

UAMMD

A CUDA/C++ library for complex fluids

Raúl P. Peláez

Universidad Autónoma de Madrid

May 25, 2022



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

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

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
- $k_B T \partial_q \cdot \mathcal{M} dt$:
Thermal drift. Zero depending on the BCs

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

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 $O(N^2)$

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

Problems

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

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

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.

Hydrodynamics

Fluctuating Stokes equations

Stokes equations

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

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

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

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

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

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} (\mathbf{f}(\mathbf{r}) + \nabla \cdot \mathcal{Z})$$

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

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

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} (\mathbf{f}(\mathbf{r}) + \nabla \cdot \mathcal{Z})$$

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

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

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} (\mathbf{f}(\mathbf{r}) + \nabla \cdot \mathcal{Z})$$

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

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

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} (\mathbf{f}(\mathbf{r}) + \nabla \cdot \mathcal{Z})$$

Force Coupling Method (FCM)

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

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

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

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

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

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

$$\mathbf{v}(\mathbf{r}) = \mathcal{G} (\mathbf{f}(\mathbf{r}) + \nabla \cdot \mathcal{Z})$$

Force Coupling Method (FCM)

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

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

Force Coupling Method

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

Triply Periodic

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

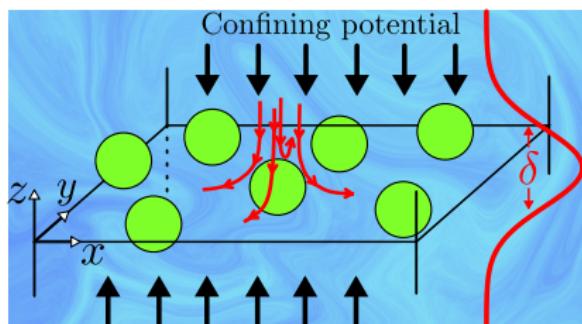
Hydrodynamics in confined geometries

New algorithms for Brownian Dynamics with Hydrodynamic Interactions (BDHI)

Force Coupling Method

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

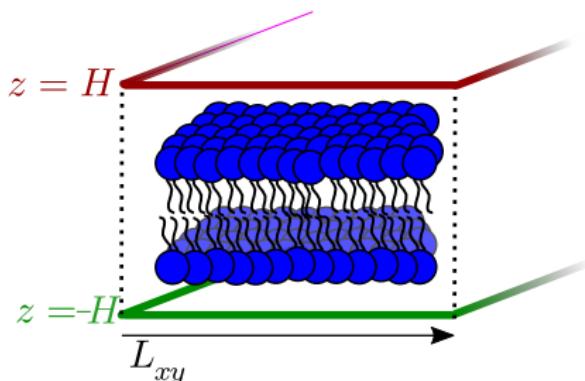
- Quasi two-dimensional (q2D)



q2D: $\delta \rightarrow 0$.

3D fluid, 2D particles

- Doubly Periodic Stokes (DPStokes)



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

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

Supervector notation. Charges $\mathbf{Q} := \{Q_1, \dots, Q_N\}$ $\hat{\mathcal{G}}_P = (\varepsilon_0 k^2)^{-1}$

