SAND2015-5407C

A Highly-Scalable Implicit SPH Code for Simulating Single- and Multi-phase Flows in Geometrically Complex Bounded Domains

Nathaniel Trask
Brown University
182 George St
Providence, RI
nathaniel_trask@brown.edu

Kyungjoo Kim
Center for Computing Research
Sandia National Laboratories
P.O. Box 5800, MS 1320
Albuquerque, NM 87185

Alexandre Tartakovsky
Pacific Northwest National Laboratory
PO Box 999
Richland, WA 99352

Mauro Perego
Center for Computing Research
Sandia National Laboratories
P.O. Box 5800, MS 1320
Albuquerque, NM 87185

Michael L. Parks
Center for Computing Research
Sandia National Laboratories
P.O. Box 5800, MS 1320
Albuquerque, NM 87185

ABSTRACT

We present 3D pore-scale simulations of multi-phase flows in a porous medium using a consistent smoothed particle hydrodynamics (SPH) formulation. Unlike grid-based Lagrangian methods, classical SPH formulations are trivially scalable but introduce stiff timestep restrictions and are inconsistent with the governing equations. The present work recovers consistency by using local compact corrections to achieve second order spatial convergence with less overall interprocessor communication. We use an implicit projection scheme to implement the continuum surface force model for multi-phase flow. To achieve a scalable implementation, an interface between two high-performance libraries has been developed: LAMMPS to handle discretization and particle dynamics, and Trilinos to solve the discretized equations using algebraic multigrid. The resulting approach has been demonstrated to scale to more than 150 million particles on 11k cores. Benchmarks demonstrating the accuracy of the method and large scale single-phase and two-phase simulations are used to demonstrate the capabilities of the code.

Categories and Subject Descriptors

X.XX [**Applications**]: [Computational earth and atmospheric sciences, Computational materials science and engineering, Computational astrophysics/astronomy, chemistry, fluid dynamics, mechanics and physics]; X.X.XX [**Performance**]: [Empirical measurement of performance, power and/or resilience on real-world systems Workload characterization and benchmarking]

1. INTRODUCTION

Lagrangian methods provide a natural framework for solving fluid dynamics problems. By advecting degrees of freedom with the streamlines of the underlying flow, the Navier-Stokes equations reduce to a linear system of equations and advection is handled exactly. For multi-phase flow problems and geometries with moving boundaries, this allows for direct tracking of the interface without the need for solving additional equations for the interface dynamics or introducing diffuse boundary representations. For moving mesh discretizations of the Lagrangian Navier-Stokes equations, this strategy is often limited to flows with simple interface dynamics or small boundary motion since the cost associated with maintaining a high quality mesh is prohibitive and scales poorly.

Smooth particle hydrodynamics (SPH) and other meshless particle methods provide a more natural framework, as discretization of the underlying PDE relies only on assembling lists of nearby particles to calculate compact interaction forces. For this reason, SPH has been successfully employed for large scale problems by using a fully explicit discretization and artificial equation of state in the weakly compressible SPH (WCSPH) formulation[14]. This approach is embarrassingly parallel and well suited to implementation on GPU architectures, where simulations of up to a billion particles have been performed[7, 8]. Both the spatial discretization and the artificial equation of state introduce inconsistencies to the system, and convergence can only be recovered by taking increasingly large numbers of neighbors or an increasingly stiff equation of state [17, 2, 9, 21]. As a result, the scalability of the method can only be exploited to study increasingly large problems, rather than study more challenging physics with increased fidelity.

In the current work, small local correction matrices are inverted for each particle to ensure convergent differential operators, while projection methods are used to remove the stiff equation of state in a variation of the so-called incompressible SPH (ISPH) approach[6]. The resulting discretization gives second order convergence in space and time and, by using algebraic multigrid to solve the resulting systems of equations, the resulting approach is weakly scalable in the number of particles. Because the accuracy of the method relies on the correction matrices and not increased numbers of neighbors to control accuracy, the resulting discretization allows a smaller interaction radius between particles, leading to smaller bandwidth when solving the global systems of equations and reduced communication between processors[21].

We have developed a 3D parallel ISPH code within the LAMMPS framework by developing an interface to Trilinos solver libraries[16, 11]. LAMMPS is a molecular dynamics (MD) code originally designed for traditional MD simulations that provides software infrastructure for any particle based simulation with efficient parallel algorithms implemented using message-passing interface and domain decomposition. Trilinos is a package including libraries for solving large matrices with algebraic multigrid, provided that a graph of the underlying matrix can be efficiently constructed. The performance of the current implementation stems from the fact that the graph of the matrix can be easily extracted from the neighbor lists in LAMMPS, allowing the integration of the two libraries with little additional overhead communication. To introduce an implicit time integration capability into LAMMPS, we use Trilinos solver libraries: Belos for a GMRES linear solver[3] and ML for algebraic multigrid (AMG) preconditioner[10].

While the discretizations used in this paper have been outlined carefully in a previous work investigating single-phase flow in relatively simple geometries [21], the current work uses a variant of the continuum surface force (CSF) method to discretize multi-phase flow and account for fluid-fluid-solid interactions in complex geometries[1, 5]. After introducing validation results with benchmarks for the multiphase flow solver, a large pore-scale geometry is used to demonstrate the ability of the method to maintain scalability for extremely complex geometries for both single-phase and multi-phase flow. While recent work done using ISPH schemes to study multiphase flow have been restricted to 2D simulations in relatively simple geometries (e.g. [20, 23, 19]), the capabilities developed here marks the first large scale fully 3D multiphase ISPH implementation.

