Bluebottle: A high-performance GPU-centric resolved-particle flow solver

Adam Sierakowski & Andrea Prosperetti

Motivation

Instances of systems of particles dispersed in fluid occur often in nature: **dust storms**, **agricultural runoff**, and **beach erosion** all exhibit particulate matter being moved by fluids. Additionally, we use particle-laden fluid flows in technological applications such as **fluidized bed reactors**, which are used in producing oil products such as petroleum and plastics. In the future, we may take advantage of the energy-generation technology **chemical looping combustion** to reduce the release of carbon bi-products into the atmosphere.

Figure: Fluidized sand in the Mojave Desert (Wikipedia)

Background

Using experimental methods, it is very difficult to track the trajectories of individual particles, and it is nearly impossible to measure the forces on the particles and understand the fluid motion. We seek to learn about these fluid/particle systems using **numerical simulations**, but computers cannot accurately simulate an entire dust storm because of their wide range of length scales. To develop models of typical flow behaviors, we use **ensemble averaging** to assimilate many instances of various particle configurations into a course-grained model that is more easily investigated.

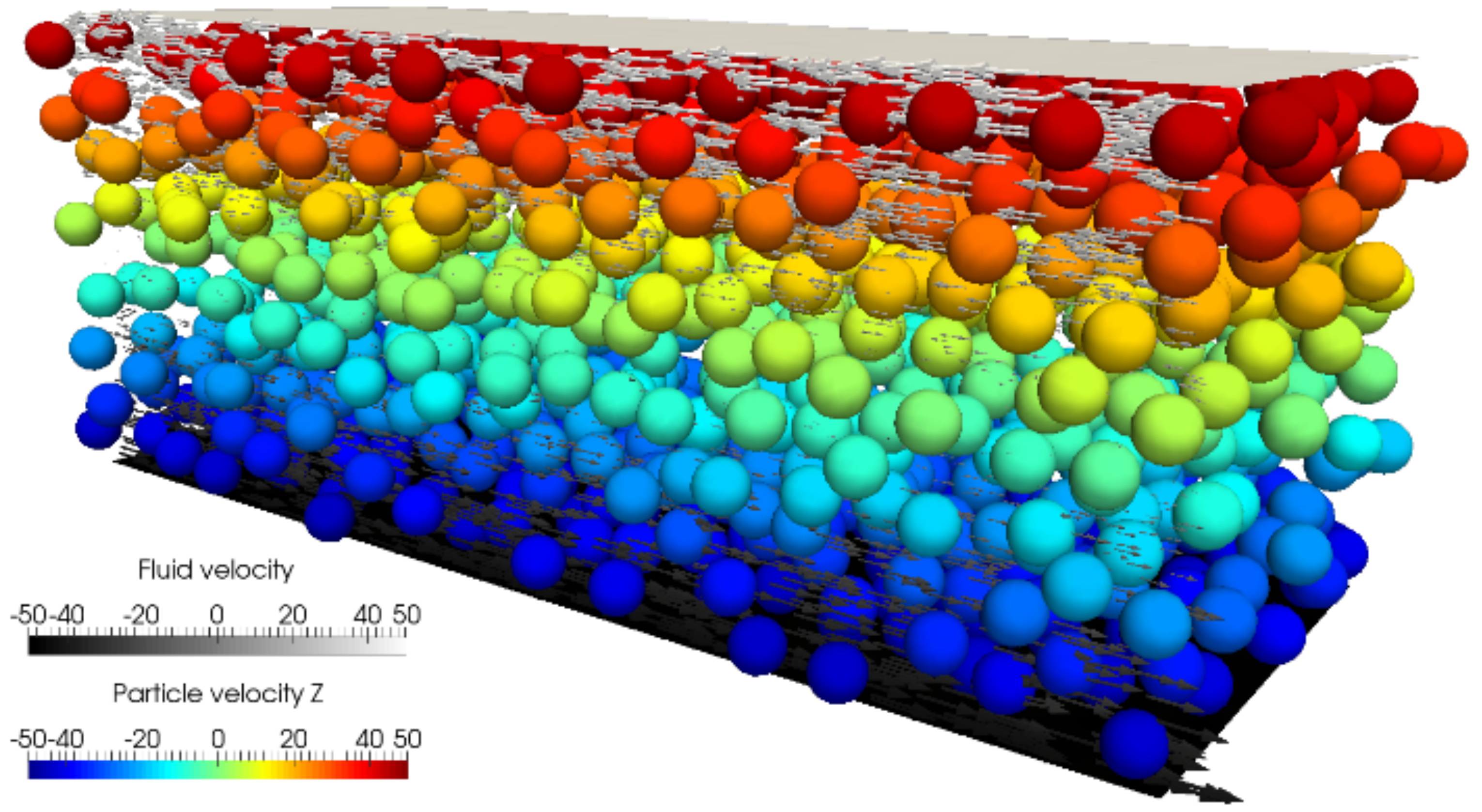


Figure: Shearing flow with 1,000 particles

The incompressible Navier-Stokes equations (1) and (2) describe the motion of the fluid while the particles impose no-slip (friction) and no-penetration boundary conditions on the fluid.

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g} \qquad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \qquad (2)$$

State of the art

Many numerical methods exist for the simulation of disperse-phase flows. **Point particle methods** are capable of simulating millions of particles, but at the cost of physical accuracy. **Body-fitted mesh methods** achieve high accuracy at the prohibitive cost of re-meshing when particles move. In both cases, calculation of the particle forces requires **error-inducing interpolations**.

