Compatible high-order meshless schemes for viscous fluid flows through ℓ_2 optimization

by

Nathaniel Trask

B.S., University of Massachusetts; Amherst, MA, 2008

M.S., University of Massachusetts; Amherst, MA, 2010

M.Sc., Brown University; Providence, RI, 2012

A dissertation submitted in partial fulfillment of the

requirements for the degree of Doctor of Philosophy

in The Division of Applied Mathematics at Brown University

PROVIDENCE, RHODE ISLAND

May 2016

This dissertation by Nathaniel Trask is accepted in its present form
by The Division of Applied Mathematics as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date_____ _____

Martin Maxey, Ph.D., Advisor

Recommended to the Graduate Council

Date_____ _____

George Karniadakis, Ph.D., Reader

Date_____ _____

Chi-Wang Shu, Ph.D., Reader

Approved by the Graduate Council

Date_____ _____

Peter Weber, Dean of the Graduate School

Vitae

Personal Information

American citizen.

Born August 18^{th}, 1985 in Norton, Massachusetts.

Education

Brown University, Providence, Rhode Island USA

M.S., Applied Mathematics, May 2011

University of Massachusetts, Amherst, Massachusetts USA

M.Sc., Mechanical Engineering, May 2010

Dual B.S., Mechanical Engineering/Mathematics, May 2008

Publications

A scalable consistent second-order SPH solver for unsteady low Reynolds number

flows

N. Trask, M. Maxey, K. Kim, M. Perego, M.L. Parks, K. Yang, J. Xu. Computer Methods in Applied Mechanics and Engineering 289 (2015): 155-178.

Smoothed particle hydrodynamics and its applications for multiphase flow and reactive transport in porous media

A.M. Tartakovsky, N. Trask, K. Pan, B. Jones, W. Pan, J.R. Williams. Computational Geosciences (2015):1-28

Intercomparison of 3D pore-scale flow and solute transport simulation methods.

X. Yang, Y. Mehmani, W. Perkins, A. Pasquali, M. Schoernherr, K. Kim, M. Perego, M.L. Parks, N.Trask, M. Balhoff, M. Richmond, M. Geier, M. Krafczyk, L. Luo, A.M. Tartakovsky, T. Scheibe. Advances in Water Resources (Accepted 2015)

Compressible modeling of the internal two-phase flow in a gas-centered swirl coaxial fuel injector

N. Trask, D.P. Schmidt, M. Lightfoot, S. Danczyk. Journal of Propulsion and Power. 28, no. 4 (2012): 685-693.

Diesel spray CFD simulations based on the $\Sigma - Y$ Eulerian atomization model

J.M. Garcia-Oliver, J.M. Pastor, A. Pandal, N. Trask, E. Baldwin, D.P. Schmidt. Atomization and Sprays 23, no. 1 (2013).

Multidimensional modeling of condensing two-phase ejector flow

M. Colarossi, N. Trask, D.P. Schmidt, M.J. Bergander. International Journal of Refrigeration 35, no. 2 (2012): 290-299.

Papers in review

Compact moving least squares: An optimization framework for generating high order compact meshless discretizations

N. Trask, M. Maxey, X. Hu (In review)

A monolithic meshfree discretization for electrophoretic suspensions

N.Trask, M. Maxey (In preparation)

A mixed meshless quasi-divergence free method for the Stokes equations

N.Trask, M. Maxey (In preparation)

A high-order compatible meshless method for elliptic problems

N. Trask, M. Perego, P. Bochev (In preparation)

Acknowledgements

I would like to thank the following people for supporting me through graduate school:

My advisor Martin Maxey, who put a great deal of trust in me and encouraged me to spend a substantial amount of time experimenting with a lot of approaches that didn't pan out. Thank you for always welcoming me into your office and ramble about my ideas, and giving me the time and support I needed to come up with something that finally did work.

My co-advisor George Karniadakis, who challenged me to think outside of my comfort zone to be creative and ambitious. Thank you for exposing me to such a broad range of material; I've probably learned just as much sitting in on your research group meetings and seminars as I have in my own research.

My committee member Chi-Wang Shu, for pushing me to carefully figure out what it was that I was doing in my work and teaching me the value of going through details.

The members of the CM4 grant, who George encouraged me to reach out to and whose collaboration, support and friendship over the past several years has contributed substantially to this thesis: Alex Tartakovsky, Wenxiao Pan, Kai Yang, Xiaozhe Hu, Jinchao Xu, Mike Parks, Kyungjoo Kim, Mauro Perego, and Pavel Bochev. I am very fortunate to have been able to meet and learn from so many

established members of the numerical analysis and scientific computing community.

My buddies Joe Klobusicky, Dan Johnson and Pat Chung, for several years of enlightened conversation, debate, and comraderie.

And most importantly thank you to my brilliant and supportive wife Adrian. Thank you for sticking around with me in Providence and inspiring me every day with your creativity and drive. I'm excited for whatever happens to us next - I was always curious what life was like after school when people get grown-up jobs.

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.

Abstract of "Compatible high-order meshless schemes for viscous fluid flows through ℓ_2 optimization" by Nathaniel Trask, Ph.D., Brown University, May 2016

Meshless methods provide an ideal framework for scalably simulating Lagrangian hydrodynamics in domains undergoing large deformation. For these schemes, interfaces can easily be treated without introducing artificial diffusion, and boundary deformation is handled without costly topological updates to the mesh. Unfortunately, abandoning a mesh also means abandoning a natural framework for performing analysis and as a result meshless methods have historically lacked the stability and conservation properties of their finite element/volume counterparts. In this work, we present a number of new schemes that use a combination of ℓ_2-optimization and graph theory to achieve highly accurate and robust meshless discretizations. These schemes mimic the algebraic structure of compatible finite-element methods, and as a result inherit many of their favorable properties. We use these methods to develop monolithic schemes for suspension flows driven by electrokinetic effects. While these flows are challenging due to the presence of singular pressure forces and a range of relevant length scales of several orders of magnitude, we demonstrate that these new techniques easily resolve analytic benchmarks without the need for sub-grid scale lubrication models.

Contents

List of Figures

Chapter One

Meshless methods and mimetic principles

1.1 Introduction

The purpose of this thesis is to develop a series of modern meshless methods for numerically solving partial differential equations. The word modern can mean several things in this context. We consider it to mean the development of a meshless method with some of the following properties: high-order approximation for smooth functions and complex geometries, simple treatment of any boundary conditions, stability for saddle point problems, and a notion of conservation. Before considering how to obtain these properties in a meshfree context, it is worthwhile to consider what it is about mesh-based methods that have made it so straightforward to obtain these properties and then identify what it is about meshless methods that have made it so difficult to achieve similar results.

Numerical methods for solving partial differential equations (PDE) have classically relied heavily on meshes to generate discrete representations of physical domains. A mesh provides a natural framework to perform analysis: as a subset of $\mathbb{R}^n$ it is natural to perform analysis in terms of either the L^2-norm or energy norm when studying elliptic problems, and the fact that a mesh forms a homogeneous simplicial complex (Figure 1.1) allows a notion of compatibility in terms of how well discrete approximations to differential operators satisfy Stokes theorem (e.g. Gauss divergence theorem, Green's theorem, etc.).

Most modern numerical methods can be interpreted from this viewpoint[5]. The relation between the discretization and Stokes theorems are immediately obvious in the case of finite volume methods where the problem is actually postulated directly from Stokes theorem by associating moments of functions with the cells and faces of a mesh. More subtly however, the analysis of other methods using a similar framework

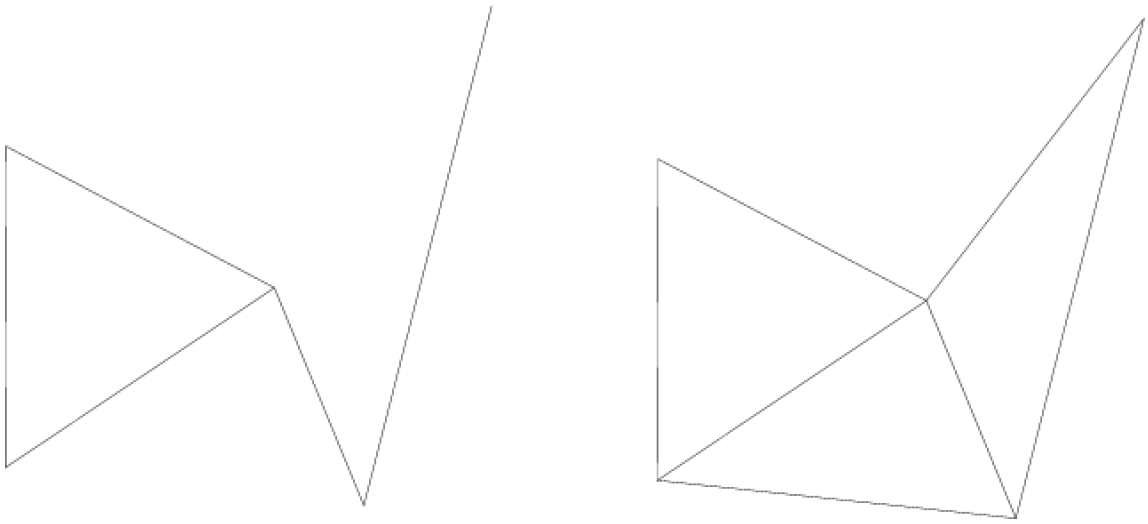


Figure 1.1: A general simplicial complex (left) and homogeneous simplicial complex (right) in $\mathbb{R}^2$. For a homogeneous simplicial complex, the boundary of each simplex is conforming.

has provided a unique new approach to numerical analysis [5]. The development of the finite element exterior calculus has allowed robust discretization of a variety of problems that have been difficult to solve stably with traditional finite elements and has provided the means to design efficient preconditioners [6, 4]. While at first glance finite difference methods seem to lack a mesh, it was not until the introduction of staggered grids and the marker-and-cell scheme that the method was able to operate robustly for high Reynolds number fluid mechanics problems [103]. This interpretation relies on the identification of the nodes of the finite difference gridpoints with cells of a finite-volume mesh, and the staggered grid-points with corresponding faces. The staggered grid arrangement used in these methods has been generalized to finite-element/finite volume type schemes in the primal/dual finite volume approach[31], in which two topologically dual meshes are used to discretize differential operators and their dual. Alternatively, in the mimetic finite difference method the problem is discretized on a single mesh, but operators are defined in such a way that, for a continuous differential operator and its adjoint, their discretization is also discretely adjoint[88].

All of these approaches have been coined mimetic methods due to the fact that at a discrete level the discrete differential operators mimic some property of the continuum operators. The specific property is dependent on the method: it could for example mean that dicretizations of operators are discretely adjoint to the discretizations of their duals (e.g. $div_h = grad_h^{\intercal}$) or that they are faithful to the vector identities underpinning Stokes theorems and exterior calculus at a discrete level(i.e. $div_h \circ curl_h = 0$ or $curl_h \circ grad_h = 0$). Mimetic discretizations typically achieve enhanced stability, discrete maximal principles, or allow efficient pre-conditioning. From an aesthetic point of view, these methods are beneficial because they are faithful to the underlying physics of a problem. For problems derived from a conservation law, it is desirable that fluxes are exactly conserved, allowing the approximation error to be not associated with conservation but instead lumped into the flux approximation, where some physical modelling error is already being made [124]. For example, when solving a diffusion equation for some conserved quantity, conservation mandates that the amount of quantity transferred among control volumes be equal and opposite, but the assumption of the form that the flux should take is often empirical. In light of that, it is better to calculate divergences exactly via the Gauss divergence theorem and associate truncation error only with the calculation of fluxes.

There is a class of methods for numerically solving PDE called meshless methods which we informally characterize as representing a function in terms of a collection of point values at a set of locations throughout the domain. The nature of the data sites (or particles) is completely unstructured; there is no assumed structure in the spatial distribution of the data sites. This lack of structure makes these methods ideal for solving many specialized problems by allowing the solution of the underlying physics in a Lagrangian reference frame relative to each data site and moving the particles with the dynamics of the problem. Such a framework is ideal

for problems with large boundary deformation or difficult interface tracking and allow an exact treatment of advection: typical applications include fluid structure interactions[71, 14], fracture[130], impact dynamics[122], multiphase flow [140, 68, 1], and mesoscale phenomena [43, 92, 15]. To use a moving mesh based discretization in this context requires costly upkeep to maintain the quality of the mesh and prevent cells from becoming degenerate or tangling as they deform under the dynamics of the problem[104, 35, 49, 154, 42, 26]. For large scale parallel implementations, the problem of mesh management is unscalable, because mesh degeneracy is an inherently local feature of a solution that impacts load balancing. Even for serial problems with static meshes, practitioners solving numerical PDE for engineering applications know full well that generating a high-quality mesh for a non-trivial problem often takes more time than actually solving the resulting discrete system. For these reasons moving mesh simulations (or arbitrary Lagrangian-Eulerian methods) are typically restricted to problems with small deformation or mild topology change. Instead, mesh based methods typically handle boundary deformation by representing interfaces diffusely [81, 125, 108, 99, 59, 123], which in turn often limits convergence to second-order in space and sacrifices accuracy for ease of boundary treatment.