2. NUMERICAL APPROACH

2.1 Governing equations

We seek a discretization of the Navier-Stokes equations in Lagrangian form on Ω with boundary $\partial\Omega$ with no-slip boundary conditions:

$$\begin{cases} \dfrac{d\mathbf{u}}{dt} = -\dfrac{1}{\rho}\nabla p + \nabla\cdot(\nu\nabla\mathbf{u}) + \mathbf{F}, \\ \nabla\cdot\mathbf{u} = 0, \\ \dfrac{d\mathbf{x}}{dt} = \mathbf{u}, \end{cases} \tag{1}$$

for velocity ($\mathbf{u}$), position ($\mathbf{x}$), pressure (p), density (ρ), dynamic viscosity (ν). $\mathbf{F}$ is the net body force, e.g. the gravity acceleration $\mathbf{g}$ and in the case of multiphase flows the volumetric force $\mathbf{F}^{ST}$ used to model the effect of surface tension. To discretize this, we associate with the set of particles $\mathbf{x}_i \subseteq \Omega$ the point values $\{\mathbf{u}_i, p_i, \rho_i, \nu_i, \phi_i\}$, where ϕ is a flag denoting the particle phase. For the current work, we assume uniform density in each phase. We first solve a Helmholtz equation for each particle i to obtain an intermediate velocity

$$\begin{cases} \dfrac{\mathbf{u}_i^* - \mathbf{u}_i^n}{\Delta t} = -\dfrac{1}{\rho}\nabla p_i^n + \nabla\cdot\left(\dfrac{\nu}{2}\nabla\left(\mathbf{u}_i^* + \mathbf{u}_i^n\right)\right) + \mathbf{F}_i & \mathbf{x}_i \in \Omega, \\ \mathbf{u}_i^* = 0 & \mathbf{x}_i \in \partial\Omega, \end{cases} \tag{2}$$

after which a Poisson equation is solved for a pressure increment $q = p^{n+1} - p^n$ to make the velocity $\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho}\nabla q$ divergence-free,

$$\begin{cases} \nabla\cdot\left(\dfrac{1}{\rho}\nabla q_i\right) = \dfrac{\nabla\cdot\mathbf{u}_i^*}{\Delta t} & \mathbf{x}_i \in \Omega, \\ \partial_n q_i = 0 & \mathbf{x}_i \in \partial\Omega \end{cases} \tag{3}$$

and finally the pressure and positions are updated, taking into account the fact that the predictor pressure p_i^n and the corrected pressure p_i^{n+1} are associated with different particle positions $\mathbf{x}_i^n$ and $\mathbf{x}_i^{n+1}$.

$$\begin{cases}\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \dfrac{\mathbf{u}_i^{n+1} + \mathbf{u}_i^n}{2}, \\ p_i^{n+1}(\mathbf{x}_i^{n+1}) = p_i^n(\mathbf{x}_i^n) + q_i^{n+1}(\mathbf{x}_i^n) + \nabla p_i^{n+1}(\mathbf{x}_i^n) \cdot (\mathbf{x}_i^{n+1} - \mathbf{x}_i^n). \end{cases} \tag{4}$$

While the particle degrees of freedom $\{\mathbf{x}_i, \mathbf{u}_i, p_i, \phi_i\}$ are available, the derivatives of these functions must be reconstructed from neighboring values. Defining the neighbour set $N_{\epsilon,i} = \{\mathbf{x}_j s.t. |\mathbf{x}_j - \mathbf{x}_i| < \epsilon\}$ and a positive radially symmetric function $W(r)$ with compact support ϵ, the differential operators ∇ and ∇^2 are defined as:

$$\nabla_h f_i = \sum_{j \in N_{\epsilon,i}} (f_j - f_i) \mathbf{G}_i \nabla_{x_i} W_{ij} V_j \tag{5}$$

and

$$\nabla_h^2 f_i = 2 \sum_{j \in N_{\epsilon,i}} (\mathbf{L}_i \colon \mathbf{e}_{ij} \otimes \nabla_{x_i} W_{ij}) \left(\frac{f_i - f_j}{|\mathbf{r}_{ij}|} - \mathbf{e}_{ij} \cdot \nabla_h f_i \right) V_j \tag{6}$$

where $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$, $r_{ij} = ||x_{ij}||$, $\mathbf{e}_{ij} = \mathbf{x}_{ij}/r_{ij}$, $W_{ij} = W(r_{ij})$, V_i denotes the particle volume

$$V_i = \left(\sum_{j \in N_{\epsilon,i}} W_{ij} \right)^{-1} \tag{7}$$

and $\mathbf{G}_i$ and $\mathbf{L}_i$ are correction tensors used to guarantee exact reproduction of linear functions for the gradient and Laplacian, respectively. For the sake of brevity, readers are referred to [21] for details of their derivation, relation to classical SPH, and efficient calculation. We note however that their construction requires only the neighbour list $N_{\epsilon,i}$ and the inversion of a 3×3 and a 6×6 matrix and therefore no additional interprocessor communication. Further, this scheme provides second order accuracy for a smaller value of ϵ, therefore simultaneously providing higher accuracy and reduced communication as compared to classical SPH approaches. To handle discontinuities in material parameters for multiphase problems, the elliptic operators in Eqns. 2 and 3 are discretized using the identity $\nabla \cdot (\alpha \nabla f) = \alpha \nabla^2 f + \nabla \alpha \cdot \nabla f$ and the discrete operators ∇_h and ∇_h^2 defined in Eqns. 5 and 6.