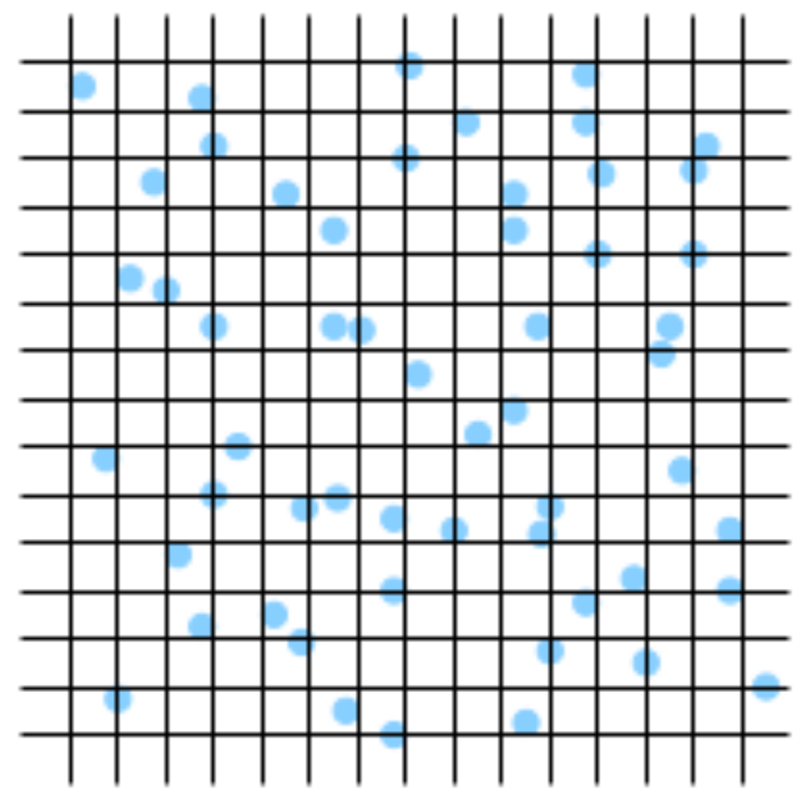


Figure: Under-resolved point particles

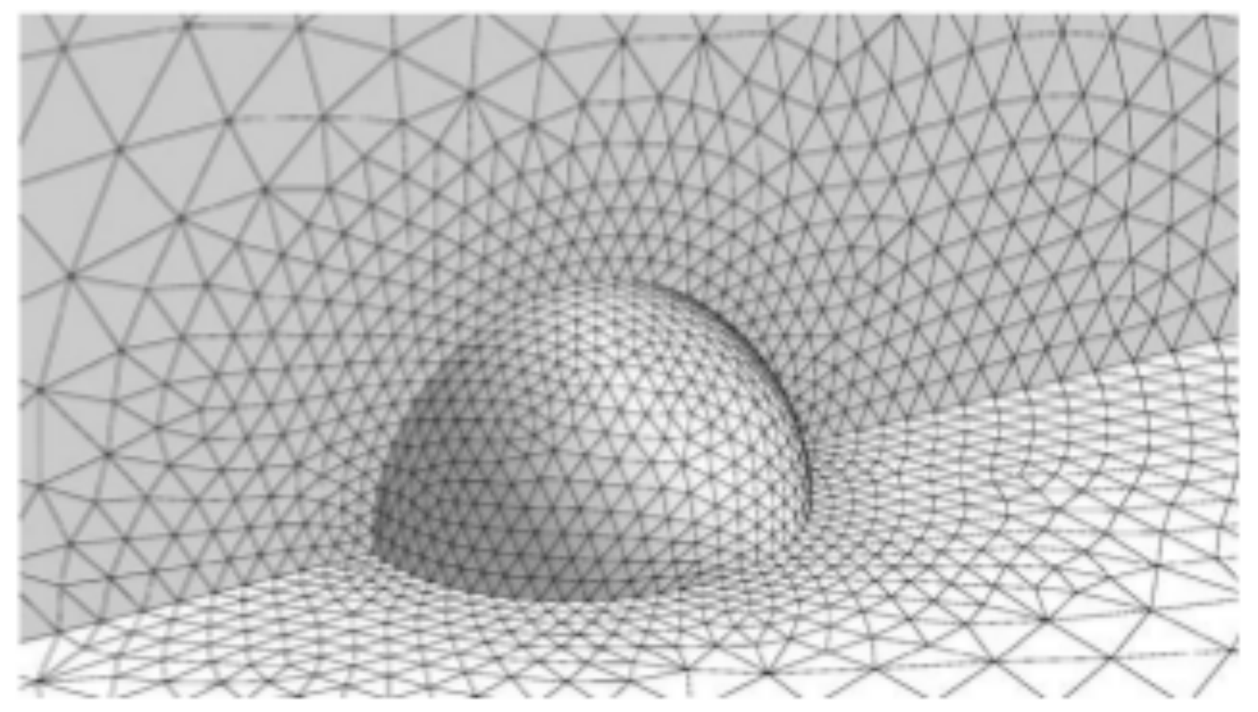
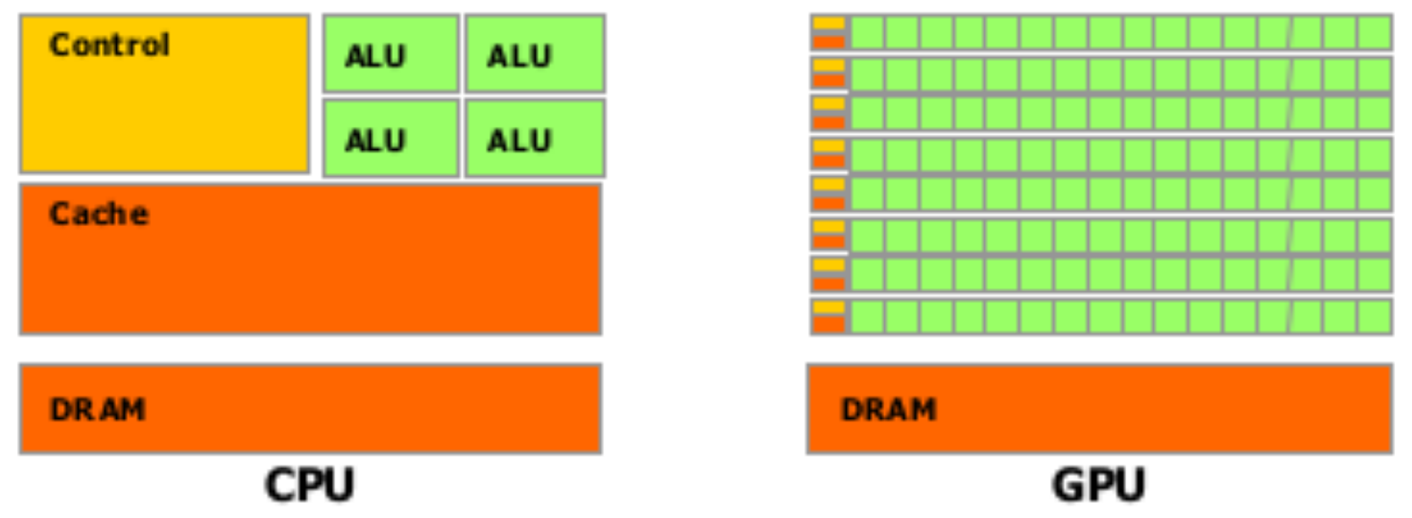