For numerical simulation of multiphase flows, it is necessary to track the interface between fluid phases to follow the spatial variation of fluid properties and to calculate surface tension forces from the interfacial curvature. In many flows the dynamics of the interface can be topologically convoluted, with droplets and ligaments pinching off from the interface across a range of length and time scales [56]. In a strategy similar to immersed boundary methods, most approaches to solving multiphase flow track the interface by advecting an indicator function that takes unit value in one phase and zero value outside of the phase. The interface of this indicator function can then be used to diffusely identify the fluid interface. For exam-

ple, volume-of-fluid methods[63] and their moment-of-fluid generalizations[41] pursue this by directly solving for integral moments of indicator functions in each computational cell and using artificial numerical techniques to control numerical smearing of the interface. Level set methods[121, 145] identify the interface as the iso-contour of an auxilliary function that satisfies a Hamilton-Jacobi equation. Thirdly, phase field methods[28, 39] function by artificially imposing a finite thickness to the interface and imposing energy arguments that would typically hold at the microscale. All three of these approaches encounter difficulty at coarse resolution when the interfacial radius of curvature is of the same order as the discretization lengthscale and the diffuse interface representation cannot adequately be resolved. This lack of resolution results in a range of negative effects: artificial surface tension forces for volume-of-fluid methods, a loss of mass conservation for level-set-methods, or a loss of accuracy in phase field methods. Moving mesh methods that track the interface explicitly by using a mesh conforming to the interface and enforcing stress continuity boundary conditions are able to alleviate these issues and provide excellent resolution for capillary flows. Unfortunately this approach is only feasible for simple capillary flows at low Reynolds and Weber number where the growth of instabilities at the interface are simple and do not lead to mesh entanglement. Meshfree methods in this context provide an attractive means to maintain a sharp representation of the interface without the prohibitive cost of maintaining a high-quality mesh.

Unfortunately, abandoning a mesh also means abandoning the framework for functional analysis that has allowed the development of mimetic mesh-based methods. Without a mesh, not only is it difficult to analyze compatibility of operators and conservation, the lack of an obvious inner-product space makes it difficult to prove even simple stability results. Where conservation or stability would typically rely on integration by parts of a weak form of the governing PDE over each computa-

tional cell, without a mesh it is not possible to similarly move between the interior of domains and their boundaries. Throughout the history of meshless methods, many techniques have been developed in an attempt to reintroduce either a variational form to obtain stability or a skew-symmetry in the discretization to obtain conservation. As will be shown in a later chapter, meshless discretization naturally leads to approximation via rational functions for which no efficient quadrature rules exist. As a result many of these techniques become overly expensive and have only been used so far for simple problems. One path to regain a framework for functional analysis is to reintroduce a mesh by generating the Voronoi diagram associated with a set of data sites. This approach has shown success in astrophysics applications solving conservation laws[40], and in the mimetic finite difference method for simulating Lagrangian hydrodynamics in the study of implosions[88]. This strategy will be pursued briefly in a later chapter, but we will see that even for two-dimensional (2D) problems, the cost of maintaining a Voronoi mesh, while scalable, overwhelms the cost of solving the Poisson problem that typically poses the bottleneck of most CFD solvers. As a result, papers pursuing this strategy in the literature are typically restricted to two dimensions. In this work we pursue a "truly meshless method" in which no mesh is reintroduced to the problem.

One of the oldest meshless methods that has proven surprisingly persistent is smooth particle hydrodynamics (SPH). Originally developed to solve the Euler equations in an astrophysics context[54, 97, 110], SPH has subsequently been modified to solve a wide range of problems including the multiphase simulation of breaking water waves, viscous flow, mesoscale flows with thermal fluctuations, and solid mechanics[109]. The discretization process in SPH possesses an anti-symmetry when approximating derivatives that can be used to develop schemes with conservation principles independent of particle arrangements. While most analysis of SPH in the

literature pursues an interpretation of SPH as a dynamical system in which particles evolve via Hamiltonian principles, we will later show that SPH can alternatively be viewed as a discretization of divergence and gradient operators that are adjoint in a diagonally weighted l_2 norm. Traditionally, accuracy arguments for SPH focused on probabilistic arguments with interpretations of the approximation process as a kernel density estimation with Monte Carlo quadrature[53, 93]. More recent work has proven that the truncation error for SPH for general particle distributions lacks even zeroth order consistency, and that for viscous flows the approximation is in fact singular and diverges for highly resolved problems[129, 47]. This introduces complications at boundaries where special care must be taken to avoid contaminating the global convergence rate of the method[100, 64]. Despite this lack of accuracy, the SPH discretization is trivially scalable and possesses a number of mimetic properties that make it ideal for applications in which an accurate transfer of energy can be more important than accurately resolving derivatives[128]. For example, in the smoothed dissipative particle hydrodynamics method, energy stability and positive definite viscous operators are used to design a scheme for fluctuating hydrodynamics at the mesoscale that discretely satisfies the fluctuation-dissipation theorem[43]. Another example is in wave breaking applications, where the accurate transfer of inertia from waves is of primary importance[38]. Another interesting area where SPH has become the de-facto standard is in computer graphics, where conservation of mass and momentum are sufficient to provide visually realistic fluid simulations. In the "eyeball norm", the average movie-goer is unable to discern that the lack of zeroth-order consistency has provided simulations that does not converge to the Navier-Stokes equations.

Optimization, and in particular constrained quadratic programming, is an ideal tool for developing a meshless discretization when no mesh is available to attain

theoretical properties, and will be the main tool used in this work. When developing a compact meshless approximation to differential operators, a derivative is approximated by taking a linear combination of function values at nearby data sites to derive a finite difference-like stencil. The coefficients in this linear combination can be determined by solving an inexpensive optimization problem that minimizes their ℓ_2-norm while applying constraints to endow the approximation with desired theoretical properties. For example: consistency can be enforced be requiring that the approximation be exact for polynomials of a given degree, and conservation can be obtained by requiring a skew-symmetry of fluxes between neighbors. Obtaining polynomial consistency through a weighted least squares optimization can be found under many names in the literature: moving least squares[90, 158], diffuse derivative moving least squares[106, 105], finite point method[45, 116, 118, 117], generalized finite difference method, and others[77]. To our knowledge, with the exception of a few techniques that will be discussed in detail in a later chapter, in all of these cases optimization has only been used to obtain polynomial consistency. The techniques developed in this thesis mark the first attempt to achieve mimetic properties for a meshless method through optimization.

1.2 Overview of the thesis

The goal of this work is to develop a series of meshless methods that possess the properties that are associated with "modern" methods for numerically solving elliptic PDE. We broadly define these properties as high order convergence, accurate treatment of general boundary conditions, efficient preconditioning, conservation properties, and compatibility in the sense of mimetic methods. This goal is pursued by generalizing strategies from classical finite-difference/finite-volume methods to

meshless methods using easily solvable constrained optimization problems.

We begin by motivating the usefulness of meshless methods using SPH as a case study, and stress the trade-off in the method between the symmetry which endows the method with mimetic principles and techniques to modify the method to retain polynomial consistency. We present a massively parallel framework that has been developed to use the algebraic multigrid method to efficiently solve the linear systems arising from general meshless discretizations. We demonstrate the unique capabilities of this method to solve multiphase Lagrangian hydrodynamics for large-scale 3D problems using a modified version of SPH as a meshless discretization. We then gather some fundamental results for developing high-order meshless methods and provide details of how moving least squares can be used to numerically solve elliptic and parabolic PDE. We show that while schemes using this technique maintain high-order polynomial consistency, a lack of compatibility in the differential operators can lead to large qualitative error in under-resolved solutions for even simple problems.

To alleviate these issues, we pursue two strategies for the remainder of the work: in the first we achieve a high-order approximation where the discrete operators mimic the continuous ones by simply increasing accuracy. This approach generalizes compact finite difference methods to general pointsets. In the other approach we generalize staggered finite difference methods, for which the reconstruction is typically chosen carefully to exploit symmetries and maintain nilpotency conditions (e.g. $curl \circ grad = 0$) for polynomials at a discrete level. In order to motivate how to construct this in a meshless framework, we will first pursue analogues between least squares finite volumes on Voronoi meshes, and then demonstrate how similar algebraic structures can be defined on ϵ-neighborhood graphs without constructing any mesh. We then pursue a staggered arrangement of variables on this graph to solve a model div-grad system. When combined with a divergence-free velocity recon-

struction and adaptive resolution, this technique allows a pointwise, divergence-free solution of the Stokes equations. We use these to solve some challenging problems in colloidal suspensions driven by both hydrodynamic and electrokinetic forces.

All of the methods developed in this work are completely scalable and are compatible with the massive scale LAMMPS/Trilinos framework. Where relevant, we have introduced effective preconditioners that will allow a practical $O(N)$ implementation of all techniques. It is our hope that the collection of methods presented in this work will provide a foundation for solving a class of problems that have so far only been accessable for methods incorporating a mesh, allowing a robust, accurate, and scalable framework for simulating Lagrangian hydrodynamics.

Chapter Two

Smoothed Particle Hydrodynamics: The simplest mimetic meshless method

In this chapter, we will present the foundations of smoothed particle hydrodynamics (SPH) for solving incompressible viscous fluid flows. While a number of variations of SPH exist in the literature, for simplicity we focus here on a single formulation that highlights the interplay between conservation properties and accuracy in the method and demonstrate that they are at odds. There is a skew-symmetry to the discretization that will provide either conservation or consistency but never both. A projection scheme for solving incompressible multiphase flow will then be presented which we will use in the following chapter to perform massive-scale simulations of flow single-phase and multi-phase flow in porous media. The details presented in this chapter are discussed more thoroughly in the author's previous works [149, 147]; we present here only enough details to motivate the development of more sophisticated meshless methods in the remainder of the thesis.

2.1 An introduction to SPH

Smoothed particle hydrodynamics is a meshless method originating in the astrophysics community that identifies point values of a function at a distributed set of points as discrete parcels of fluid with fixed mass that can be tracked in a Lagrangian fashion[97, 54, 53, 110]. In its original formulation, the Euler equations are discretized at each particle using a kernel density estimate. This estimate can be interpreted as a combination of a mollification of an underlying function coupled with an application of a quadrature rule, providing the basis for the "smoothed" in SPH. Since its inception the method has found success in diverse fields in many variations of the original formulation [109, 93, 95, 147]. For the purposes of this work, we will focus on an interpretation of the method in terms of particle volume, fixed fluid density, and velocity but note that there is a wealth of alternative ways

to define the method that are effectively equivalent in application, although many have subtle differences in terms of accuracy and stability that can give dramatically different qualitative solutions.

Given a set of points $\mathbf{X}_h := \{x_i\}_{i=1,\dots,N}$ contained in an open region $\Omega \subset \mathbb{R}^n$ with known function values $\{u_i\}_{i=1,\dots,N} := \{u(x_i)\}_{i=1,\dots,N}$ we seek an approximation to the derivatives at each point. The function is approximated with the convolution

$$\langle u \rangle (x) = \int_{supp(W_\epsilon)} u(y) W_\epsilon(||x-y||) dy \tag{2.1}$$

where $W_\epsilon(r)$ is a positive, radially symmetric approximation to the Dirac delta function satisfying $supp(W_\epsilon) = \epsilon$ and $\int W_\epsilon = 1$. It is simple to show by symmetry arguments and Taylor series that, for the ball of radius ϵ centered at x $B_\epsilon(x) \subset \Omega$, this convolution is second order accurate for sufficiently smooth u at any point x. To evaluate derivatives at a point x_i, the derivative can be approximated in a weak sense using integration by parts and the compact support of W_ϵ to obtain

$$\langle \nabla u \rangle_i := \langle \nabla u \rangle (x_i) = \int_{supp(W_\epsilon)} \nabla u(y) W_\epsilon(||x_i-y||) dy \tag{2.2}$$

$$= \int_{supp(W_\epsilon)} u(y) \nabla W_\epsilon(||x_i-y||) dy \tag{2.3}$$

To discretize the quadrature in the smoothing process, we seek a set of quadrature weights V_i associated with each particle which we will interpret as particle volumes

that can be used to approximate Equations 2.1 and 2.2 as

$$\widetilde{u}_i := \sum_{j \in supp(W_\epsilon)} u_j W_{ij} V_j \tag{2.4}$$

$$\widetilde{\nabla u}_i := \sum_{j \in supp(W_\epsilon)} u_j \nabla W_{ij} V_j \tag{2.5}$$

where $W_{ij} := W_\epsilon(||x_i - x_j||)$, $\nabla W_{ij} := \frac{\partial}{\partial x_i} W_\epsilon(||x_i - x_j||)$, and for brevity we will drop the dependence of the sum on the support of the kernel in what follows. To determine values for these quadrature weights, their reciprocal is substituted into Equation 2.4 to obtain

$$\frac{1}{V_i} = \widetilde{V_i^{-1}} = \sum_j W_{ij} \tag{2.6}$$

While the smoothed approximation of function values and their derivatives in Equations 2.1 and 2.2 are able to exactly reproduce linear polynomial functions, for general particle distributions their discrete counterparts (Equation 2.4) are not exact for even constant functions (i.e. $\widetilde{\nabla 1} \neq 0$) and hence we say that SPH lacks zeroth order consistency. We note that despite this lack of consistency, due to the radial symmetry of the kernel function W_ϵ, the gradient of the kernel possesses the antisymmetry

$$\nabla W_{ij} = W'(||x_i - x_j||)\mathbf{e}_{ij} = -\nabla W_{ji} \tag{2.7}$$

where $\mathbf{e}_{ij} = \frac{\mathbf{x_{ij}}}{||\mathbf{x_{ij}}||}$ and $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. This symmetry will be the key factor in maintaining conservation properties in SPH.

2.2 Conservation vs. consistency: SPH as a mimetic method

First we note that to within truncation error, the following product rule applies for the smoothing process.

$$\langle fg \rangle = \langle f \rangle \langle g \rangle \tag{2.8}$$

The following identities may be applied to modify the derivative approximation. To achieve zeroth order consistency we use the quotient rule before discretizing to obtain

$$\nabla \frac{f}{1} = \frac{1\nabla f - f\nabla 1}{1^2} \tag{2.9}$$

$$\widetilde{\nabla f}_i = \sum_j (f_j - f_i)\nabla W_{ij} V_j. \tag{2.10}$$

To achieve what will be shown to be a conservative scheme we apply the product rule

$$\nabla(f \cdot 1) = 1\nabla f + f\nabla 1 \tag{2.11}$$

$$\widetilde{\nabla f}_i = \sum_j (f_j + f_i)\nabla W_{ij} V_j. \tag{2.12}$$

We note that after a multiplication by V_i the conservative gradient operator Equation 2.12 is anti-symmetric in i, j due to the anti-symmetry of the gradient of the kernel ($\nabla W_{ij} = -\nabla W_{ji}$). However, this form is not able to reproduce exactly the zero derivatives a of constant function. There are numerous variations of differential operators in SPH after the initial convolution step, either seeking varying degrees of polynomial reproduction or maintaining anti-symmetry. In the following section we will demonstrate how anti-symmetric discretizations give rise to conservation

properties and in the section following that we demonstrate how the introduction of consistency corrections breaks the symmetry leading to a loss of conservation.