As the particles advect under the flow, near stagnation points in the flow particles will clump together or form voids. Following the approach in Xu et al. [18], at the end of every timestep an anisotropy indicator is used to shift particles to enforce a uniform particle number density in the domain. Details of the calculation of the anisotropy shift δr can be found in [18], but after shifting the particles the pressure and velocity are corrected using the consistent gradient

$$\mathbf{x}_i^{n+1,corr} = \mathbf{x}_i^{n+1} + \delta r_i, \tag{8}$$

$$\mathbf{u}_i^{n+1,corr} = \mathbf{u}_i^{n+1} + \nabla \mathbf{u}_i^{n+1} \cdot \delta r_i, \tag{9}$$

$$p_i^{n+1,corr} = p_i^{n+1} + \nabla p_i^{n+1} \cdot \delta r_i. \tag{10}$$

2.2 Boundary conditions

The accuracy of approximation in SPH relies on having full support of the kernel contained in the domain ($\forall i$, $supp(W(|x - x_i|) \subseteq \Omega$). For particles near the wall, the truncation is remedied by introducing several layers of fixed dummy particles in the wall and performing a linear extrapolation using the boundary conditions to impose a consistent extension of the function on the interior [15]. So for example, for a Dirichlet boundary condition, the function f is extended

$$f_D = \frac{d_D}{d_I}(f_W - f_I) + f_W \tag{11}$$

where the subscript $\{D, I, W\}$ denote dummy, interior and wall values, and d_D and d_I denote closest perpendicular distances to the wall for the dummy and interior particle, respectively. Computing d_D and d_I for general geometries would require a spline representation of the boundary, reintroducing a mesh into the problem. In practice, these distances can be approximated using a smoothed approximation[12]. Defining a state specific particle density differentiating fluid and solid particles

$$\chi_i^\alpha = \frac{\sum_{\{j, \phi_j = \alpha\}} W_{ij}}{\sum_j W_{ij}} \tag{12}$$

the distance to the wall can be approximated for either fluid or dummy particles using

$$d_\alpha = 2\epsilon \left(\chi_i^\alpha - 0.5 \right). \tag{13}$$

In practice this allows the specification of an arbitrarily complex geometry by simply placing a lattice of particles over a domain Ω and marking particles as either fluid or dummy particles. For many applications this framework allows trivial discretization of experimentally available datasets (e.g. geometry specified from voxel data such as MRI applications).

2.3 Surface tension force

To calculate the surface tension force $\mathbf{F}^{ST}$, we follow the work by Adami et al. [1] for fluid-fluid interaction and extension by Breinlinger [5] to handle fluid-fluid-solid interaction. We discretize the exact surface tension

$$\mathbf{F}^{ST} = -\sigma \kappa \hat{n} \delta_\Sigma \tag{14}$$

where σ is the surface tension coefficient, κ is the local interfacial curvature, and δ_Σ is a Dirac delta function concentrating the force at the fluid interface. Rather than track the location of the interface, a smoothed representation is calculated by defining the color function

$$c_i^j = \begin{cases} 1 & \phi_i \neq \phi_j, \\ 0 & \phi_i = \phi_j. \end{cases} \tag{15}$$

The diffuse interface normal in the bulk is then calculated at each particle as $\hat{n}_i^b = \nabla c_i / ||\nabla c_i||$. ∇c_i can be computed using the corrected gradient operator defined in Eqn. 5. However, when the densities of the different phases are significantly different, we use the following gradient operator that ensures that particles of different phases experience the same acceleration at the phase interface, following the approach in [1]

$$\nabla c_i = \frac{1}{V_i} \sum_{j \in N_{\epsilon_i}} \left(V_i^2 + V_j^2 \right) \tilde{c}_{ij} \nabla_{x_i} W_{ij}, \tag{16}$$

$$\tilde{c}_{ij} = \frac{\rho_j}{\rho_i + \rho_j} c_i^i + \frac{\rho_i}{\rho_i + \rho_j} c_j^i. \tag{17}$$

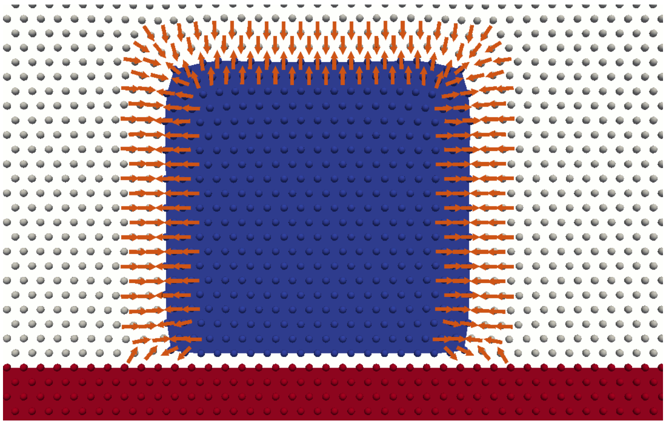

Figure 1: Diffuse normals transition smoothly between ∇c in bulk and normal imposed by contact angle (in this case, $\theta_{eq} = 150°$).

For particles near walls, the Young-Laplace relation fixes an equilibrium contact angle Θ_{eq} which in turn fixes the normal near the triple point of the fluid-fluid-solid interface. Near the wall, the diffuse normal should satisfy

$$\hat{n}_i^w = \hat{n}_i^t \sin\Theta_{eq} + \hat{n}_i^p \cos\Theta_{eq} \tag{18}$$

where $\hat{n}_i^p$ is a diffuse approximation of the wall normal computed from wall number density

$$\hat{n}_i^p = \frac{\sum_{j\in\partial\Omega} \nabla_{x_i} W_{ij}}{||\sum_{j\in\partial\Omega} \nabla_{x_i} W_{ij}||} \tag{19}$$

and $\hat{n}_i^t$ is a diffuse approximation of the normal in the bulk projected tangent to the wall

$$\hat{n}_i^t = \frac{\hat{n}_i^b - \left(\hat{n}_i^b \cdot \hat{n}_i^p\right)\hat{n}_i^p}{||\hat{n}_i^b - \left(\hat{n}_i^b \cdot \hat{n}_i^p\right)\hat{n}_i^p||} \tag{20}$$

