ~ 1 particle per grid point

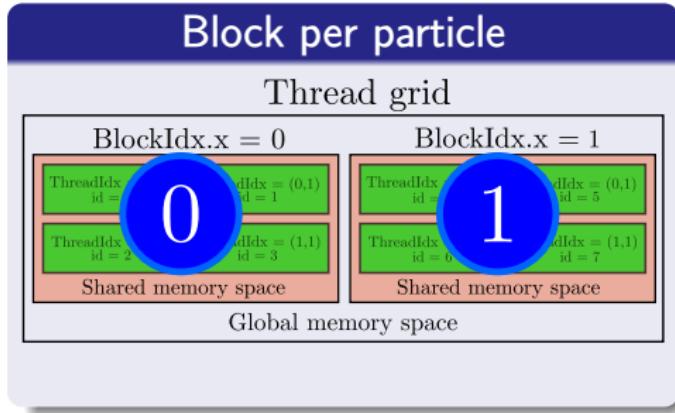
- To test both atomics and the effect of the memory pattern, I will study a really dense random distribution of particles.
- In particular, with as many particles as grid points.
- Random means that first, there is a lot of overlap.
- And second, there is no correlation between nearby particles and their memory locations.

Particle-grid coupling

Spreading: Performance of a random distribution



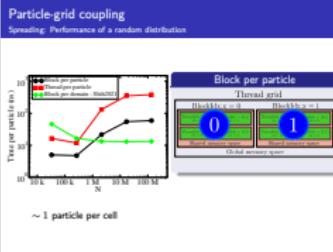
~ 1 particle per cell



Complex fluids in the GPU era

Elements of a complex fluid simulation

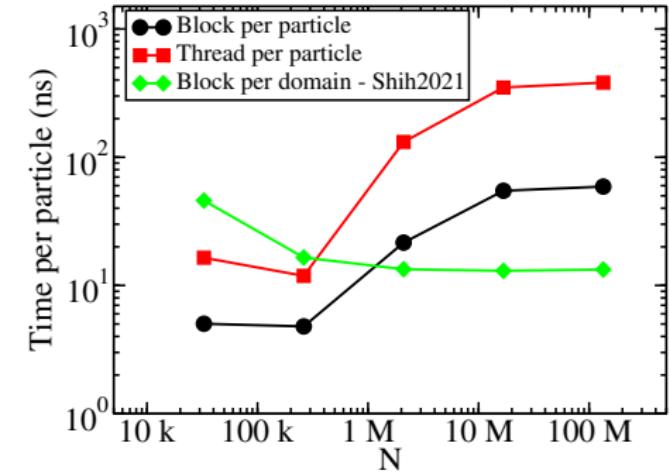
Particle-grid coupling



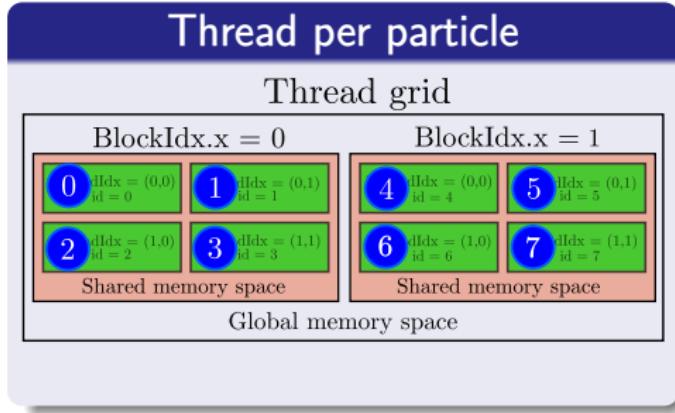
- This is the average time it takes to spread one particle as we increase the system size (keeping the concentration constant).
 - Here block per particle represents the algorithm I just described.

Particle-grid coupling

Spreading: Performance of a random distribution



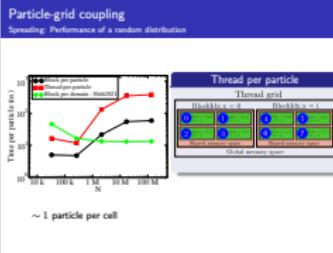
~ 1 particle per cell



Complex fluids in the GPU era

- Elements of a complex fluid simulation
 - Particle-grid coupling
 - Particle-grid coupling

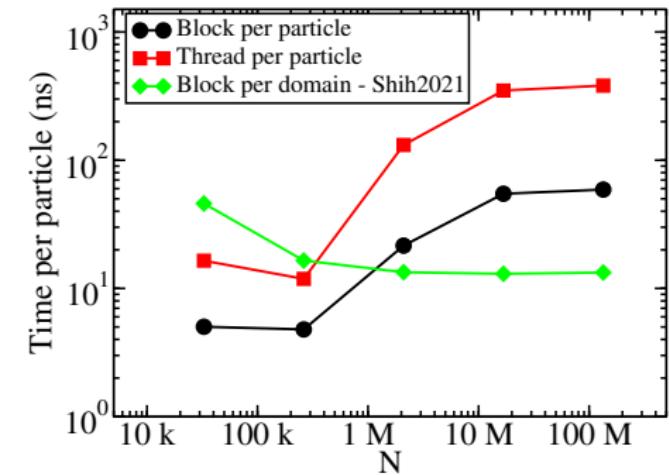
2022-05-29



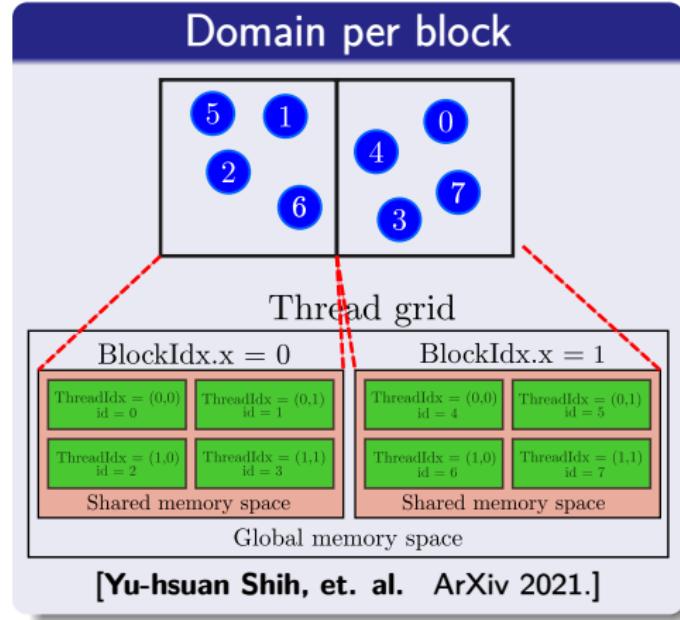
- Thread per particle is a naive implementation where we assign a particle to each thread, without them collaborating in any way.

Particle-grid coupling

Spreading: Performance of a random distribution



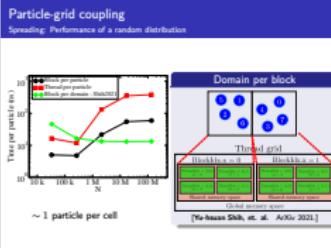
~ 1 particle per cell



Complex fluids in the GPU era

- Elements of a complex fluid simulation
 - Particle-grid coupling
 - Particle-grid coupling

2022-05-29



- Block per domain is a recent sophisticated alternative that assigns groups of threads to a piece of the domain instead of a particle, trying to avoid atomic operations.
- Although my strategy seems to be best for small systems. It appears to fail after 2 million particles.
- A block per particle naturally reduces atomic operations. On the other hand the particles being randomly distributed in memory also help with that at the cost of a much less cache friendly access pattern.
- I claim that this trade-off is not worth it and a better access pattern is more important than avoiding atomics. Luckily I have a really cool trick that can help me prove my point.

Increasing data locality

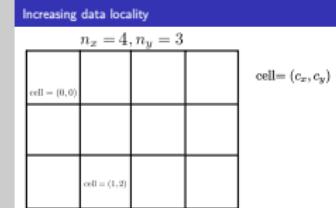
$$n_x = 4, n_y = 3$$

cell = (0, 0)			

cell= (c_x, c_y)

- Complex fluids in the GPU era
- └ Elements of a complex fluid simulation
 - └ Particle-grid coupling
 - └ Increasing data locality

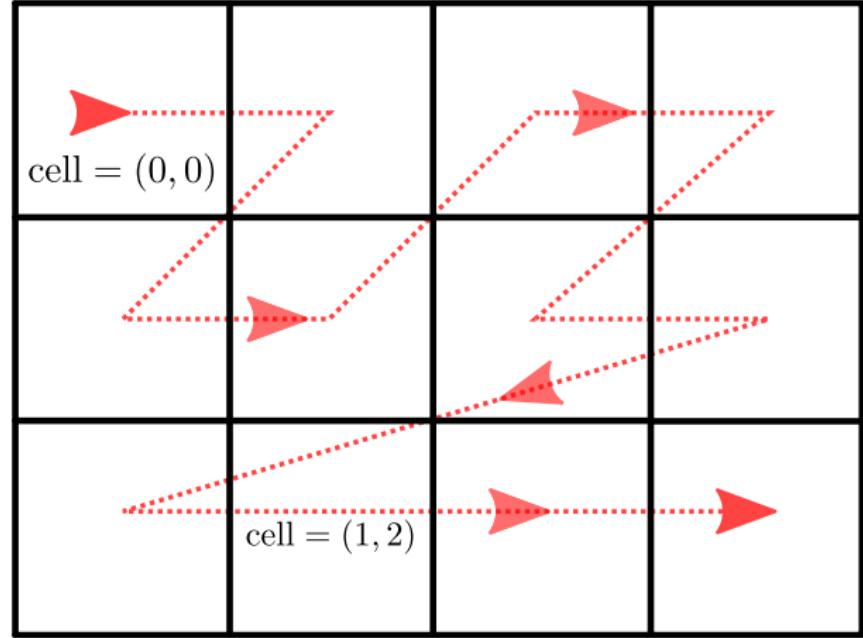
2022-05-29



Let us virtually bin the domain, where each cell is identified by unique coordinates. And note that this binning is decoupled from the target discretized field.

Increasing data locality

$$n_x = 4, n_y = 3$$

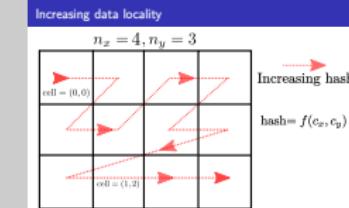


Increasing hash

$$\text{hash} = f(c_x, c_y)$$

2022-05-29

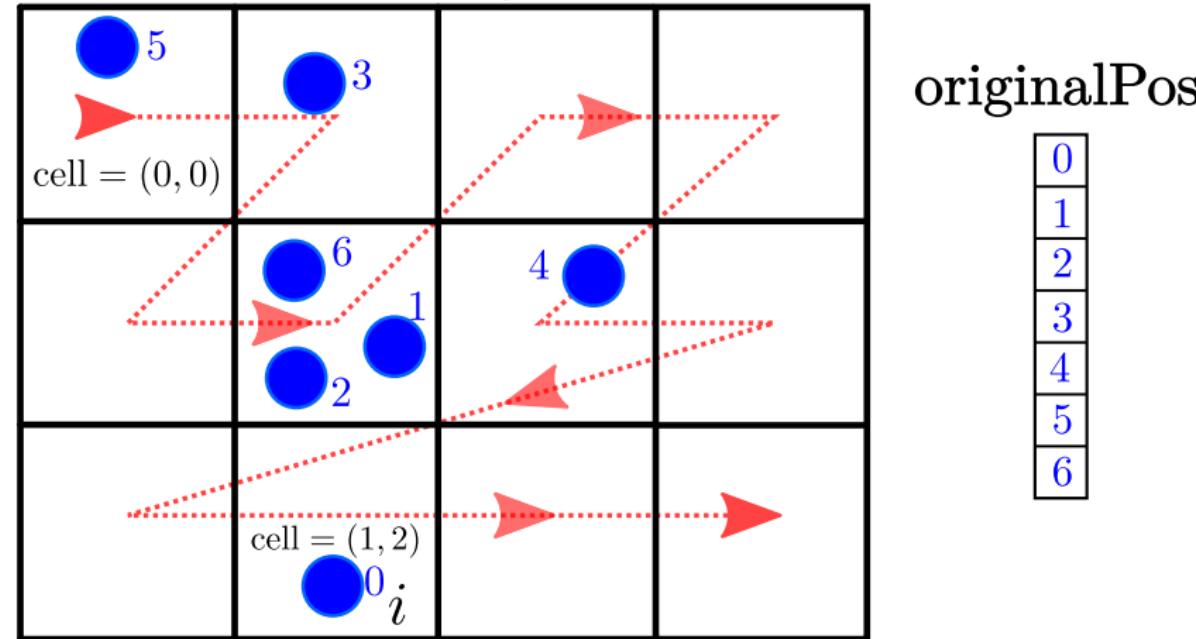
- Complex fluids in the GPU era
 - Elements of a complex fluid simulation
 - Particle-grid coupling
 - Increasing data locality



Then, using the cell coordinates to hash them, we choose a space filling curve that maximizes local volume exploration with an optimal tortuosity. Here the hash value increases following the red line.

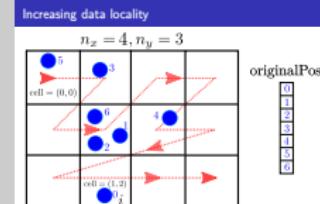
Increasing data locality

$$n_x = 4, n_y = 3$$



- Complex fluids in the GPU era
- └ Elements of a complex fluid simulation
 - └ Particle-grid coupling
 - └ Increasing data locality

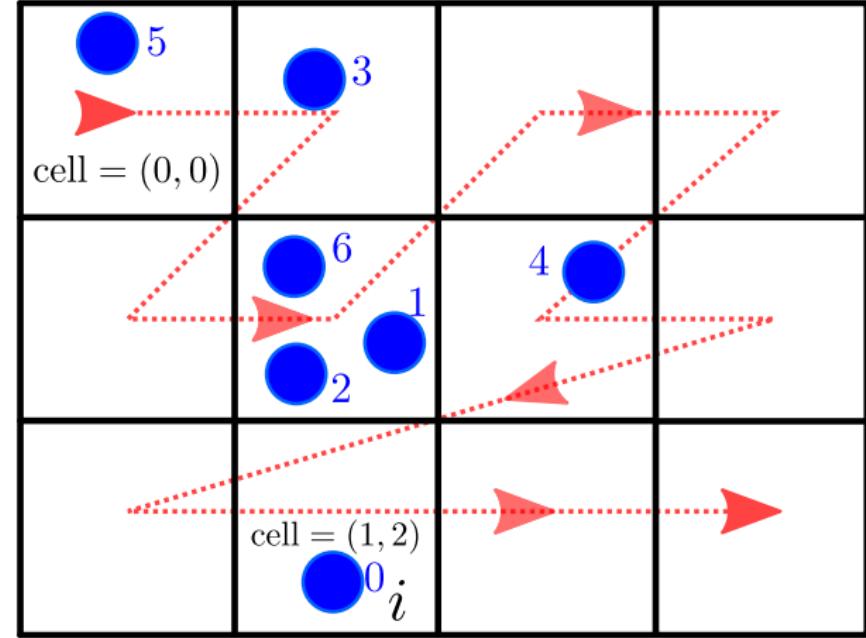
2022-05-29



For any set of random particles, we hash and sort them along the curve. The numbers above particles represent their initial location in memory. Before this sorting, there is no correlation between a particle's location in memory and space.

Increasing data locality

$$n_x = 4, n_y = 3$$

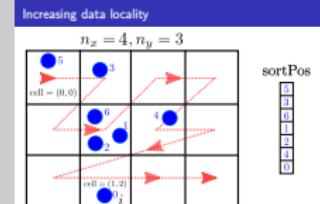


sortPos

5
3
6
1
2
4
0

- Complex fluids in the GPU era
- └ Elements of a complex fluid simulation
 - └ Particle-grid coupling
 - └ Increasing data locality

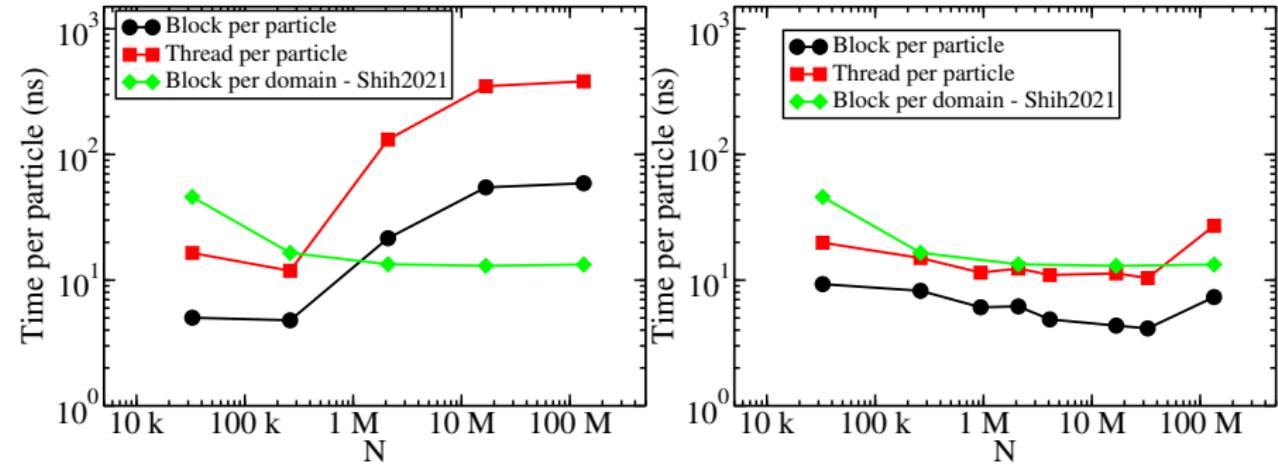
2022-05-29



But after sorting, particles with similar hashes (close in space by construction) have a similar memory location.

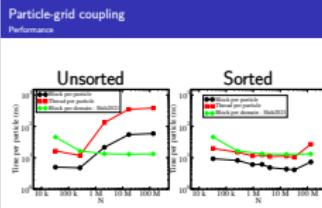
Particle-grid coupling

Performance



Complex fluids in the GPU era
└ Elements of a complex fluid simulation
 └ Particle-grid coupling
 └ Particle-grid coupling

2022-05-29



- Turns out this works amazingly well.
- Just by sorting we increase performance ten fold in some cases.
- The black line being quasi horizontal means that we have achieved linear scaling, and with quite a good timing too.
- In this case simplicity beats complexity.
- Now, being a closed-sourced architecture I have no way of knowing what it is exactly, but I am pretty sure there is something going on with atomics in the GPU that makes them way less expensive than you would expect.

Talk outline

1 Introduction

2 Elements of a complex fluid simulation

3 New doubly periodic solvers

- Hydrodynamics
- Electrostatics

4 UAMMD

5 Conclusions

Complex fluids in the GPU era

New doubly periodic solvers

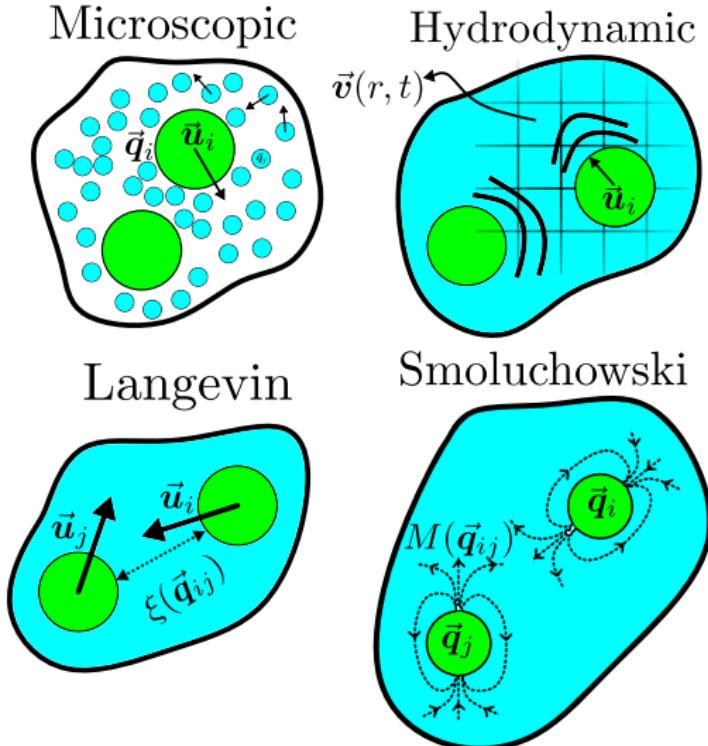
Talk outline

2022-05-29

- Let us put what we just learned to practice, by discussing some algorithms, developed during my phd for computing hydrodynamics and electrostatics in confined geometries. Both of them, long-ranged interactions.
- I will start with two algorithms for hydrodynamics.

Hydrodynamics in confined geometries

New algorithms for Brownian Dynamics with Hydrodynamic Interactions (BDHI)

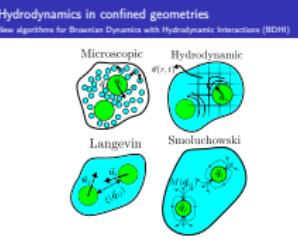


Complex fluids in the GPU era

└ New doubly periodic solvers

└ Hydrodynamics in confined geometries

2022-05-29



- And note that my contributions focus on the Smoluchowski level.

Complex fluids

The spatio-temporal landscape of numerical techniques.