2.3 Conservation and mimetic properites of SPH

We will now show that if the SPH particles are viewed as a N-body dynamical system rather than a discretization of a PDE, the particles obey a Hamiltonian principle. We will then show that if the particles are viewed as a finite dimensional approximation of a PDE, this principle manifests itself as a conservation statement that divergence and gradient operators are adjoint.

In the derivation that follows we adopt a variation of the SPH formulation used in [43, 128]. In keeping with the original aim of SPH for simulating inviscid flow for astrophysics applications, we seek an approximation to the Lagrangian form of the Euler equations. We assume that each particle of fluid represents a discrete control volume of fluid with fixed mass m_i, and that the quadrature weights previously introduced can be identified as the volume associated with each particle. This interpretation of particles as discrete volumes of fluid is what will allow the treatment of the discretization as a dynamical system. The density is therefore given by $\rho_i = m_i/V_i$. We assume that each parcel is in thermodynamic equilibrium and does not exchange mass with other particles; the specific internal energy of each particle therefore can be considered in terms of particle volume and entropy, $e_i(V_i, s_i)$.

In terms of these variables we can write the standard Navier-Stokes equations for

a general compressible Newtonian fluid as

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\mathbf{u} \tag{2.13}$$

$$\rho\frac{d\mathbf{u}}{dt} = -\nabla p + \eta\nabla^2\mathbf{u} + (\psi+\eta/3)\nabla\nabla\cdot\mathbf{u} \tag{2.14}$$

$$T\rho\frac{ds}{dt} = \phi + \kappa\nabla^2 T \tag{2.15}$$

where η and ψ are dynamic and bulk viscosities, κ is the thermal conductivity, and ϕ is production of dissipation by viscous heating. We consider the inviscid reversible case where the viscous and dissipation terms are zero and discretize the resulting Euler equations with SPH as follows

$$\frac{dV_i}{dt} = V_i(\nabla\cdot\mathbf{u})_i \tag{2.16}$$

$$m_i\frac{d\mathbf{u}_i}{dt} = -V_i(\nabla p)_i \tag{2.17}$$

$$T_i m_i\frac{ds_i}{dt} = 0 \tag{2.18}$$

where $(\nabla p)_i$ and $(\nabla\cdot\mathbf{u})_i$ will be suitably chosen SPH discretizations of the pressure gradient and velocity divergence. The careful choice of these operators will provide the mimetic properties in SPH.

The Lagrangian of the system of SPH particles is defined as

$$\mathcal{L} = \sum_j m_j\left(\frac{\mathbf{u}_j^2}{2} - e_j\right) \tag{2.19}$$

and by applying the Euler-Lagrange equations

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \mathbf{u}_i}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{r_i}} = 0 \tag{2.20}$$

we obtain the following equation of motion

$$m_i \frac{d\mathbf{u}_i}{dt} = -\sum_j m_j \frac{\partial e}{\partial V_j}\Big|_s \frac{\partial V_j}{\partial \mathbf{r}_i}. \tag{2.21}$$

From the first law of thermodynamics at constant entropy, $de = -(p/m)dV$, it follows that

$$m_i \frac{d\mathbf{u}_i}{dt} = \sum_j m_j p_j \frac{\partial V_j}{\partial \mathbf{r}_i}. \tag{2.22}$$

From the definition of volume as $V_j = \left(\sum_k W_{jk}\right)^{-1}$, the gradient with respect to particle position is given by

$$\frac{\partial V_j}{\partial \mathbf{r}_i} = -V_j^2 \sum_k \nabla W_{jk}(\delta_{ji} - \delta_{ki}). \tag{2.23}$$

We then eventually obtain the following expression for the evolution of momentum.

$$m_i \frac{d\mathbf{u}_i}{dt} = \sum_j \left(p_i V_i^2 + p_j V_j^2\right) \nabla W_{ij} \tag{2.24}$$

If we compare Equation 2.24 with Equation 2.16, we see that this is consistent with a definition of the pressure gradient as

$$(\nabla p)_i = -V_i^{-1} \sum_j \left(p_i V_i^2 + p_j V_j^2\right) \nabla W_{ij} \tag{2.25}$$

while the continuity equation in Equation 2.16 motivates a definition of the velocity

divergence as

$$(\nabla \cdot \mathbf{u})_i = V'/V = V_i \sum_j \nabla W_{ij} \cdot (\mathbf{u}_j - \mathbf{u}_i) \tag{2.26}$$

By construction, this discretization conserves energy by virtue of its derivation via the Euler-Lagrange equation. Linear momentum is also conserved, by recognizing that via skew-symmetry

$$\sum_i m_i \frac{d\mathbf{u}_i}{dt} = \sum_{i,j} \left(p_i V_i^2 + p_j V_j^2\right) \nabla W_{ij} = 0 \tag{2.27}$$

Meanwhile, the divergence operator is consistent for constant velocities. Though similar, neither of these operators are exactly the same as that given be the original SPH discretization in Equation 2.4, but we see that through a symmetrization conservation has been obtained.

Through this energetic interpretation of SPH a number of other conservation principles can be derived by virtue of the derivation from Lagrangian principles, and this interpretation is how SPH is often presented in the literature. To draw parallels to mesh-based methods, we provide an alternative interpretation of the method as follows.

We consider the following discrete system of equations that can be used variously to model the solution of the Euler equations, the Stokes equations, or the div-grad problem.

$$\begin{bmatrix} \mathbf{M} & \mathbf{G} \\ \mathbf{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ g \end{bmatrix} \tag{2.28}$$

Here, $\mathbf{M}$ is a symmetric-positive definite matrix, and could be for example an identity

matrix in the case of solving the Euler equations or the div-grad system, or it could be a Laplacian operator in the case of solving Stokes equations. $\mathbf{G}$ and $\mathbf{D}$ are the matrices corresponding to the discretized gradient and divergence operator. $\mathbf{u}$ and p are the column vectors of velocity and pressure degrees of freedom, and $\mathbf{f}$ and g are general forcing terms. We say that $\mathbf{G}$ and $\mathbf{D}$ are skew-adjoint with respect to the diagonal norms $\mathbf{A}$ and $\mathbf{B}$ if for any $\mathbf{u}$ or p, there exist diagonal matrices $\mathbf{A}$ and $\mathbf{B}$ such that

$$p^\intercal \mathbf{A}\mathbf{D}\mathbf{u} = -\mathbf{u}^\intercal \mathbf{B}\mathbf{G}p \tag{2.29}$$

If such diagonal norms exist then they can be used to prove energy conservation and stability. For example taking the right hand side of Equation 2.28 to be zero, we can prove that the velocity must also be zero by first multiplying the second equation by $p^\intercal \mathbf{A}$ to obtain

$$p^\intercal \mathbf{A}\mathbf{D}\mathbf{u} = 0 \tag{2.30}$$

This can then be used with the adjoint condition and Equation 2.28 to show that

$$\mathbf{u}^\intercal \mathbf{B}\mathbf{M}\mathbf{u} = 0 \tag{2.31}$$

which, for the case in which $\mathbf{BM}$ is positive definite, implies that $\mathbf{u} = 0$. This interpretation of stability at a discrete level can be seen as analagous to the inf-sup condition used to prove stability of finite element methods. The identity typically used in proofs of energy stability in SPH methods (see for example [43]):

$$\sum_i V_i \left(\nabla p \cdot \mathbf{u} + p\nabla \cdot \mathbf{u}\right) = 0 \tag{2.32}$$

can therefore be viewed as a specific instance of $\mathbf{M} = \mathbf{I}$ and divergence and gradient

operators skew-adjoint under the diagonal norms

$$\mathbf{A} = \mathbf{B} = diag(V_1, \ldots, V_N) \tag{2.33}$$

Interpreting energy stability in this way is useful because standard measures of stability (e.g. in L^2 or the energy norm) are not available without a mesh. Although the dynamical system interpretation of SPH is how conservation is generally presented in the SPH literature, this alternative definition highlights the role of divergence and gradient operators as adjoints and allows a much simpler derivation that highlights the incompatibility between consistency and conservation. For example, we can use Equation 2.10 to develop the zeroth-order consistent divergence operator

$$\nabla \cdot \mathbf{u}_i = \sum_j (\mathbf{u}_j - \mathbf{u}_i) \nabla W_{ij} V_j \tag{2.34}$$

The corresponding adjoint gradient operator under the diagonal volume norm can be derived from Equations 2.34 and 2.32 as follows:

$$\sum_i V_i p_i \nabla \cdot \mathbf{u}_i = \sum_{ij} V_i V_j p_i (\mathbf{u}_j - \mathbf{u}_i) \cdot \nabla W_{ij} \tag{2.35}$$

$$= \sum_{ij} V_i V_j p_i \mathbf{u}_j \cdot \nabla W_{ij} - \sum_{ij} V_i V_j p_i \mathbf{u}_i \cdot \nabla W_{ij} \tag{2.36}$$

$$= \sum_{ji} V_j V_i p_j \mathbf{u}_i \cdot \nabla W_{ji} - \sum_{ij} V_i V_j p_i \mathbf{u}_i \cdot \nabla W_{ij} \tag{2.37}$$

$$= -\sum_{ij} V_j V_i p_j \mathbf{u}_i \cdot \nabla W_{ij} - \sum_{ij} V_i V_j p_i \mathbf{u}_i \cdot \nabla W_{ij} \tag{2.38}$$

$$= \sum_i V_i \mathbf{u}_i \cdot \left(-\sum_j (p_j + p_i) \nabla W_{ij} V_j \right) \tag{2.39}$$

$$\tag{2.40}$$

Consistency with Equation 2.32 then requires that the pressure gradient be dis-

cretized as

$$\nabla p_i := -\sum_j (p_j + p_i)\nabla W_{ij} V_j \tag{2.41}$$

In other words, under the diagonal volume norm the zeroth order consistent divergence operator (Equation 2.34) gives rise to the antisymmetric inconsistent pressure gradient in Equation 2.12. Similarly, starting with a zeroth order consistent gradient operator gives rise to an antisymmetric inconsistent divergence operator. In the SPH literature, the first case is preferable because, in the absence of external forces when solving the Euler equations, the following momentum conservation statement is obtained.

$$\sum_i m_i \frac{d\mathbf{u}_i}{dt} = -\sum_i V_i \nabla p_i \tag{2.42}$$

$$= \sum_{ij} V_i (p_j + p_i)\nabla W_{ij} V_j \tag{2.43}$$

$$= \sum_{ij} V_i V_j p_j \nabla W_{ij} + \sum_{ji} V_j V_i p_j \nabla W_{ji} \tag{2.44}$$

$$= 0 \tag{2.45}$$

While SPH has been useful in a certain niche of applications due to these conservation principles, the duality between conservation and consistency raises an interesting question that we will pursue for the remainder of this work: is it possible to derive a meshless method which is simultaneously consistent and conservative/stable?

2.4 Restoring linear consistency for incompressible viscous flow

While standard SPH approaches often rely on developing compatible pairs of operators to obtain conservation despite poor accuracy, an alternative approach is to sacrifice compatibility in the pursuit of higher order consistency. This is achieved by introducing correction matrices that enforce an exact result when applied to linear functions and therefore obtain second order truncation error. Details of these operators' derivation, implementation, and analysis in the literature can be found in the paper by Trask et al. [149]. We present here a brief overview of the general results.

The zeroth order consistent gradient operator (Equation 2.10) is modified by introducing in $\mathbb{R}^d$ the $d \times d$ correction tensor $\mathbf{G}_i$

$$\nabla f = \sum_j (f_j - f_i)\mathbf{G}_i \nabla W_{ij} V_j \tag{2.46}$$

Provided that the correction tensor

$$\mathbf{G}_i = \left[\sum_j (\mathbf{x}_j - \mathbf{x}_i) \otimes \nabla W_{ij} V_j \right]^{-1} \tag{2.47}$$

is invertible, Equation 2.46 will exactly reproduce derivatives of linear functions. Similar to the previous section, a diagonally skew-adjoint divergence operator can be defined;

$$\nabla \cdot \vec{u}_i = \sum_j \left(\vec{u}_i \mathbf{G}_i + \vec{u}_j \mathbf{G}_j \right) \nabla W_{ij} V_j \tag{2.48}$$

Similarly, this operator preserves conservation but loses linear consistency, and for simplicity we will not consider this operator in this section.

To discretize Laplacians, the following operator is often used in the literature for viscous flows [111].

$$\nabla^2 f_i = \sum_j 2\frac{f_j - f_i}{r_{ij}}\mathbf{e}_{ij} \cdot \nabla W_{ij} V_j \tag{2.49}$$

While this is consistent for constant functions, linear consistency can be recovered by applying a correction [47]

$$\nabla^2 f_i = \sum_j 2\left(\mathbf{L}_i : \mathbf{e}_{ij} \otimes \nabla W_{ij}\right)\left(\frac{f_j - f_i}{r_{ij}} - \mathbf{e}_{ij} \cdot \nabla f_i\right) V_j \tag{2.50}$$

where $\mathbf{L}_i$ is a $d \times d$ correction tensor derived from estimates of leading truncation error. The gradient is then discretized using Equation 2.46.