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

Universally Adaptable Multiscale Molecular Dynamics

What makes UAMMD stand out

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

- Header only and library-like

Universally Adaptable Multiscale Molecular Dynamics

What makes UAMMD stand out

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

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

Universally Adaptable Multiscale Molecular Dynamics

What makes UAMMD stand out

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

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

Universally Adaptable Multiscale Molecular Dynamics

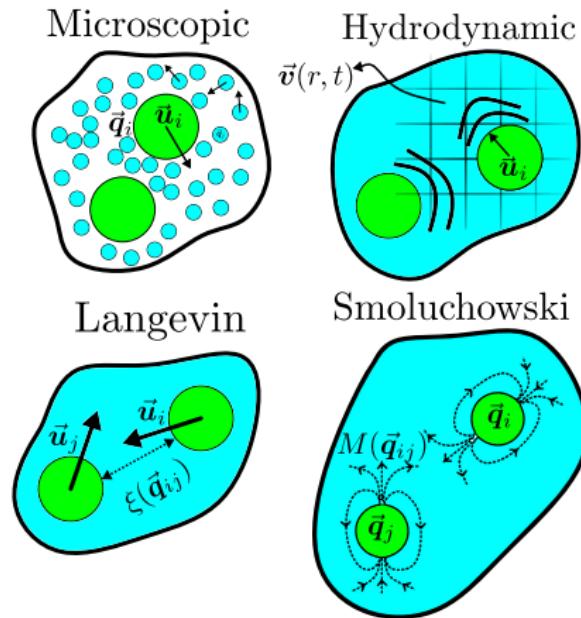
What makes UAMMD stand out

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

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

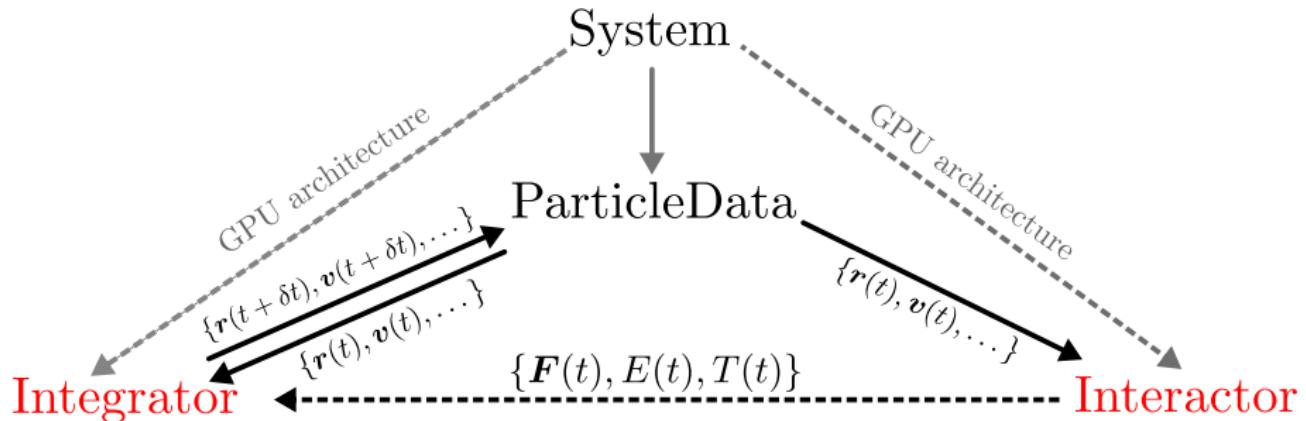
Complex fluids

Usual levels of coarse-grained description



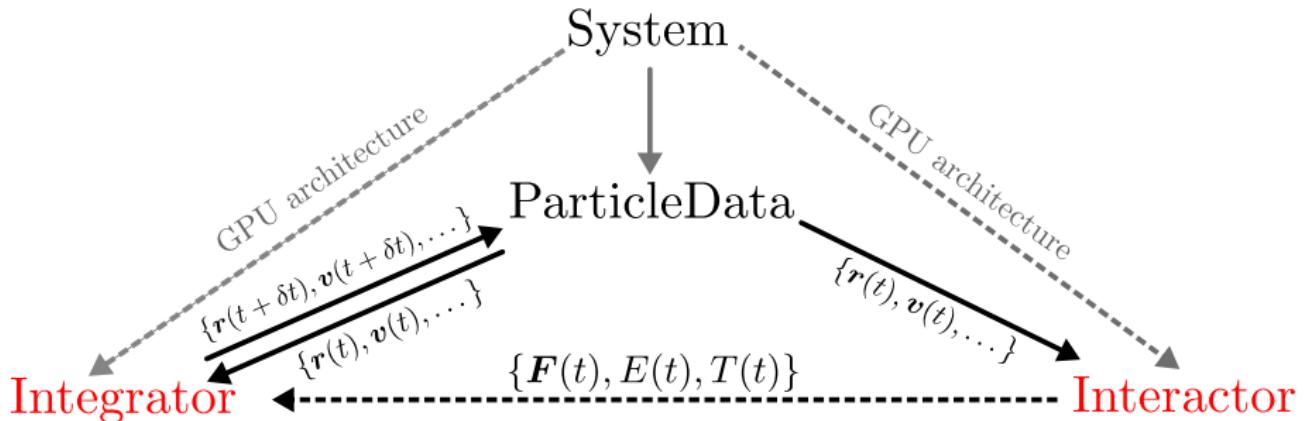
We always have

"Interacting *particles* with a state that evolves"