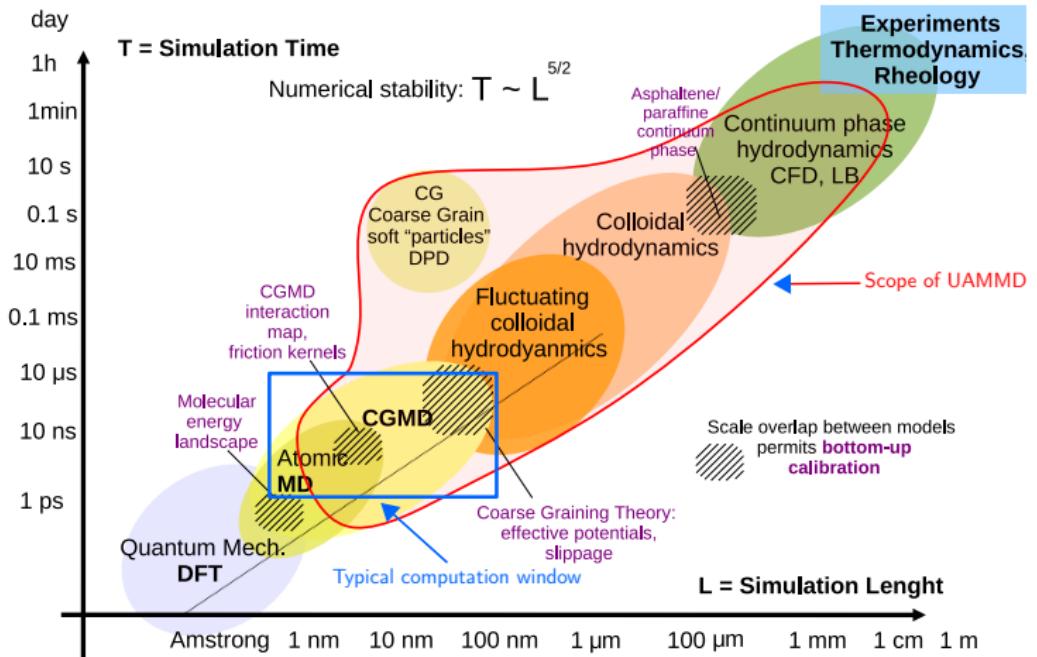


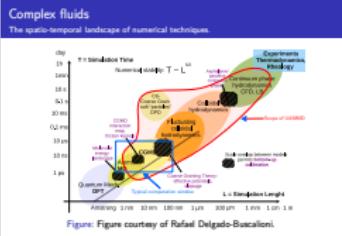
Figure: Figure courtesy of Rafael Delgado-Buscalioni.

Complex fluids in the GPU era

└ New doubly periodic solvers

└ Complex fluids

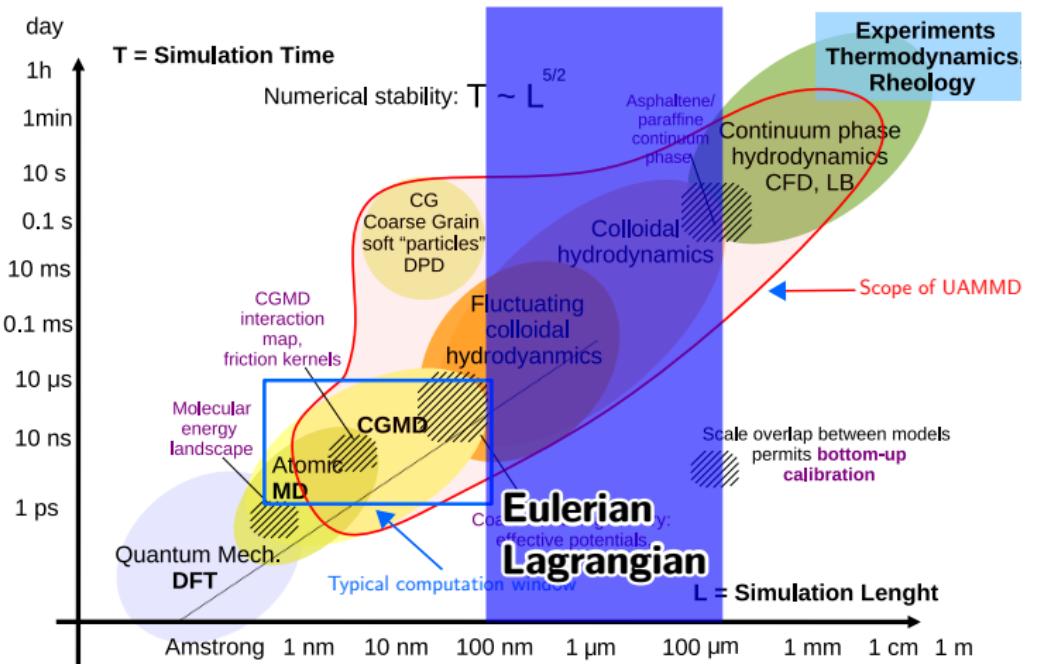
2022-05-29



- Which will require techniques that mix both continuous fields and particles.

Complex fluids

The spatio-temporal landscape of numerical techniques.

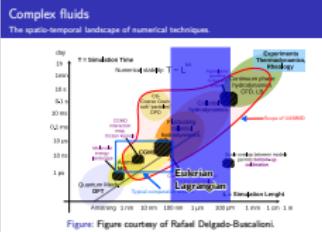


Complex fluids in the GPU era

└ New doubly periodic solvers

└ Complex fluids

2022-05-29



- Which will require techniques that mix both continuous fields and particles.

Smoluchowski level

Brownian Dynamics with Hydrodynamics Interactions (BDHI)

$$dq = \mathcal{M}Fdt + \sqrt{2k_B T \mathcal{M}}d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

- $\mathcal{M}(q)$:

Mobility tensor. Configuration- and BC-dependent

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Smoluchowski level

2022-05-29

Smoluchowski level
Brownian Dynamics with Hydrodynamics Interactions (BDHI)

$$dq = \mathcal{M}Fdt + \sqrt{2k_B T \mathcal{M}}d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

• $\mathcal{M}(q)$:
Mobility tensor. Configuration- and BC-dependent

- The Smoluchowski level is governed by the Brownian Dynamics equations of motion. Which can be written as follows.
- Here, M is a Mobility tensor, which in general is a symmetric matrix with elements for all pairs of particles.

Smoluchowski level

Brownian Dynamics with Hydrodynamics Interactions (BDHI)

$$dq = \mathcal{M}F dt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

- $\mathcal{M}(q)$:
Mobility tensor. Configuration- and BC-dependent
- $d\tilde{W}$: Brownian motion

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Smoluchowski level

2022-05-29

We also have some thermal fluctuations.

Smoluchowski level
Brownian Dynamics with Hydrodynamics Interactions (BDHI)
 $dq = \mathcal{M}F dt + \sqrt{2k_B T \mathcal{M}} dW + k_B T \partial_q \cdot \mathcal{M} dt.$
• $\mathcal{M}(q)$:
 Mobility tensor. Configuration- and BC-dependent
• dW : Brownian motion

Smoluchowski level

Brownian Dynamics with Hydrodynamics Interactions (BDHI)

$$dq = \mathcal{M}Fdt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

- $\mathcal{M}(q)$:
Mobility tensor. Configuration- and BC-dependent
- $d\tilde{W}$: Brownian motion
- $k_B T \partial_q \cdot \mathcal{M} dt$:
Thermal drift. Zero depending on the BCs

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Smoluchowski level

2022-05-29

And in general, a so-called thermal drift term that is luckily, many times null.

$$dq = \mathcal{M}Fdt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

Problems

Smoluchowski level

Brownian Dynamics with Hydrodynamics Interactions (BDHI)

$$dq = \mathcal{M}Fdt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

Problems

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Smoluchowski level

Sadly, applying this equation directly is far from trivial in most instances.

2022-05-29

$$dq = \mathcal{M}F dt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

Problems

• $\mathcal{M}F$: Matrix-vector multiplication is $O(N^2)$

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Smoluchowski level

2022-05-29

For starters, applying the mobility involves multiplying a $3N \times 3N$ matrix and a $3N$ vector.

Problems

- $\mathcal{M}F$: Matrix-vector multiplication is $O(N^2)$

$$dq = \mathcal{M}F dt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

Problems

- $\mathcal{M}F$: Matrix-vector multiplication is $O(N^2)$
- $\sqrt{\mathcal{M}}$: Quite expensive in general. $O(N^{[2.25-3]})$

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Smoluchowski level

2022-05-29

Smoluchowski level
Brownian Dynamics with Hydrodynamics Interactions (BDHI)
 $dq = \mathcal{M}F dt + \sqrt{2k_B T \mathcal{M}} dW + k_B T \partial_q \cdot \mathcal{M} dt.$
Problems
• $\mathcal{M}F$: Matrix-vector multiplication is $O(N^2)$
• $\sqrt{\mathcal{M}}$: Quite expensive in general. $O(N^{[2.25-3]})$

Moreover, we need to compute the square root of the mobility, whatever that means.

$$dq = \mathcal{M}F dt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

Problems

- $\mathcal{M}F$: Matrix-vector multiplication is $O(N^2)$
- $\sqrt{\mathcal{M}}$: Quite expensive in general. $O(N^{[2.25-3]})$
- \mathcal{M} might be unknown/not analytical.

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Smoluchowski level

2022-05-29

Smoluchowski level
Brownian Dynamics with Hydrodynamics Interactions (BDHI)
 $dq = \mathcal{M}F dt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$
Problems

- $\mathcal{M}F$: Matrix-vector multiplication is $O(N^2)$
- $\sqrt{\mathcal{M}}$: Quite expensive in general. $O(N^{[2.25-3]})$
- \mathcal{M} might be unknown/not analytical.

And to make things worse, many times we do not even have an expression for the mobility. Which is an object that emerges from macroscopic hydrodynamics.

$$dq = \mathcal{M}F dt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

Problems

- $\mathcal{M}F$: Matrix-vector multiplication is $O(N^2)$
- $\sqrt{\mathcal{M}}$: Quite expensive in general. $O(N^{[2.25-3]})$
- \mathcal{M} might be unknown/not analytical.

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Smoluchowski level

2022-05-29

$$dq = \mathcal{M}F dt + \sqrt{2k_B T \mathcal{M}} d\tilde{W} + k_B T \partial_q \cdot \mathcal{M} dt.$$

Problems

- $\mathcal{M}F$: Matrix-vector multiplication is $O(N^2)$
- $\sqrt{\mathcal{M}}$: Quite expensive in general. $O(N^{[2.25-3]})$
- \mathcal{M} might be unknown/not analytical.

Instead of using this equation, we can arrive at the same dynamics by solving the Stokes equations coupled with particles. This opens the way to a family of Eulerian-Lagrangian algorithms, our specialty.

2022-05-29

- The Stokes equations are the inertia-less version of the Incompressible Navier Stokes equations. And can be understood as a balance between the momentum diffusion of the fluid and the stress that it is subjected to.
- This stress comes from its internal pressure and any external stress, such as forces and thermal fluctuations.
- Here, Z is a tensor that is designed to comply with fluctuation-dissipation balance.

momentum diffusion

$$-\overbrace{\eta \nabla^2 v}^{\text{momentum diffusion}}$$

2022-05-29

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Hydrodynamics

momentum diffusion

$$-\overbrace{\eta \nabla^2 v}^{\text{momentum diffusion}}$$

$$\underbrace{-\eta \nabla^2 v}_{\text{momentum diffusion}} = \underbrace{-\nabla \pi}_{\text{internal stress}}$$

2022-05-29

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Hydrodynamics

Hydrodynamics
Stokes equations

$$\underbrace{-\eta \nabla^2 v}_{\text{momentum diffusion}} = \underbrace{-\nabla \pi}_{\text{internal stress}}$$

- The Stokes equations are the inertia-less version of the Incompressible Navier Stokes equations. And can be understood as a balance between the momentum diffusion of the fluid and the stress that it is subjected to.
- This stress comes from its internal pressure and any external stress, such as forces and thermal fluctuations.
- Here, Z is a tensor that is designed to comply with fluctuation-dissipation balance.

Hydrodynamics

Stokes equations

$$\underbrace{-\eta \nabla^2 v}_{\text{momentum diffusion}} = \underbrace{-\nabla \pi}_{\text{internal stress}} + \underbrace{\tilde{f}}_{\text{external stress}}$$

Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Hydrodynamics

2022-05-29

Hydrodynamics
Stokes equations

$$\underbrace{-\eta \nabla^2 v}_{\text{momentum diffusion}} = \underbrace{-\nabla \pi}_{\text{internal stress}} + \underbrace{\tilde{f}}_{\text{external stress}}$$

- The Stokes equations are the inertia-less version of the Incompressible Navier Stokes equations. And can be understood as a balance between the momentum diffusion of the fluid and the stress that it is subjected to.
- This stress comes from its internal pressure and any external stress, such as forces and thermal fluctuations.
- Here, Z is a tensor that is designed to comply with fluctuation-dissipation balance.

Hydrodynamics

Stokes equations

$$\underbrace{-\eta \nabla^2 v}_{\text{momentum diffusion}} = \underbrace{-\nabla \pi}_{\text{internal stress}} + \underbrace{\tilde{f}}_{\text{external stress}}$$
$$\underbrace{\nabla \cdot v = 0}_{\text{incompressibility}}$$

2022-05-29

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Hydrodynamics

Hydrodynamics
Stokes equations

$$\begin{aligned} \text{momentum diffusion} \\ -\eta \nabla^2 v &= \underbrace{-\nabla \pi}_{\text{internal stress}} + \underbrace{\tilde{f}}_{\text{external stress}} \\ \nabla \cdot v &= 0 \end{aligned}$$

incompressibility

- The Stokes equations are the inertia-less version of the Incompressible Navier Stokes equations. And can be understood as a balance between the momentum diffusion of the fluid and the stress that it is subjected to.
- This stress comes from its internal pressure and any external stress, such as forces and thermal fluctuations.
- Here, Z is a tensor that is designed to comply with fluctuation-dissipation balance.

Hydrodynamics

Fluctuating Stokes equations

$$\underbrace{-\eta \nabla^2 v}_{\text{momentum diffusion}} = \underbrace{-\nabla \pi}_{\text{internal stress}} + \underbrace{\tilde{f}}_{\text{external stress}}$$

$$\underbrace{\nabla \cdot v = 0}_{\text{incompressibility}}$$

$$\tilde{f}(r) := \underbrace{f(r)}_{\text{external forces}} + \underbrace{\nabla \cdot Z}_{\text{thermal noise}}$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Hydrodynamics

2022-05-29

Hydrodynamics
Fluctuating Stokes equations

$$\begin{aligned} \text{momentum diffusion} \quad -\eta \nabla^2 v &= -\nabla \pi + \underbrace{j}_{\substack{\text{internal stress} \\ \text{external stress}}} \\ \nabla \cdot v &= 0 \\ \tilde{f}(r) &:= \underbrace{f(r)}_{\substack{\text{external forces} \\ \text{thermal noise}}} + \underbrace{\nabla \cdot Z}_{\substack{\text{thermal noise} \\ \text{external forces}}} \end{aligned}$$

- The Stokes equations are the inertia-less version of the Incompressible Navier Stokes equations. And can be understood as a balance between the momentum diffusion of the fluid and the stress that it is subjected to.
- This stress comes from its internal pressure and any external stress, such as forces and thermal fluctuations.
- Here, Z is a tensor that is designed to comply with fluctuation-dissipation balance.

Hydrodynamics

Fluctuating Stokes equations

momentum diffusion

$$\overbrace{-\eta \nabla^2 v}^{\text{momentum diffusion}} = \underbrace{-\nabla \pi}_{\text{internal stress}} + \underbrace{\tilde{f}}_{\text{external stress}}$$

$\nabla \cdot v = 0$

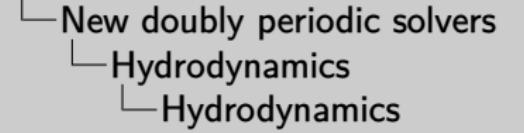
incompressibility

$$\tilde{f}(r) := \underbrace{f(r)}_{\text{external forces}} + \overbrace{\nabla \cdot Z}^{\text{thermal noise}}$$

$$\langle Z_{ij} \rangle = 0$$

$$\langle Z_{ik}(r, t) Z_{jm}(r', t') \rangle = 2k_B T \eta (\delta_{ij} \delta_{km} + \delta_{im} \delta_{kj}) \delta(r - r') \delta(t - t')$$

Complex fluids in the GPU era



2022-05-29

Hydrodynamics
Fluctuating Stokes equations

momentum diffusion

$$\overbrace{-\eta \nabla^2 v}^{\text{momentum diffusion}} = \underbrace{-\nabla \pi}_{\text{internal stress}} + \underbrace{\tilde{f}}_{\text{external stress}}$$

incompressibility

$$\nabla \cdot v = 0$$

thermal noise

$$\tilde{f}(r) := \underbrace{f(r)}_{\text{external forces}} + \overbrace{\nabla \cdot Z}^{\text{thermal noise}}$$
$$\langle Z_{ij} \rangle = 0$$
$$\langle Z_{ik}(r, t) Z_{jm}(r', t') \rangle = 2k_B T \eta (\delta_{ij} \delta_{km} + \delta_{im} \delta_{kj}) \delta(r - r') \delta(t - t')$$

- The Stokes equations are the inertia-less version of the Incompressible Navier Stokes equations. And can be understood as a balance between the momentum diffusion of the fluid and the stress that it is subjected to.
- This stress comes from its internal pressure and any external stress, such as forces and thermal fluctuations.
- Here, Z is a tensor that is designed to comply with fluctuation-dissipation balance.

Stokes equations

$$\begin{aligned}\nabla \pi - \eta \nabla^2 \mathbf{v} &= \mathbf{f} + \nabla \cdot \mathcal{Z} \\ \nabla \cdot \mathbf{v} &= 0\end{aligned}$$

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Hydrodynamics

2022-05-29

Stokes equations
 $\nabla \pi - \eta \nabla^2 \mathbf{v} = \mathbf{f} + \nabla \cdot \mathcal{Z}$
 $\nabla \cdot \mathbf{v} = 0$

- Interesting for us is that, if we couple the Stokes equations with a series of particles we can prove that the equations for the dynamics of the particles are equivalent to those of Brownian Dynamics.
- In particular, we are using the Immersed Boundary formalism. First by spreading particle forces as an external fluid forcing. Then imposing a no-slip condition on the surface of the particles by making them follow the local fluid exactly.

Hydrodynamics

Fluctuating Stokes equations

Stokes equations

$$\begin{aligned}\nabla \pi - \eta \nabla^2 \mathbf{v} &= \mathbf{f} + \nabla \cdot \mathcal{Z} \\ \nabla \cdot \mathbf{v} &= 0\end{aligned}$$

Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$$

Particle velocity

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Hydrodynamics

2022-05-29

Hydrodynamics
Fluctuating Stokes equations

Stokes equations
 $\nabla \pi - \eta \nabla^2 \mathbf{v} = \mathbf{f} + \nabla \cdot \mathcal{Z}$
 $\nabla \cdot \mathbf{v} = 0$

Fluid forcing
 $\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$

Particle velocity
 $\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$

- Interesting for us is that, if we couple the Stokes equations with a series of particles we can prove that the equations for the dynamics of the particles are equivalent to those of Brownian Dynamics.
- In particular, we are using the Immersed Boundary formalism. First by spreading particle forces as an external fluid forcing. Then imposing a no-slip condition on the surface of the particles by making them follow the local fluid exactly.

Stokes equations

$$\nabla \pi - \eta \nabla^2 \mathbf{v} = \mathbf{f} + \nabla \cdot \mathcal{Z}$$

$$\nabla \cdot \mathbf{v} = 0$$

Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r})\mathbf{F}$$

Particle velocity

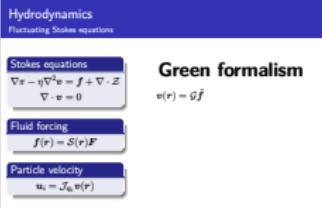
$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

Green formalism

$$\mathbf{v}(\mathbf{r}) = \mathcal{G}\tilde{\mathbf{f}}$$

2022-05-29

- Complex fluids in the GPU era
 - └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Hydrodynamics



Let me dwell a little bit more into this. We use the Green formalism to construct an operator, G , that transforms forces to velocities in the fluid.

Hydrodynamics

Fluctuating Stokes equations

Stokes equations

$$\begin{aligned}\nabla \pi - \eta \nabla^2 \mathbf{v} &= \mathbf{f} + \nabla \cdot \mathcal{Z} \\ \nabla \cdot \mathbf{v} &= 0\end{aligned}$$

Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$$

Particle velocity

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Hydrodynamics

2022-05-29

Where this operator is applied via a convolution.

Hydrodynamics
Fluctuating Stokes equations

Stokes equations
$\nabla \pi - \eta \nabla^2 \mathbf{v} = \mathbf{f} + \nabla \cdot \mathcal{Z}$
$\nabla \cdot \mathbf{v} = 0$
$v(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$
Fluid forcing
$f(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$
Particle velocity
$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$

Green formalism
$v(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$
Fluid forcing
$f(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$

Hydrodynamics

Fluctuating Stokes equations

Stokes equations

$$\begin{aligned}\nabla \pi - \eta \nabla^2 \mathbf{v} &= \mathbf{f} + \nabla \cdot \mathcal{Z} \\ \nabla \cdot \mathbf{v} &= 0\end{aligned}$$

Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$$

Particle velocity

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

Green formalism

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$$

Force Coupling Method (FCM)

$$\frac{d\mathbf{q}_i}{dt} = \mathbf{u}_i = \mathcal{J}_{q_i} \mathcal{G}(\mathcal{S} \mathbf{F} + \nabla \cdot \mathcal{Z})$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Hydrodynamics

2022-05-29

From the particle's perspective, putting all the operators together we can write the following equation. Where we input particle forces and get back particle displacements. We call this the Force Coupling Method.