Figure: Time-consuming body-fitted mesh (Popiolek & Awruch, 2005)

GPU-centric code design



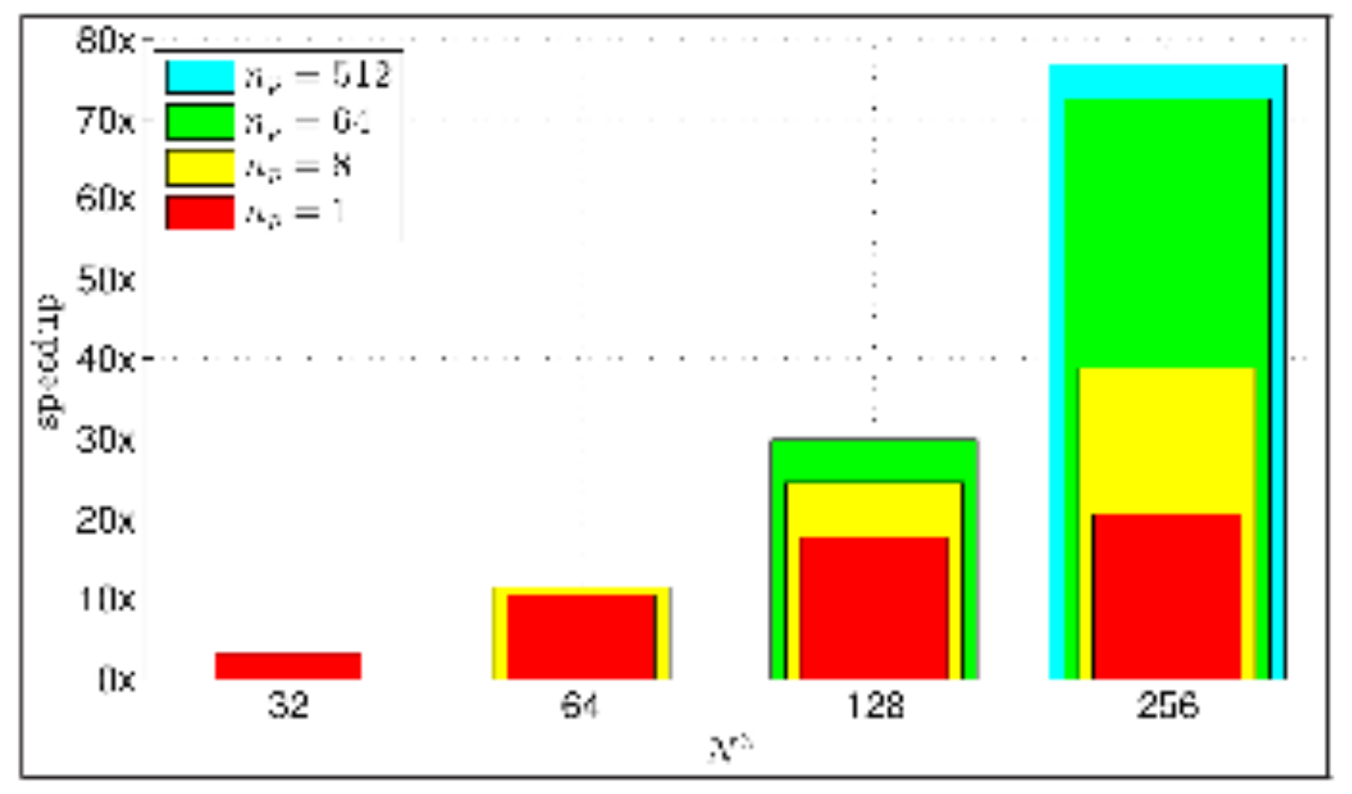


Figure (left): GPUs have fewer "brains" and more "brawn" than CPUs. Figure (right): Bluebottle performance compared to CPU implementation