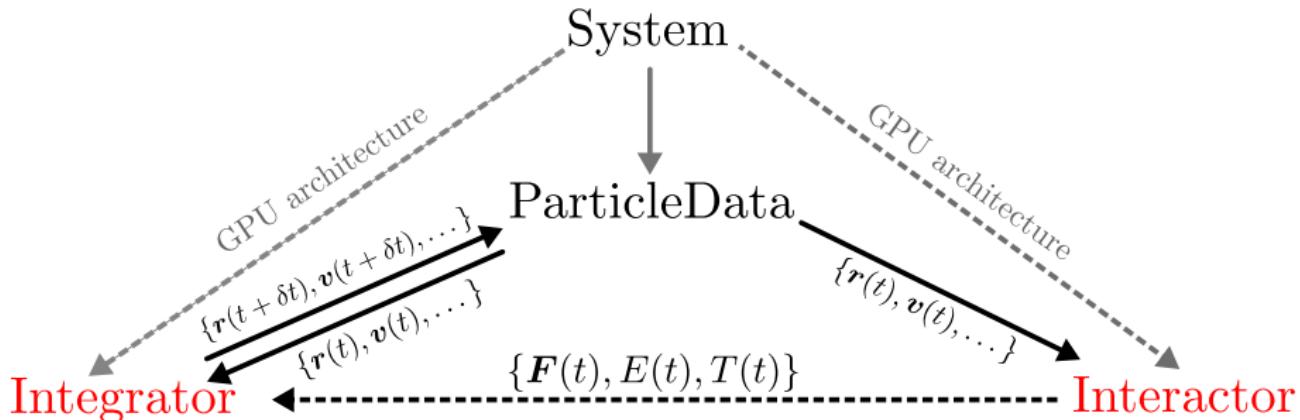


Remember

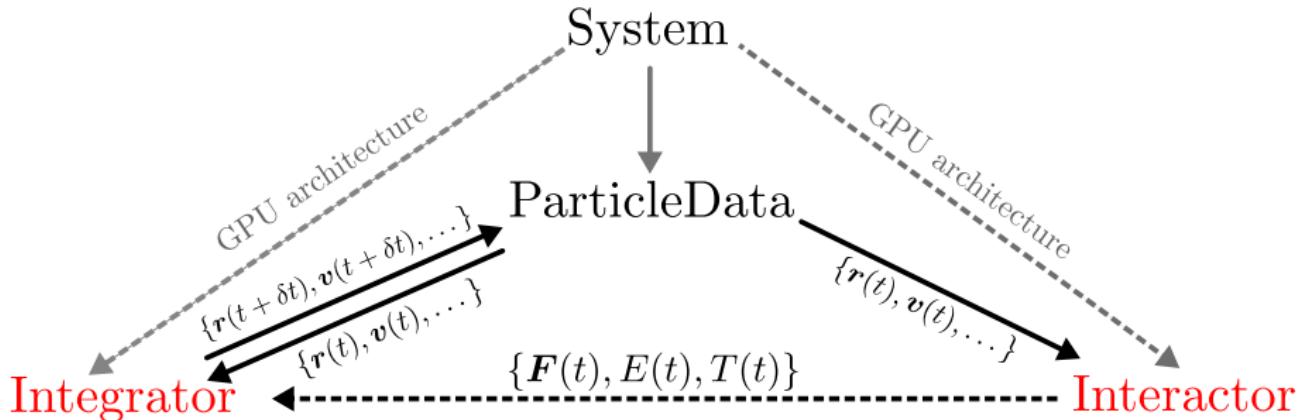
“Interacting *particles* with a state that *evolves*”



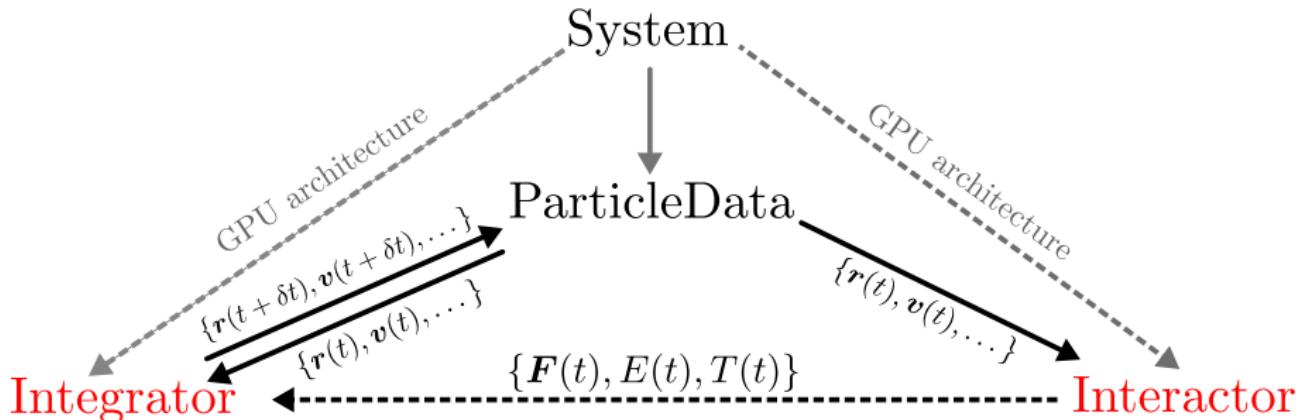
```
#include <uammd.cuh>
int main(int argc, char* argv[]){
    auto sys = std::make_shared<System>(argc, argv);
    ...
}
```