In order to give a brief sense of the difference the corrected operators make in terms of accuracy, three representative particle configurations are presented in Figure 2.1 corresponding to varying degrees of particle anisotropy. The root-mean-square error corresponding to the uncorrected operators and corrected operators applied to the smooth function $f(x, y) = \langle \sin x \cos y, -\cos x \sin y \rangle$ are presented in Figures 2.2 and 2.3, respectively. The particles are distributed in one of three ways: a Cartesian arrangement; a random particle arrangement obtained by shifting each particle from an initially Cartesian lattice by a random vector of uniformly distributed magnitude $\chi \Delta x$; or a quasi-uniform particle arrangement representative of particle distributions typically found in SPH simulations. Figure 2.2 demonstrates that as more particles are added to the system, the discretization of the gradient only converges to an error of roughly 5%, while the Laplacian actually diverges as more resolution is added. On the other hand, for the corrected operators the error converges with

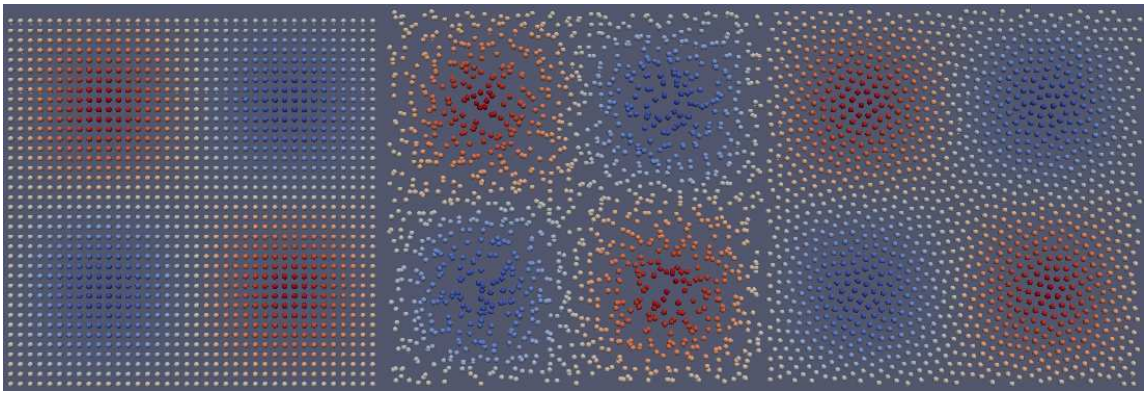


Figure 2.1: Laplacian approximation for representative particle distributions of Cartesian, randomly perturbed with magnitude $\chi = 0.1$, and quasi-uniform particle configurations.

effectively second order to the exact solution (Figure 2.3). This highlights key points regarding the behavior of SPH: without corrections the method is inconsistent and in fact divergent for viscous flows, and that for both cases the convergence behavior is sensitive to particle anisotropy. Of course, the desired application is to advect particles with the flow so that there is no direct way of controlling anisotropy, and as a result in classical conservative versions of SPH the pressure field suffers from "noise"; oscillatory errors on the length scale of the smoothing length that must be artificially controlled with filters or remeshing. On the other hand, if following the previous section we start with a zeroth order consistent divergence operator and find the diagonal volume norm induced gradient operator, the gradient is also singular [47, 149]. For a first-order consistent divergence operator the singularity is more drastic.

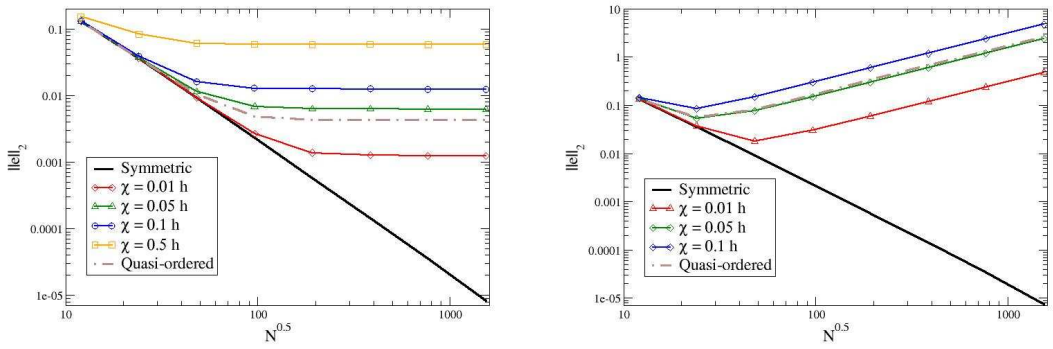


Figure 2.2: L^2 error for uncorrected gradient (left) and Laplacian (right) operators with fixed $supp(W_{ij}) = 4.5\Delta x$.

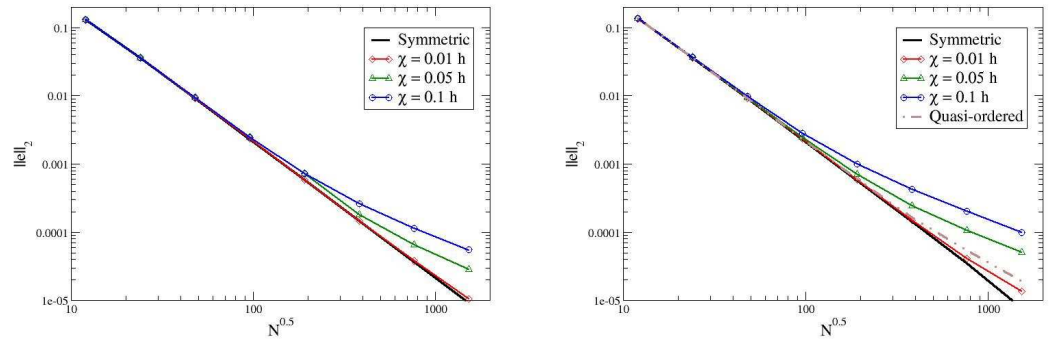


Figure 2.3: L^2 error for corrected gradient (left) and Laplacian (right) operators with fixed $supp(W_{ij}) = 4.5\Delta x$.

2.5 A second order consistent projection scheme for incompressible flow

When using the SPH discretization to simulate incompressible flows there are two strategies that can be found in the literature: weakly compressible SPH (WCSPH) and incompressible SPH (ISPH). In WCSPH, a linearized equation of state is used to penalize variation in density. This is implemented as an explicit scheme that, although prone to temporal errors, is explicit and only requires data structures to maintain particle neighbor lists. Because of this, parallelization is trivial and well-

suited to implementation on GPU architectures[37, 38]. In the ISPH formulation, a divergence-free constraint is imposed by instead solving a global Poisson problem at every timestep. This requires significantly more computational machinery, but the use of stiffly stable implicit schemes removes the expensive viscous timestep restriction. The details and tradeoffs of these two approaches are discussed in the authors previous work [149], and for the purposes here we focus on presenting the modeling advantages an implicit scheme such as ISPH allows to motivate the further development of implicit meshfree schemes beyond SPH.

We extend the original ISPH scheme by Cummins and Rudman [32] to obtain a scheme for multiphase flow that is second order in time and space by applying a combination of the previously introduced spatial corrections and a stiffly stable Crank-Nicolson scheme. After outlining the scheme in the following section, we discuss in the next chapter details of its computational implementation, how the resulting linear systems are solved, the excellent scalability of the method, and examples using the scheme to study multiphase flow in porous media.

We seek a discretization of the Navier-Stokes equations in Lagrangian form in a domain Ω

$$\begin{cases} \dfrac{d\mathbf{u}}{dt} = -\dfrac{1}{\rho}\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) + \mathbf{F}, \\ \nabla \cdot \mathbf{u} = 0, \\ \dfrac{d\mathbf{x}}{dt} = \mathbf{u}, \end{cases} \tag{2.51}$$

for velocity ($\mathbf{u}$), position ($\mathbf{x}$), pressure (p), density (ρ), dynamic viscosity (ν) with a no-slip condition applied at the boundary $\partial\Omega$. $\mathbf{F}$ is the net body force density, e.g. the acceleration due to gravity $\mathbf{g}$ or 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

$$\begin{cases} \dfrac{\mathbf{u}_i^* - \mathbf{u}_i^n}{\Delta t} = -\dfrac{1}{\rho}\nabla p_i^n + \nabla \cdot \left(\dfrac{\nu}{2}\nabla\left(\mathbf{u}_i^* + \mathbf{u}_i^n\right)\right) + \mathbf{F}_i & \mathbf{x}_i \in \Omega, \\ \mathbf{u}_i^* = 0 & \mathbf{x}_i \in \partial\Omega. \end{cases} \tag{2.52}$$

After this 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,

$$\begin{cases} \nabla \cdot \left(\dfrac{1}{\rho}\nabla q_i\right) = \dfrac{\nabla \cdot \mathbf{u}_i^*}{\Delta t} & \mathbf{x}_i \in \Omega, \\ \partial_n q_i = 0 & \mathbf{x}_i \in \partial\Omega \end{cases} \tag{2.53}$$

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 $\mathbf{x}_i^{n+1}$:

$$\begin{cases} \mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \dfrac{\mathbf{u}_i^{n+1} + \mathbf{u}_i^n}{2}, \\ p_i^{n+1}(\mathbf{x}_i^{n+1}) = p_i^n(\mathbf{x}_i^n) + q_i^{n+1}(\mathbf{x}_i^n) + \nabla p_i^{n+1}(\mathbf{x}_i^n) \cdot (\mathbf{x}_i^{n+1} - \mathbf{x}_i^n). \end{cases} \tag{2.54}$$

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 using Equations 2.46 and 2.50. To handle discontinuities in material parameters for multiphase problems, the elliptic operators in Equations 2.52 and 2.53 are discretized using the identity $\nabla \cdot (\alpha \nabla f) = \alpha \nabla^2 f + \nabla\alpha \cdot \nabla f$.

As the particles advect under the flow, particles will clump together or form

voids near stagnation points. Following the approach in Xu et al. [164], at the end of every timestep an anisotropy indicator is used to shift particles to enforce a uniform particle number density in the domain. There are several strategies in the literature for finding an appropriate shift vector δr_i: the original approach [164] used an inter-particle potential, some authors shift particles along the negative of the particle number density, while others solve a global system to enforce a constant number density exactly. For the current work we will use the approach in [164], defining

$$\delta \mathbf{r}_i = C_{shift} \sum_{j \in supp(W)} \frac{\bar{r}_i^2}{r_{ij}^2} \mathbf{e}_{ij} \tag{2.55}$$

where $C_{shift} = C U_{max} \delta t$ is a scale factor to ensure particles are never shifted more than the characteristic lengthscale of the flow, and the centroid of nearby neighbors is given by

$$\bar{r}_i = \frac{1}{\#\left\{j \in supp(W)\right\}} \sum_{j \in supp(W)} ||\mathbf{r}_{ij}|| \tag{2.56}$$

For a Cartesian arrangement of particles this will result in no shift, and typical values of C are $0.0001 - 0.001$.

After shifting the particles the pressure and velocity are corrected using the consistent gradient

$$\mathbf{x}_i^{n+1,corr} = \mathbf{x}_i^{n+1} + \delta \mathbf{r}_i, \tag{2.57}$$

$$\mathbf{u}_i^{n+1,corr} = \mathbf{u}_i^{n+1} + \nabla \mathbf{u}_i^{n+1} \cdot \delta \mathbf{r}_i, \tag{2.58}$$

$$p_i^{n+1,corr} = p_i^{n+1} + \nabla p_i^{n+1} \cdot \delta \mathbf{r}_i. \tag{2.59}$$

2.6 Boundary conditions

The accuracy of approximation in SPH relies on having full support of the kernel contained in the domain when performing the initial mollification in Equation 2.1 (i.e. $\forall i, supp(W(|x - x_i|) \subseteq \Omega)$, otherwise first order errors are introduced [100]. 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 [111]. So for example, for a Dirichlet boundary condition, the function f is extended

$$f_D = \frac{d_D}{d_I}(f_W - f_I) + f_W \tag{2.60}$$

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[64]. Defining a state specific particle density, differentiating fluid and solid particles,

$$\chi_i^\alpha = \frac{\sum_{\{j,\phi_j=\alpha\}} W_{ij}}{\sum_j W_{ij}} \tag{2.61}$$

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

$$d_\alpha = 2\epsilon\,(\chi_i^\alpha - 0.5)\,. \tag{2.62}$$

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 ex-

perimentally available datasets with complex geometry. This will be seen in the following chapter where the scheme is used to simulate flow through porous media.

2.7 Multiphase flow

In a Lagrangian framework it is trivial to track interfaces for multiphase flow; assuming immiscible interaction between phases the evolution equation for species volume fraction in a Lagrangian reference frame is given by

$$\frac{d\phi_i}{dt} = 0 \tag{2.63}$$

and therefore at the beginning of a simulation a particle is assigned a phase and it will maintain that phase for the whole simulation. This avoids all of the difficulty inherent in mesh-based methods where the need for interface tracking leads to artificial smearing, loss of mass conservation, and artificial surface tension (see Section 1.1). Instead the indicator for species volume fraction can simply be advected along with the flow. While this approach gives trivial interface tracking, it is still necessary to introduce a diffuse thickness to the interface to calculate interfacial curvature to evaluate surface tension forces. For the purposes of this work we will present a recent reformulation of the continuum surface force method (CSF) because of its ubiquity in the literature. We note that this approach requires enough resolution to adequately resolve the curvature of the interface; for an alternative more robust approach appropriate for under-resolved flows we refer to the author's review article on the pairwise force method [147].

To calculate the surface tension force $\mathbf{F}^{ST}$, we follow the work by Adami et al.