Specially designed for rendering graphics and driven by the consumer gaming industry, the Graphic Processing Unit (GPU) is a massively parallel, multithreaded, many-core processor. Compared to a CPU, a GPU has many transistors devoted to performing calculations and very few that are dedicated to data caching and flow control. Distinct pieces of hardware, GPUs are controlled by CPUs, and all data must be copied from system RAM to GPU RAM for GPU computation. Thus, algorithm development for a GPU involves both parallelization and careful memory management. GPU-accelerated codes use GPUs for only the most expensive parts of a legacy CPU code and frequent memory transfers between host and device wash away all performance benefits. **GPU-centric code can run up to 100 times faster than CPU or GPU-accelerated code** if carefully designed to minimize the need for these memory transfers. **We have designed Bluebottle to utilize GPUs for simulating particle-laden flows.**

The flow solver

We numerically solve the incompressible Navier-Stokes equations (1) and (2) in the bulk flow on a staggered Cartesian grid using the projection method:

1. Calculate intermediate velocity $\mathbf{u}^*$ using the second-order Adams-Bashforth method for the convective term and an explicit forward-time central-space method for the diffusive term:
$$\mathbf{u}^* = \mathbf{u}^n + \Delta t \left[-(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1/2} + \nu \nabla^2 \mathbf{u}^n + \mathbf{g} \right] \qquad (3)$$
2. Solve for the pressure that will project $\mathbf{u}^*$ into a divergence-free space
$$\nabla^2 p^{n+1/2} = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}^* \qquad (4)$$
3. Step $\mathbf{u}^n$ forward to $\mathbf{u}^{n+1}$ by projecting $\mathbf{u}^*$ onto a divergence-free space to satisfy (2)
$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla p^{n+1/2} \qquad (5)$$

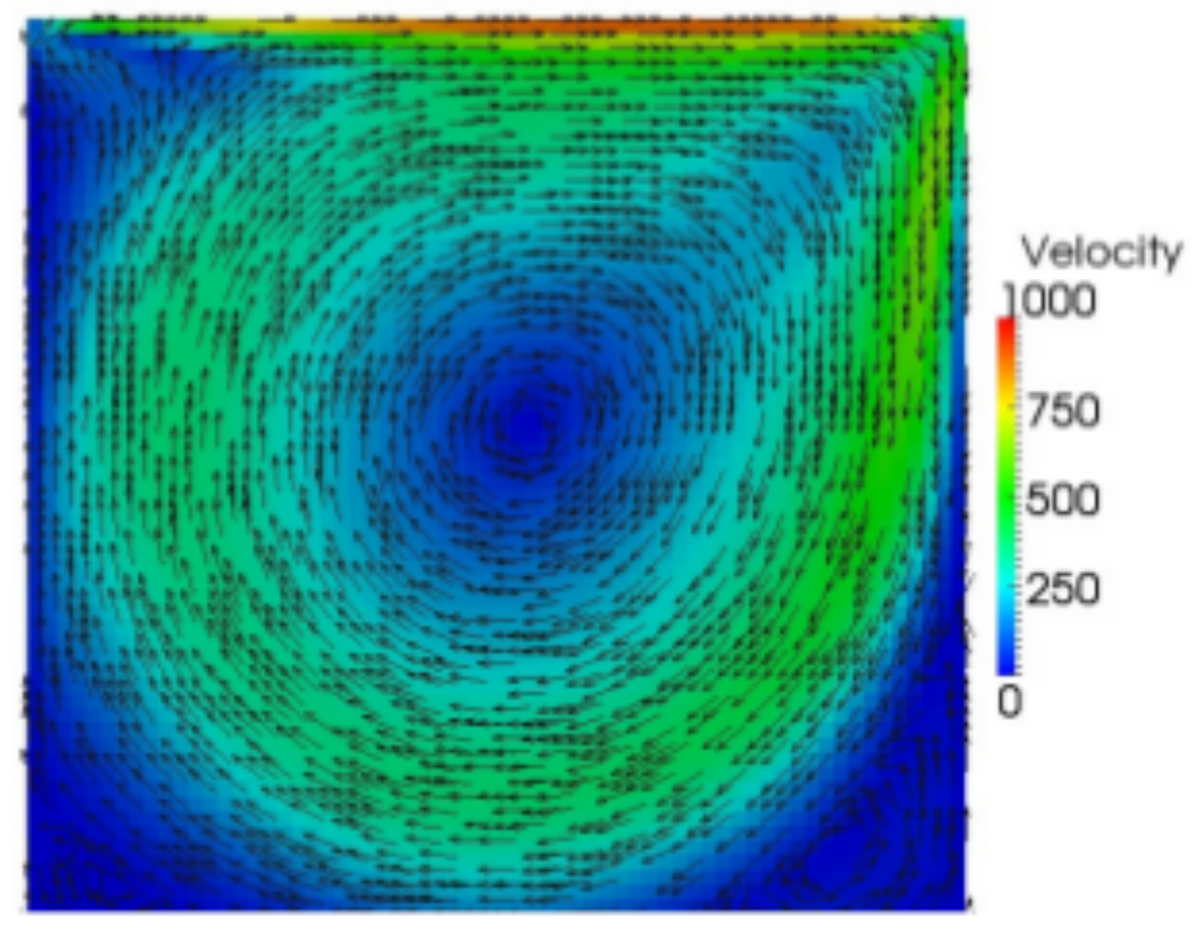


Figure: A lid-driven cavity validation at Re = 1000

To incorporate the Physalis method into the projection method, we repeat the Physalis steps and the projection steps until solutions (8) and (5) converge before proceeding to the next time step.