Hydrodynamics

Fluctuating Stokes equations

Stokes equations

$$\nabla \pi - \eta \nabla^2 \mathbf{v} = \mathbf{f} + \nabla \cdot \mathcal{Z}$$

$$\nabla \cdot \mathbf{v} = 0$$

Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$$

Particle velocity

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

Green formalism

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$$

Force Coupling Method (FCM)

$$\frac{d\mathbf{q}_i}{dt} = \mathbf{u}_i = \mathcal{J}_{q_i} \mathcal{G}(\mathcal{S} \mathbf{F} + \nabla \cdot \mathcal{Z})$$

Hydrodynamics

Fluctuating Stokes equations

Stokes equations

$$\begin{aligned}\nabla \pi - \eta \nabla^2 \mathbf{v} &= \mathbf{f} + \nabla \cdot \mathcal{Z} \\ \nabla \cdot \mathbf{v} &= 0\end{aligned}$$

Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$$

Particle velocity

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

Green formalism

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$$

Force Coupling Method (FCM)

$$\frac{d\mathbf{q}_i}{dt} = \mathbf{u}_i = \mathcal{J}_{q_i} \mathcal{G}(\mathcal{S} \mathbf{F} + \nabla \cdot \mathcal{Z})$$

$$\mathcal{M} = \mathcal{J} \mathcal{G} \mathcal{S}$$

$$\mathcal{M}^{1/2} = \mathcal{J} \mathcal{G} \nabla \cdot$$

$$\partial_q \cdot \mathcal{M} = 0 \leftarrow \text{Incompressible}$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Hydrodynamics

2022-05-29

Hydrodynamics
Fluctuating Stokes equations

Stokes equations
 $\nabla \pi - \eta \nabla^2 \mathbf{v} = \mathbf{f} + \nabla \cdot \mathcal{Z}$
 $\nabla \cdot \mathbf{v} = 0$

Fluid forcing
 $\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$

Particle velocity
 $\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$
 $\mathcal{M} = \mathcal{J} \mathcal{G} \mathcal{S}$
 $\mathcal{M}^{1/2} = \mathcal{J} \mathcal{G} \nabla \cdot$
 $\partial_q \cdot \mathcal{M} = 0 \leftarrow \text{Incompressible}$

Green formalism
 $\mathbf{v}(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$

Force Coupling Method (FCM)
 $\frac{d\mathbf{q}_i}{dt} = \mathbf{u}_i = \mathcal{J}_{q_i} \mathcal{G}(\mathcal{S} \mathbf{F} + \nabla \cdot \mathcal{Z})$

- As I mentioned earlier, by carefully defining the mobility like this we can see that this form is, in fact, equivalent to the Brownian Dynamics equation.
- We are still far from done, though. Writing the mobility operator in a functional form...

Hydrodynamics

Fluctuating Stokes equations

Stokes equations

$$\begin{aligned}\nabla \pi - \eta \nabla^2 \mathbf{v} &= \mathbf{f} + \nabla \cdot \mathcal{Z} \\ \nabla \cdot \mathbf{v} &= 0\end{aligned}$$

Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$$

Particle velocity

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

Green formalism

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$$

Force Coupling Method (FCM)

$$\frac{d\mathbf{q}_i}{dt} = \mathbf{u}_i = \mathcal{J}_{q_i} \mathcal{G}(\mathcal{S} \mathbf{F} + \nabla \cdot \mathcal{Z})$$

$$\mathcal{M}_{ij} = \iint \delta_a(\mathbf{r} - \mathbf{q}_j) \mathcal{G}(\mathbf{r}, \mathbf{r}') \delta_a(\mathbf{q}_i - \mathbf{r}') d\mathbf{r}' d\mathbf{r}$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Hydrodynamics

2022-05-29

Hydrodynamics

Fluctuating Stokes equations

Stokes equations

$$\begin{aligned}\nabla \pi - \eta \nabla^2 \mathbf{v} &= \mathbf{f} + \nabla \cdot \mathcal{Z} \\ \nabla \cdot \mathbf{v} &= 0\end{aligned}$$

Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$$

Particle velocity

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

Green formalism

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$$

Force Coupling Method (FCM)

$$\frac{d\mathbf{q}_i}{dt} = \mathbf{u}_i = \mathcal{J}_{q_i} \mathcal{G}(\mathcal{S} \mathbf{F} + \nabla \cdot \mathcal{Z})$$

$$\mathcal{G} := -\eta^{-1} \nabla^{-2} (\mathbb{I} - \nabla \nabla^{-2} \nabla)$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Hydrodynamics

2022-05-29

Hydrodynamics	
Fluctuating Stokes equations	
Stokes equations	Green formalism
$\nabla \pi - \eta \nabla^2 \mathbf{v} = \mathbf{f} + \nabla \cdot \mathcal{Z}$	$v(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$
$\nabla \cdot \mathbf{v} = 0$	Force Coupling Method (FCM)
$f(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$	$\frac{d\mathbf{q}_i}{dt} = \mathbf{u}_i = \mathcal{J}_{q_i} \mathcal{G}(\mathcal{S} \mathbf{F} + \nabla \cdot \mathcal{Z})$
$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$	$\mathcal{G} := -\eta^{-1} \nabla^{-2} (\mathbb{I} - \nabla \nabla^{-2} \nabla)$

And I have not even showed you yet the form of the Green operator. Yes, that is an inverse laplacian, whatever that means.

Hydrodynamics

Fluctuating Stokes equations

Stokes equations

$$\begin{aligned}\nabla \pi - \eta \nabla^2 \mathbf{v} &= \mathbf{f} + \nabla \cdot \mathcal{Z} \\ \nabla \cdot \mathbf{v} &= 0\end{aligned}$$

Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$$

Particle velocity

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

Green formalism

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$$

Force Coupling Method (FCM)

$$\frac{d\mathbf{q}_i}{dt} = \mathbf{u}_i = \mathcal{J}_{q_i} \mathcal{G}(\mathcal{S} \mathbf{F} + \nabla \cdot \mathcal{Z})$$

$$\mathcal{G} := -\eta^{-1} \nabla^{-2} (\mathbb{I} - \nabla \nabla^{-2} \nabla)$$

Open boundaries

$$\mathcal{G} \rightarrow \mathcal{O}(\mathbf{r}) = \frac{1}{8\pi\eta r} \left(\mathbb{I} - \frac{\mathbf{r} \otimes \mathbf{r}}{r^2} \right)$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Hydrodynamics

2022-05-29

Hydrodynamics

Fluctuating Stokes equations

Stokes equations

$$\begin{aligned}\nabla \pi - \eta \nabla^2 \mathbf{v} &= \mathbf{f} + \nabla \cdot \mathcal{Z} \\ \nabla \cdot \mathbf{v} &= 0\end{aligned}$$

Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r}) \mathbf{F}$$

Particle velocity

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

Open boundaries

$$\mathcal{G} \rightarrow \mathcal{O}(\mathbf{r}) = \frac{1}{8\pi\eta r} \left(\mathbb{I} - \frac{\mathbf{r} \otimes \mathbf{r}}{r^2} \right)$$

Green formalism

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} \tilde{\mathbf{f}} := \int \mathcal{G}(\mathbf{r}, \mathbf{r}') \tilde{\mathbf{f}}(\mathbf{r}') d\mathbf{r}'$$

Force Coupling Method (FCM)

$$\frac{d\mathbf{q}_i}{dt} = \mathbf{u}_i = \mathcal{J}_{q_i} \mathcal{G}(\mathcal{S} \mathbf{F} + \nabla \cdot \mathcal{Z})$$

Open boundaries

$$\mathcal{G} \rightarrow \mathcal{O}(\mathbf{r}) = \frac{1}{8\pi\eta r} \left(\mathbb{I} - \frac{\mathbf{r} \otimes \mathbf{r}}{r^2} \right)$$

To be fair, sometimes we do have a "simple" expression for G, but integrating over all space for every point in the domain is still a no go.

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathcal{G}(\mathcal{S}\mathbf{F} + \nabla \cdot \mathcal{Z})$$

2022-05-29

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Hydrodynamics

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathcal{G}(\mathcal{S}\mathbf{F} + \nabla \cdot \mathcal{Z})$$

- Turns out there is a quite efficient way to overcome this if we impose periodic boundary conditions.

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathcal{G}(\mathcal{S}\mathbf{F} + \nabla \cdot \mathcal{Z})$$

\downarrow

\mathfrak{F} : Fourier transform

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \quad \mathcal{G}(\mathcal{S}\mathbf{F} + \nabla \cdot \mathcal{Z})$$

2022-05-29

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Hydrodynamics
 - └ Hydrodynamics

Hydrodynamics
Triply periodic FCM

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathcal{G}(\mathcal{S}\mathbf{F} + \nabla \cdot \mathcal{Z})$$

\mathfrak{F} : Fourier transform

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \quad \mathcal{G}(\mathcal{S}\mathbf{F} + \nabla \cdot \mathcal{Z})$$

- In that case we can go to Fourier space (which is fast in the GPU).

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathcal{G}(\mathcal{S}\mathbf{F} + \nabla \cdot \mathcal{Z})$$

\downarrow
 \mathfrak{F} : Fourier transform

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \widehat{\mathcal{G}}(\mathfrak{F} \mathcal{S}\mathbf{F} + \mathbf{k} \cdot \mathcal{Z}_k)$$

2022-05-29

Complex fluids in the GPU era
└ New doubly periodic solvers
 └ Hydrodynamics
 └ Hydrodynamics

Hydrodynamics
Triply periodic FCM

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathcal{G}(\mathcal{S}\mathbf{F} + \nabla \cdot \mathcal{Z})$$

\mathfrak{F} : Fourier transform

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \widehat{\mathcal{G}}(\mathfrak{F} \mathcal{S}\mathbf{F} + \mathbf{k} \cdot \mathcal{Z}_k)$$

- And in Fourier space convolutions are just algebraic multiplications.
So we can convolve "for free".
- Bonus point, fluctuations are also free for us to compute.

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathfrak{F}^{-1} \widehat{\mathcal{G}} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z}_k)$$

Triply Periodic

$$\widehat{\mathcal{G}}_{3D} := \eta^{-1} k^{-2} \left(\mathbb{I} - \frac{\mathbf{k} \otimes \mathbf{k}}{k^2} \right)$$

Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Hydrodynamics

2022-05-29

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathfrak{F}^{-1} \widehat{\mathcal{G}} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z}_k)$$

Triply Periodic

$$\widehat{\mathcal{G}}_{3D} := \eta^{-1} k^{-2} \left(\mathbb{I} - \frac{\mathbf{k} \otimes \mathbf{k}}{k^2} \right)$$

- So we do the cheap stuff, spreading and interpolation, in real space and leave the convolutions and the noise to Fourier space.
- Finally, in this specific case, we have an analytic form of the Greens function in Fourier space.

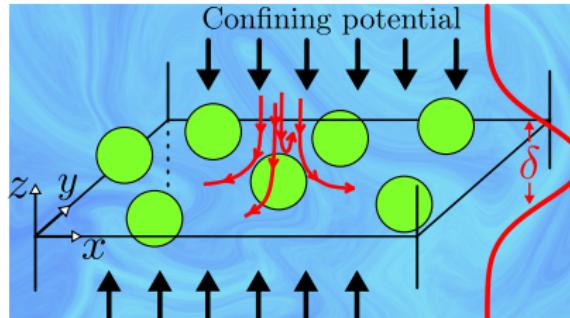
Hydrodynamics in confined geometries

New algorithms for Brownian Dynamics with Hydrodynamic Interactions (BDHI)

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$$

- Quasi two-dimensional (q2D)



q2D: $\delta \rightarrow 0$.

3D fluid, 2D particles

[Raul P. Pelaez et. al. JSTAT 2018.]

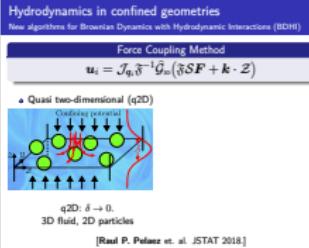
Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Hydrodynamics in confined geometries

2022-05-29



- Now I want to introduce some modifications to the Force Coupling Method for other kinds of geometries.
- I will start with what we call a quasi two dimensional system.
- Imagine a fluid that is periodic in the plane but open in the perpendicular direction.
- We embed particles in this fluid and confine them with some external potential. The quasi two dimensional limit happens when this force becomes a hard constraint and the particles move strictly in a plane.

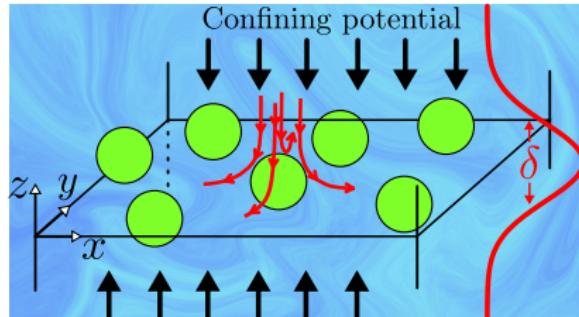
Hydrodynamics in confined geometries

New algorithms for Brownian Dynamics with Hydrodynamic Interactions (BDHI)

Force Coupling Method

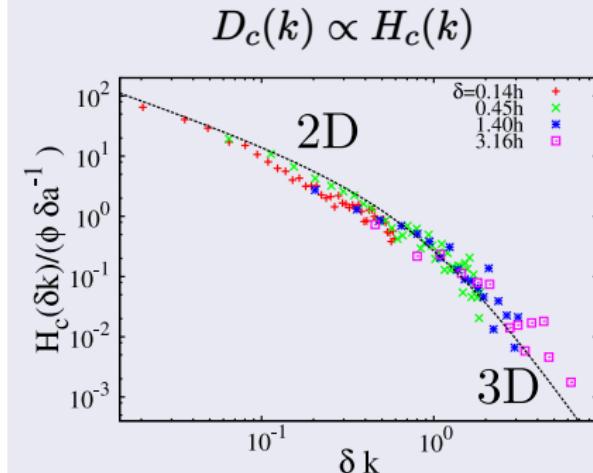
$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$$

- Quasi two-dimensional (q2D)



q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

Collective diffusion



[Raul P. Pelaez et. al. JSTAT 2018.]

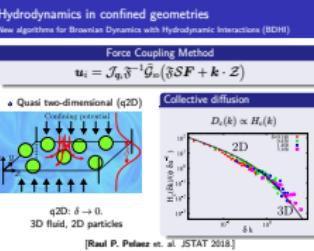
Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Hydrodynamics in confined geometries

2022-05-29



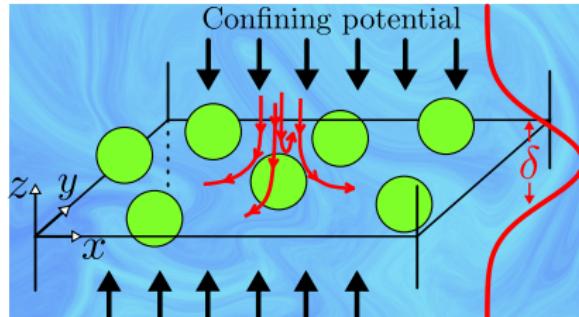
Hydrodynamics in confined geometries

New algorithms for Brownian Dynamics with Hydrodynamic Interactions (BDHI)

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$$

- Quasi two-dimensional (q2D)

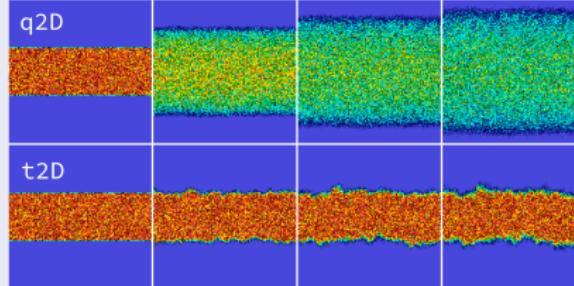


q2D: $\delta \rightarrow 0$.

3D fluid, 2D particles

$D_c(k) > 0$ in q2D

Correlated density fluctuations



t2D: 2D fluid and particles

[Raul P. Pelaez et. al. JSTAT 2018.]

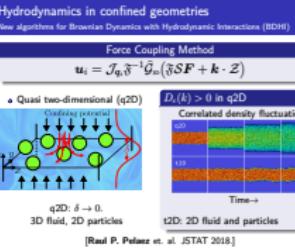
Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Hydrodynamics in confined geometries

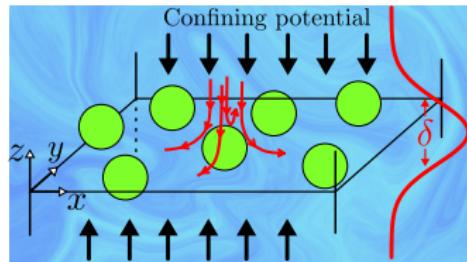
2022-05-29



Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$$



q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

[Raul P. Pelaez et. al. JSTAT 2018.]

Complex fluids in the GPU era

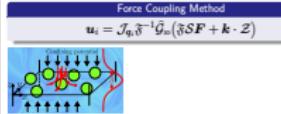
└ New doubly periodic solvers

└ Hydrodynamics

└ Force Coupling Method in quasi 2D

2022-05-29

Force Coupling Method in quasi 2D

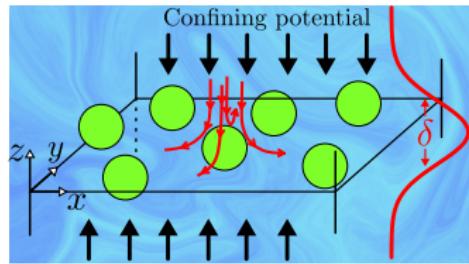


[Raul P. Pelaez et. al. JSTAT 2018]

Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$$



q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

[Raul P. Pelaez et. al. JSTAT 2018.]

Complex fluids in the GPU era

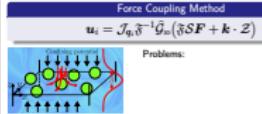
└ New doubly periodic solvers

└ Hydrodynamics

└ Force Coupling Method in quasi 2D

2022-05-29

Force Coupling Method in quasi 2D



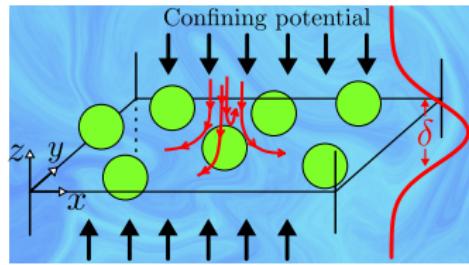
[Raul P. Pelaez et. al. JSTAT 2018]

We cannot use triply periodic algorithm directly because in doing so we would have to...

Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$$



q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

[Raul P. Pelaez et. al. JSTAT 2018.]

Complex fluids in the GPU era

└ New doubly periodic solvers

└ Hydrodynamics

└ Force Coupling Method in quasi 2D

2022-05-29

...make the confining force infinite

Force Coupling Method in quasi 2D

Force Coupling Method
 $\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$

Problems:
■ 2D limit implies $F_z \rightarrow \infty$

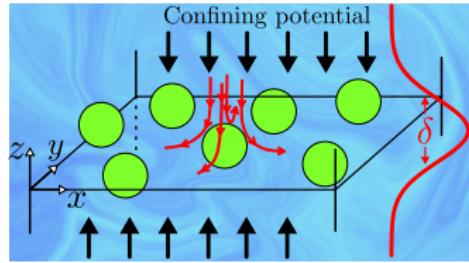
q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

[Raul P. Pelaez et. al. JSTAT 2018]

Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$$



q2D: $\delta \rightarrow 0.$
3D fluid, 2D particles

[Raul P. Pelaez et. al. JSTAT 2018.]

Complex fluids in the GPU era

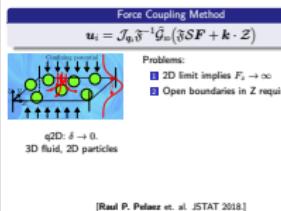
└ New doubly periodic solvers

└ Hydrodynamics

└ Force Coupling Method in quasi 2D

2022-05-29

Force Coupling Method in quasi 2D

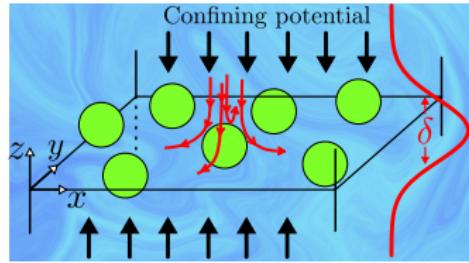


And increase the simulation domain to mimic an open boundary.

Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$$



Problems:

- 1 2D limit implies $F_z \rightarrow \infty$
- 2 Open boundaries in Z require $L_z \rightarrow \infty$

Solution:

q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

[Raul P. Pelaez et. al. JSTAT 2018.]

Complex fluids in the GPU era

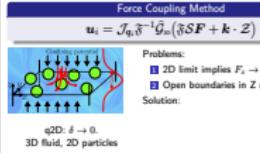
New doubly periodic solvers

Hydrodynamics

Force Coupling Method in quasi 2D

2022-05-29

Force Coupling Method in quasi 2D

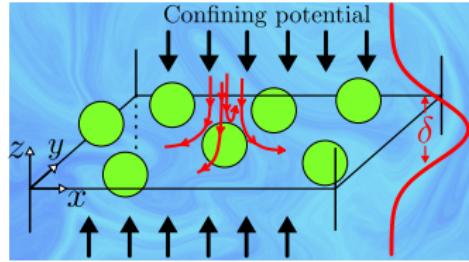


Both of these problems disappear if we eliminate the Z direction from our description.

Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$$



q2D: $\delta \rightarrow 0.$
3D fluid, 2D particles

Problems:

- 1 2D limit implies $F_z \rightarrow \infty$
- 2 Open boundaries in Z require $L_z \rightarrow \infty$

Solution:

- 1 Eliminate Z direction

2022-05-29

Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Force Coupling Method in quasi 2D

Doing so requires a series of changes.