[1] for fluid-fluid interaction and the extension by Breinlinger [21] to handle fluid-fluid-solid interaction. We discretize the exact surface tension

$$\mathbf{F}^{ST} = -\sigma\kappa\hat{n}\delta_{\Sigma} \tag{2.64}$$

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

$$c_i^j = \begin{cases} 1 & \phi_i \neq \phi_j, \\ 0 & \phi_i = \phi_j. \end{cases} \tag{2.65}$$

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. 2.46. 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]

$$\nabla c_i = \frac{1}{V_i} \sum_{j \in N_{\epsilon_i}} \left(V_i^2 + V_j^2\right) \tilde{c}_{ij} \nabla_{x_i} W_{ij}, \tag{2.66}$$

$$\tilde{c}_{ij} = \frac{\rho_j}{\rho_i + \rho_j} c_i^i + \frac{\rho_i}{\rho_i + \rho_j} c_j^i. \tag{2.67}$$

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

$$\hat{n}_i^w = \hat{n}_i^t \sin\Theta_{eq} + \hat{n}_i^p \cos\Theta_{eq} \tag{2.68}$$

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

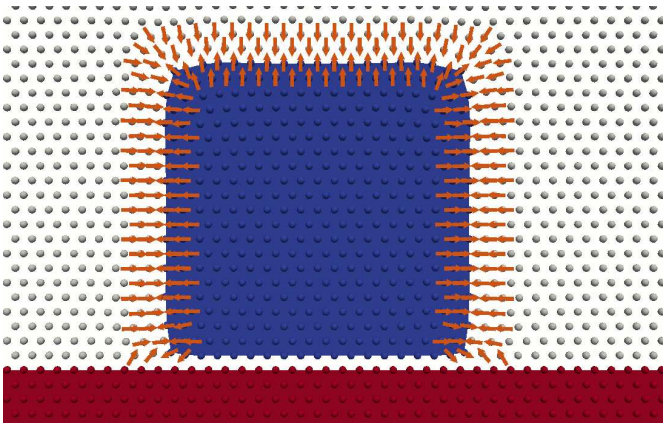


Figure 2.4: Diffuse normals transition smoothly between ∇c in bulk and normal imposed by contact angle (in this case, $\theta_{eq} = 150°$).

density

$$\hat{n}_i^p = \frac{\sum_{j \in \partial\Omega} \nabla_{x_i} W_{ij}}{|| \sum_{j \in \partial\Omega} \nabla_{x_i} W_{ij} ||} \tag{2.69}$$

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

$$\hat{n}_i^t = \frac{\hat{n}_i^b - \left(\hat{n}_i^b \cdot \hat{n}_i^p\right) \hat{n}_i^p}{||\hat{n}_i^b - \left(\hat{n}_i^b \cdot \hat{n}_i^p\right) \hat{n}_i^p||} \tag{2.70}$$

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

$$f_i(d_I) = \begin{cases} d_I/d_{max} & d_I < d_{max} \\ 1 & d_I > d_{max} \end{cases} \tag{2.71}$$

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 transition smoothly between the bulk and near wall estimates of the interfacial normal. In Figure 2.4 we illustrate the diffuse normal layer for a square droplet in the vicinity of a wall. In the layer of particles near the wall, the imposed contact angle is apparent, while in the bulk away from the wall the imposed normal is governed by the Young-Laplace relation and

$$\hat{n}_i = \frac{f_i \hat{n}_i^b + (1 - f_i)\hat{n}_i^w}{||f_i \hat{n}_i^b + (1 - f_i)\hat{n}_i^w||}. \tag{2.72}$$

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

$$\kappa_i = \nabla \cdot \hat{n}_i \approx d \frac{\sum_{\{j,\phi_i=\phi_j\}} (\hat{n}_i - \hat{n}_j) \cdot \nabla_{x_i} W_{ij} V_j}{\sum_{\{j,\phi_i=\phi_j\}} \mathbf{x}_{ij} \cdot \nabla_{x_i} W_{ij} V_j}. \tag{2.73}$$

The surface tension force is then estimated as

$$\mathbf{F}_i^{ST} = -\sigma \kappa_i \hat{n}_i \nabla c_i. \tag{2.74}$$

Chapter Three

LAMMPS/Trilinos: A massively parallel framework for solving meshfree methods with algebraic multigrid

In this chapter we collect results from a series of papers detailing a parallel implementation of the scheme presented in the previous chapter. Two codes have been developed: firstly a serial 2D code using auxilliary space preconditioning to solve the resulting Helmholtz and Poisson problems, and secondly a parallel 3D framework combining existing massive-scale codes for molecular simulation and algebraic multigrid to achieve an implementation that scales to about thirty two thousand processors. We present validation cases for single and multiphase flows with scaling results for their parallel implementation, and some results from a comparative study in which this code was compared to a number of other parallel implementations of differing numerical methods to study flow through porous media.

3.1 Fast solvers: Serial prototyping with FASP

The SPH discretization of both the first and second-order schemes requires the solution of both a Helmholtz equation for the predictor step and a Poisson problem to determine the pressure or pressure increment. Both of the resulting elliptic matrices are of M-form [134] for Cartesian particle distributions, for which the geometric multigrid method and its variant algebraic multigrid (AMG) have been shown to be the most efficient solvers. Generating a hierarchy of meshes is not feasible in SPH, and due to the method's Lagrangian nature meshes may become severely distorted as the flow evolves. Thus, AMG methods are more suitable than geometric MG methods. For details of the MG method, we refer to [153, 155] and references therein.

In our verification of the two-dimensional serial code, we apply the AMG preconditioner implemented in the FASP linear algebra package [161, 163]. We have

investigated two different AMG methods for solving the SPH discretization including: (1) the classical AMG (C-AMG) method [131] with Gauss-Seidel smoother and V-cycle and (2) the unsmoothed aggregation AMG (UA-AMG) method [78] with Gauss-Seidel smoother and nonlinear AMLI-cycle [69]. Both methods serve as a preconditioner for a Krylov subspace iterative method (e.g. GMRES) to accelerate the overall performance of the linear solver. In Table 3.1 we present results of a scaling study in solving the Helmholtz problem with a large viscosity ($Re = 0.01$,$Co = 0.5$) where the Helmholtz operator is dominated by the Laplacian term. We can observe that AMG methods are efficient for solving the Helmholtz equation, and the scaling of the computational cost is nearly optimal, in the sense that the CPU time is almost linear in the degrees of freedom.

Table 3.1: Performance of AMG methods for solving Helmholtz equation in large viscosity limit with Dirichlet data

	UA-AMG		C-AMG	
DoF	#Iter.	CPU (sec.)	#Iter.	CPU (sec.)
2^{12}	7	0.03	5	0.03
2^{14}	8	0.14	6	0.16
2^{16}	8	0.64	7	0.75
2^{18}	9	2.90	8	3.42
2^{20}	9	14.15	9	16.37

3.2 Fast solvers: Parallel implementation using LAMMPS/Trilinos

We have developed a parallel 3D implicit SPH code using existing libraries to leverage the proposed second-order projection scheme for large scale simulations. LAMMPS is originally developed for molecular dynamics simulations and provides computational

infrastructures for massive scale particle simulations [126]. However, LAMMPS lacks the capability to perform either implicit time integration or to implement the parallel linear algebra solvers required for solving the Poisson and Helmholtz problems in our scheme. For an efficient, parallel solution of these linear problems, we have developed an interface integrating LAMMPS with packages from the Trilinos library [62]. Specifically, we use the Epetra package [61] for distributed memory linear algebra, the Belos package [9] for a GMRES linear solver, and the ML package [51] for smoothed aggregation AMG preconditioners for block GMRES. The interfaced code lets LAMMPS handle the particle simulation data and Trilinos handle the distributed memory linear solves, allowing each code to do what it was designed to do well.

As both LAMMPS and the Trilinos libraries have demonstrated massively parallel scalability, our interfaced code inherits that performance. To the best of our knowledge, at the time this work was done the computational results that follow represented the largest implicit SPH simulations ever performed. For both the Helmholtz and Poisson systems, we use ML's smoothed aggregation algebraic multigrid with 4 sweeps (pre and post) of Gauss-Seidel as a smoother on each level, except the coarsest one for which a serial direct solve is performed. To improve parallel performance we repartition levels onto a subset of processors using Zoltan when the number of unknowns per core falls below 500 [34]. In each case ML is used to precondition GMRES (Poisson) or block GMRES (Helmholtz) where the convergence tolerance of the Krylov method is set to 10^{-10}.

3.3 Results: Validation and Benchmarking

We demonstrate in this section validation of the code in simulating simple single-phase and multi-phase benchmarks for which either analytic solutions are available or a reference benchmark can easily be generated. For the single-phase validation, further details and validation can be found in our previous work[149]. We then demonstrate the performance of the code by investigating its performance when studying massive scale problems in porous media flow and multiphase flow. All results for this section were run on NERSC's Cray XE6 (Hopper).

3.3.1 Single-phase validation

We first demonstrate convergence of the method by considering the 2D Taylor-Green vortex, for which the following analytic solution exists.

$$\begin{aligned} U_x(x,y,t) &= U_0 e^{-2\nu t}\sin(x)\cos(y) \\ U_y(x,y,t) &= -U_0 e^{-2\nu t}\cos(x)\sin(y) \\ P(x,y,t) &= \frac{U_0^2}{4}e^{-4\nu t}\left(\cos(2x)+\cos(2y)\right) \end{aligned} \tag{3.1}$$

We extrude the problem domain to be fully 2π-periodic in the z dimension, so that the computation is fully 3D. First, results corresponding to $h = 1.5\Delta x$ are presented to investigate scaling for the regime where second-order convergence was previously observed in the Laplacian operators (See Figure 2.3). As expected, second-order convergence was observed, as reflected in Figure 3.1. Parallel scalability for up to 134 million particles computed using 32,768 cores is reported in Figure 3.2. The number of iterations per linear solve is shown on the left of Figure 3.2. It shows

that the iteration counts grow modestly with increasing problem size for both the Helmholtz and Poisson solves. The right-hand plot in Figure 3.2 shows the weak scaling efficiency per linear solver iteration, namely the cost per iteration at n cores divided by the cost per iteration at 64 cores. The ideal method would have a weak scaling efficiency of one, but in practice we expect that to decrease slightly as the problem size increases. While AMG theoretically converges in a constant number of iterations for finite element discretizations and high quality meshes, in practice the number of iterations required for convergence increases mildly for massive scale problems. The efficiencies shown for the Helmholtz solver are very good, as they are near and sometimes even above one. Efficiencies greater than one are usually caused by machine "jitter" or network effects and should not be taken to imply that the underlying algorithms are more efficient on larger core counts. The Poisson solver efficiencies are more typical, declining to about 0.7 at 32,768 cores.

As previously noted in Section 2.4, the consistency operators allow for a much smaller support radius than that typically used in the SPH literature. By setting $h = 0.8\Delta x$, the increased sparsity of the operators leads to better memory bandwidth utilization and we see a four-fold decrease in the total wall-time of the method. As discussed previously, this gives more accurate results since the truncation error scales with $O(h^2)$ and transitions to $O(h)$ for large numbers of particles. This leads to the somewhat counter-intuitive conclusion that the extra linear algebra necessary in inverting the small local matrices for the consistency corrections actually leads to a much faster method, since the global matrices become cheaper to solve. The scaling results corresponding to this reduced support radius are presented in Figure 3.3. While the number of iterations is similar to what is shown in the previous result, the parallel efficiency per iteration illustrated in the right of Figure 3.3 shows larger fluctuations. This is likely because the reduced time complexity, due to the sparser

operators, amplifies the network or hardware noise.

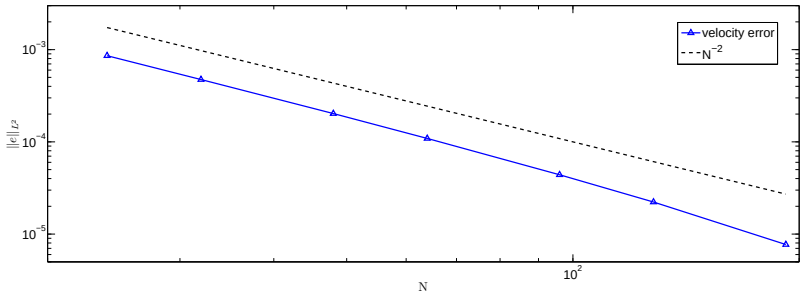


Figure 3.1: Taylor-Green Vortex 3D test: velocity RMS error as a function of N^3 total particles .

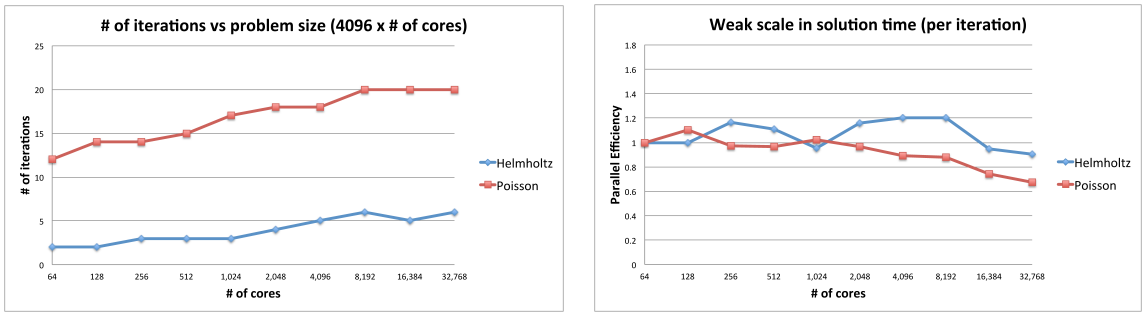


Figure 3.2: Weak scalability results for Helmholtz (blue) and Poisson (red) solves with a smoothing length $h = 1.5\Delta x$, having 4096 particle per core. Left: number of linear iterations per solve as a function of number of cores. Right: weak scaling efficiency per iteration as a function of number of cores.