Finally, we define an indicator function transitioning smoothly between the bulk and the wall region

$$f_i(d_I) = \begin{cases} d_I/d_{max} & d_I < d_{max} \\ 1 & d_I > d_{max} \end{cases} \tag{21}$$

where d_I is the smoothed approximation to the wall distance defined in the previous section, and d_{max} is a length scale used to define the transition region and is set to 2ϵ in this work. This function is used to smoothly transition between the bulk and near wall estimates of the interfacial normal (See Fig. 1)

$$\hat{n}_i = \frac{f_i\hat{n}_i^b + (1-f_i)\hat{n}_i^w}{||f_i\hat{n}_i^b + (1-f_i)\hat{n}_i^w||}. \tag{22}$$

With a diffuse normal finally defined that is faithful to the Young-Laplace equation, the curvature can be estimated as from particles of identical phase

$$\kappa_i = \nabla\cdot\hat{n}_i \approx d\frac{\sum_{\{j,\phi_i=\phi_j\}}(\hat{n}_i - \hat{n}_j)\cdot\nabla_{x_i}W_{ij}V_j}{\sum_{\{j,\phi_i=\phi_j\}}\mathbf{x}_{ij}\cdot\nabla_{x_i}W_{ij}V_j} \tag{23}$$

and the surface tension force can be estimated as

$$\mathbf{F}_i^{ST} = -\sigma\kappa_i\hat{n}_i\nabla c_i. \tag{24}$$

2.4 Linear algebra

As the discretization of the Poisson equation (Eqn. 3) contains a constant vector in its kernel, and the resulting global system is asymmetric, there exists a vector in the cokernel of the matrix to which the right hand side of the equation must be orthogonal for a solution to exist[4, 6]. Although theoretically this left null vector can be computed with a singular value decomposition and projected out of the right hand side, this is prohibitively expensive. Instead we use Trilinos' capability to project out the kernel during each matrix-vector product occurring in the matrix-vector product of the Krylov subspace iterations, i.e. solving the problem

$$\mathbf{PAP}p = \mathbf{P}b \tag{25}$$

where $\mathbf{A}$ is the discretized Poisson equation with boundary conditions, p is the pressure, b is the discretized right hand side, and $\mathbf{P}$ is a projection matrix to remove constant vectors

$$\mathbf{P} = \mathbf{I} - \frac{\mathbf{c}\mathbf{c}^\intercal}{\mathbf{c}^\intercal\mathbf{c}} \tag{26}$$

where $\mathbf{c}$ is any constant vector. This forces the right hand side to lie in the range of $\mathbf{A}$, and since $\mathbf{A}$ is a consistent approximation to the continuous (symmetric) Laplacian, the error introduced by projecting out the kernel as an approximation to the cokernel is negligible. This approach is expected to be more faithful to the spectrum of the operator (and therefore provide better convergence), as opposed to approaches such as using Lagrange multipliers to remove the singularity or replacing a single row with a one on the diagonal. We note that the generation of the AMG preconditioner is based on matrix $\mathbf{A}$, as the matrix $\mathbf{PAP}$ is never formed.

When constructing the correction matrices (Eqns. 5 and 6) the process is entirely local with no extra communication with neighboring processors. Correction matrices are assembled by looping over nearest neighbors and matrix inversion is done in place using LAPACK's LU-decomposition with partial pivoting. When solving the global Poisson and Helmholtz matrices, the matrix graph is constructed in LAMMPS and then passed to Trilinos to handle the solution. In Trilinos the Epetra package is used for distributed memory linear algebra, the Belos package for a GMRES linear solver, and the ML package for smoothed aggregation AMG preconditioners for block GMRES. Although the Poisson and Helmholtz matrices are M-matrices only for uniform Cartesian particle arrangements, standard preconditioners perform well without special treatment.

3. RESULTS: VALIDATION AND BENCHMARKING

We begin by briefly demonstrating the accuracy of the current implementation for equilibrium multi-phase benchmarks, as the single-phase behavior of the solver has been documented in a previous work [21], and continuum surface force type models have been thoroughly verified elsewhere[1, 13, 5]. Then, we present weak scaling studies for single-phase and multi-phase fluid applications. All simulations are run on NERSC's Cray XE6 (Hopper).

3.1 Multi-phase validation

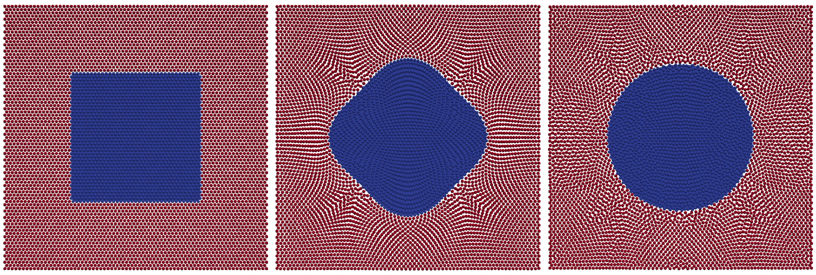

Figure 2: Relaxation of a square drop to circle with uniform pressure at equilbrium.

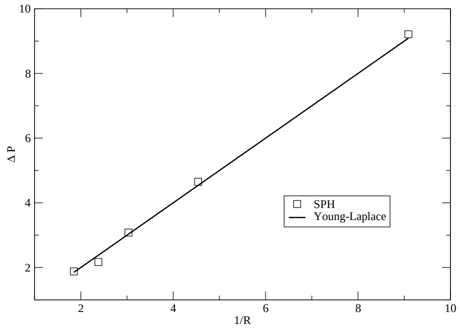

Figure 3: Scaling of droplet pressure with droplet radius in accordance with Young-Laplace relation $p_{out} - p_{in} = \sigma/R$.