Force Coupling Method in quasi 2D

$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$

Problems:

- 1 2D limit implies $F_z \rightarrow \infty$
- 2 Open boundaries in Z require $L_z \rightarrow \infty$

Solution:

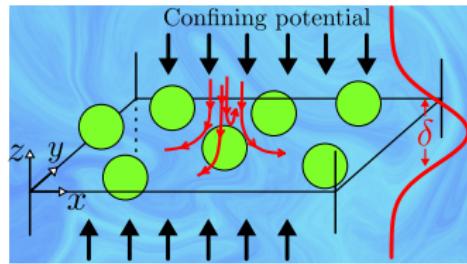
- 1 Eliminate Z direction

[Raul P. Pelaez et. al. JSTAT 2018]

Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} \hat{\mathcal{G}}_{3D} (\mathfrak{F} \mathcal{S} \mathbf{F} + \mathbf{k} \cdot \mathcal{Z})$$



q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

[Raul P. Pelaez et. al. JSTAT 2018.]

Complex fluids in the GPU era

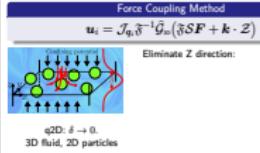
└ New doubly periodic solvers

└ Hydrodynamics

└ Force Coupling Method in quasi 2D

2022-05-29

Force Coupling Method in quasi 2D

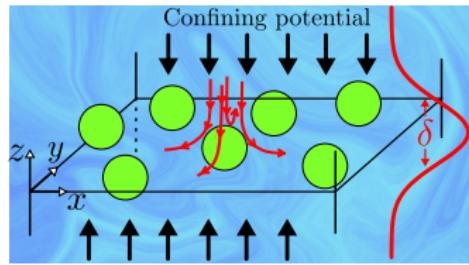


First, let's get fluctuations out of the convolution. We can reintroduce them later ad-hoc with some form that satisfies fluctuation-dissipation balance.

Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathfrak{F}^{-1} (\hat{\mathcal{G}}_{\text{3D}} \mathfrak{F} \mathcal{S} \mathbf{F} + \hat{\mathbf{w}})$$



q2D: $\delta \rightarrow 0.$

3D fluid, 2D particles

$$\langle \hat{\mathbf{w}} \otimes \hat{\mathbf{w}} \rangle = 2k_B T \hat{\mathcal{G}}$$

Eliminate Z direction:

- 1 Fluctuations via FDT

Complex fluids in the GPU era

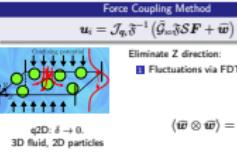
└ New doubly periodic solvers

└ Hydrodynamics

└ Force Coupling Method in quasi 2D

2022-05-29

Force Coupling Method in quasi 2D



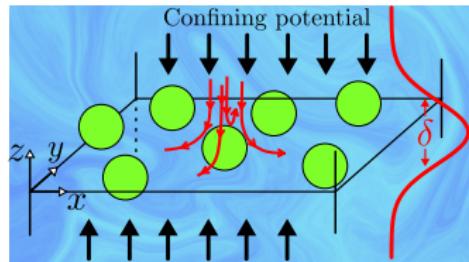
[Raul P. Pelaez et. al. JSTAT 2018]

First, let's get fluctuations out of the convolution. We can reintroduce them later ad-hoc with some form that satisfies fluctuation-dissipation balance.

Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathfrak{F}^{-1} (\hat{\mathcal{G}}_{3D} \mathfrak{F} \mathcal{S} \mathbf{F} + \hat{\mathbf{w}})$$



q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

Eliminate Z direction:

- 1 Fluctuations via FDT
- 2 Preconvolve \mathcal{G} in Z

Remember

$\delta_a(\mathbf{r}) := \phi(r_x)\phi(r_y)\phi(r_z) \rightarrow$ Smeared delta

[Raul P. Pelaez et. al. JSTAT 2018.]

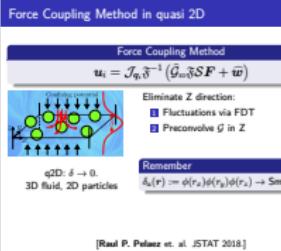
Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Force Coupling Method in quasi 2D

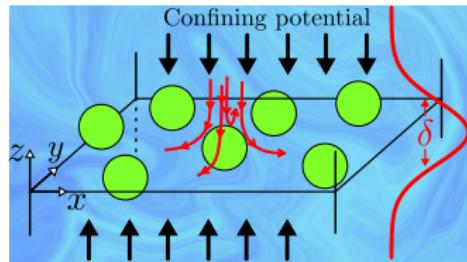
2022-05-29



Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{q_i}^{\parallel} \mathfrak{F}^{-1} (\hat{\mathcal{G}}_{\text{q2D}} \mathfrak{F} \mathcal{S}^{\parallel} \mathbf{F} + \hat{\mathbf{w}})$$



q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

Eliminate Z direction:

- 1 Fluctuations via FDT
- 2 Preconvolve \mathcal{G} in Z

Remember

$\delta_a(\mathbf{r}) := \phi(r_x)\phi(r_y)\phi(r_z) \rightarrow \text{Smeared delta}$

$$\hat{\mathcal{G}}_{\text{q2D}}(\mathbf{k}^{\parallel}) = \frac{1}{2\pi} \int \hat{\phi}(k_z)^2 \hat{\mathcal{G}}_{\text{3D}}(\mathbf{k}^{\parallel}, k_z) dk_z$$

[Raul P. Pelaez et. al. JSTAT 2018.]

Complex fluids in the GPU era

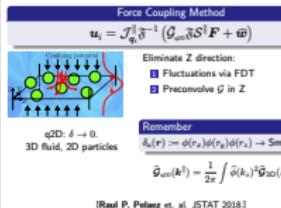
New doubly periodic solvers

Hydrodynamics

Force Coupling Method in quasi 2D

2022-05-29

Force Coupling Method in quasi 2D

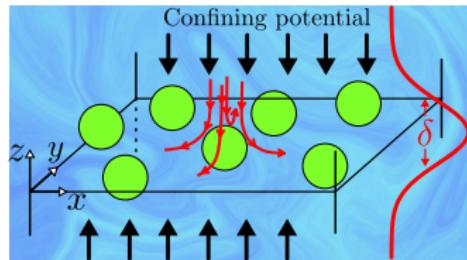


Since the kernel is analytical (in particular we use a Gaussian), and we have the expression of the three-dimensional Greens function, we can simply integrate in the Z direction to get a new Greens function defined only in the plane. Note that spreading and interpolation now happen also in the plane.

Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{q_i}^{\parallel} \mathfrak{F}^{-1} (\hat{\mathcal{G}}_{\text{q2D}} \mathfrak{F} \mathcal{S}^{\parallel} \mathbf{F} + \hat{\mathbf{w}})$$



q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

$$\mathcal{M}_{\text{q2D}} := \mathcal{J}^{\parallel} \mathcal{G}_{\text{q2D}} \mathcal{S}^{\parallel} \quad \hat{\mathcal{G}}_{\text{q2D}}(\mathbf{k}) = \eta^{-1} (g(k) \mathbf{k}^{\perp} \otimes \mathbf{k}^{\perp} + f(k) \mathbf{k} \otimes \mathbf{k})$$
$$k_B T \partial_q \cdot \mathcal{M}_{\text{q2D}} \neq 0$$

[Raul P. Pelaez et. al. JSTAT 2018.]

Complex fluids in the GPU era

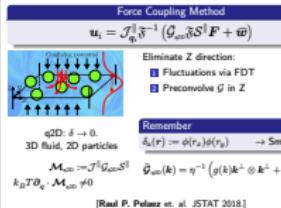
New doubly periodic solvers

Hydrodynamics

Force Coupling Method in quasi 2D

2022-05-29

Force Coupling Method in quasi 2D

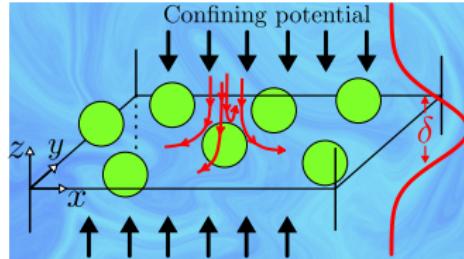


We end up with a purely two-dimensional description with an analytic form for G . Alas, if we study the mobility, defined as we did before, we see that its divergence, the thermal drift, is different from zero. This speaks of the fact that the flow in the plane is effectively compressible.

Force Coupling Method in quasi 2D

Force Coupling Method

$$\mathbf{u}_i = \mathcal{J}_{\mathbf{q}_i}^{\parallel} \mathfrak{F}^{-1} \left(\hat{\mathcal{G}}_{\text{q2D}} \mathfrak{F} [\mathcal{S}^{\parallel} \mathbf{F} + \partial_{\mathbf{q}} \mathcal{S}^{\parallel} (k_B T)] + \hat{\mathbf{w}} \right)$$



q2D: $\delta \rightarrow 0$.
3D fluid, 2D particles

$$\mathcal{M}_{\text{q2D}} := \mathcal{J}^{\parallel} \mathcal{G}_{\text{q2D}} \mathcal{S}^{\parallel}$$

$$k_B T \partial_{\mathbf{q}} \cdot \mathcal{M}_{\text{q2D}} \neq 0$$

Remember

$$\delta_a(\mathbf{r}) := \phi(r_x) \phi(r_y) \quad \rightarrow \text{Smeared delta}$$

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}^{\parallel} \mathbf{F} + \partial_{\mathbf{q}} \mathcal{S}^{\parallel} [k_B T]$$

[Raul P. Pelaez et. al. JSTAT 2018.]

Complex fluids in the GPU era

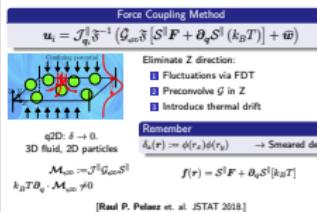
New doubly periodic solvers

Hydrodynamics

Force Coupling Method in quasi 2D

2022-05-29

Force Coupling Method in quasi 2D

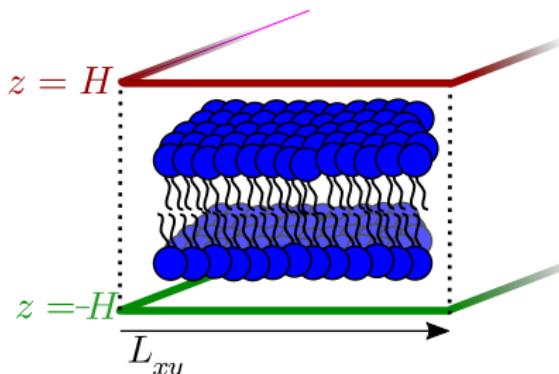


- It is not trivial to show that the thermal drift, that comes from the confining force, appears as an entropic force coming from a pressure gradient and has this particular shape.
- Maybe more intuitively, the force coming from the confining potential induces a flow which acts as a source of fluid momentum in the plane. And in fact enhances collective diffusion.
- This concludes the adaptation of the Force Coupling method to quasi two-dimensional geometries. And it is worth mentioning that we have some leeway when choosing the Green's function here, allowing to reuse the same algorithm for other kinds of strongly confined systems. Such as when the fluid is also two dimensional.

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$



Fluid forcing

$$\mathbf{f}(\mathbf{r}) = \mathcal{S}(\mathbf{r})\mathbf{F}$$

Particle velocity

$$\mathbf{u}_i = \mathcal{J}_{q_i} \mathbf{v}(\mathbf{r})$$

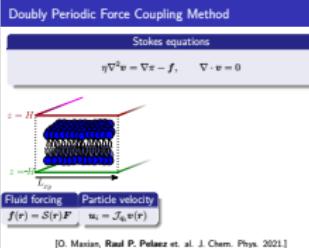
Complex fluids in the GPU era

└ New doubly periodic solvers

└ Hydrodynamics

└ Doubly Periodic Force Coupling Method

2022-05-29

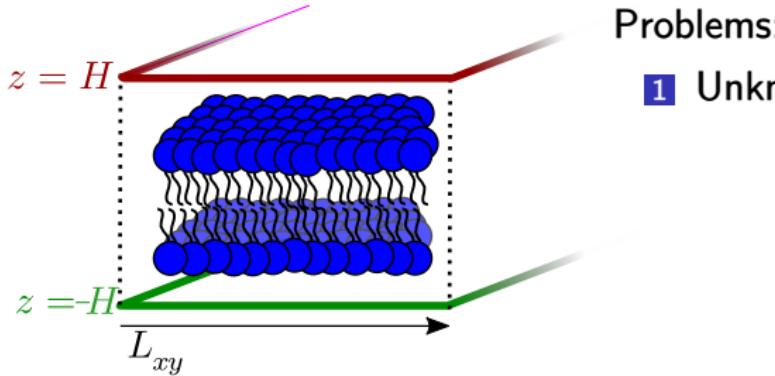


- The quasi two-dimensional framework is based on the assumption that the embedded particles are not allowed to leave the plane. It is of no use if we want to simulate something anisotropic, like a lipid bilayer.
- I will now introduce a Force coupling-like method to simulate a doubly periodic domain. That is periodic in the plane and open in the third direction, while having an arbitrary width.
- We will again use the same Stokes equations, but changing the boundary conditions requires a lot of rework.

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$



Problems:

- 1 Unknown Green's function

Complex fluids in the GPU era

└ New doubly periodic solvers

 └ Hydrodynamics

 └ Doubly Periodic Force Coupling Method

2022-05-29

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$

Problems:

- Unknown Green's function

A diagram of a fluid simulation domain. The domain is a rectangular prism with dimensions L_x , L_y , and L_z . The top boundary is labeled $z = H$ and the bottom boundary is labeled $z = -H$. The front boundary is labeled $x = 0$ and the back boundary is labeled $x = L_x$. The left boundary is labeled $y = 0$ and the right boundary is labeled $y = L_y$. The top surface contains a grid of points representing a boundary condition. A legend indicates that a red square represents "Unknown Green's function".

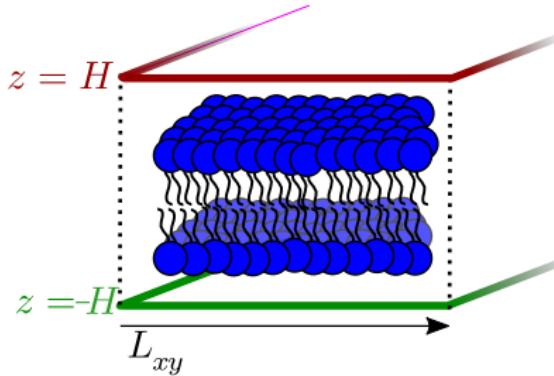
[O. Maxian, Raúl P. Peláez et al. J. Chem. Phys. 2021]

For starters, we do not know the Green's function, and our previous pre-convolution trick does not apply.

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$



Problems:

- 1 Unknown Green's function
- 2 Cannot use \mathfrak{F} in Z

Complex fluids in the GPU era

└ New doubly periodic solvers

└ Hydrodynamics

└ Doubly Periodic Force Coupling Method

2022-05-29

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$

Problems:

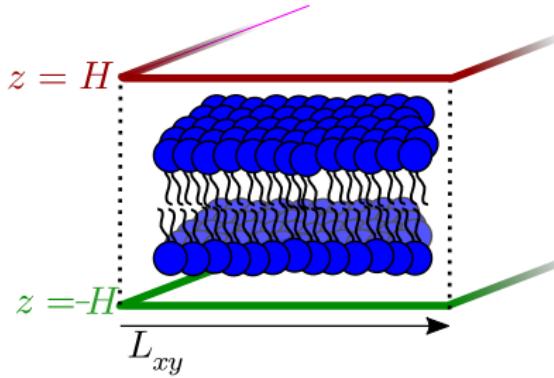
- Unknown Green's function
- Cannot use \mathfrak{F} in Z

[O. Maxian, Raúl P. Peláez et. al. J. Chem. Phys. 2021.]

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$



Problems:

- 1 Unknown Green's function
- 2 Cannot use \mathfrak{F} in Z

Solutions:

- 1 Solve Stokes directly

Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Doubly Periodic Force Coupling Method

2022-05-29

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$

Problems:

- Unknown Green's function
- Cannot use \mathfrak{F} in Z

Solutions:

- Solve Stokes directly

$z = H$

$z = -H$

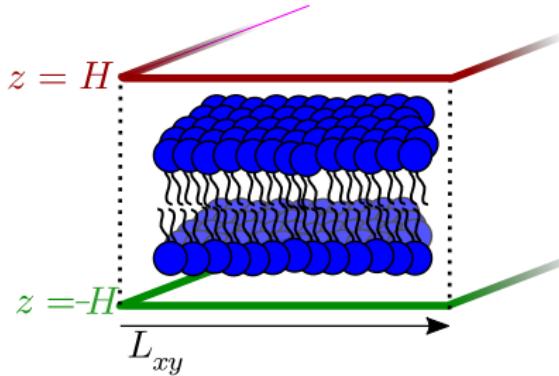
L_{xy}

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$



Problems:

- 1 Unknown Green's function
- 2 Cannot use \mathfrak{F} in Z

Solutions:

- 1 Solve Stokes directly
- 2 \mathfrak{F} only in XY plane

Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Doubly Periodic Force Coupling Method

2022-05-29

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$

Problems:

- Unknown Green's function
- Cannot use \mathfrak{F} in Z

Solutions:

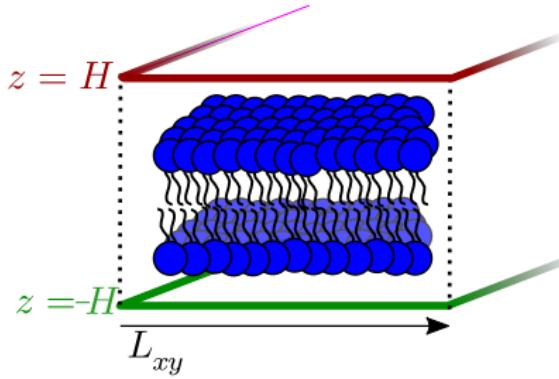
- Solve Stokes directly
- \mathfrak{F} only in XY plane

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$



\mathfrak{F} only in XY plane

$$\nabla \rightarrow [ik_x, ik_y, \partial_z] := [i\mathbf{k}, \partial_z]$$

$$\mathbf{f} \rightarrow [\hat{f}_x, \hat{f}_y, f_z] := \hat{\mathbf{f}}(\mathbf{k}, z)$$

2022-05-29

Complex fluids in the GPU era

└ New doubly periodic solvers

 └ Hydrodynamics

 └ Doubly Periodic Force Coupling Method

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \nabla^2 \mathbf{v} = \nabla \pi - \mathbf{f}, \quad \nabla \cdot \mathbf{v} = 0$$

\mathfrak{F} only in XY plane

$$\nabla \rightarrow [ik_x, ik_y, \partial_z] := [i\mathbf{k}, \partial_z]$$
$$\mathbf{f} \rightarrow [\hat{f}_x, \hat{f}_y, f_z] := \hat{\mathbf{f}}(\mathbf{k}, z)$$

[O. Maxian, Raúl P. Peláez et. al. J. Chem. Phys. 2021]

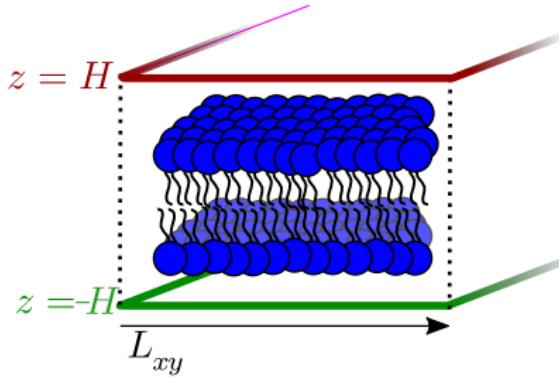
A diagram showing a rectangular domain in the XY plane. The top boundary is labeled $z = H$ and the bottom boundary is labeled $z = -H$. The width of the domain is labeled L_{xy} . Inside the domain, there is a layer of blue spheres representing a fluid, with wavy lines indicating its motion. A pink arrow points upwards from the XY plane towards the top boundary. A green arrow points downwards from the bottom boundary towards the bottom boundary. A red arrow points outwards from the right boundary.

Lets do just that. Now the different quantities depend on a wave vector and a height.

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} i\mathbf{k} \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [i\mathbf{k}, \partial_z] \cdot \hat{\mathbf{v}} = 0$$



$\hat{\mathbf{f}}$ only in XY plane

$$\nabla \rightarrow [ik_x, ik_y, \partial_z] := [i\mathbf{k}, \partial_z]$$

$$\mathbf{f} \rightarrow [\hat{f}_x, \hat{f}_y, f_z] := \hat{\mathbf{f}}(\mathbf{k}, z)$$

2022-05-29

Complex fluids in the GPU era

└ New doubly periodic solvers

 └ Hydrodynamics

 └ Doubly Periodic Force Coupling Method

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} i\mathbf{k} \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [i\mathbf{k}, \partial_z] \cdot \hat{\mathbf{v}} = 0$$

$\hat{\mathbf{f}}$ only in XY plane

$\nabla \rightarrow [ik_x, ik_y, \partial_z] := [i\mathbf{k}, \partial_z]$

$\mathbf{f} \rightarrow [\hat{f}_x, \hat{f}_y, f_z] := \hat{\mathbf{f}}(\mathbf{k}, z)$

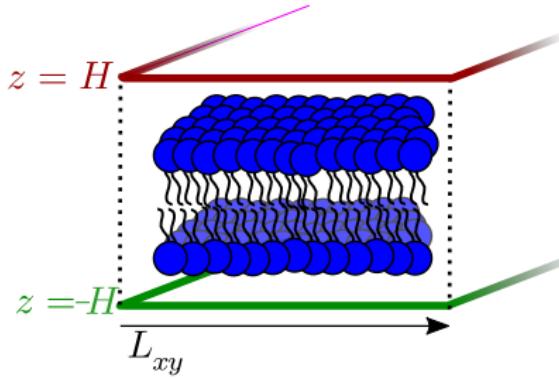
[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021]

And the Stokes equation is transformed into a set of equations independent for each wave vector.