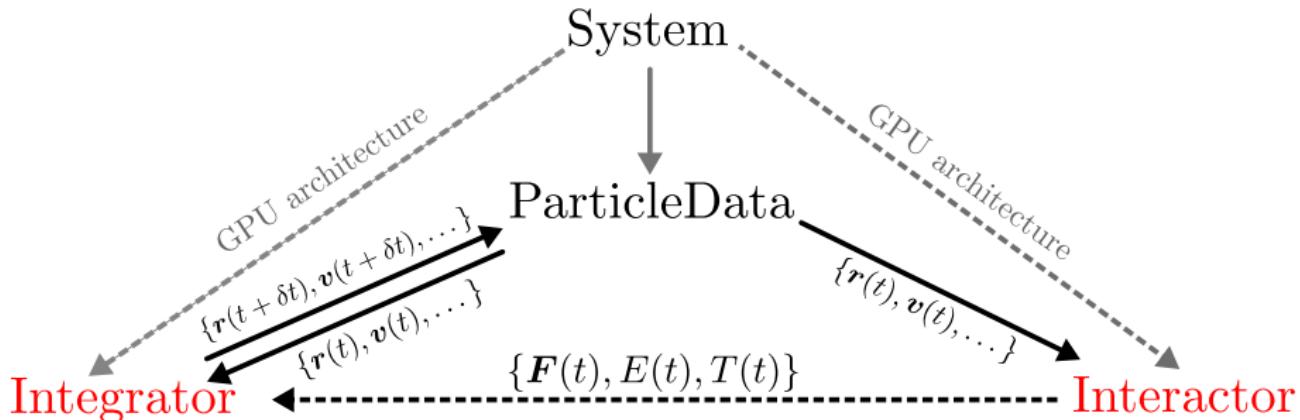
```
#include <uammd.cuh>
int main(int argc, char* argv[]){
    auto sys = std::make_shared<System>(argc, argv);
    const int nP = 1e6; //Number of particles
    auto pd = std::make_shared<ParticleData>(nP, sys);
    ...
}
```



```
...
auto pos = pd->getPos(access::cpu, access::write);
pos[0] = {1,1,1,0}; // x,y,x, color (type)
...
```

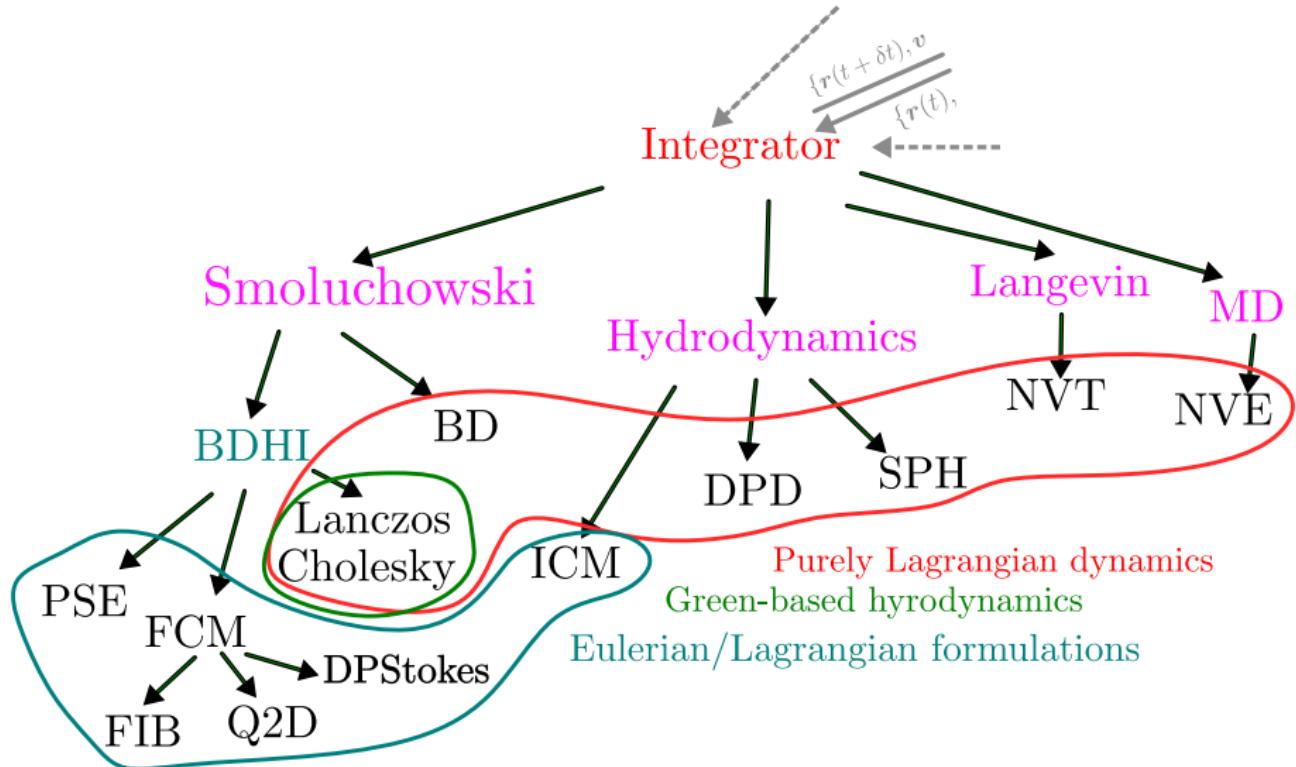


```
...
auto bd = std::make_shared<BDHI::FCM>(pd, /*Parameters*/);
bd->forwardTime(); //Exposed by every Integrator
...
```