We first validate against the relaxation of a droplet from a square shape into a sphere and demonstrate that the resulting pressure field respects the equilibrium hydrostatic pressure predicted by the Young-Laplace equation. Fig. 2 demonstrates a droplet of unit radius, equal density to the surrounding fluid, and unit surface tension coefficient relaxing to a spherical shape. To validate the implementation for fluid-fluid interactions, a $L \times L$ square of fluid A was placed in a periodic box of fluid B with size $2L \times 2L$ for $L \in \{0.1, 0.2, 0.3, 0.4\}$. A 36×36 particle lattice is initialized and evolved forward with $dt = 0.2dx/U_{max}$ until reaching steady state. For parameters of $\sigma = 1, \rho_A = \rho_B = 1, \nu = 0.02$, the scaling of the droplet pressure with equilibrium droplet size is given in Figure 3 and demonstrated to obey the derivation from the equilbrium Young-Laplace equation

$$p_{out} - p_{in} = \frac{\sigma}{R_{drop}}. \tag{27}$$

To validate the implementation for fluid-fluid-solid interactions, a 0.2×0.2 m^2 drop is placed beside a wall in a 0.8×0.6 m^2 fluid channel and discretized with a 22×22 particle resolution across the droplet. Fig. 4 demonstrates the initial configuration and final equilibrium shapes for contact angles of $\theta_{eq} = 60°$ and $\theta_{eq} = 150°$.

3.2 Single-phase application and benchmarking

To further demonstrate the scalability and accuracy of the developed ISPH code for flow in bounded domains with complex geometry, we simulate a single-phase flow in a bead pack consisting of 6864 beads of diameter 0.5 mm randomly placed inside a cylinder with a diameter of 8.8 mm. Such placement resulted in a 12.8 mm long bead pack. In the simulations, SPH particles are initially placed on a Cartesian mesh with the grid size $\Delta = 23\mu m$, resulting in 151 million SPH particles. The particles lying within beads or outside of the cylinder wall are designated as dummy particles. The dummy particle positions are fixed in space and these particles are used to impose the no-slip boundary condition at the boundaries between the fluid and cylinder and glass beads. The rest of the particles are designated as fluid particles with velocities computed according to Eqns. 2 and 3, and positions advected according to Eqn. 2. At time zero, the particle velocities are set to zero. Periodic boundary conditions for pressure and velocity are applied at the top and the bottom boundaries of the cylinder, and a body force is applied in the direction parallel to the main axis of the cylinder to drive the flow. The viscosity and density of the fluid are assumed to be those of water. In the simulations, the SPH equations are integrated until the average velocity reaches steady state. Fig. 3.2 shows the steady-state velocity distribution obtained from a simulation with the body force per unit mass $g = 1.06$ m/s^2.

The simulation results are compared with a high fidelity finite volume solution validated against an experimental MRI dataset [22]. Fig. 6 shows comparison of the cross-sectional velocity profiles obtained from the SPH and FV simulations. This figure shows an overall good agreement between the ISPH and FV methods. The steady-state average velocity and point velocities (measured in the center of the column), obtained from the ISPH and FV, are found to agree within 2%. To further validate the model we simulate flow for three different body forces. Fig. 7 shows that the average velocity v scales linearly with the magnitude of the body force g according to the Darcy Law:

$$v = \frac{k\rho}{\mu} g, \tag{28}$$

where k is the permeability of the bead pack.

Results from a weak scalability study are reported in Fig. 8 and the problem setup is detailed in Tab. 1. In order to improve the load balancing we performed a rebalance operation using the recursive coordinate bisection algorithm implemented in LAMMPS. Fig. 8 shows that the computational time per time step is almost constant for the Helmholtz problem whereas it is slowly increasing for the Poisson problem, which is likely due to the increase in the

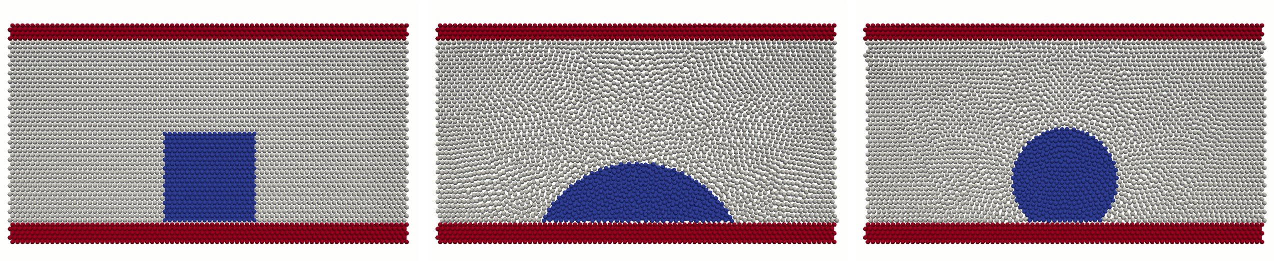

Figure 4: From left to right: Initial configuration of square droplet beside wall. Equilibrium configuration for $\theta_{eq} = 60°$. Equilibrium configuration for $\theta_{eq} = 150°$.