Next, to highlight the capability of the code to simulate complex 3D geometries, we consider flow past a 3D periodic array of spheres. The spheres are arranged in a body-centered cubic structure within a periodic box, as shown in Figure 3.4. We select the following set of parameters: the cube edge length $L = 0.1$; the spheres' radius $a = 0.02$; the viscosity $\nu = 10^{-4}$; and the forcing term $F = 5 \cdot 10^{-5}$, corresponding to a flow Reynolds number of 0.01. To enforce the Takeda/Morris no-slip boundary condition requires the evaluation of the perpendicular distances to walls for fluid particles interacting with ghost particles. In increasingly complex boundary configurations this becomes cumbersome and in general requires some sort of piecewise spline representation of the boundaries. To circumvent this we have implemented the smoothed perpendicular distance approximation introduced by Holmes et al. [64]. This technique approximates the perpendicular distances using only in-

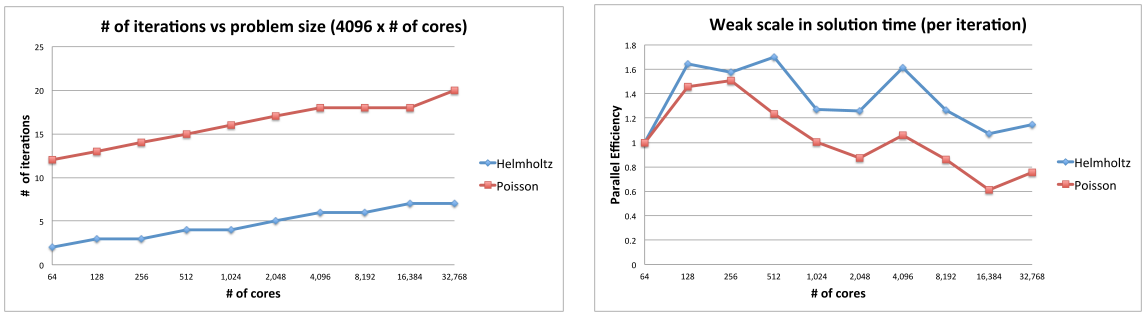


Figure 3.3: Weak scalability results for Helmholtz (blue) and Poisson (red) solves with a smoothing length $h = 0.8\Delta x$, having 4096 particle per core. Left: number of linear iterations per solve as a function of number of cores. Right: weak scaling efficiency per iteration as a function of number of cores.

formation corresponding to the distribution and ID (either interior or ghost particle) of nearest neighbors and preserves a truly meshfree boundary representation. Again, we refer to our previous work for details[149].

To validate these results we present a comparison with a finite volume solver. Grid convergence for the benchmark was conducted by performing a refinement study for 9,408, 71,632, and 768,000 hexahedra cell meshes. In Figure 3.5 and Figure 3.6 we compare using results from an SPH simulation with $N = 64^3$ particles with smoothing length $h = 0.8\Delta x$ at time $t = 20s$. To highlight the agreement between the results, the Shepard interpolant

$$u_{shep}(\vec{x}) = \frac{\sum_j u_j(1 - \frac{||x-x_j||}{h_{shep}})_+}{\sum_k (1 - \frac{||x-x_k||}{h_{shep}})_+}$$

with support $h_{shep} = 0.0025$ is used to compare SPH streamwise velocity and pressure to the finite volume results along the line starting at $y = 0.05, z = 0.05$ in the streamwise direction. Figure 3.7 demonstrates excellent agreement between the two results, with a small post-processing discrepancy near the particle boundary that is attributable to the truncation of the support of the Shepard interpolant. Having validated the solver for a single pore-scale flow, we consider in a later section a

massive scale simulation of flow through a porous material in which each individual pore is fully resolved.

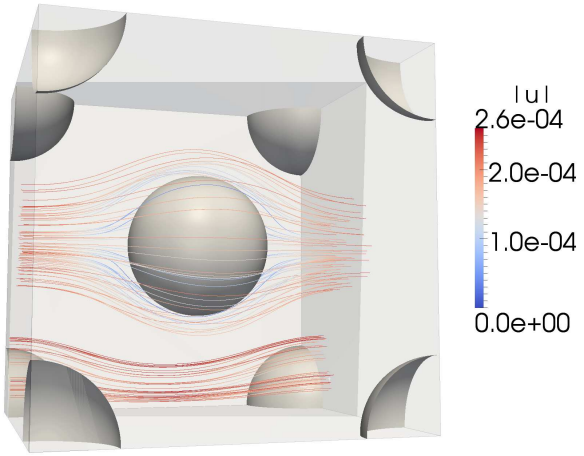


Figure 3.4: Streamlines starting from selected points colored with magnitude of velocity generated from SPH results.

3.3.2 Multiphase validation

We begin by briefly demonstrating the accuracy of the current implementation for equilibrium multi-phase benchmarks. We use the continuum surface force model for surface tension effects described in the previous chapter. The consistency of this approach has been thoroughly verified elsewhere[1, 67, 21].

We first validate against the relaxation of a cylindrical droplet from a square shape into a circle and demonstrate that the resulting pressure field respects the equilibrium hydrostatic pressure predicted by the Young-Laplace equation. Figure 3.8 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

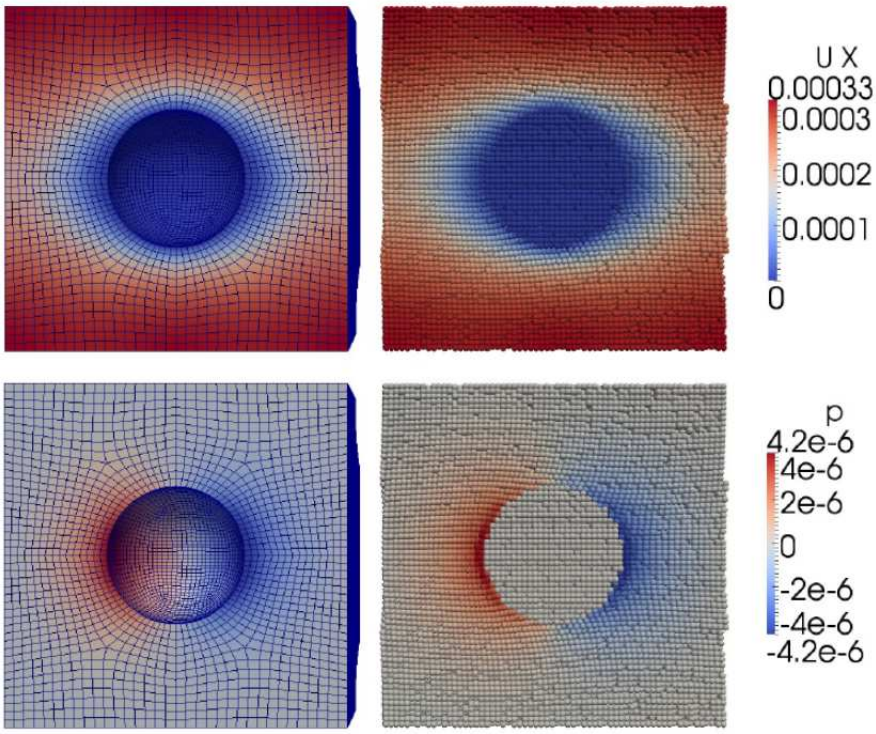


Figure 3.5: Streamwise velocity (m/s) and pressure (Pa) in plane z = 0.05 with comparison to finite volume results. Mesh shown for 71632 cell case to demonstrate mesh structure.

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 3.9 and demonstrated to obey the equilbrium Young-Laplace equation

$$p_{out} - p_{in} = \frac{\sigma}{R_{drop}}. \tag{3.2}$$

To validate the implementation for fluid-fluid-solid interactions, a $0.2 \times 0.2\ m^2$ drop is placed beside a wall in a $0.8 \times 0.6\ m^2$ fluid channel and discretized with a 22×22 particle resolution across the droplet. Figure 3.10 demonstrates the initial configuration and final equilibrium shapes for contact angles of $\theta_{eq} = 60°$ and $\theta_{eq} = 150°$.

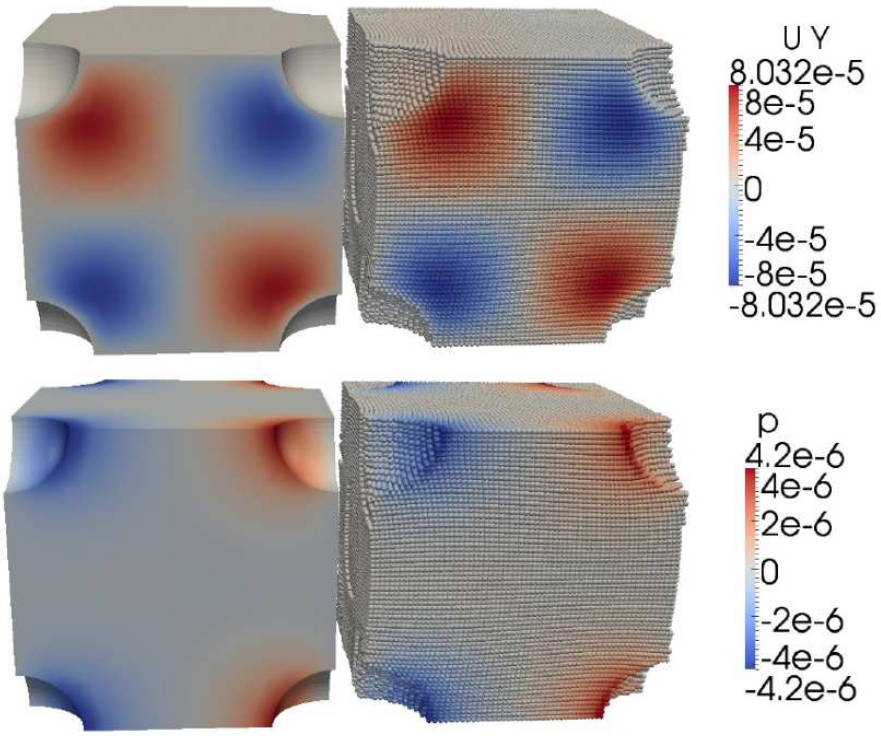


Figure 3.6: Normal velocity (m/s) and pressure (Pa) in plane z = 0.0 with comparison to finite volume results.

3.4 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 of diameter 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 Equations 2.52 and 2.53, and positions advected according

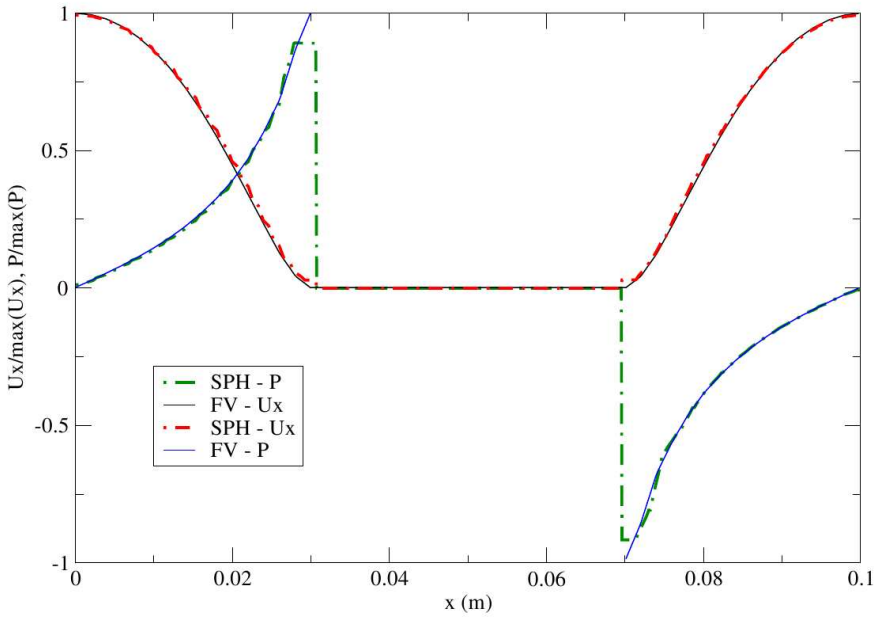


Figure 3.7: Streamwise probe along center of domain comparing interpolated SPH and finite volume velocity and pressures normalized by maximum value.

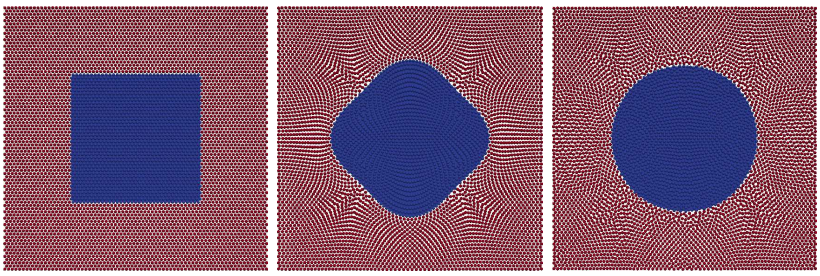


Figure 3.8: Relaxation of a square, cylindrical drop to a circle with uniform pressure at equilbrium.

to Equation 2.52. The particle velocities are initially set to zero. Periodic boundary conditions for pressure and velocity are applied at the top and the bottom 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. Figure 3.11 shows the steady-state velocity distribution obtained from a simulation for which the body force per unit mass $g = 1.06$ m/s^2 is imposed.

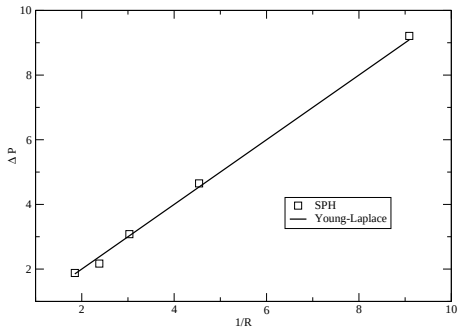


Figure 3.9: Scaling of droplet pressure with droplet radius in accordance with Young-Laplace relation $p_{out} - p_{in} = \sigma/R$.