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} i\mathbf{k} \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [i\mathbf{k}, \partial_z] \cdot \hat{\mathbf{v}} = 0$$



$\hat{\mathbf{f}}$ only in XY plane

$$\nabla \rightarrow [ik_x, ik_y, \partial_z] := [i\mathbf{k}, \partial_z]$$

$$\mathbf{f} \rightarrow [\hat{f}_x, \hat{f}_y, f_z] := \hat{\mathbf{f}}(\mathbf{k}, z)$$

$$\nabla^2 \pi = \nabla \cdot \mathbf{f}$$

Complex fluids in the GPU era

└ New doubly periodic solvers

└ Hydrodynamics

└ Doubly Periodic Force Coupling Method

2022-05-29

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} i\mathbf{k} \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [i\mathbf{k}, \partial_z] \cdot \hat{\mathbf{v}} = 0$$

$\hat{\mathbf{v}}$ only in XY plane

$$\nabla \rightarrow [ik_x, ik_y, \partial_z] := [i\mathbf{k}, \partial_z]$$

$$\mathbf{f} \rightarrow [\hat{f}_x, \hat{f}_y, f_z] := \hat{\mathbf{f}}(\mathbf{k}, z)$$

$$\nabla^2 \pi = \nabla \cdot \mathbf{f}$$

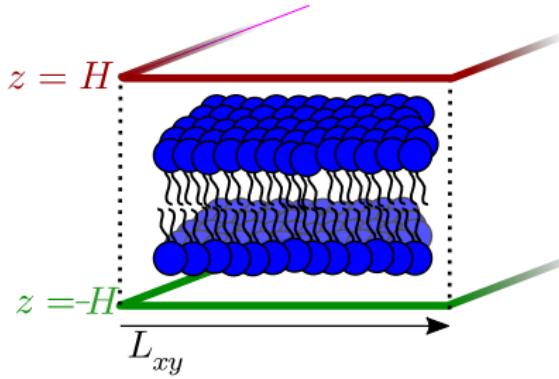
[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021]

There are two unknowns in the Stokes equation: velocity and pressure.
But we can use the incompressibility condition to eliminate the pressure.

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} ik \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [ik, \partial_z] \cdot \hat{\mathbf{v}} = 0$$



BVP for each \mathbf{k} . Embarrassingly parallel

$$\begin{aligned} \left(\partial_z^2 - k^2 \right) \hat{\pi} &= [ik, \partial_z] \cdot \hat{\mathbf{f}} \\ (\partial_z \pm k) \hat{\pi}(\mathbf{k}, z = \pm H) &= 0 \end{aligned}$$

Complex fluids in the GPU era

└ New doubly periodic solvers

└ Hydrodynamics

└ Doubly Periodic Force Coupling Method

2022-05-29

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} ik \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [ik, \partial_z] \cdot \hat{\mathbf{v}} = 0$$

BVP for each \mathbf{k} : Embarrassingly parallel

$z = H$

$z = -H$

L_{xy}

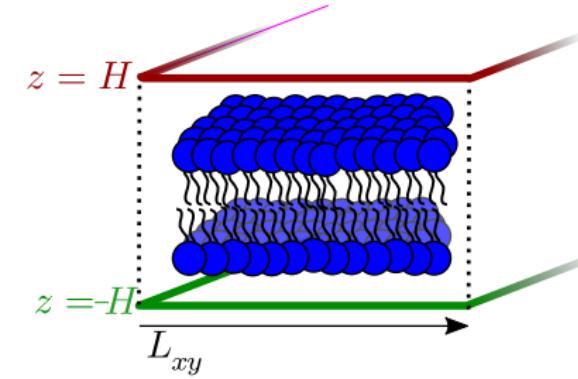
[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021]

We end up with a set of independent Boundary Value Problems for the pressure (one for each wave vector). Once we have the pressure, we replace it in the Stokes equation and solve for the velocity. Then we transform that velocity back to real space and interpolate to the particles as usual.

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} ik \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [ik, \partial_z] \cdot \hat{\mathbf{v}} = 0$$



Hardships

2022-05-29

Complex fluids in the GPU era

└ New doubly periodic solvers

 └ Hydrodynamics

 └ Doubly Periodic Force Coupling Method

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} ik \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [ik, \partial_z] \cdot \hat{\mathbf{v}} = 0$$

Hardships

[O. Maxian, Raúl P. Peláez et. al. J. Chem. Phys. 2021.]

This is easier said than done, though.

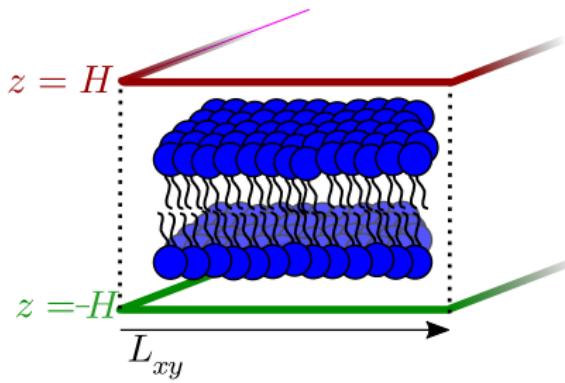
Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} ik \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [ik, \partial_z] \cdot \hat{\mathbf{v}} = 0$$

Hardships

1 BVP solver



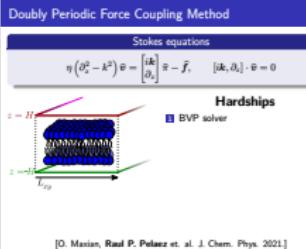
Complex fluids in the GPU era

└ New doubly periodic solvers

└ Hydrodynamics

└ Doubly Periodic Force Coupling Method

2022-05-29

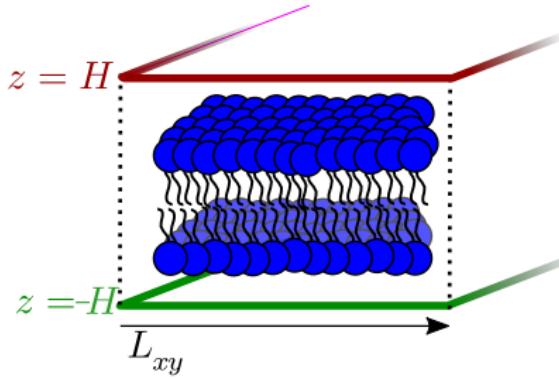


Although embarrassingly parallel, the BVP solver is not straight-forward to encode in a GPU.

Doubly Periodic Force Coupling Method

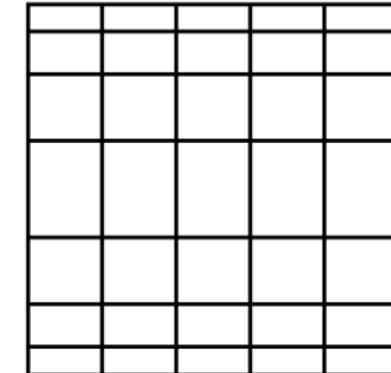
Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} ik \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [ik, \partial_z] \cdot \hat{\mathbf{v}} = 0$$



Hardships

- 1 BVP solver
- 2 Requires Chebyshev space in Z



[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

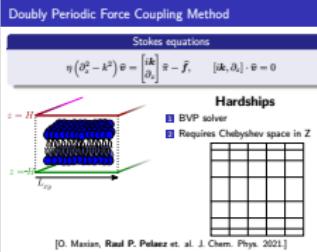
Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Doubly Periodic Force Coupling Method

2022-05-29

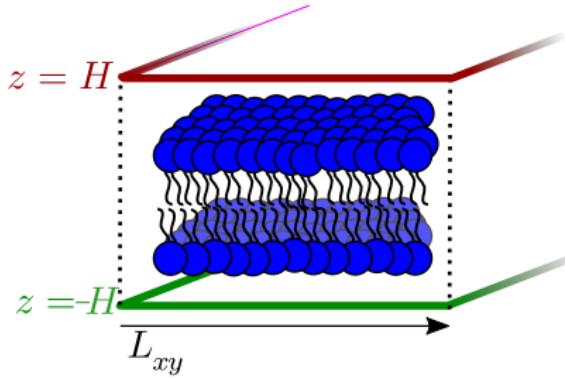


In particular, it involves defining the Eulerian fields in a Chebyshev grid in the Z direction and performing a so-called fast Chebyshev transform, which is similar to a Fast Fourier transform.

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta \left(\partial_z^2 - k^2 \right) \hat{\mathbf{v}} = \begin{bmatrix} ik \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [ik, \partial_z] \cdot \hat{\mathbf{v}} = 0$$



Hardships

- 1 BVP solver
- 2 Requires Chebyshev space in Z
- 3 Walls added as corrections

$$\mathbf{v}^* = \mathbf{v} + \mathbf{v}_{\text{corr}} \rightarrow \mathbf{v}^*(z = \pm H) = 0$$

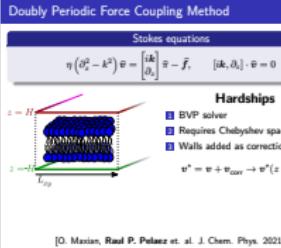
Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Doubly Periodic Force Coupling Method

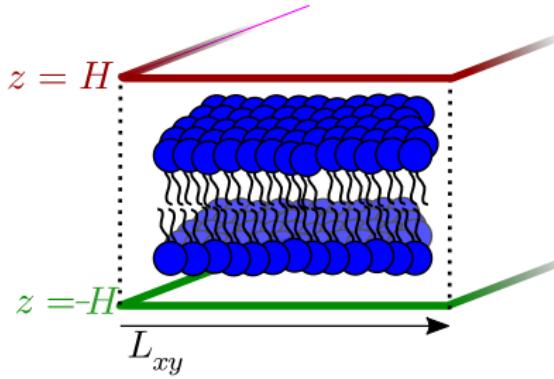
2022-05-29



Doubly Periodic Force Coupling Method

Stokes equations

$$\eta (\partial_z^2 - k^2) \hat{\mathbf{v}} = \begin{bmatrix} ik \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [ik, \partial_z] \cdot \hat{\mathbf{v}} = 0$$



Hardships

- 1 BVP solver
- 2 Requires Chebyshev space in Z
- 3 Walls added as corrections
- 4 Fluct. require iterative solver
 - Lanczos: ~ 10 Stokes solves
 - Thermal drift: 2 Stokes solves

Complex fluids in the GPU era

New doubly periodic solvers

Hydrodynamics

Doubly Periodic Force Coupling Method

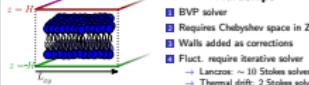
2022-05-29

Doubly Periodic Force Coupling Method

Stokes equations

$$\eta (\partial_z^2 - k^2) \hat{\mathbf{v}} = \begin{bmatrix} ik \\ \partial_z \end{bmatrix} \hat{\pi} - \hat{\mathbf{f}}, \quad [ik, \partial_z] \cdot \hat{\mathbf{v}} = 0$$

Hardships



[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

- Finally, we only know how to add fluctuations via iterative solvers, which require several Stokes solves.
- And similarly to the quasi 2D case, we also have to add thermal drift, which involves another couple of solves.
- All in all, we end up again with a mechanism to transform particle forces to velocities that scales linearly with the number of particles. This time for a doubly periodic environment.

Triply Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

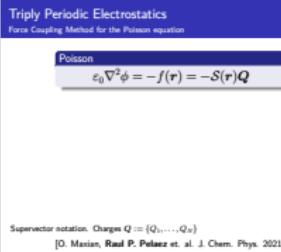
Supervector notation. Charges $Q := \{Q_1, \dots, Q_N\}$

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Electrostatics
 - └ Triply Periodic Electrostatics

2022-05-29



- Now I will shift the focus to electrostatics.
- I want to show you how we can employ our recently learned tricks to other equations.
- In particular, the Force Coupling Method for the Stokes equation has an almost one-to-one simile for the Poisson equation.
- Here I am showing the Poisson equation for the potential of a domain in the presence of a charge density.
- epsilon is the permittivity of the medium.
- We will start with triply periodic boundary conditions.

Triply Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

Spreading

$$\mathcal{S}(\mathbf{r})Q = \sum_i Q_i \delta_a(\mathbf{q}_i - \mathbf{r})$$

Supervector notation. Charges $Q := \{Q_1, \dots, Q_N\}$

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Electrostatics
 - └ Triply Periodic Electrostatics

2022-05-29

Triply Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

Spreading

$$\mathcal{S}(\mathbf{r})Q = \sum_i Q_i \delta_a(\mathbf{q}_i - \mathbf{r})$$

Supervector notation. Charges $Q := \{Q_1, \dots, Q_N\}$
[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Triply Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r}) \mathbf{Q}$$

Spreading

$$\mathcal{S}(\mathbf{r}) \mathbf{Q} = \sum_i Q_i \delta_a(\mathbf{q}_i - \mathbf{r})$$

Gaussian sources

$$\delta_a(\mathbf{r}) \propto e^{-r^2/(2a^2)}$$

Supervector notation. Charges $\mathbf{Q} := \{Q_1, \dots, Q_N\}$

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Electrostatics
 - └ Triply Periodic Electrostatics

2022-05-29

Triply Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson
 $\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r}) \mathbf{Q}$

Spreading
 $\mathcal{S}(\mathbf{r}) \mathbf{Q} = \sum_i Q_i \delta_a(\mathbf{q}_i - \mathbf{r})$

Gaussian sources
 $\delta_a(\mathbf{r}) \propto e^{-r^2/(2a^2)}$

Supervector notation. Charges $\mathbf{Q} := \{Q_1, \dots, Q_N\}$
[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Triply Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r}) \mathbf{Q}$$

Spreading

$$\mathcal{S}(\mathbf{r}) \mathbf{Q} = \sum_i Q_i \delta_a(\mathbf{q}_i - \mathbf{r})$$

Gaussian sources

$$\delta_a(\mathbf{r}) \propto e^{-r^2/(2a^2)}$$

$$\mathbf{E}(\mathbf{r}) = \partial_r \phi \rightarrow \hat{\mathbf{E}} = i \mathbf{k} \hat{\phi}$$

$$\mathbf{F}_i = Q_i \mathcal{J}_{\mathbf{q}_i} \mathbf{E}(\mathbf{r})$$

Complex fluids in the GPU era

└ New doubly periodic solvers

└ Electrostatics

└ Triply Periodic Electrostatics

2022-05-29

Triply Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r}) \mathbf{Q}$$

E(r) = $\partial_r \phi \rightarrow \hat{\mathbf{E}} = ik\hat{\phi}$

$$\mathcal{S}(\mathbf{r}) \mathbf{Q} = \sum_i Q_i \delta_a(\mathbf{q}_i - \mathbf{r})$$

$$\mathbf{F}_i = Q_i \mathcal{J}_{\mathbf{q}_i} \mathbf{E}(\mathbf{r})$$

Gaussian sources

$$\delta_a(\mathbf{r}) \propto e^{-r^2/(2a^2)}$$

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021]

Then, instead of interpolating velocities, we now interpolate electric fields. Which are trivial to compute once we solve the potential, moreover in Fourier space.

Triply Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r}) \mathbf{Q}$$

Spreading

$$\mathcal{S}(\mathbf{r}) \mathbf{Q} = \sum_i Q_i \delta_a(\mathbf{q}_i - \mathbf{r})$$

$$\mathbf{E}(\mathbf{r}) = \partial_r \phi \rightarrow \hat{\mathbf{E}} = i \mathbf{k} \hat{\phi}$$

$$\mathbf{F}_i = Q_i \mathcal{J}_{\mathbf{q}_i} \mathbf{E}(\mathbf{r})$$

Gaussian sources

$$\delta_a(\mathbf{r}) \propto e^{-r^2/(2a^2)}$$

Charges to forces

$$\mathbf{F}_i = Q_i \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} i \mathbf{k} \hat{\mathcal{G}}_P \mathfrak{F} \mathcal{S} \mathbf{Q}$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Electrostatics
 - └ Triply Periodic Electrostatics

2022-05-29

Triply Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson
 $\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r}) \mathbf{Q}$

Spreading
 $\mathcal{S}(\mathbf{r}) \mathbf{Q} = \sum_i Q_i \delta_a(\mathbf{q}_i - \mathbf{r})$

Gaussian sources
 $\delta_a(\mathbf{r}) \propto e^{-r^2/(2a^2)}$

$E(\mathbf{r}) = \partial_r \phi \rightarrow \hat{\mathbf{E}} = i \mathbf{k} \hat{\phi}$

$\mathbf{F}_i = Q_i \mathcal{J}_{\mathbf{q}_i} E(\mathbf{r})$

$\mathbf{F}_i = Q_i \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} i \mathbf{k} \hat{\mathcal{G}}_P \mathfrak{F} \mathcal{S} \mathbf{Q}$

Charges to forces

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021]

In the end, we end up with a mechanism to transform particle charges to particle forces that is quite similar to what we have been seeing up until now.

Triply Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r}) \mathbf{Q}$$

Spreading

$$\mathcal{S}(\mathbf{r}) \mathbf{Q} = \sum_i Q_i \delta_a(\mathbf{q}_i - \mathbf{r})$$

$$\mathbf{E}(\mathbf{r}) = \partial_r \phi \rightarrow \hat{\mathbf{E}} = i\mathbf{k}\hat{\phi}$$

$$\mathbf{F}_i = Q_i \mathcal{J}_{\mathbf{q}_i} \mathbf{E}(\mathbf{r})$$

Gaussian sources

$$\delta_a(\mathbf{r}) \propto e^{-r^2/(2a^2)}$$

Charges to forces

$$\mathbf{F}_i = Q_i \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} i\mathbf{k} \hat{\mathcal{G}}_P \mathfrak{F} \mathcal{S} \mathbf{Q}$$

$$\hat{\mathcal{G}}_P = (\varepsilon_0 k^2)^{-1}$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Electrostatics
 - └ Triply Periodic Electrostatics

2022-05-29

Triply Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson	$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r}) \mathbf{Q}$
Spreading	$E(\mathbf{r}) = \partial_r \phi \rightarrow \hat{E} = ik\hat{\phi}$ $\mathbf{F}_i = Q_i \mathcal{J}_{\mathbf{q}_i} E(\mathbf{r})$
Gaussian sources	$\delta_a(\mathbf{r}) \propto e^{-r^2/(2a^2)}$ $\mathbf{F}_i = Q_i \mathcal{J}_{\mathbf{q}_i} \mathfrak{F}^{-1} ik \hat{\mathcal{G}}_P \mathfrak{F} \mathcal{S} \mathbf{Q}$
Charges to forces	$\hat{\mathcal{G}}_P = (\varepsilon_0 k^2)^{-1}$

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021]

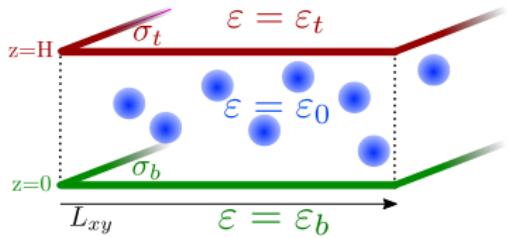
Bonus point, the (triply periodic) Green's function is now much more simple.

Doubly Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$



Supervector notation. Charges $Q := \{Q_1, \dots, Q_N\}$

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Complex fluids in the GPU era

└ New doubly periodic solvers
 └ Electrostatics
 └ Doubly Periodic Electrostatics

2022-05-29

Doubly Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

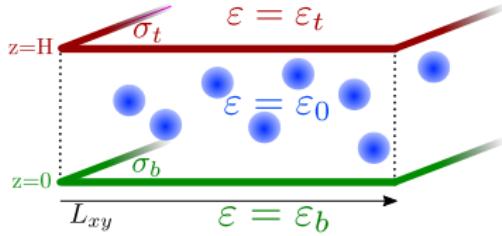
A 2D diagram of a unit cell for doubly periodic electrostatics. It shows a square domain with periodic boundary conditions. The top boundary is labeled $\varepsilon = \varepsilon_t$, the bottom $\varepsilon = \varepsilon_b$, the left σ_b , and the right σ_t . A supervector of charges $Q := (Q_1, \dots, Q_N)$ is shown as a series of points. The caption below reads: "Supervector notation. Charges $Q := (Q_1, \dots, Q_N)$ [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Doubly Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$



$$\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$$

$$\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Electrostatics
 - └ Doubly Periodic Electrostatics

2022-05-29

Doubly Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

$\varepsilon_t E^z(z=H) - \varepsilon_0 E^z(z=H) = \sigma_t$

$\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Doubly Periodic Electrostatics

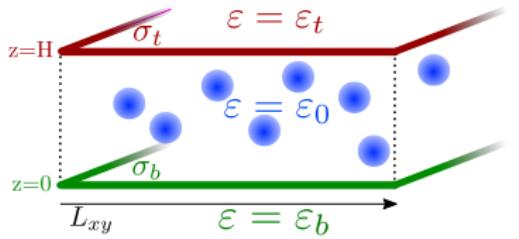
Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

Problems:

1 Unknown Green's function



$$\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$$

$$\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$$

Complex fluids in the GPU era

└ New doubly periodic solvers
└ Electrostatics
└ Doubly Periodic Electrostatics

2022-05-29

Doubly Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

Problems:
Unknown Green's function

A 3D diagram of a unit cell with dimensions L_x , L_y , and L_z . The top surface is labeled $\varepsilon = \varepsilon_t$ and the bottom surface is labeled $\varepsilon = \varepsilon_b$. The left and right boundaries are periodic. Inside the cell, there are several blue spheres representing charges. Boundary conditions are shown: $\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$ and $\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$.

[O. Maxian, Raúl P. Peláez et. al. J. Chem. Phys. 2021.]