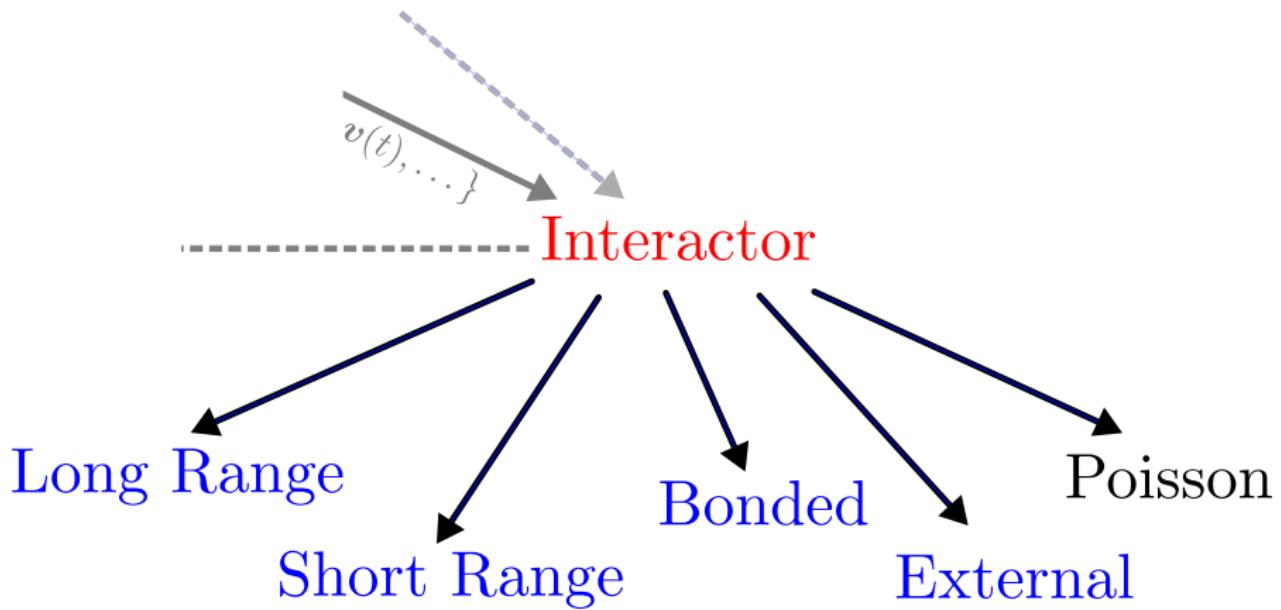


```
...
auto elec = std::make_shared<Poisson>(pd, /*Parameters*/);
elec->sum({.force=true, .energy=false, .virial=false});
//Exposed by every Integrator
bd->addInteractor(elec);
bd->forwardTime();
...
```

Available solvers



Available interactions



UAMMD's online presence

Git repository

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

The screenshot shows the GitHub repository page for 'RaulPPelaez/uammd'. The repository is public and has 11 branches and 0 tags. The 'Code' tab is selected, showing a list of commits. The most recent commit is 'Merge branch 'docs' into v2.x' by RaulPPelaez, 3 days ago, with 798 commits. Other commits include adding images to README.md, updating sketch images in docs, and various updates to examples, extensions, src, test, .gitignore, .gitmodules, CHANGELOG.txt, LICENSE.txt, README.md, and compile_flags.txt. The 'About' section describes UAMMD as a CUDA project for Molecular Dynamics, Brownian Dynamics, and Hydrodynamics, intended to simulate a very generic system using modules. It includes tags for cuda, molecular-dynamics, hydrodynamics, and cuda-molecular-dynamics. The 'Readme' section links to the file, and the 'License' section shows it uses the GPL-3.0 license. The repository has 25 stars, 4 watchers, and 7 forks. The 'Releases' section indicates no releases have been published, with a link to 'Create a new release'. The 'Packages' section shows no packages have been published, with a link to 'Publish your first package'. The footer features the text 'Universally Adaptable Multiscale Molecular'.

RaulPPelaez Merge branch 'docs' into v2.x 6ea244c 3 days ago 798 commits

.res Add images to README.md 4 years ago

docs Docs: Update sketch image 3 days ago

examples Examples: Update CUDA root detection in Makefiles last month

extensions @ b8f6346 Add preamble.h to extensions 6 months ago

src Merge branch 'docs' into v2.x 3 days ago

test Update tests to new interfaces last month

.gitignore Update .gitignore 2 months ago

.gitmodules Change default uammd-extensions submodule to https 12 months ago

CHANGELOG.txt Update CHANGELOG 3 months ago

LICENSE.txt Add LICENSE.txt 4 years ago

README.md Add Doc badge to README.md 2 months ago

compile_flags.txt Refractor the new Integrator/Interactor creation into the wh... 5 months ago

README.md

Universally Adaptable Multiscale Molecular

UAMMD's online presence

Git repository

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

The screenshot shows a GitHub repository page for 'uammd'. At the top, there is a dropdown menu set to 'v2.x' and a breadcrumb navigation bar with 'UAMMD / examples /'. Below this, a user profile picture of Raúl P. Peláez is shown next to the text 'RaúlPPelaez Examples: Update CI'. The main content area displays a list of files and folders:

- ..
- advanced
- basic_concepts
- generic_md
- integration_schemes
- interaction_modules
- misc
- uammd_as_a_library
- Makefile
- README.md

UAMMD's online presence

Git repository

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

The image shows two side-by-side screenshots of GitHub repository pages. Both pages have a header with a dropdown menu set to 'v2.x' and the URL 'UAMMD / examples / basic_concepts /'. The left screenshot shows a list of files under the 'basic_concepts' folder, while the right screenshot shows a list of subfolders.