The Physalis method

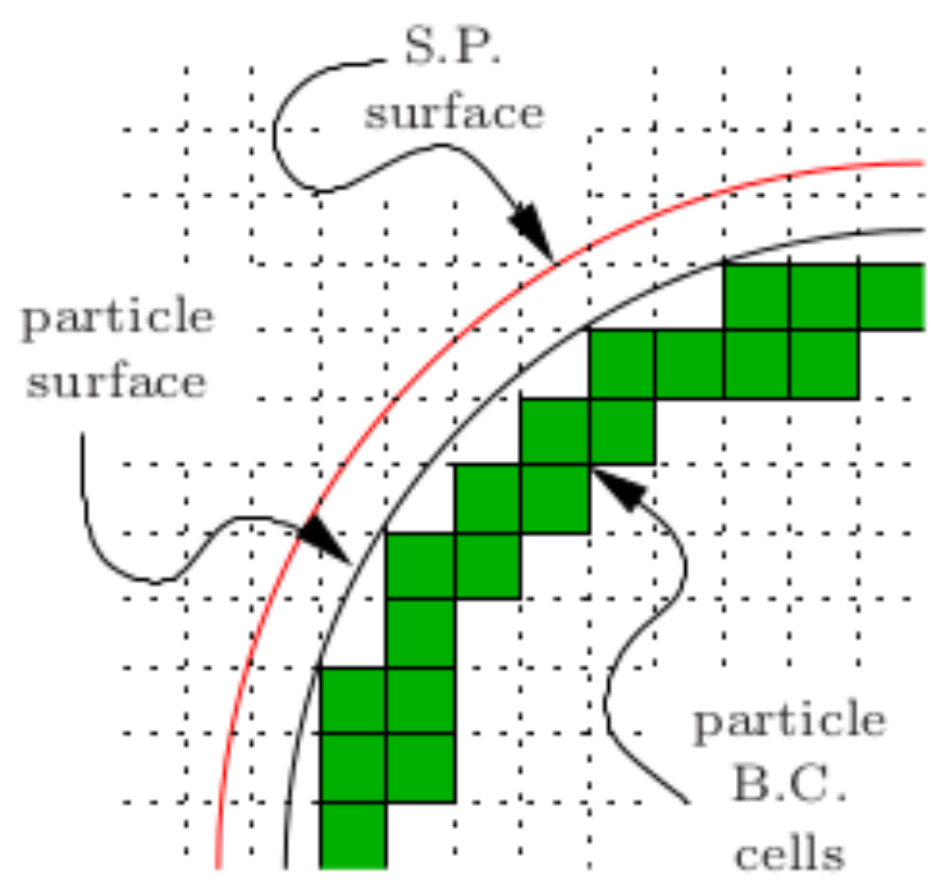


Figure: Physalis boundary conditions

The Physalis method takes advantage of the efficiency of a regular Cartesian finite-difference mesh while applying **spectrally-accurate particle boundary conditions**. In a particle reference frame in a region near the surface of a particle, due to the no slip condition, (1) can be approximated as the Stokes equation

$$-\nabla p + \mu \nabla^2 \mathbf{u} = 0. \qquad (6)$$

Using p_n, ϕ_n, and χ_n solid harmonics of order n,

$$p_n = \left(\frac{r}{a} \right)^n \sum_{m=-n}^{n} p_{nm} \, Y_n^m (\theta, \phi), \qquad (7)$$

Lamb provided the general solution of (6) in the presence of a spherical boundary with radius a:

$$\mathbf{u} = \frac{\nu}{a^2} \sum_{n=-\infty}^{\infty} \frac{1}{(n-1)(2n+3)} \left[\frac{1}{2} (n+3) r^2 \nabla p_n - n \mathbf{x} p_n \right] + \frac{\nu}{a} \sum_{n=-\infty}^{\infty} \left[a \nabla \phi_n + \nabla \times (\mathbf{x} \chi_n) \right]. \qquad (8)$$

After finding the complex-valued dimensionless coefficients p_{nm}, ϕ_{nm}, and χ_{nm} from the flow field using scalar products (S.P.), (8) determines particle boundary conditions (B.C.) at grid nodes near the particle surface; these boundary conditions are used to solve the flow field and the process is iterated until convergence. At convergence, the low-order p_{nm}, ϕ_{nm}, and χ_{nm} give the **hydrodynamic force** (9) **and moment** (10) on the particle without any additional work:

$$\mathbf{F}_{\mathrm{hd}} = \nu (\rho_p - \rho_f)(\dot{\mathbf{w}} - \mathbf{g}) - \pi \mu \nu \left[2N_1^1 \left(p_{11}^{\Re} + 6\phi_{11}^{\Re} \right) \hat{\imath} - 2N_1^1 \left(p_{11}^{\Im} + 6\phi_{11}^{\Im} \right) \hat{\jmath} - N_0^1 \left(p_{10}^{\Re} + 6\phi_{10}^{\Re} \right) \hat{k} \right] \qquad (9)$$

$$\mathbf{L}_{\mathrm{hd}} = \nu a^2 (\rho_p - \rho_f) \dot{\mathbf{\Omega}} - 8\pi \mu \nu a \left[N_1^1 \chi_{11}^{\Re} \hat{\imath} - N_1^1 \chi_{11}^{\Im} \hat{\jmath} - N_1^0 \chi_{10}^{\Re} \hat{k} \right] \qquad (10)$$

Example simulations

Bluebottle is capable of simulating configurations ranging from particles in Stokes flow to flows at moderate Reynolds number. Since the Physalis method resolves the particles fully within the Navier-Stokes flow solver, Bluebottle accurately captures the physical behavior of particle-laden flows at arbitrary particle volume fraction. **Short-range particle-particle interactions are modeled using lubrication theory and collisions are modeled as elastic spheres.** A turbulent concurrent precursor may be used to inject turbulent inflow.