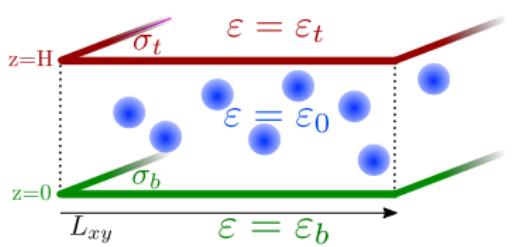
We have the same problems as before, and we solve them in the exact same way.

Doubly Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$



Problems:

- 1 Unknown Green's function
- 2 Cannot use \mathfrak{F} in Z

$$\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$$

$$\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$$

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Electrostatics
 - └ Doubly Periodic Electrostatics

2022-05-29

Doubly Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

Problems:

- ─ Unknown Green's function
- ─ Cannot use \mathfrak{F} in Z

$\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$

$\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$

[O. Maxian, Raúl P. Peláez et. al. J. Chem. Phys. 2021.]

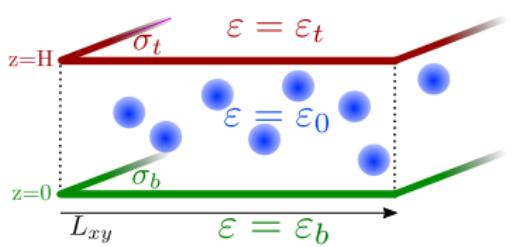
We have the same problems as before, and we solve them in the exact same way.

Doubly Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$



Problems:

- 1 Unknown Green's function
- 2 Cannot use $\tilde{\mathfrak{F}}$ in Z

Solutions:

$$\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$$

$$\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$$

Complex fluids in the GPU era

└ New doubly periodic solvers
└ Electrostatics
└ Doubly Periodic Electrostatics

2022-05-29

Doubly Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

Problems:

- Unknown Green's function
- Cannot use $\tilde{\mathfrak{F}}$ in Z

Solutions:

- $\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$
- $\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$

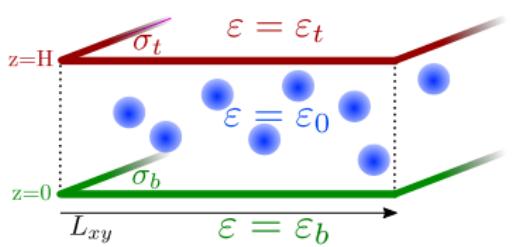
[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Doubly Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$



Problems:

- 1 Unknown Green's function
- 2 Cannot use $\tilde{\mathfrak{F}}$ in Z

Solutions:

- 1 Solve Poisson directly

$$\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$$

$$\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$$

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Complex fluids in the GPU era

- └ New doubly periodic solvers
 - └ Electrostatics
 - └ Doubly Periodic Electrostatics

2022-05-29

Doubly Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

Problems:

- Unknown Green's function
- Cannot use $\tilde{\mathfrak{F}}$ in Z

Solutions:

- Solve Poisson directly

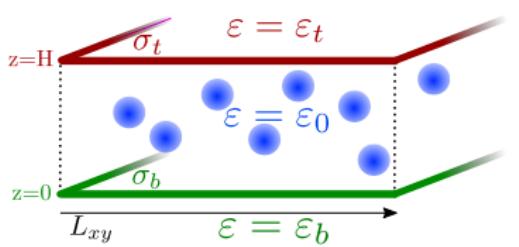
[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Doubly Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\epsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$



Problems:

- 1 Unknown Green's function
- 2 Cannot use $\tilde{\mathfrak{F}}$ in Z

Solutions:

- 1 Solve Poisson directly
- 2 $\tilde{\mathfrak{F}}$ only in XY plane

$$\epsilon_0 E^z(z=0) - \epsilon_b E^z(z=0) = -\sigma_b$$

$$\epsilon_0 E^z(z=H) - \epsilon_t E^z(z=H) = \sigma_t$$

Complex fluids in the GPU era

New doubly periodic solvers

Electrostatics

Doubly Periodic Electrostatics

2022-05-29

Doubly Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson

$$\epsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

Problems:

- 1 Unknown Green's function
- 2 Cannot use $\tilde{\mathfrak{F}}$ in Z

Solutions:

- 1 Solve Poisson directly
- 2 $\tilde{\mathfrak{F}}$ only in XY plane

$\epsilon_0 E^z(z=0) - \epsilon_b E^z(z=0) = -\sigma_b$

$\epsilon_0 E^z(z=H) - \epsilon_t E^z(z=H) = \sigma_t$

[O. Maxian, Raúl P. Peláez et. al. J. Chem. Phys. 2021.]

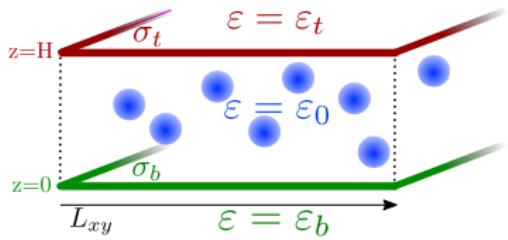
We have the same problems as before, and we solve them in the exact same way.

Doubly Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$



\mathfrak{F} only in XY plane

$$\begin{aligned}\nabla &\rightarrow [ik_x, ik_y, \partial_z] := [i\mathbf{k}, \partial_z] \\ f &\rightarrow \hat{f}(\mathbf{k}, z)\end{aligned}$$

$$\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$$

$$\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$$

Complex fluids in the GPU era

New doubly periodic solvers

Electrostatics

Doubly Periodic Electrostatics

2022-05-29

Doubly Periodic Electrostatics

Force Coupling Method for the Poisson equation

Poisson

$$\varepsilon_0 \nabla^2 \phi = -f(\mathbf{r}) = -\mathcal{S}(\mathbf{r})Q$$

\mathfrak{F} only in XY plane

$$\nabla \rightarrow [ik_x, ik_y, \partial_z] := [i\mathbf{k}, \partial_z]$$

$$f \rightarrow \hat{f}(\mathbf{k}, z)$$

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

We have the same problems as before, and we solve them in the exact same way.

Doubly Periodic Electrostatics

Force Coupling Method for the Poisson equation

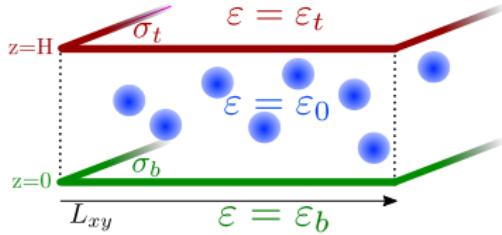
Poisson

$$\varepsilon_0 (\partial_z^2 - k^2) \hat{\phi} = -\hat{f}$$

\Im only in XY plane

$$\nabla \rightarrow [ik_x, ik_y, \partial_z] := [i\mathbf{k}, \partial_z]$$

$$f \rightarrow \hat{f}(\mathbf{k}, z)$$



$$\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$$

$$\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$$

Complex fluids in the GPU era

New doubly periodic solvers

Electrostatics

Doubly Periodic Electrostatics

2022-05-29

Doubly Periodic Electrostatics
Force Coupling Method for the Poisson equation

Poisson
 $\varepsilon_0 (\partial_z^2 - k^2) \hat{\phi} = -\hat{f}$

\Im only in XY plane
 $\nabla \rightarrow [ik_x, ik_y, \partial_z] := [i\mathbf{k}, \partial_z]$
 $f \rightarrow \hat{f}(\mathbf{k}, z)$

$\varepsilon_0 E^z(z=0) - \varepsilon_b E^z(z=0) = -\sigma_b$
 $\varepsilon_0 E^z(z=H) - \varepsilon_t E^z(z=H) = \sigma_t$

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

- Eventually, we end up with a BVP that is equivalent to the one we had to solve for the pressure in Stokes. This algorithm also comes with similar hardships as the Stokes one: We need to solve BVPs, different boundary conditions are added as corrections (which can be quite dense mathematically).

Talk outline

- 1 Introduction
- 2 Elements of a complex fluid simulation
- 3 New doubly periodic solvers
- 4 UAMMD
 - Universally Adaptable Multiscale Molecular Dynamics
 - Basic code structure
- 5 Conclusions

2022-05-29

Complex fluids in the GPU era

UAMMD

Talk outline

In this final section I will introduce UAMMD, the GPU software framework for complex fluids that encases, among others, all the algorithms I have previously introduced.

Talk outline

- Introduction
- Elements of a complex fluid simulation
- New doubly periodic solvers
- UAMMD**
 - Universally Adaptable Multiscale Molecular Dynamics
 - Basic code structure
- Conclusions

Raúl P. Peláez

Complex fluids in the GPU era

35 / 46

The sea of packages

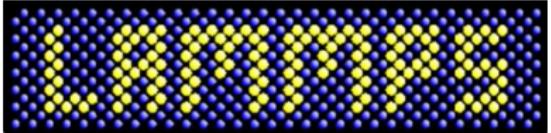


AMBER MD



CHARMM
Chemistry at HARvard Macromolecular Mechanics

HOOMD-blue



Complex fluids in the GPU era
└ UAMMD

└ The sea of packages

2022-05-29



UAMMD is born in a crowded sea of molecular dynamics packages. Although most of the commercial packages focus on the microscopic level, in other words, molecular dynamics, many times the limits between molecular dynamics and other descriptions proper to complex fluids are kind of blurred from an algorithmic point of view. So in principle these packages, some of them more than others, can be adapted to other descriptions.

UAMMD: A CUDA/C++ library for complex fluids.

- Header only and library-like

└ What makes UAMMD stand out

- I would like to lay out some key points that make UAMMD, in my opinion, stand out.
- The first one is that UAMMD is header-only library, as opposed to a monolithic single-binary program.
- This facilitates the use of UAMMD as an accelerator library in other projects. For instance, exposing some of its utilities to python.

UAMMD: A CUDA/C++ library for complex fluids.

- Header only and library-like
- Lightweight with minimal dependencies

2022-05-29

The UAMMD source code is lightweight and, besides an NVIDIA GPU with its required software development kit UAMMD does not depend on any code that is not included in its repository.

UAMMD: A CUDA/C++ library for complex fluids.

- Header only and library-like
- Lightweight with minimal dependencies
- Hackable

Complex fluids in the GPU era

└ UAMMD

└ What makes UAMMD stand out

2022-05-29

What makes UAMMD stand out

UAMMD: A CUDA/C++ library for complex fluids.

- Header only and library-like
- Lightweight with minimal dependencies
- Hackable

I made UAMMD not only adaptable, but hackable. I tried to reduce internal dependencies to a minimum to encourages users to copy or take away parts of it.

UAMMD: A CUDA/C++ library for complex fluids.

- Header only and library-like
- Lightweight with minimal dependencies
- Hackable
- Focus on hydrodynamics

Complex fluids in the GPU era

└ UAMMD

└ What makes UAMMD stand out

Finally, there are not a lot of packages out there that put the focus on hydrodynamics and/or complex fluids.

What makes UAMMD stand out

UAMMD: A CUDA/C++ library for complex fluids.

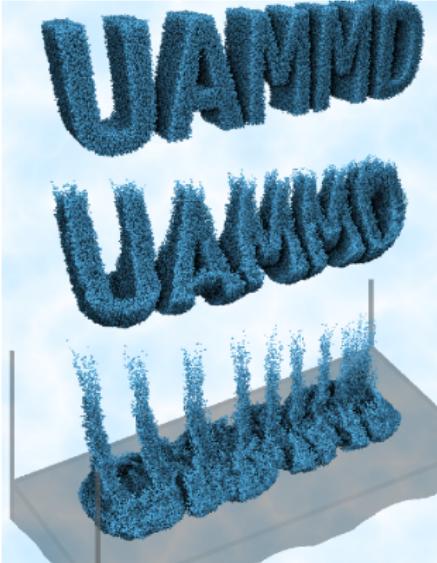
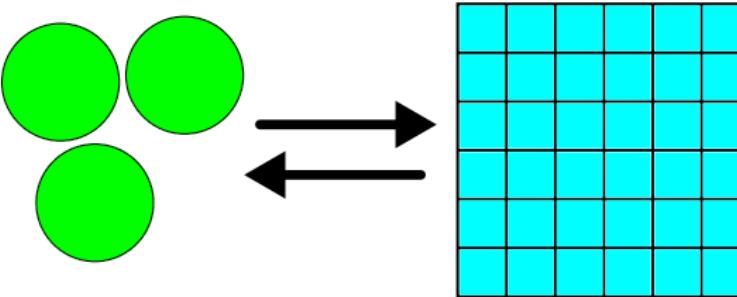
- Header only and library-like
- Lightweight with minimal dependencies
- Hackable
- Focus on hydrodynamics

Universally Adaptable Multiscale Molecular Dynamics

Universally Adaptable Multiscale Molecular Dynamics

Universally Adaptable Multiscale Molecular Dynamics

Universal
Lagrangian and Eulerian descriptions.

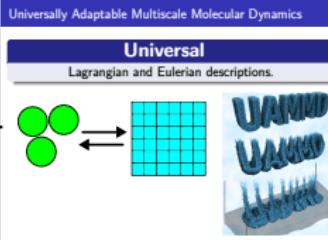


2022-05-29

Complex fluids in the GPU era

- └ UAMMD
 - └ Universally Adaptable Multiscale Molecular Dynamics
 - └ Universally Adaptable Multiscale Molecular Dynamics

I like to use the UAMMD acronym to introduce its core philosophies. Universal means that both Eulerian and Lagrangian descriptions are supported.



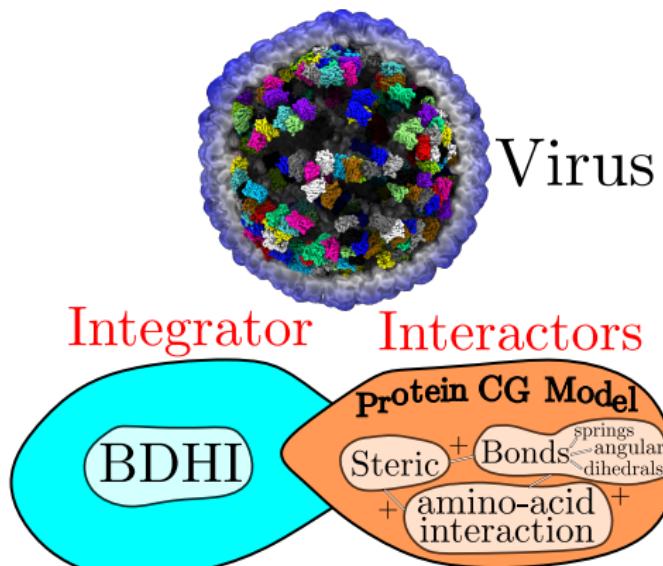
Raúl P. Peláez

Complex fluids in the GPU era

38 / 46

Adaptable

Independent building blocks.



Complex fluids in the GPU era

└ UAMMD

└ Universally Adaptable Multiscale Molecular Dyna-

ics

└ Universally Adaptable Multiscale Molecular
Dynamics

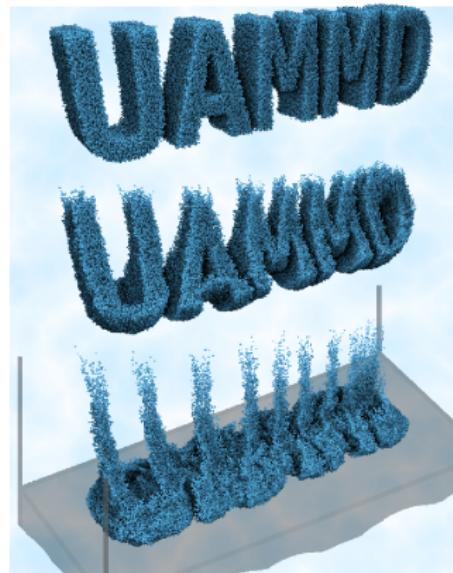
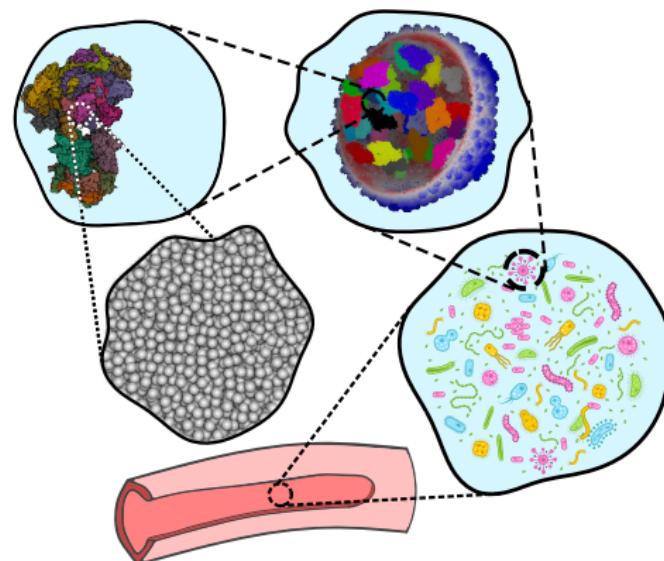
2022-05-29



Adaptable speaks about the modular design of the code, which I will get into more detail soon. The library is composed of a series of independent modules that can be interconnected to create one simulation or another.

Multiscale

Solvers for many scales.

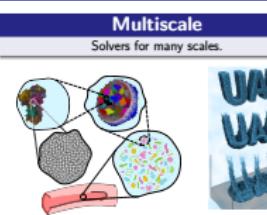
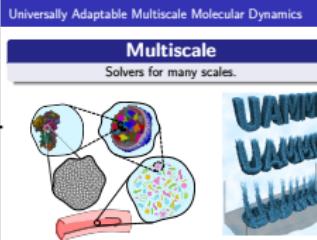


Complex fluids in the GPU era

└ UAMMD

└ Universally Adaptable Multiscale Molecular Dynamics
└ Universally Adaptable Multiscale Molecular Dynamics

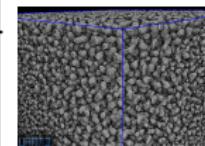
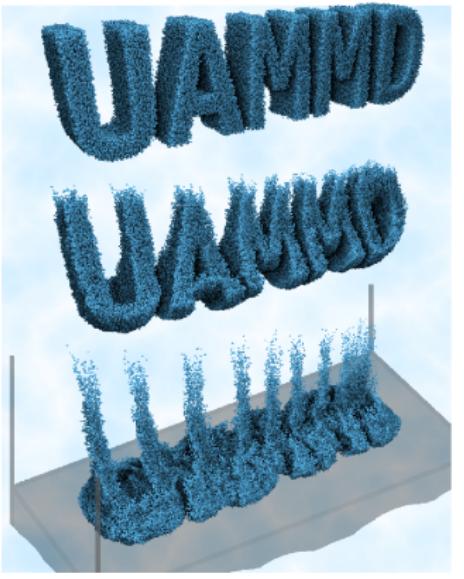
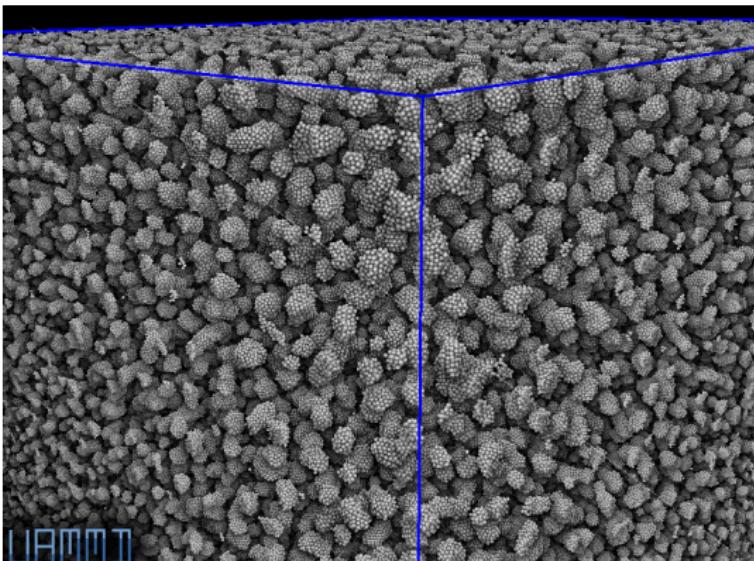
2022-05-29



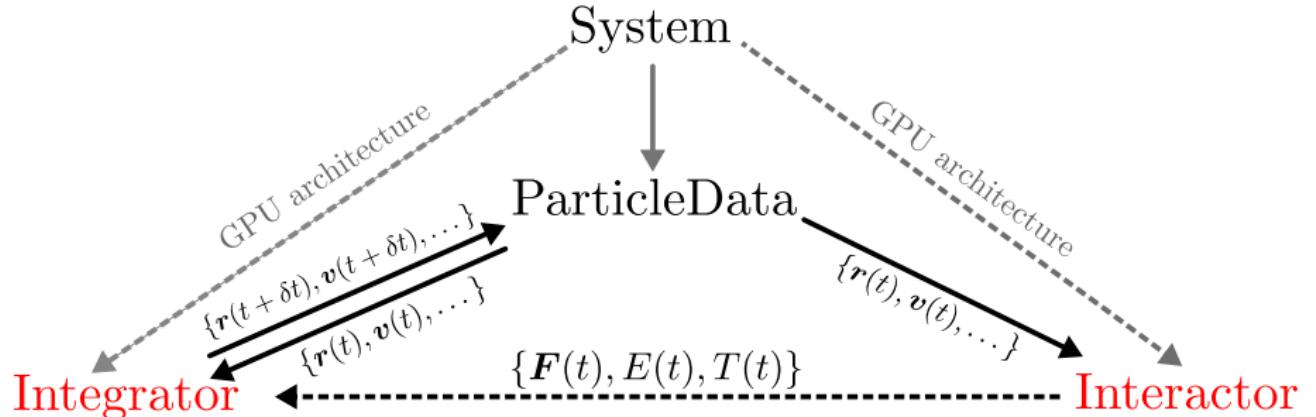
Multiscale means that UAMMD offers solvers for all the levels of description that I introduced at the start.