Left Screenshot (basic_concepts folder contents):

- ..
- 1-system.cu
- 10-initial_configuration.cu
- 11-measuring_things.cu
- 2-hello_world.cu
- 3-more_system.cu
- 4-uammd_types.cu
- 5-particle_data.cu
- 6-particle_data2.cu
- 7-moving_particles.cu
- 8-interacting_particles.cu
- 9-reading_parameters.cu
- Makefile
- README.md

Right Screenshot (basic_concepts subfolders):

- ..
- advanced
- basic_concepts
- generic_md
- integration_schemes
- interaction_modules
- misc
- uammd_as_a_library
- Makefile
- README.md

UAMMD's online presence

Git repository

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

The image shows two GitHub code editor panes side-by-side. Both panes have a 'v2.x' dropdown at the top left and a 'History' button at the top right.

Left Pane: The title is 'UAMMD / examples / basic_concepts /'. It shows a list of files:

- 1-system.cu
- 10-initial_configuration.cu
- 11-measuring_things.cu
- 2-hello_world.cu (highlighted)
- 3-more_system.cu
- 4-uammd_types.cu
- 5-particle_data.cu
- 6-particle_data2.cu
- 7-moving_particles.cu
- 8-interacting_particles.cu
- 9-reading_parameters.cu
- Makefile
- README.md

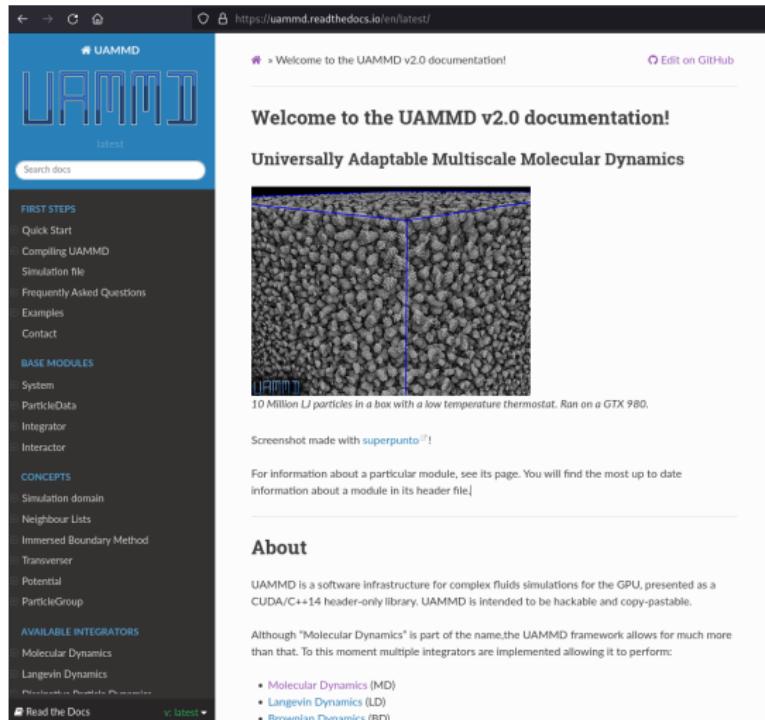
Right Pane: The title is 'UAMMD / examples / basic_concepts / 2-hello_world.cu'. It shows the content of the selected file:

```
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 // to error level. In this example we will use System::log to print a simple message along with the date.
7 */
8
9 //uammd.h is the basic uammd include containing, among other things, the System struct.
10 #include "uammd.h"
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->logSystem("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->logSystem(WARNING,"Current time is: %s",ctime(&currentTime));
28     //The maximum log level printed can be controlled through the MAXLOGLEVEL compile macro (which is
29     //The special level CRITICAL will terminate the execution of the program with an error code.
30     //Unless the log level STDOUT is used, all messages will be issued to stderr.
31     //Since it is known at compile time, any log calls with levels above the maximum one will
32     //no performance penalty
33     //Destroy the UAMMD environment and exit
34     sys->finish();
35     return 0;
36 }
```

UAMMD's online presence

Documentation

<https://uammd.readthedocs.io>



The screenshot shows the homepage of the UAMMD v2.0 documentation. At the top, there is a navigation bar with icons for back, forward, search, and refresh, followed by the URL <https://uammd.readthedocs.io/en/latest/>. To the right of the URL is an "Edit on GitHub" link.

The main content area has a blue header with the "UAMMD" logo and the word "latest". Below the header, the title "Welcome to the UAMMD v2.0 documentation!" is displayed, followed by the subtitle "Universally Adaptable Multiscale Molecular Dynamics".

On the left side, there is a sidebar with a "Search docs" input field and several sections:

- 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.

At the bottom of the sidebar, there are links for "Read the Docs" and "v. latest".

The main content area features a large image of a simulation box containing 10 million LJ particles. Below the image, the caption 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[®]!"

For more information about a module, it is mentioned that you can see its page or find it in its header file.