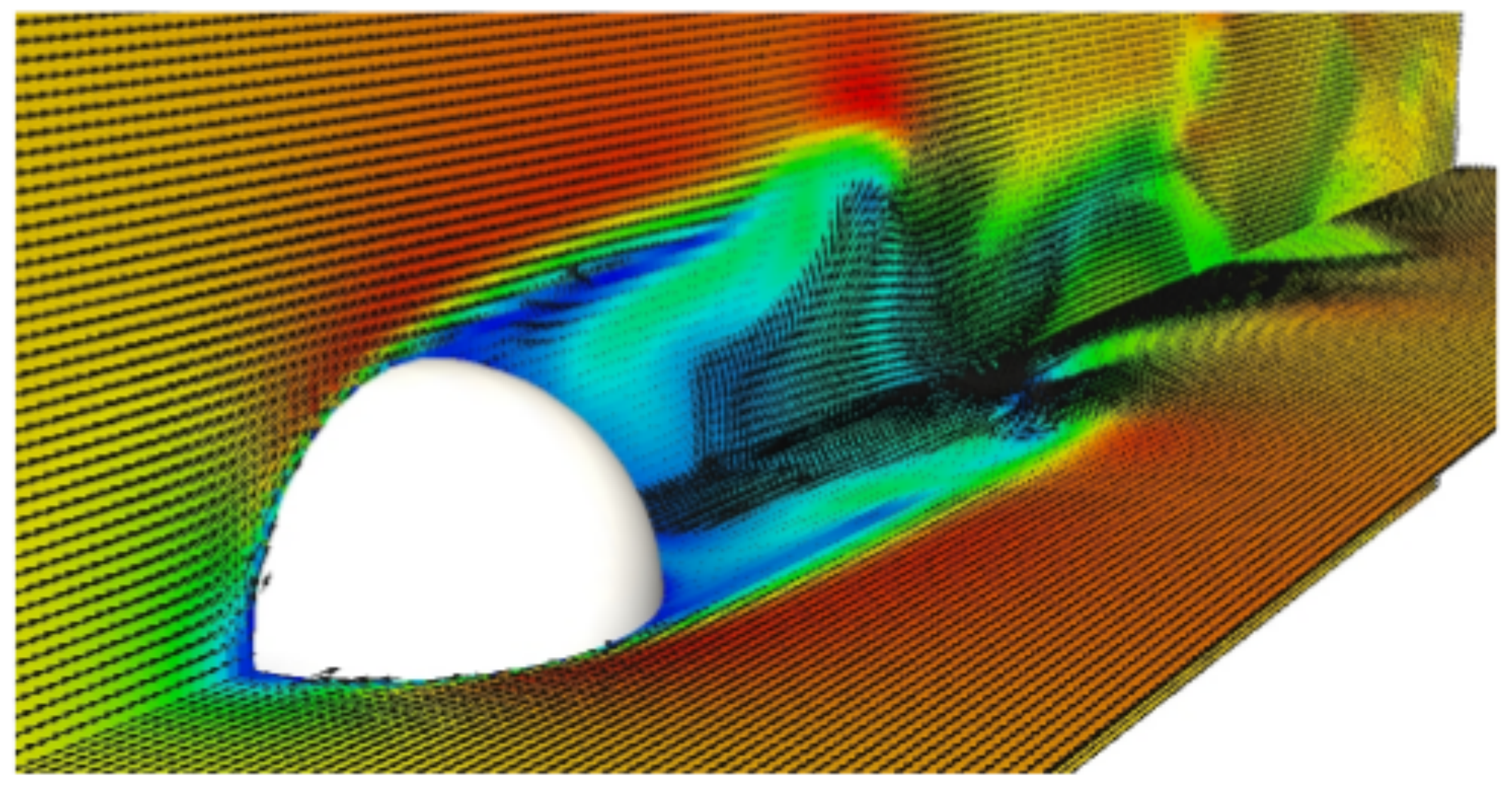


Figure: Fluid flow around a sphere at Re = 500

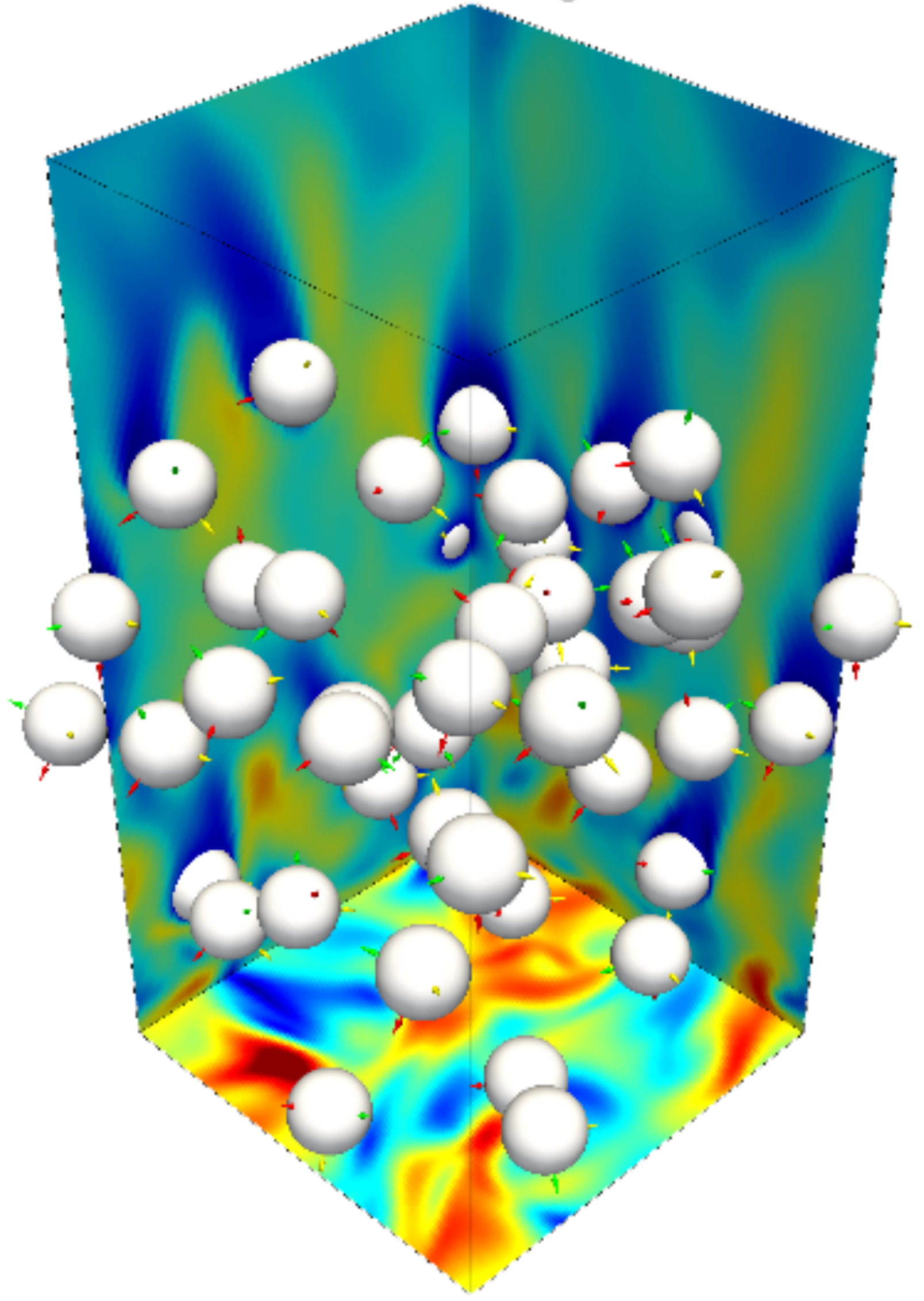


Figure: Sedimentation of 32 particles into decaying homogeneous isotropic turbulence