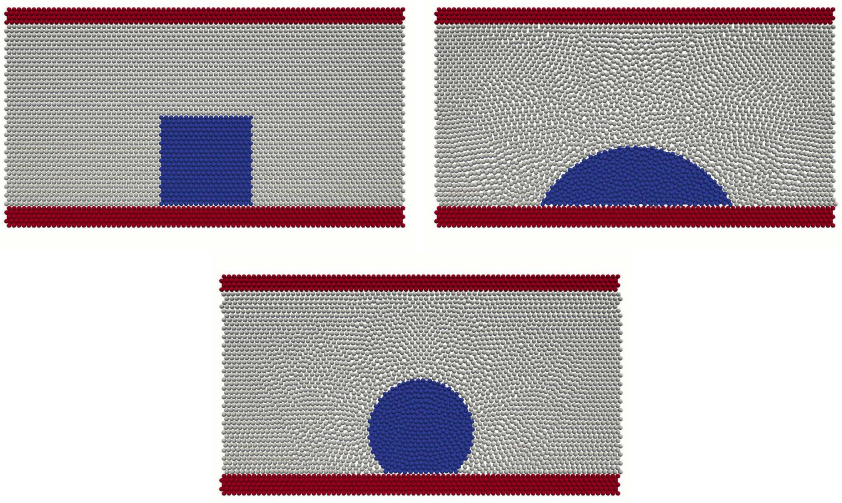


Figure 3.10: From left to right: Initial configuration of square droplet beside wall. Equilibrium configuration for $\theta_{eq} = 60°$. Equilibrium configuration for $\theta_{eq} = 150°$.

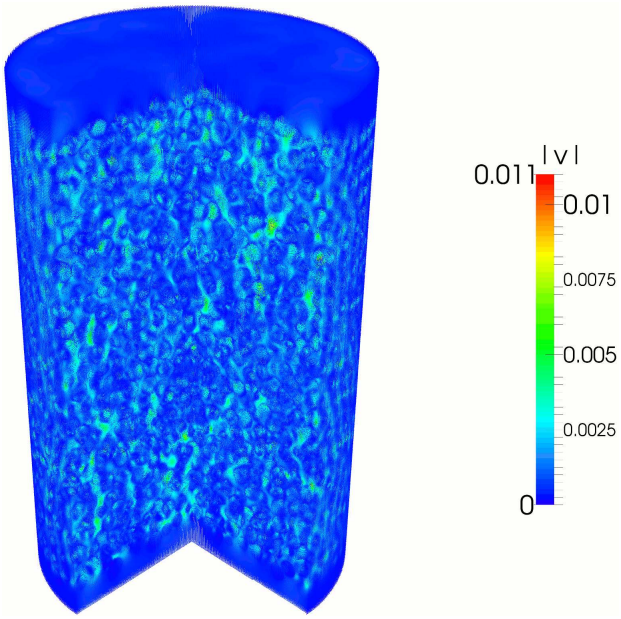


Figure 3.11: Steady-state solution of flow in the bead pack.

The simulation results are compared with a high fidelity finite-volume (FV) solution validated against an experimental MRI dataset [165]. Figure 3.12 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 at the center of the column are obtained from both the ISPH and FV results, and were found to agree to within 2%. To further validate the model we simulate flow for three different body forces. Figure 3.13 shows that the average velocity v scales linearly with the magnitude of the body force g according to the Darcy Law:

$$v = \frac{k\rho}{\mu} g, \tag{3.3}$$

where k is the permeability of the bead pack.

Results from a weak scalability study are reported in Figure 3.14 and the problem setup is detailed in Table 3.2. In order to improve the load balancing we performed a

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

rebalance operation using the recursive coordinate bisection algorithm implemented in LAMMPS. Figure 3.14 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 number of iterations per time step.

3.5 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 the presence of a moving interface separating the 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 of diameter 0.5 mm randomly placed inside a cylinder of diameter 4.4 mm and length 6.4 mm (see Figure 3.15). As in the single-phase flow simulations, SPH particles are initially placed on a BCC lattice of grid size $\Delta = 23\mu m$. Periodic boundary conditions for pressure and velocity are applied at the top and the bottom boundaries of the cylinder whereas no-slip Dirichlet conditions are enforced at the cylinder wall. The particle velocities are initially set to zero and a body force of $g = 4.9$ 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 that of water and the CO_2 viscosity is ten times smaller than that of water. Figure 3.16 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 Table 3.2 in and report the results in Figure 3.17. The results, in terms of scalability, are comparable with those for single-phase flow, showing that the parallel implementation is rather robust and is not significantly affected by the intrinsic nonlinearity of the multi-phase problem.

3.6 Conclusion

We have presented in this chapter details of a massive scale implementation of the SPH method outlined in the previous section. This method provides comparable accuracy to a standard second-order finite volume solver but with the flexibility of a meshless method. For the simulation of pore scale geometry presented here particles can simply be placed on a lattice and identified as being either fluid or solid. Meanwhile, for the finite volume discretization a mesh has to be carefully constructed to avoid the formation of degenerate cells and requires tedious user interaction during the meshing stage. The results presented here were meant to be a precursor to simulating suspension flows in which the beads were allowed to advect under the flow. This problem provides the ideal application for a meshless method, in which the construction of meshes conforming to each colloid at each timestep is intractable. Unfortunately, for the regime of vanishing Reynolds number flows we were interested in, the method was found to be unstable. Projection methods have a well-known splitting error that scales with the inverse of the Reynolds number, and a common remedy for this is to use high-order backward difference formula (BDF) schemes coupled with high-order discretization[75]. This pursuit of high-order meshless schemes provided the motivation for the remainder of the work in this thesis, and in the following chapter we present fundamentals of the moving least

square method, which we selected as the ideal approach for extending our current code to high-order.

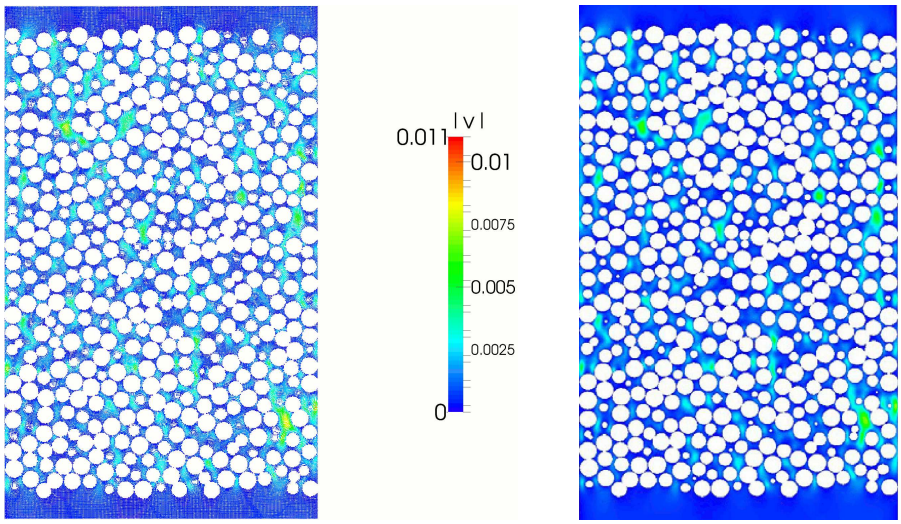


Figure 3.12: Steady state flow for the bead pack problem using the current approach (left) and a finite volume solution (right) computed with 30 million tetrahedral element smoothed-surface boundary-fitted mesh solver computed in STAR-CCM+ [165].

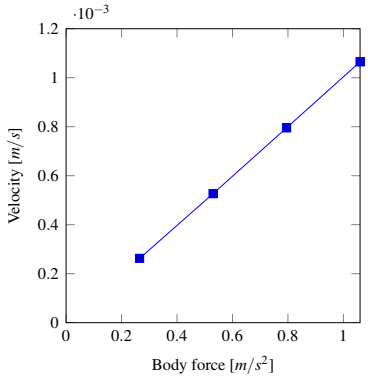


Figure 3.13: Average velocity as a function of applied body force. The results are in agreement with the linear Darcy law in Eqn. 3.3 and consistent with results in [165].

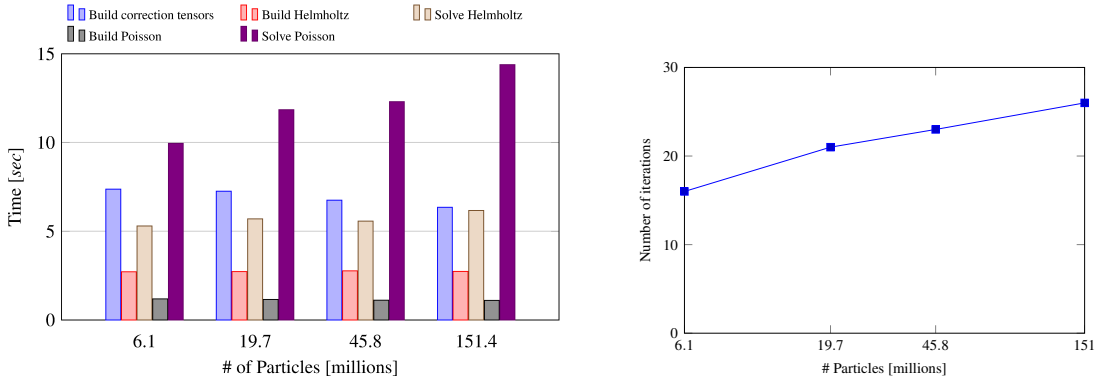


Figure 3.14: Weak scalability for single-phase problem (the number of particles per process is kept approximately constant, see Tab. 3.2). Left: computational average times per time step for assembling terms and solving linear systerms vs number of particles. Right: Number of iterations of the GMRES solver vs the number of particles for the Poisson problem.

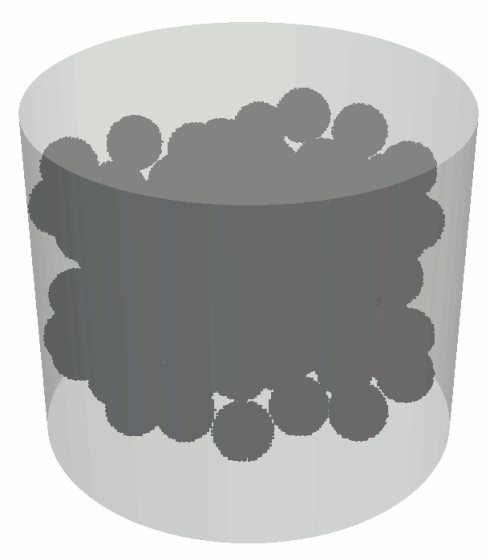


Figure 3.15: Initial configuration of the spherical CO_2 bubbles for multi-phase problem.

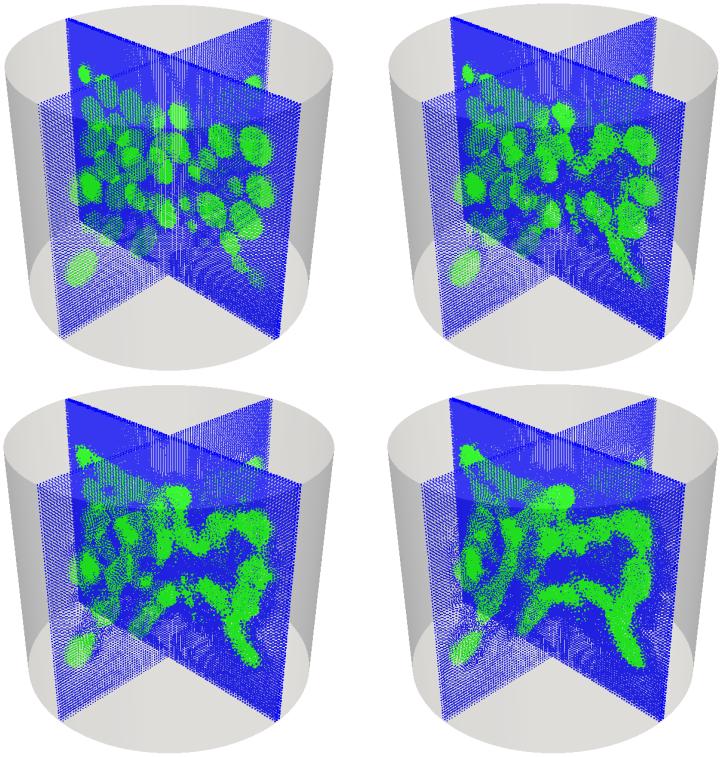


Figure 3.16: Fluid phases at different time steps, CO_2 in green and water in blue. From left to right, top to bottom, t=1.125ms, 2.25ms, 3.375ms, 4.5ms.

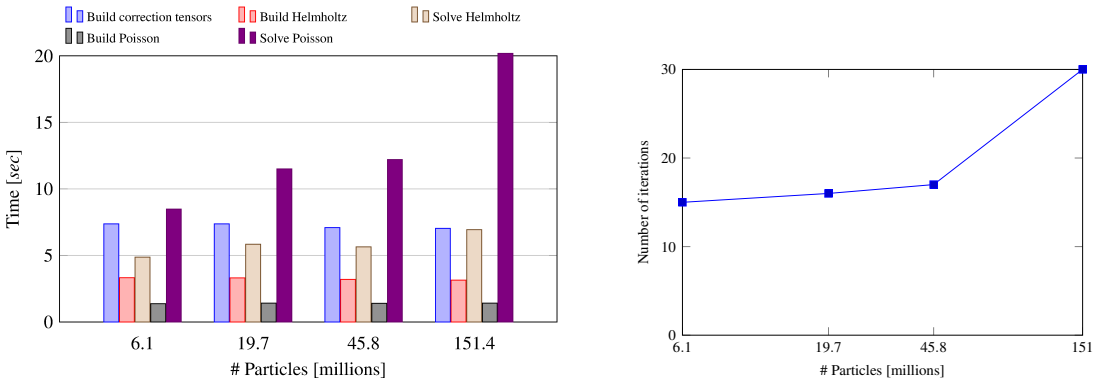


Figure 3.17: Weak scalability for multi-phase problem. Left: computational average times per time step for assembling terms and solving linear systerms vs number of particles. Right: Number of iterations of the GMRES solver vs the number of particles for the Poisson problem.