About

UAMMD is a software infrastructure for complex fluids simulations for the GPU, presented as a CUDA/C++14 header-only library. UAMMD is intended to be hackable and copy-pastable.

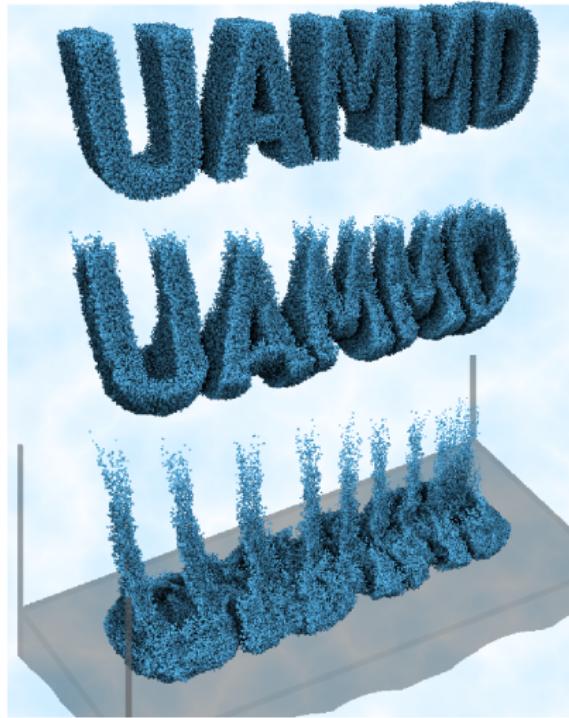
Although "Molecular Dynamics" is part of the name, the UAMMD framework allows for much more than that. To this moment multiple integrators are implemented allowing it to perform:

- Molecular Dynamics (MD)
- Langevin Dynamics (LD)
- Brownian Dynamics (BD)

Lets clone

```
$ git clone git@github.com:RaulPPelaez/2022Bilbao
```

The falling logo

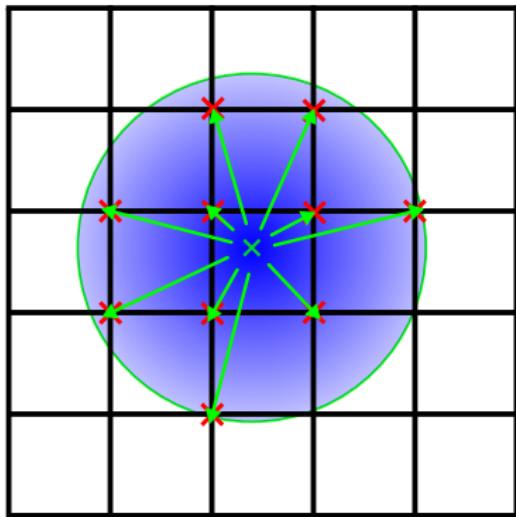


Particle-grid coupling

Particle-grid coupling

Spreading (\mathcal{S}):

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



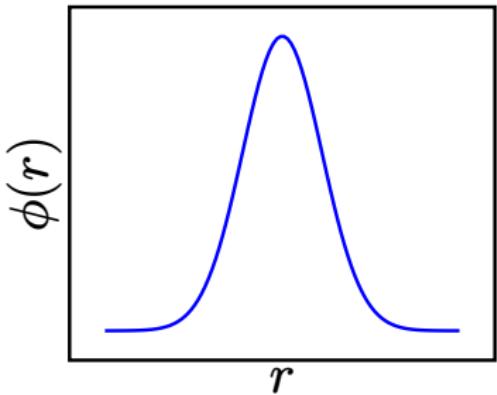
Supervector notation $\mathbf{F} := \{\mathbf{F}_1, \dots, \mathbf{F}_N\}$

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

Particle-grid coupling

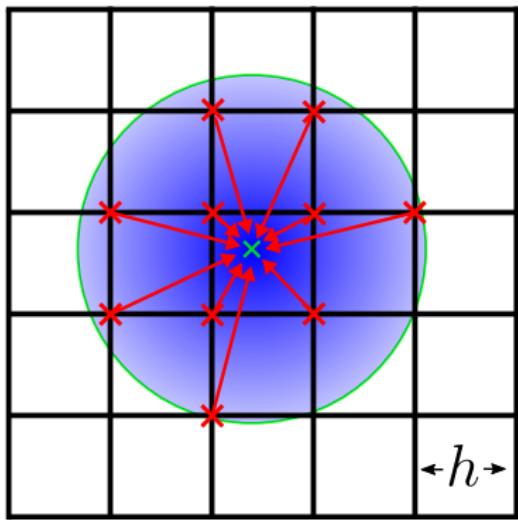
Spreading (\mathcal{S}):

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

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

Interpolation (\mathcal{J}):