Molecular Dynamics

Mainly particle based.



Finally, Molecular Dynamics, not the best choice maybe, conveys that particles are the main player. Remember, even when a continuous field, such as a fluid, exists its only as a proxy to communicate the particles.



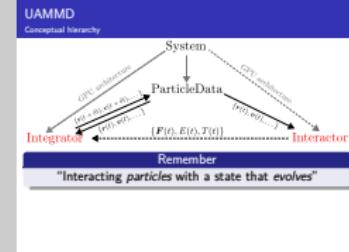
Complex fluids in the GPU era

UAMMD

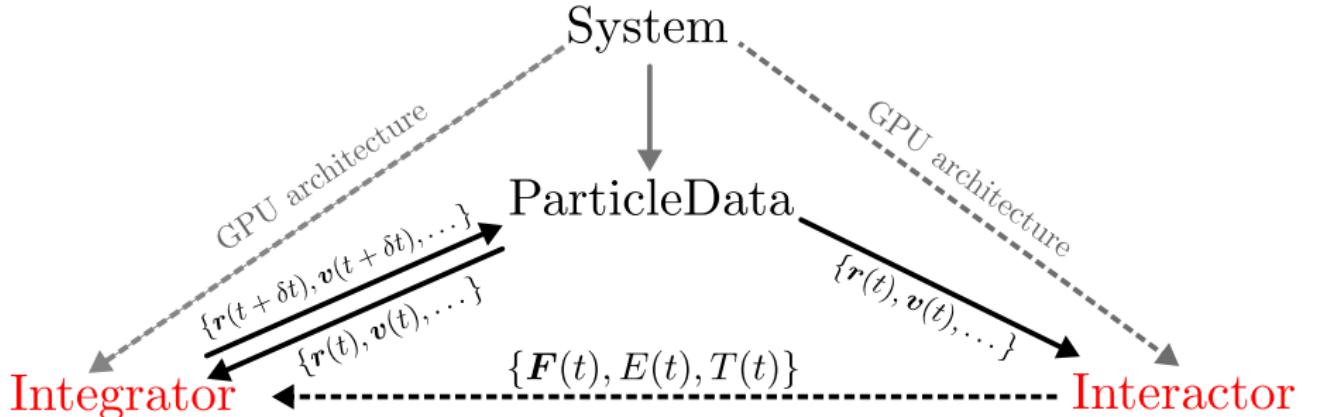
Basic code structure

UAMMD

2022-05-29



- When introducing complex fluids, I stated that the majority of levels of description in complex fluids can be interpreted as a series of interacting particles with a state that evolves.
- This sentence is the foundation of UAMMD. A first layer abstracts the actual machine. Then, we enter into a continuous loop that represents a simulation.
- Particles feed their properties (positions, velocities...) downwards to rest of the modules, and UAMMD distinguishes between two types of modules.
- Integrators, which read the state of the particles at one time and take them to the next.
- And Interactors, which are used by Integrators to compute Forces, Energies or virials.

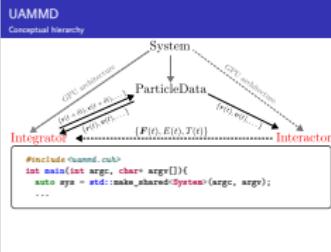


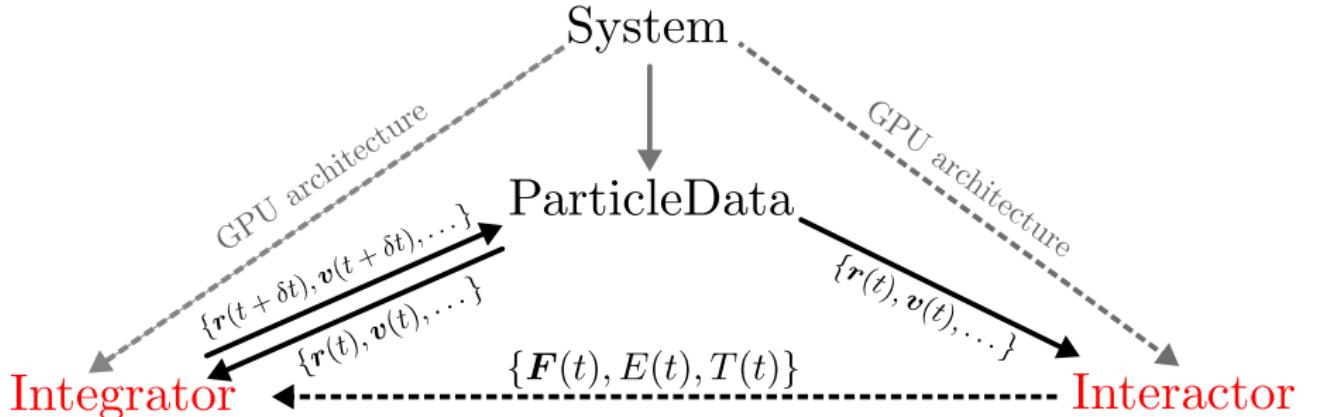
```
#include <uammd.cuh>
int main(int argc, char* argv[]){
    auto sys = std::make_shared<System>(argc, argv);
    ...
}
```

Complex fluids in the GPU era
 └── UAMMD
 └── Basic code structure
 └── UAMMD

2022-05-29

This conceptual separation is reflected in actual code, where **System** is a C++ class that can be instantiated.





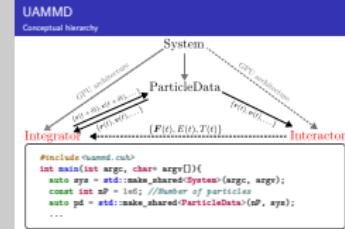
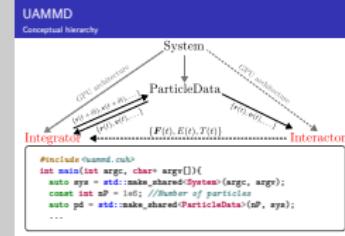
```

#include <uammd.cuh>
int main(int argc, char* argv[]){
    auto sys = std::make_shared<System>(argc, argv);
    const int nP = 1e6; //Number of particles
    auto pd = std::make_shared<ParticleData>(nP, sys);
    ...
  
```

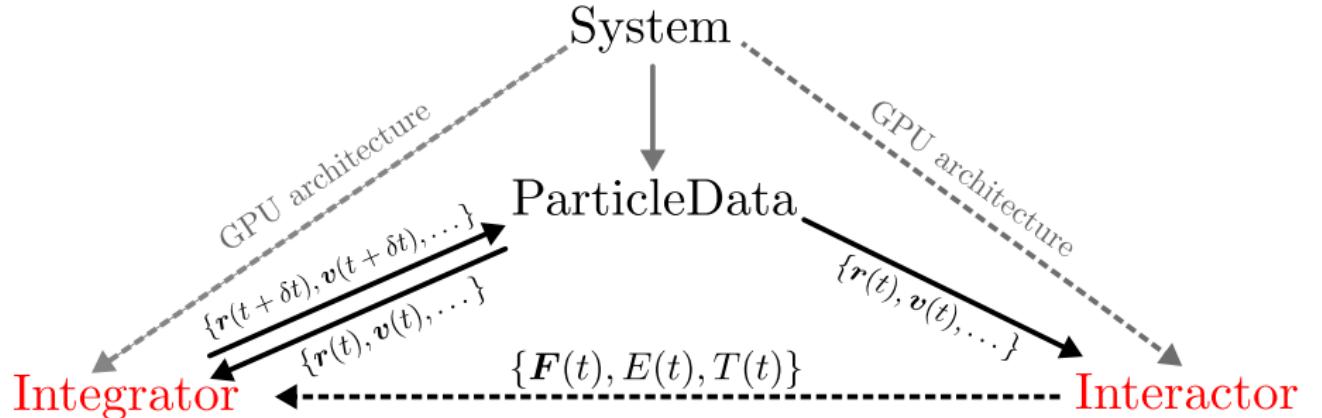
Complex fluids in the GPU era

2022-05-29

- └ UAMMD
 - └ Basic code structure
 - └ UAMMD

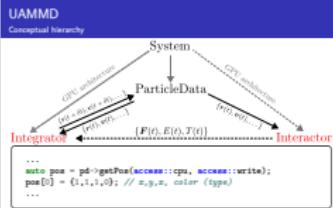


In the same way we can initialize an instance of **ParticleData**, a container for handling the particle properties.

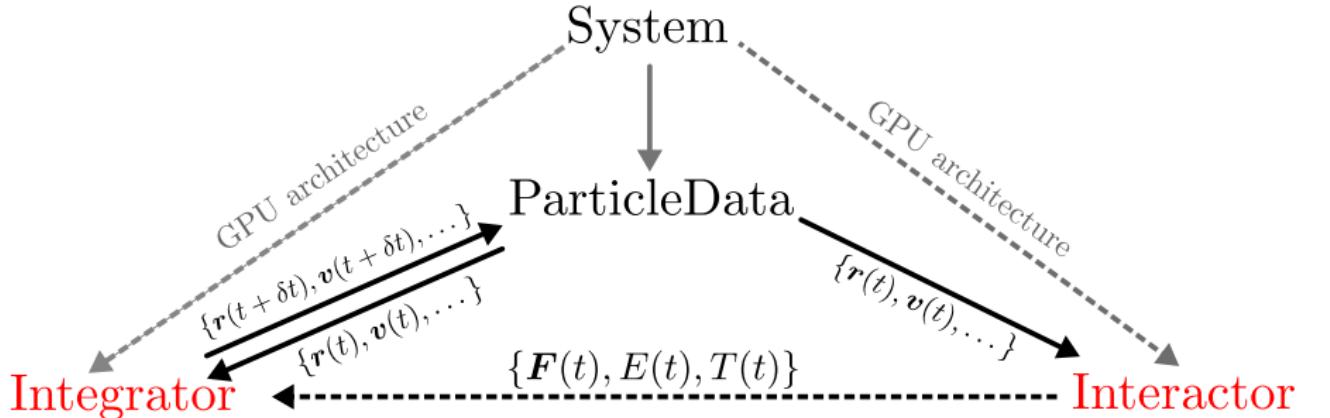


Complex fluids in the GPU era
└ UAMMD
 └ Basic code structure
 └ UAMMD

2022-05-29



Properties that can be easily accessed at any time.



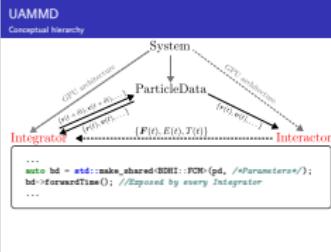
```

...
auto bd = std::make_shared<BDHI::FCM>(pd, /*Parameters*/);
bd->forwardTime(); //Exposed by every Integrator
...

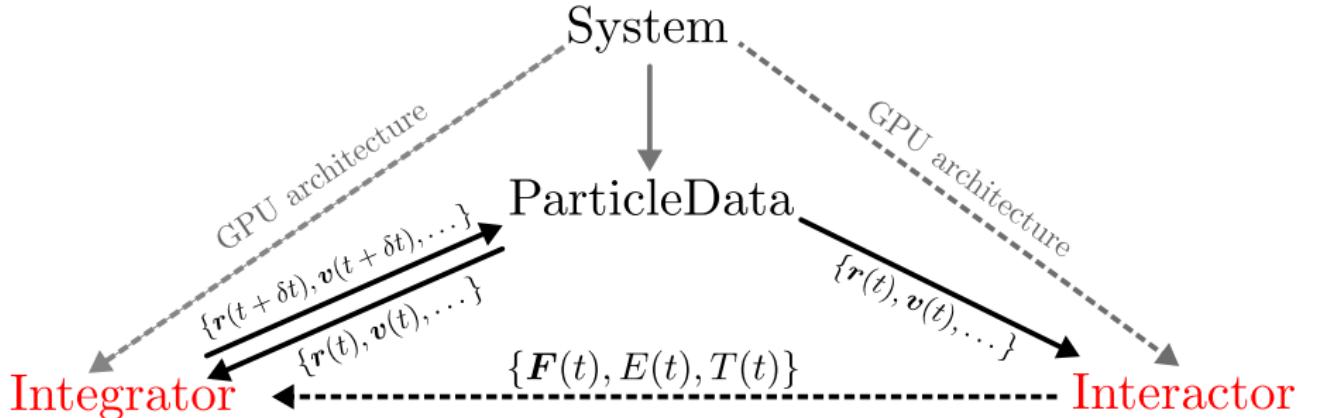
```

Complex fluids in the GPU era
 └── UAMMD
 └── Basic code structure
 └── UAMMD

2022-05-29



Integrators are fed a Particle container and are in charge of forwarding them to the next time step. Integrator is a virtual class, and thus can be specialized for each solver. Here FCM is the UAMMD module for hydrodynamics with the force coupling method.



```

...
auto elec = std::make_shared<Poisson>(pd, /*Parameters*/);
elec->sum({.force=true, .energy=false, .virial=false});
//Exposed by every Integrator
bd->addInteractor(elec);
bd->forwardTime();
...

```

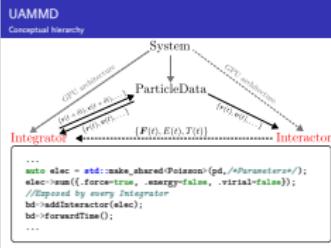
Complex fluids in the GPU era

UAMMD

Basic code structure

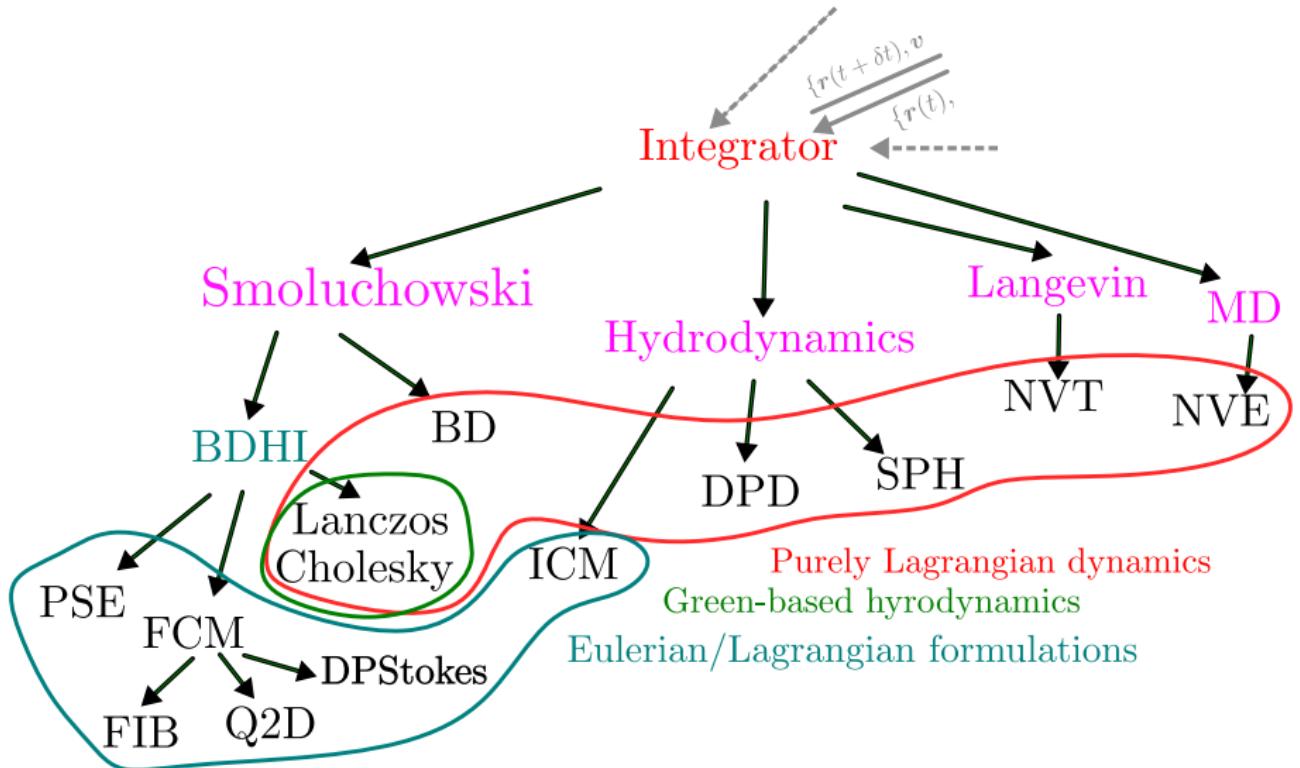
UAMMD

2022-05-29



Finally, **Interactors**, also a virtual class, can be issued to compute things like forces, which will be added at the particle container. and can be added to an **Integrator**, which in turn will use it as needed when forwarding the simulation. Poisson here is the UAMMD Interactor for triply periodic electrostatics.

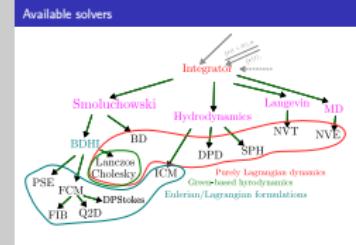
Available solvers



Complex fluids in the GPU era

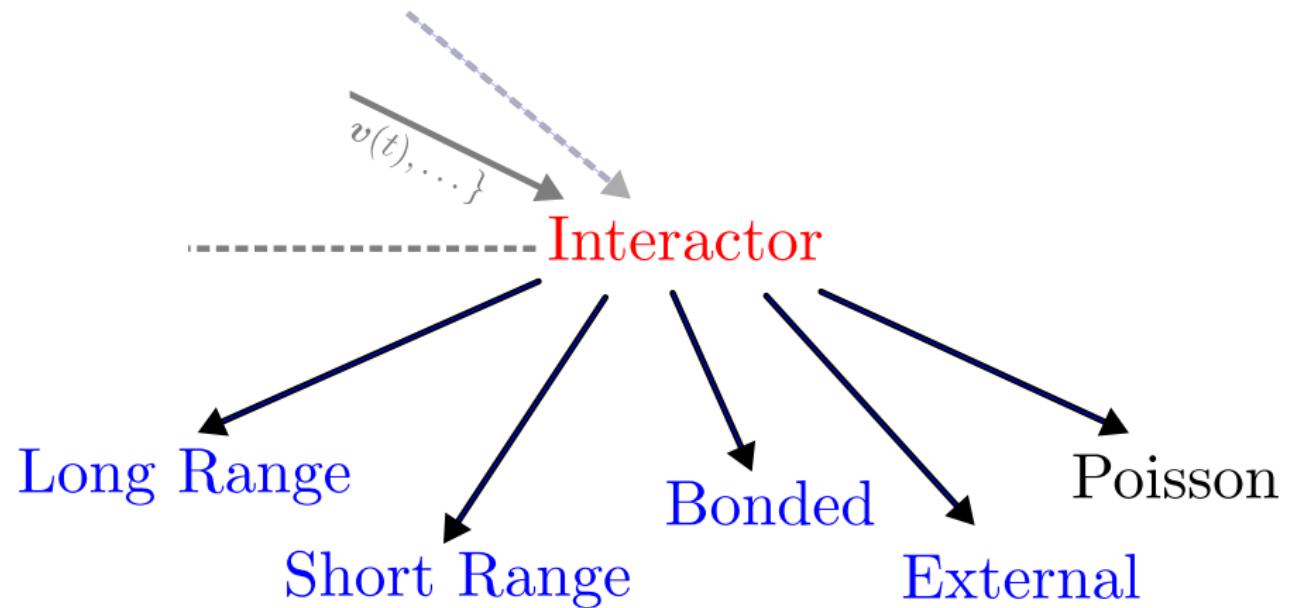
2022-05-29

- └ UAMMD
 - └ Basic code structure
 - └ Available solvers



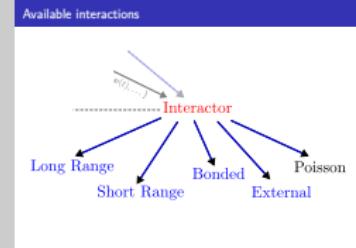
- This is a list of Integrators available in UAMMD to this date.
- As you can see, UAMMD exposes solvers for all levels of description that we discussed, and includes Lagrangian and Eulerian/Lagrangian formulations.
- From molecular dynamics to dissipative particle dynamics and Doubly periodic hydrodynamics.

Available interactions



Complex fluids in the GPU era
└ UAMMD
 └ Basic code structure
 └ Available interactions

2022-05-29

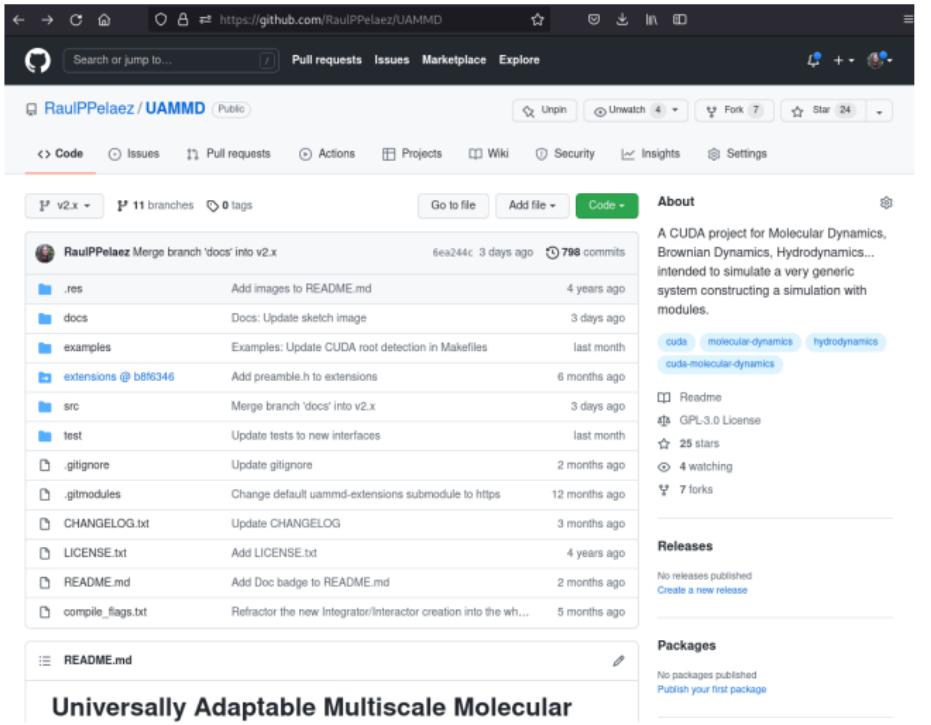


- Moreover, the familiar interactions are available as independent **Interactor** modules, neighbour lists, bonds, electrostatics in several geometries...
- And the list grows, with collaborators working on **Interactors** for things like: Optohydrodynamics, magnetism, coarse-grained potentials...