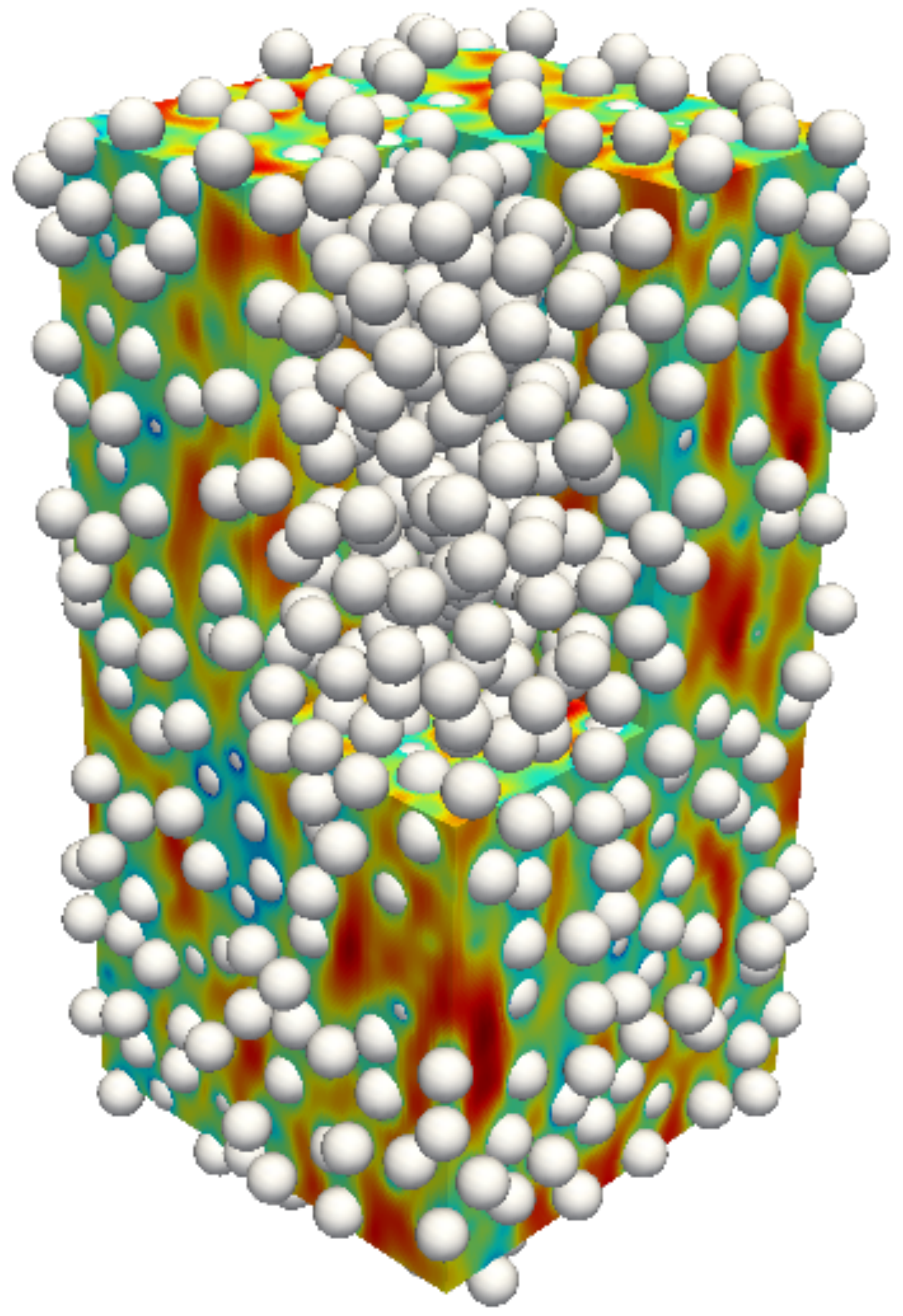


Figure: 1,000 particles (21% volume fraction) sedimenting under gravity

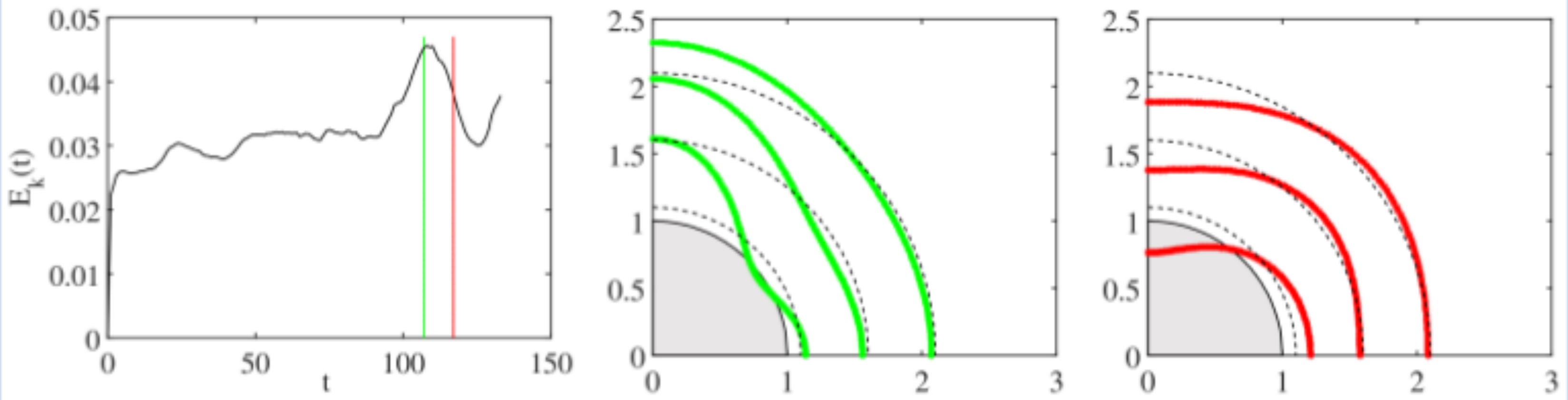
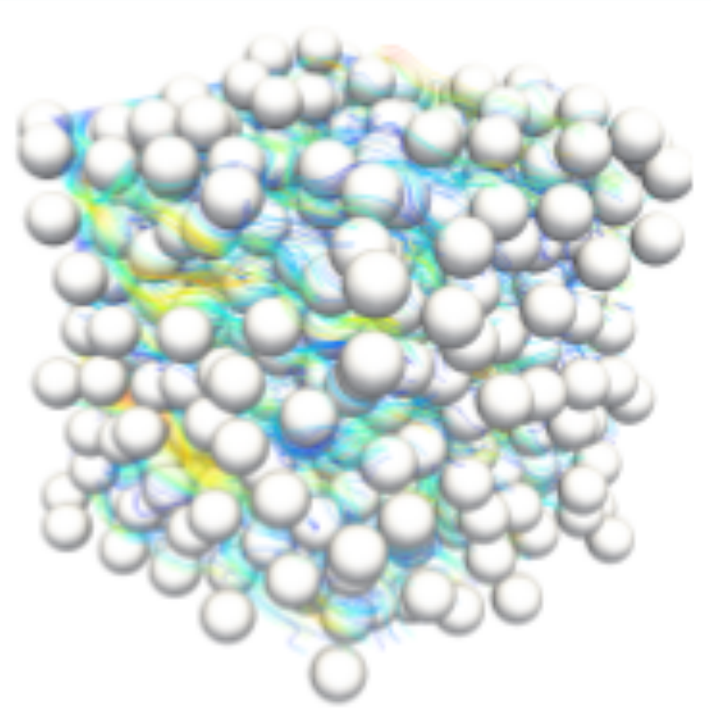


Figure: Particle pair distributions for sedimentation (plots courtesy of Yayun Wang)

Open source code available



http://lucan.me.jhu.edu

Are you interested in learning more about Bluebottle or the Physalis method? Visit our website for a more detailed development. Would you like to run some particle-laden flow simulations of your own? **We have released Bluebottle under an open source license** and we encourage you to download, share, and/or contribute to the source code.

NSF CBET Award Number 1335965, Program Manager William Olbricht

Contact Adam Sierakowski at sierakowski@jhu.edu.