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



Particle-grid coupling

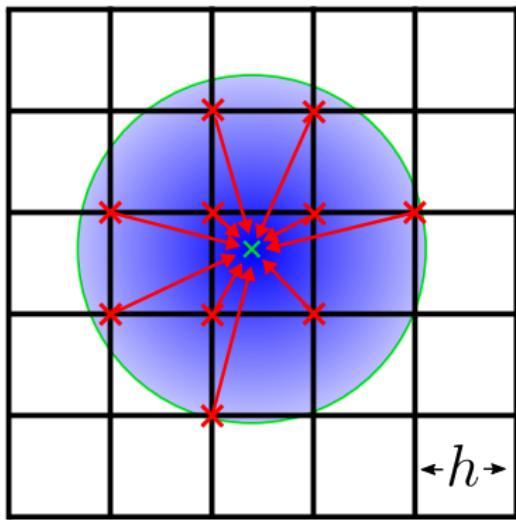
Spreading (\mathcal{S}):

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

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

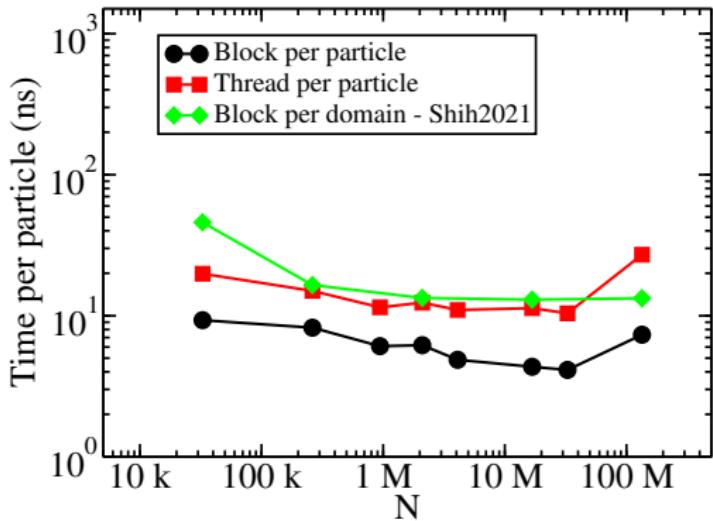
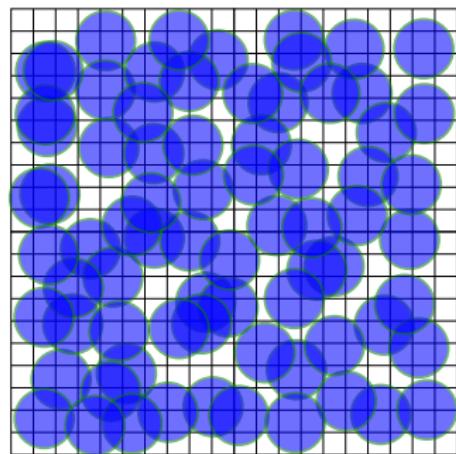
Interpolation (\mathcal{J}):

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



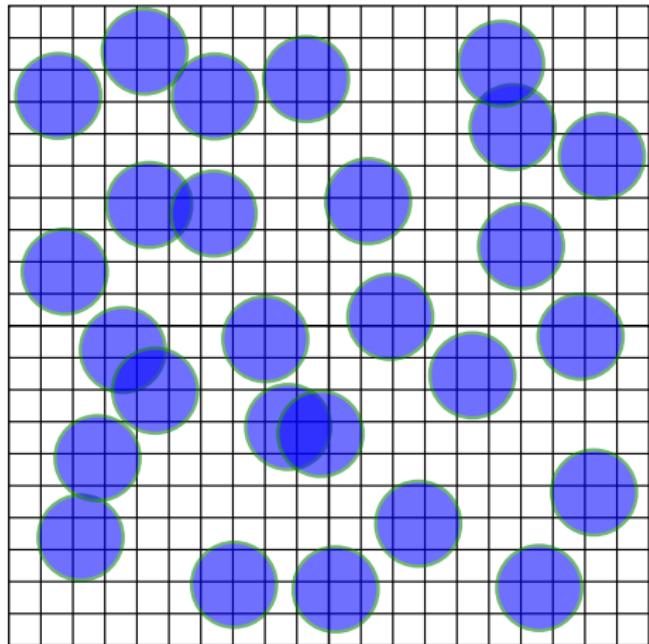
Particle-grid coupling

Performance

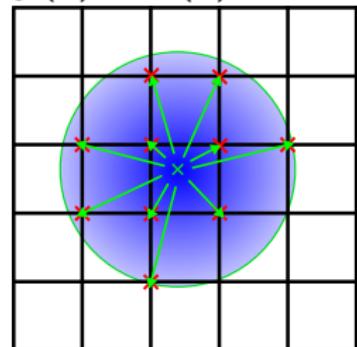


3D random distribution. Ran with a single RTX2080Ti GPU.

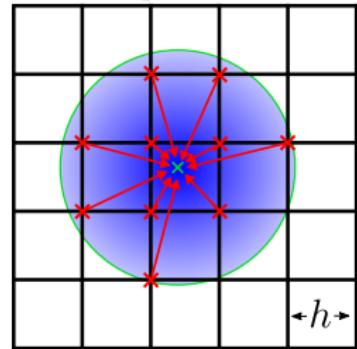
Particle-grid coupling



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



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



Thanks for your attention!