UAMMD's online presence

Git repository

<https://github.com/RaulPPelaez/uammd>



A screenshot of the GitHub repository page for 'RaulPPelaez/UAMMD'. The page shows a list of commits from 'RaulPPelaez' over the past few months. The commits include updates to README.md, documentation, examples, and source code. The repository has 11 branches and 0 tags. It features CUDA, molecular-dynamics, hydrodynamics, and cuda-molecular-dynamics tags. The repository is licensed under GPL-3.0 License and has 25 stars, 4 watchers, and 7 forks. There are no releases published. The README.md file is the active tab at the bottom.

Universally Adaptable Multiscale Molecular

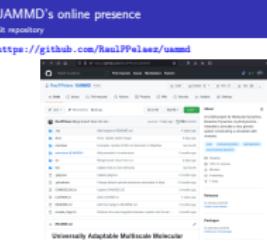
Complex fluids in the GPU era

└ UAMMD

└ Basic code structure

└ UAMMD's online presence

2022-05-29

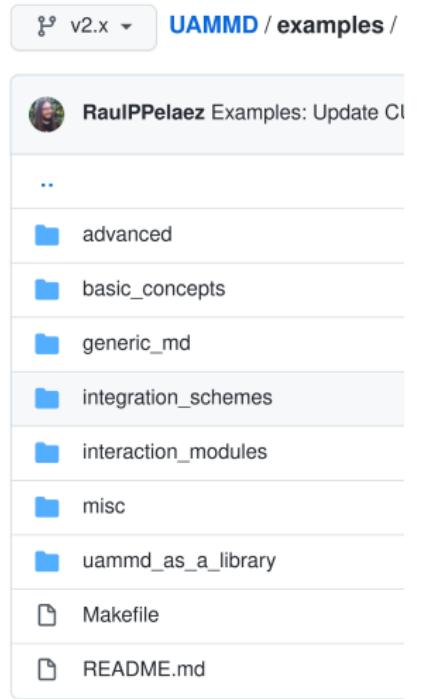


UAMMD is open-source and freely available as a repository at github.

UAMMD's online presence

Git repository

<https://github.com/RaulPPelaez/uammd>



Complex fluids in the GPU era
└ UAMMD
 └ Basic code structure
 └ UAMMD's online presence

2022-05-29

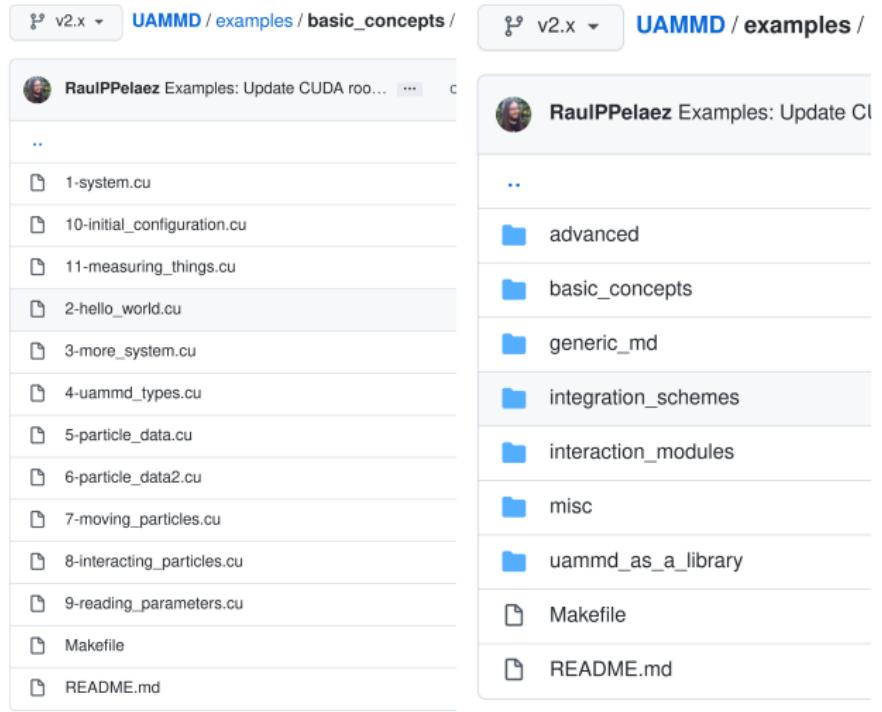
This repository is rich with examples, including an in depth series of them that increasingly introduce the UAMMD concepts in a literary, tutorial fashion.



UAMMD's online presence

Git repository

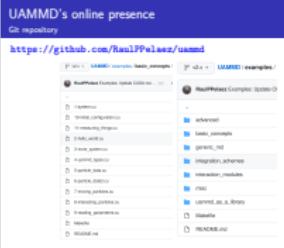
<https://github.com/RaulPPelaez/uammd>



Complex fluids in the GPU era
└ UAMMD
 └ Basic code structure
 └ UAMMD's online presence

2022-05-29

This repository is rich with examples, including an in depth series of them that increasingly introduce the UAMMD concepts in a literary, tutorial fashion.



UAMMD's online presence

Git repository

<https://github.com/RaulPPelaez/uammd>

v2.x

UAMMD / examples / basic_concepts / 2-hello_world.cu

RaulPPelaez Examples: Update CUDA root... History

1 contributor

36 lines (35 sloc) | 2.2 KB

```
1 /* Raul P. Pelaez 2021
2 * Hello world with UAMMD
3 * We will start from the previous example and see the first utility provided by System, the
4 * System::log allows to, well, log messages with several levels of priority.
5 * Internally UAMMD modules use this utility to print everything ranging from informational
6 * In this example we will use System::log to print a simple message along with the date.
7 */
8
9 //uammd.cuh is the basic uammd include containing, among other things, the System struct.
10 #include "uammd.cuh"
11 #include <ctime> //For time and ctime
12 using namespace uammd;
13 //The main function will initialize an UAMMD environment, print a message, then destroy it.
14 int main(int argc, char* argv[])
15 {
16     //Initialize System
17     auto sys = std::make_shared<System>(argc, argv);
18     //Unless something goes wrong System creation logs messages using level MESSAGE, which prints
19     //There are a lot more levels to choose from, each associated with a number. From highest
20     //CRITICAL=0, ERROR, EXCEPTION, WARNING, MESSAGE, STDERR, STDOUT, DEBUG1, DEBUG2, DEBUG3...
21     //For example, MESSAGE is associated with log level number 5.
22     //Let's print something using the MESSAGE level:
23     sys->log<System>("Hello from UAMMD");
24     //Let's also print today's date, this time as a WARNING:
25     auto currentTime = time(nullptr);
26     //Notice that System::log works as C's printf, with a format string and then arguments.
27     sys->log<System>("WARNING: Current time is: %s", ctime(&currentTime));
28     //The maximum log level printed can be controlled through the MAXLOGLEVEL compile macro (which
29     //The default is 5, which will print up to MESSAGE.
30     //The special level CRITICAL will terminate the execution of the program with an error code.
31     //Unless the log level STDOUT is used, all messages will be issued to stderr.
32     //Since it is known at compile time, any log calls with levels above the maximum one will
33     //no performance penalty
34     //Destroy the UAMMD environment and exit
35     sys->finish();
36 }
```

Complex fluids in the GPU era

UAMMD

Basic code structure

UAMMD's online presence

2022-05-29

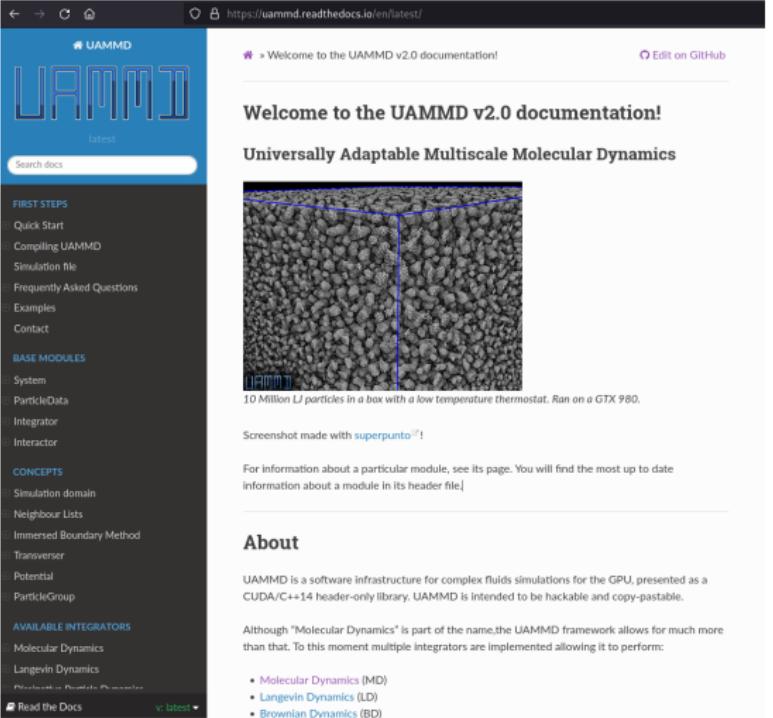
This repository is rich with examples, including an in depth series of them that increasingly introduce the UAMMD concepts in a literary, tutorial fashion.



UAMMD's online presence

Documentation

<https://uammd.readthedocs.io>



The screenshot shows the homepage of the UAMMD v2.0 documentation. The left sidebar contains navigation links for 'FIRST STEPS' (Quick Start, Compiling UAMMD, Simulation file, Frequently Asked Questions, Examples, Contact), 'BASE MODULES' (System, ParticleData, Integrator, Interactor), 'CONCEPTS' (Simulation domain, Neighbour Lists, Immersed Boundary Method, Transverser, Potential, ParticleGroup), 'AVAILABLE INTEGRATORS' (Molecular Dynamics, Langevin Dynamics), and 'Read the Docs' and 'v. latest' buttons. The main content area features a large image of a simulation box containing 10 million LJ particles. Below the image, text reads: '10 Million LJ particles in a box with a low temperature thermostat. Run on a GTX 980.' A note below states: 'Screenshot made with superpunto[®]!'. A link to 'About' is present.

Complex fluids in the GPU era

UAMMD

Basic code structure

UAMMD's online presence

2022-05-29

UAMMD's online presence
Documentation
<https://uammd.readthedocs.io>



Finally, a complete documentation is available, with in-depth practical and theoretical information about each existing module and utility in UAMMD.

This thesis main contributions

1 Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

2 Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
[Raul P. Pelaez et. al. JSTAT 2018.]
[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
- Eulerian-Lagrangian coupling.

3 Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
[Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
[Raul P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]
- Intra-cellular viscosity.
[Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

Complex fluids in the GPU era

Conclusions

2022-05-29

This thesis main contributions

This thesis main contributions

Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
[Raul P. Pelaez et. al. JSTAT 2018.]
[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
- Eulerian-Lagrangian coupling.

Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
[Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
[Raul P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]
- Intra-cellular viscosity.
[Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

Raúl P. Peláez

Complex fluids in the GPU era

44 / 46

This thesis main contributions

1 Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

2 Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
[Raul P. Pelaez et. al. JSTAT 2018.]
[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
- Eulerian-Lagrangian coupling.

3 Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
[Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
[Raul P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]
- Intra-cellular viscosity.
[Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

Complex fluids in the GPU era

Conclusions

2022-05-29

This thesis main contributions

This thesis main contributions

- Software.
 - UAMMD: Complex fluids in the GPU
 - Superpunto: A particle visualizer
 - GPU post-processing tools: RDF, MSD, correlations...

Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
 - [Raul P. Pelaez et. al. JSTAT 2018]
 - [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021]
 - Eulerian-Lagrangian coupling.

Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
 - [Raul P. Pelaez et. al. JSTAT 2018]
 - Dynamics of star-polymer solutions.
 - [Raul P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]
 - Intra-cellular viscosity.
 - [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

Complex fluids in the GPU era

Conclusions

2022-05-29

This thesis main contributions

- **Software.**
 - UAMMD: Complex fluids in the GPU
 - Superpunto: A particle visualizer
 - GPU post-processing tools: RDF, MSD, correlations...

- **Algorithms.**
 - Electrostatics and Hydrodynamics in doubly periodic domains.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
 - Eulerian-Lagrangian coupling.

- **Physics.**
 - Colloid diffusion under soft (and hard) two-dimensional confinement.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - Dynamics of star-polymer solutions.
 - [Raul P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]
 - Intra-cellular viscosity.
 - [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

This thesis main contributions

1 Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

2 Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
- [Raul P. Pelaez et. al. JSTAT 2018.]
- [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
- Eulerian-Lagrangian coupling.

3 Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
- [Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
- [Raul P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]
- Intra-cellular viscosity.
- [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

■ Software.

Electrostatics and Hydrodynamics in doubly periodic domains.
 [Raúl P. Pelaez et. al. JSTAT 2018.]
 [O. Maxian, Raúl P. Pelaez et. al. J. Chem. Phys. 2021.]

■ Algorithms.

Colloid diffusion under soft (and hard) two-dimensional confinement.
 [Raúl P. Pelaez et. al. 2018a]
 [Raúl P. Pelaez et. al. 2018b]
 Dynamics of star-polymer solutions.
 [Raúl P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]
 Eulerian-Lagrangian coupling.
 [Rodríguez, Sanz, Raúl P. Pelaez et.al. 2019 Adv. Biosyst.]

■ Physics.

Complex fluids in the GPU era
 Conclusions

Complex fluids in the GPU era

Conclusions

└ This thesis main contributions

2022-05-29

This thesis main contributions

1 Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

2 Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.

[Raúl P. Pelaez et. al. JSTAT 2018.]

[O. Maxian, Raúl P. Pelaez et. al. J. Chem. Phys. 2021.]

- Eulerian-Lagrangian coupling.

3 Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.

[Raúl P. Pelaez et. al. JSTAT 2018.]

- Dynamics of star-polymer solutions.

[Raúl P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]

- Intra-cellular viscosity.

[Rodríguez, Sanz, Raúl P. Pelaez et.al. 2019 Adv. Biosyst.]

Complex fluids in the GPU era

Conclusions

2022-05-29

This thesis main contributions

- Software.
 - UAMMD: Complex fluids in the GPU
 - Superpunto: A particle visualizer
 - GPU post-processing tools: RDF, MSD, correlations...
- Algorithms.
 - Electrostatics and Hydrodynamics in doubly periodic domains.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
 - Eulerian-Lagrangian coupling.
- Physics.
 - Colloid diffusion under soft (and hard) two-dimensional confinement.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - Dynamics of star-polymer solutions.
 - [Raul P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]
 - Intra-cellular viscosity.
 - [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

This thesis main contributions

1 Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

2 Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.

[Raul P. Pelaez et. al. JSTAT 2018.]

[O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

- Eulerian-Lagrangian coupling.

3 Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.

[Raul P. Pelaez et. al. JSTAT 2018.]

- Dynamics of star-polymer solutions.

[Raul P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]

- Intra-cellular viscosity.

[Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

Complex fluids in the GPU era

Conclusions

2022-05-29

This thesis main contributions

- Software.
 - UAMMD: Complex fluids in the GPU
 - Superpunto: A particle visualizer
 - GPU post-processing tools: RDF, MSD, correlations...

■ Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
- Eulerian-Lagrangian coupling.

■ Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
 - [Raul P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]
- Intra-cellular viscosity.
 - [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

This thesis main contributions

1 Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

2 Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
- Eulerian-Lagrangian coupling.

3 Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
 - [Raul P. Pelaez and Delgado-Buscalloni, R. Macromolecules 2020]
- Intra-cellular viscosity.
 - [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

■ Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
- [Raul P. Pelaez et. al. JSTAT 2018.]
- [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

■ Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
- [Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
- [Raul P. Pelaez and Delgado-Buscalioni, R. Macromolecules 2020]
- Intra-cellular viscosity.
- [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

This thesis main contributions

1 Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

2 Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
- [Raul P. Pelaez et. al. JSTAT 2018.]
- [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
- Eulerian-Lagrangian coupling.

3 Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
- [Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
- [Raul P. Pelaez and Delgado-Buscalioni, R. Macromolecules 2020]
- Intra-cellular viscosity.
- [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

Complex fluids in the GPU era

└ Conclusions

└ This thesis main contributions

2022-05-29

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

Algorithms

- Electrostatics and Hydrodynamics in doubly periodic domains.
- [Raul P. Pelaez et. al. JSTAT 2018.]
- [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]

Physics

- Colloid diffusion under soft (and hard) two-dimensional confinement.
- [Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
- [Raul P. Pelaez and Delgado-Buscalioni, R. Macromolecules 2020]
- Intra-cellular viscosity.
- [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

This thesis main contributions

1 Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

2 Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
- [Raul P. Pelaez et. al. JSTAT 2018.]
- [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
- Eulerian-Lagrangian coupling.

3 Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
- [Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
- [Raul P. Pelaez and Delgado-Buscalioni, R. Macromolecules 2020]
- Intra-cellular viscosity.
- [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

Complex fluids in the GPU era

Conclusions

This thesis main contributions

2022-05-29

Complex fluids in the GPU era

Conclusions

2022-05-29

This thesis main contributions

- **Software.**
 - UAMMD: Complex fluids in the GPU
 - Superpunto: A particle visualizer
 - GPU post-processing tools: RDF, MSD, correlations...
- **Algorithms.**
 - Electrostatics and Hydrodynamics in doubly periodic domains.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
 - Eulerian-Lagrangian coupling.
- **Physics.**
 - Colloid diffusion under soft (and hard) two-dimensional confinement.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - Dynamics of star-polymer solutions.
 - [Raul P. Pelaez and Delgado-Buscalioni, R. Macromolecules 2020]
 - Intra-cellular viscosity.
 - [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

This thesis main contributions

1 Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

2 Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
- Eulerian-Lagrangian coupling.

3 Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
 - [Raul P. Pelaez and Delgado-Buscalioni, R. Macromolecules 2020]
- Intra-cellular viscosity.
 - [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

Complex fluids in the GPU era

Conclusions

2022-05-29

This thesis main contributions

- **Software.**
 - UAMMD: Complex fluids in the GPU
 - Superpunto: A particle visualizer
 - GPU post-processing tools: RDF, MSD, correlations...
- **Algorithms.**
 - Electrostatics and Hydrodynamics in doubly periodic domains.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
 - Eulerian-Lagrangian coupling.
- **Physics.**
 - Colloid diffusion under soft (and hard) two-dimensional confinement.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - Dynamics of star-polymer solutions.
 - [Raul P. Pelaez and Delgado-Buscalioni, R. Macromolecules 2020]
 - Intra-cellular viscosity.
 - [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

This thesis main contributions

1 Software.

- UAMMD: Complex fluids in the GPU
- Superpunto: A particle visualizer
- GPU post-processing tools: RDF, MSD, correlations...

2 Algorithms.

- Electrostatics and Hydrodynamics in doubly periodic domains.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
 - [O. Maxian, Raul P. Pelaez et. al. J. Chem. Phys. 2021.]
- Eulerian-Lagrangian coupling.

3 Physics.

- Colloid diffusion under soft (and hard) two-dimensional confinement.
 - [Raul P. Pelaez et. al. JSTAT 2018.]
- Dynamics of star-polymer solutions.
 - [Raul P. Pelaez and Delgado-Buscalioni, R. Macromolecules 2020]
- Intra-cellular viscosity.
 - [Rodríguez, Sanz, Raul P. Pelaez et.al. 2019 Adv. Biosyst.]

UAMMD: What others are doing

Nerea Alcazar

Anomalous diffusion in random networks



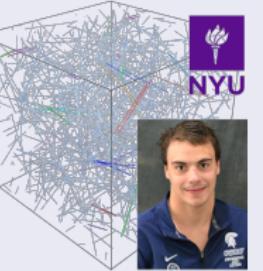
Salvatore Assenza

Coarse-grained DNA potentials



Ondrej Maxian

Hydrodynamics of actin networks



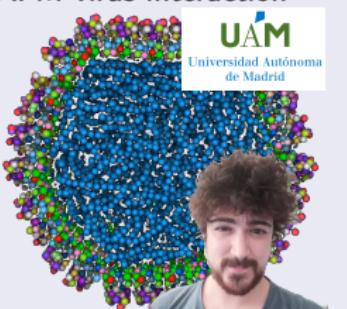
Pablo Palacios

Magnetic nanoparticles



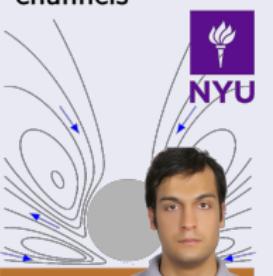
Pablo Ibañez

AFM-virus interaction



Aref Hasemi

Electrohydrodynamics in thin channels

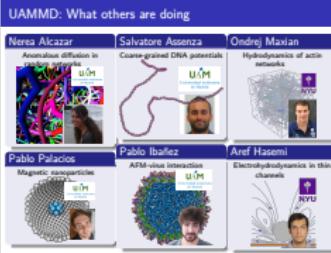


Complex fluids in the GPU era

Conclusions

2022-05-29

UAMMD: What others are doing



UAMMD: What others are doing

Thanks for your attention!