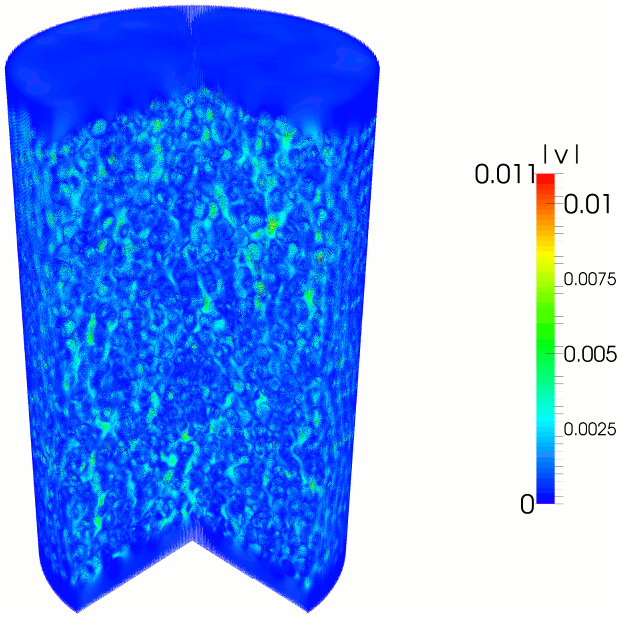

Figure 5: Steady-state solution of flow in the bead pack.

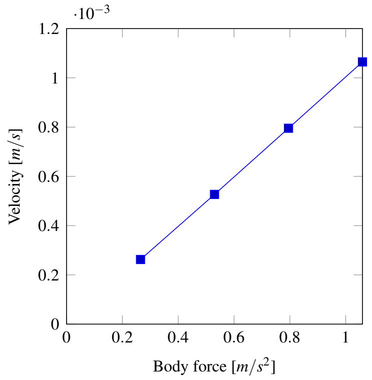

Figure 7: Average velocity as a function of applied body force. The results are in agreement with the linear Darcy law in Eqn. 28 and consistent with results in [22].

Discretization			Per Processor		
N	dx	# Particles	# Processor	# Particles	Load balance
128	6.875e-05	6,083,687	432	14,083	1.0003
192	4.583e-05	19,701,287	1,440	13,682	1.0004
256	3.437e-05	45,803,537	3,432	13,347	1.0007
384	2.291e-05	151,438,991	11,376	13,313	1.0006

Table 1: Setup for weak scaling study for both single-phase and multi-phase problems

number of iteration per time step.

3.3 Multi-phase application and benchmarking

Finally, we demonstrate the scalability of the ISPH code for modeling multiphase flow in bounded domains, which constitutes a highly non-linear problem due to presence of the moving interface separating two fluids. To do so, we simulate bubbles of supercritical carbon dioxide (CO_2) in water. The inital domain is represented by a pack of spherical bubbles with the diameter 0.5 mm randomly placed inside a cylinder with the diameter 4.4 mm and length 6.4 mm (see Fig. 9). As in the single phase flow simulations, SPH particles are initially placed on a BCC lattice with the grid size $\Delta = 23\mu m$. Periodic boundary conditions for pressure and velocity are used at the top and the bottom boundaries of the cylinder whereas we prescribe no-slip Dirichlet conditions at the cylinder lateral wall. At time zero, the particle velocities are set to zero and the body force $g = 4.9$ m/s^2 is applied in the direction parallel to the main axis of the cylinder to drive the flow. In these simulations, the CO_2 density is assumed to be half of the density of water and the CO_2 viscosity is 10 times smaller than the water viscosity. Fig. 10 shows the evolution of the two-phase fluid at different time instants and the agglomeration of the CO_2 bubbles. We perform a weak scalability study on the described problem, using the same setup described in Tab. 1 in and report the results in Fig. 11. The results, in term of scalability, are comparable with those for single-phase fluid, showing that the parallel implementation is rather robust and not significantly affected by the intrinsic nonlinearity of the multi-phase problem.

4. CONCLUSIONS

A highly scalable implementation of a consistent second order incompressible SPH scheme has been implemented using an interface between the LAMMPS and Trilinos libraries to provide a means for studying multiphase flow in complex

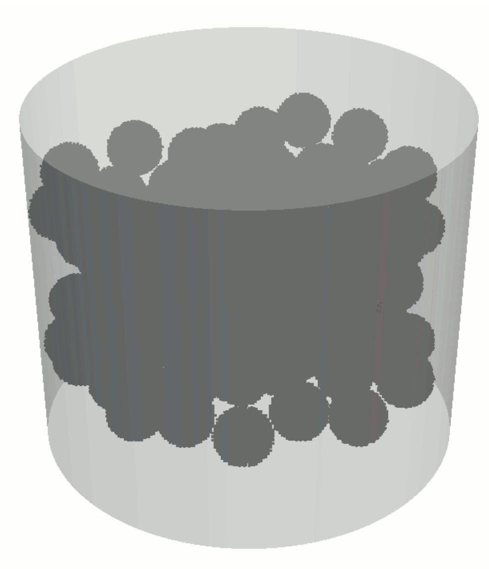

Figure 9: Initial configuration of the spherical CO_2 bubbles for multi-phase problem.

geometry. The current work has demonstrated the ability of this approach to simulate hundreds of millions of particles while maintaining load balancing and favorable weak scaling. While these results have focused primarily on demonstrating the performance of the implementation, another work is currently underway carefully benchmarking the accuracy of this approach for multiphase flows.

5. ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program as part of the Colloboratory on Mathematics for Mesoscopic Modeling of Materials (CM4), under Award Number DE-SC0009247. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under contract DE-AC04-94-AL85000.

6. REFERENCES

[1] S. Adami, X. Hu, and N. Adams. A new surface-tension formulation for multi-phase {SPH} using a reproducing divergence approximation. Journal of Computational Physics, 229(13):5011 – 5021, 2010.

[2] M. Basa, N. J. Quinlan, and M. Lastiwka. Robustness and accuracy of sph formulations for viscous flow. International Journal for Numerical Methods in Fluids, 60(10):1127–1148, 2009.

[3] E. Bavier, M. Hoemmen, S. Rajamanickam, and H. Thornquist. Amesos2 and belos: Direct and iterative solvers for large sparse linear systems. Scientific Programming, 20(3):241–255, 2012.

[4] P. Bochev and R. B. Lehoucq. On the finite element solution of the pure neumann problem. SIAM review, 47(1):50–66, 2005.

[5] T. Breinlinger, P. Polfer, A. Hashibon, and T. Kraft. Surface tension and wetting effects with smoothed particle hydrodynamics. Journal of Computational Physics, 243(0):14 – 27, 2013.

[6] S. J. Cummins and M. Rudman. An sph projection method. Journal of computational physics, 152(2):584–607, 1999.

[7] J. M. Domínguez, A. J. Crespo, and M. Gómez-Gesteira. Optimization strategies for cpu and gpu implementations of a smoothed particle hydrodynamics method. Computer Physics Communications, 184(3):617–627, 2013.

[8] J. M. Domínguez, A. J. Crespo, D. Valdez-Balderas, B. D. Rogers, and M. Gómez-Gesteira. New multi-gpu implementation for smoothed particle hydrodynamics on heterogeneous clusters. Computer Physics Communications, 184(8):1848–1860, 2013.

[9] R. Fatehi and M. Manzari. Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives. Computers & Mathematics with Applications, 61(2):482–498, 2011.

[10] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. Ml 5.0 smoothed aggregation userâĂŹs guide. Technical report, Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

[11] M. Heroux, R. Bartlett, V. Howle, et al. An overview of trilinos, sandia national laboratories. Technical report, Technical Report SAND2003-2927, 2003.

[12] D. W. Holmes, J. R. Williams, and P. Tilke. Smooth particle hydrodynamics simulations of low reynolds number flows through porous media. International Journal for Numerical and Analytical Methods in Geomechanics, 35(4):419–437, 2011.

[13] X. Hu and N. Adams. A multi-phase {SPH} method for macroscopic and mesoscopic flows. Journal of Computational Physics, 213(2):844 – 861, 2006.

[14] J. Monaghan. Smoothed particle hydrodynamics and its diverse applications. Annual Review of Fluid Mechanics, 44:323–346, 2012.

[15] J. P. Morris, P. J. Fox, and Y. Zhu. Modeling low reynolds number incompressible flows using sph. Journal of computational physics, 136(1):214–226, 1997.

[16] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. Journal of computational physics, 117(1):1–19, 1995.

[17] N. J. Quinlan, M. Basa, and M. Lastiwka. Truncation error in mesh-free particle methods. International Journal for Numerical Methods in Engineering, 66(13):2064–2085, 2006.

[18] D. L. Rui Xu, Peter Stansby. Accuracy and stability in incompressible sph (isph) based on the projection method and a new approach. Journal of Computational Physics, 2009.

[19] A. Skillen, S. Lind, P. K. Stansby, and B. D. Rogers. Incompressible smoothed particle hydrodynamics (sph) with reduced temporal noise and generalised fickian smoothing applied to body–water slam and efficient wave–body interaction. Computer Methods in Applied Mechanics and Engineering, 265:163–173,

2013.

[20] N. Tofighi and M. Yildiz. Numerical simulation of single droplet dynamics in three-phase flows using isph. Computers & Mathematics with Applications, 66(4):525–536, 2013.

[21] N. Trask, M. Maxey, K. Kim, M. Perego, M. L. Parks, K. Yang, and J. Xu. A scalable consistent second-order sph solver for unsteady low reynolds number flows. Computer Methods in Applied Mechanics and Engineering, 2015.

[22] X. Yang, T. D. Scheibe, M. C. Richmond, W. A. Perkins, S. J. Vogt, S. L. Codd, J. D. Seymour, and M. I. McKinley. Direct numerical simulation of pore-scale flow in a bead pack: Comparison with magnetic resonance imaging observations. Advances in Water Resources, 54:228–241, 2013.

[23] A. Zainali, N. Tofighi, M. S. Shadloo, and M. Yildiz. Numerical investigation of newtonian and non-newtonian multiphase flows using isph method. Computer Methods in Applied Mechanics and Engineering, 254:99–113, 2013.

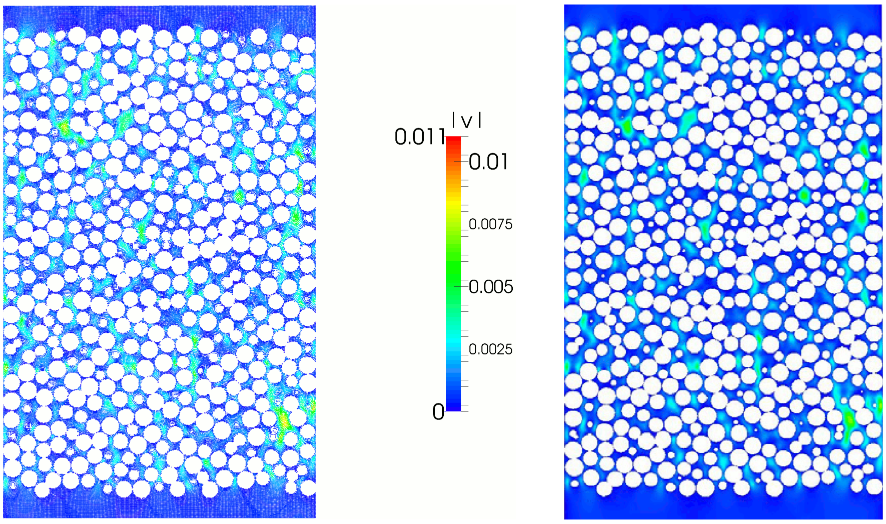

Figure 6: Steady state flow for the bead pack problem using the current approach (left) and a finite volume solution (right) computed with 30 million tetrahedral element smoothed-surface boundary-fitted mesh solver computed in STAR-CCM+ [22].

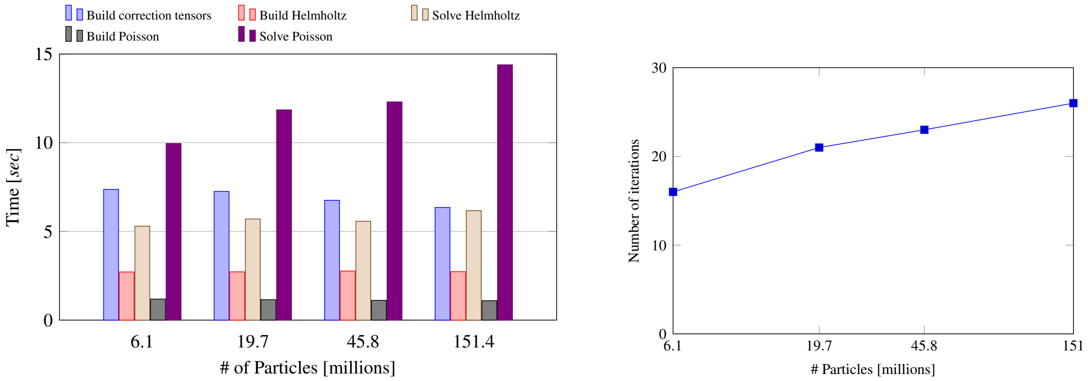

Figure 8: Weak scalability for single-phase problem (the number of particles per process is kept approximately constant, see Tab. 1). Left: computational average times per time step for assembling terms and solving linear systerms vs number of particles. Right: Number of iterations of the GMRES solver vs the number of particles for the Poisson problem.

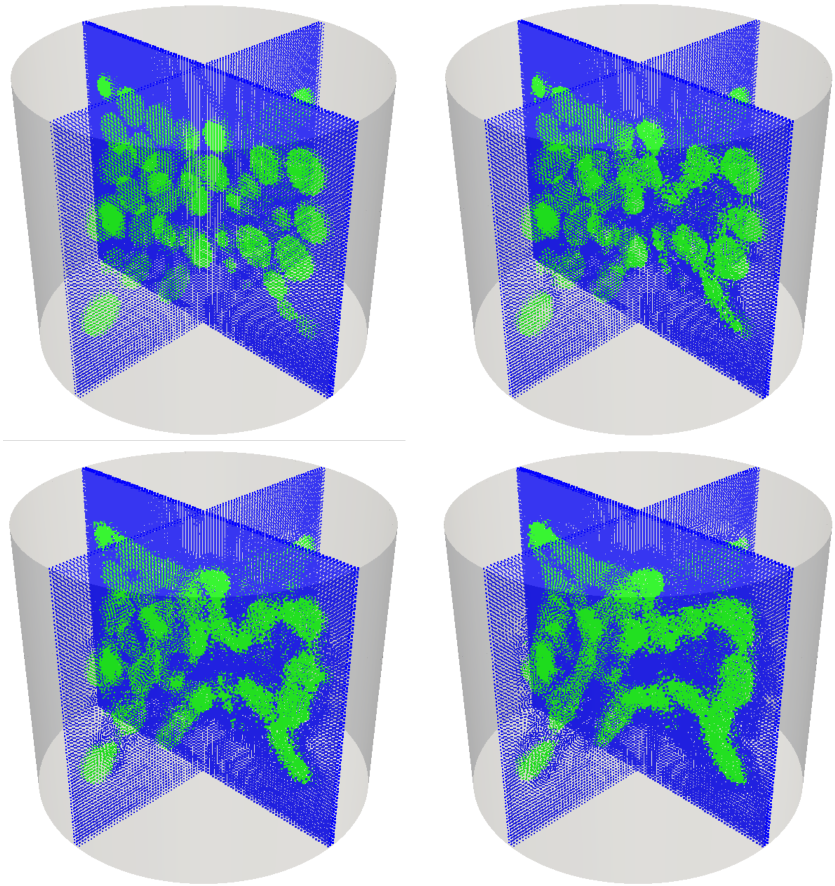

Figure 10: Fluid phases at different time steps, CO_2 in green and water in blue. From left to right, top to bottom, t=1.125ms, 2.25ms, 3.375ms, 4.5ms.

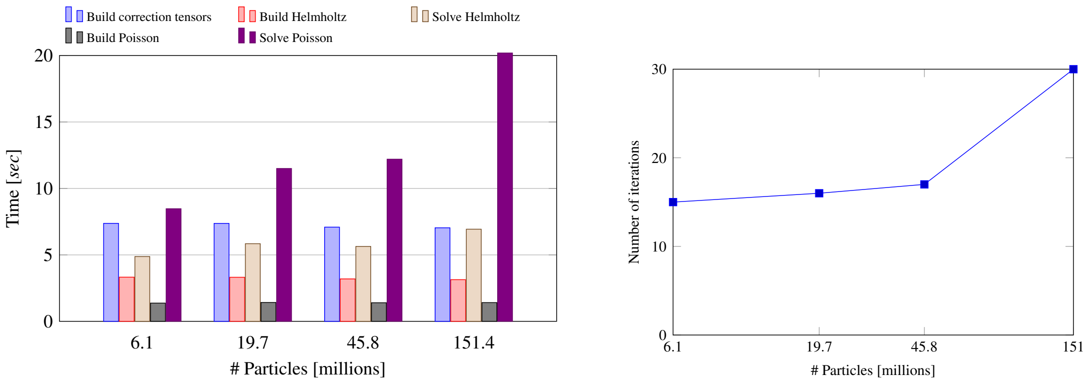

Figure 11: Weak scalability for multi-phase problem. Left: computational average times per time step for assembling terms and solving linear systerms vs number of particles. Right: Number of iterations of the GMRES solver vs the number of particles for the Poisson problem.