Chapter Four

Classical moving least squares: A framework for high-order consistent approximation

The remainder of the thesis will be concerned with developing new numerical methods, extending the previously presented code to high order. The SPH code presented thus far is robust and useful for a range of problems, but eventually exhibits stability issues at very high and very low Reynolds numbers. In this chapter we summarize some of the theoretical underpinnings of the moving least squares method (MLS), present some more advanced extensions that have recently been presented in the literature, and finally present high order stiffly stable arbitrary Lagrangian Eulerian (ALE) projection schemes that couple together with MLS discretizations to give high order in space and time.

4.1 Introduction

To define meshless methods in a broad sense, we consider the approximation theory associated with reconstruction of a function from an unstructured set of point data sampled from the underlying function. As mentioned in the first chapter, optimization provides an ideal tool to construct such approximations. If, for example, we seek a pointwise reconstruction as some linear combination of basis functions, we can pose an optimization problem seeking at any point the optimal combination of basis functions that in some sense best approximates nearby nodal data. Perhaps the simplest optimization problem is to define optimality in terms of a weighted l_2 norm and to find a collection of polynomials that best matches data; this gives the moving least squares method (MLS) [158]. Minimizing instead an entropy functional gives rise to the maximum entropy meshfree approximants [143, 8, 119, 120, 33], while assuming a normal distribution on the underlying point data and minimizing a covariance functional gives rise to kriging/co-kriging methods [25, 168, 169, 57, 86]. For this thesis we will use the moving least squares framework since minimization in l_2 has

an analytic solution allowing a cleaner presentation, however all of the approaches defined henceforth could just as easily be posed in a different norm.

Moving least squares was first introduced in the literature by Lancaster and Salkauskas [82] where the method was used to generate a surface forming a good approximation to scattered data. In computer science applications, MLS was primarily used to reconstruct surfaces from general point clouds. In the context of numerical PDE, some combination of moving least squares and finite difference methods appears under several different names. Attempts have been made since the early eighties to extend the simplicity of finite difference methods beyond Cartesian grids to general pointsets. These methods generate stencils that are exact for a given order polynomial either by solving interpolation problems or some form of moving least square optimization. Examples of approaches using interpolation are: generalized finite-differences[91, 50] and radial-basis function finite-difference methods [160]. Among MLS type methods are: finite point method [117, 118, 116, 45], collocated moving least squares[77], and direct meshless local Petrov-Galerkin [105]. While these methods have been successful in achieving high-order convergence, in some sense they resemble collocated finite difference methods, albeit on a non-uniform grid, and therefore are prone to the same instability issues. For this reason, a number of the above approaches require some form of numerical stabilization to avoid checker-boarding symptomatic of inf-sup incompatibility. In later chapters we will present generalizations of MLS that remedy this by pursuing analogues to marker and cell finite-difference methods and primal/dual approximation. However in this chapter we will first present fundamentals of classical MLS approaches to motivate the development of later techniques.

We gather a variety of concepts presented in the literature concerning fundamentals of MLS and its generalizations. The reference by Holger Wendland[158] provides

an excellent introduction to the technical aspects of scattered data approximation. We follow his approach and terminology for the remainder of this thesis but make some assumptions allowing us to skirt technical elements. We then gather some recent extensions of the approach from the literature which we will generalize in later chapters to develop new techniques for generating compact, high-order, compatible schemes for numerically solving PDE. Finally, we demonstrate a new stiffly stable scheme for solving incompressible Lagrangian hydrodynamics to arbitrary order. We will see that even though this new scheme provides high order approximation, the lack of compatibility inherent in standard MLS leads to subtle issues analagous to those found when using standard finite differences on collocated meshes, and this will motivate the need for development of more sophisticated discretizations in later chapters.

4.2 Local polynomial reproduction

Following Wendland [158] we define a scattered data approximant as follows. Given a set of data sites $\mathbf{X}_h := \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \Omega$ for some open bounded set $\Omega \subset \mathbb{R}^d$ with known function values $f_i := f(\mathbf{x}_i)$ for some function $f \in C^\infty$, we seek an approximant to f as some combination of functions $u_j : \Omega \to \mathbb{R}$ independent of f.

$$s(\mathbf{x}) = \sum_j f(\mathbf{x}_j) u_j(\mathbf{x}) \tag{4.1}$$

We define the fill distance as the largest ball that can fit in the domain without

intersecting any data sites

$$h_{X_h} = \sup_{x \in \Omega} \min_{1 \leq j \leq N} ||\mathbf{x} - \mathbf{x}_j||_2 \tag{4.2}$$

This parameter will quantify the spatial dependence of the approximant error as resolution is increased, in a similar manner to element diameter in finite elements or grid spacing in finite differences. Accuracy can be proven by showing that these combinations 4.1 can exactly reproduce polynomial functions for any $\mathbf{x} \in \Omega$.

The family of functions $\{u_j\}_{j=1,\ldots,N}$ forms a *local polynomial reproduction of order* m if there exists constants independent of data sites $h_0, C_1, C2 > 0$ such that for any polynomial of order m, $\pi_m(\mathbb{R}^d)$,

- $\sum_j p(\mathbf{x}_j)u_j = p(\mathbf{x})$ (polynomial reproduction)
- $\sum_j |u_j| < C_1$ (finite basis functions)
- $\forall \mathbf{x} \in \Omega$, $u_j(\mathbf{x}) = 0$ if $||\mathbf{x} - \mathbf{x}_j|| > C_2 h_{X_h}$ (compact support)

with $h_{X_h} < h_0$.

These conditions are simple to check and allow the following simple analysis of truncation error

Theorem 1. *Let* $\Omega^* = \cup_{x \in \Omega} B(x, C_2 h_0)$ *and consider the approximation* $s(\mathbf{x})$ *of* $f \in C^{m+1}$ *with a local polynomial reproduction of order* m*. Then there exists* $C > 0$ *such that*

$$|f(\mathbf{x}) - s(\mathbf{x})| < C h_{X_h}^{m+1} |f|_{C_{\Omega^*}^{m+1}}$$

where $|\cdot|_{C_{\Omega^*}^{m+1}}$ *denotes the* $m+1$ *order seminorm.*

Proof. The proof is given by Wendland[158]. First approximate f with p, the first m terms of its Taylor series at m. The polynomial reproduction property allows the elimination of these terms and compact support allows bounds to be placed on the remainder as follows

$$\begin{aligned}
|f(\mathbf{x}) - s(\mathbf{x})| &\leq |f(\mathbf{x}) - p(\mathbf{x})| + |p(\mathbf{x}) - s(\mathbf{x})| \\
&= |f(\mathbf{x}) - p(\mathbf{x})| + |\sum_j p(\mathbf{x}_j)u_j - \sum_j f(\mathbf{x}_j)u_j| \\
&\leq |f(\mathbf{x}) - p(\mathbf{x})| + \sum_j |p_j - f_j||u_j| \\
&\leq ||f - p||_{L^\infty(B(x, C_2 \cdot h_{X_h}))}(1 + \sum_j |u_j|) \\
&\leq ||f - p||_{L^\infty(B(x, C_2 \cdot h_{X_h}))}(1 + C_1) \\
&< h_{X_h}^{m+1} |f|_{C^{m+1}_{\Omega^*}}(1 + C_1).
\end{aligned}$$

Here $B(x, h)$ denotes the ball centered at x of radius h, and the final bound follows from standard arguments for best polynomial approximation. □

This analysis will prove fundamental to making sense of approximation in a meshless context. Without a mesh there is no way to make sense of truncation error elementwise, but other proofs will share the common feature of using polynomial consistency to eliminate the first m terms of the truncation error, and then using compact support to bound the remaining error with the fill distance. As a result, when posing our search for a meshless approximation as an optimization problem, we will add equality constraints to enforce polynomial reproduction and therefore consistency.

4.3 Moving least squares approximation

In defining the moving least squares method, we will assume $\pi_m(\mathbb{R}^d)$-unisolvency of the underlying data sites. This skirts the issue of proving existence and uniqueness results for a local polynomial reproduction; in what follows we assume that such an approximation exists and define optimization problems to find it. This allows us to avoid many of the technical details discussed in [158] for the sake of brevity. In practice this means that we will assume a non-degenerate set of data sites; that is data sites cannot occupy the same location or be colinear.

Given a $\pi_m(\mathbb{R}^d)$-unisolvent set of data sites $\mathbf{X}_h$ with $\{u_j\}_{j=1,\ldots,N}$ sampled from $u \in C^\infty$, we seek the m^{th} order polynomial $p^* \in \pi_m(\mathbb{R}^d)$ that best locally approximates u at a given point x by minimizing the weighted least squares problem

$$p^* = \underset{p \in \pi_m(\mathbb{R}^d)}{\arg\min} \sum_j (u_j - p_j)^2 \, W(||\mathbf{x} - \mathbf{x}_j||) \tag{4.3}$$

where W(r) is a a radially symmetric positive kernel with compact support ϵ, and the function $\arg\min_{x \in V} f(x)$ denotes the element x in the set V for which the functional f takes smallest value. The term "moving" in MLS refers to the fact that for each $\mathbf{x}$, as the kernel varies in space, a different pointwise polynomial reconstruction is generated.

By choosing a polynomial basis $p = \mathbf{c}^\intercal \mathbf{P}(\mathbf{x})$ with dimension $Q = dim(\pi_m(\mathbb{R}^d))$, we can equivalently state this by searching for an optimal coefficient vector $\mathbf{c}^* \in \mathbb{R}^Q$

$$\mathbf{c}^* = \underset{\mathbf{c} \in \mathbb{R}^Q}{\arg\min} \sum_j \left(u_j - \mathbf{c}^\intercal \mathbf{P}(\mathbf{x}_j)\right)^2 W(||\mathbf{x} - \mathbf{x}_j||). \tag{4.4}$$

We have omitted the implicit spatial dependence of $\mathbf{c}(\mathbf{x})$ coming from the terms involving the kernel. Taking the variation of Equation 4.4 with respect to $\mathbf{c}$, we obtain the following solution

$$\mathbf{c}^* = \mathbf{M}^{-1}(\mathbf{x}) \sum_j \mathbf{P}_j W((||\mathbf{x} - \mathbf{x}_j||) u_j \tag{4.5}$$

where $\mathbf{M} \in \mathbb{R}^{Q \times Q}$ is the symmetric positive definite system of weighted normal equations

$$\mathbf{M}(\mathbf{x}) = \sum_k \mathbf{P}_k W(||\mathbf{x} - x_k||) \mathbf{P}_k^\intercal \tag{4.6}$$

Therefore, evaluating the MLS approximant at any point x amounts to inverting Equation 4.6 and evaluating

$$s(\mathbf{x}) = \mathbf{P}^\intercal(\mathbf{x}) \mathbf{c}^* \tag{4.7}$$

For moderate order polynomial fits, this process is $O(N)$ in the number of evaluations due to the compact support of the kernel function and therefore leads to an efficient means of approximating the sampled data for all $\mathbf{x} \in \Omega$. It is straightforward to verify that Equation 4.7, can be expanded as a linear combination of rational polynomials multiplied by the kernel function; the approximation therefore inherits the same smoothness as the kernel. It is also straightforward to see that the approximation can exactly reproduce polynomials by noting that Equation 4.3 is identically equal to zero when applied to $u \in \pi_m(\mathbb{R}^d)$. Alternatively, we can see that MLS forms a local polynomial reproduction by identifying the approximation as the solution of the following dual constrained least squares problem.

Theorem 2. *Assuming a solution to Equation 4.3 exists, then Equation 4.7 can be equivalently written as*

$$s(\mathbf{x}) = \sum_{i \in supp(W)} a_i^*(x) f(x_i)$$

where the coefficients $\mathbf{a}^*$ *are found as the solution to the constrained optimization problem*

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} \sum_{i \in supp(W)} a_i(x)^2 \frac{1}{W(||x - x_i||)}$$

such that for any polynomial $p \in \pi_m(\mathbb{R}^d)$

$$p(x) = \sum_j a_j p(x_j)$$

Proof. The proof follows immediately from minimizing the optimization function using Lagrange multipliers to enforce the constraints, and standard uniqueness arguments for convex optimization. □

The details of the proof can be found in Chapter 4 of [158] along with conditions on Ω such that Equation 4.7 forms a local polynomial reproduction. We present these two interpretations of MLS here because they will each provide a different foundation for developing new meshless methods in later chapters. The first interpretation (Equation 4.3) allows a simpler implementation, and we will later see that because the optimization is over the set of polynomials, the addition of regularizing terms to the cost functional will not impact polynomial reproduction so long as they are convex. On the other hand, the structure of the second interpretation (Theorem 2) shows the real strength of optimization. The actual cost functional minimizes a quantity that is in some way irrelevant to the problem (the magnitude of the coefficients), while constraints are used to endow the approximation with desirable properties (accuracy) provided a useful ansatz for the solution can be posed (local polynomial

reproduction). In the literature, these two approaches have primarily been used only to achieve accuracy. The later chapters of this work we will attempt to modify these two strategies to develop new methods endowing meshless approximation with additional properties.

4.4 Diffuse derivative approximation

To use MLS to solve PDE numerically, it is necessary to use Equation 4.7 to generate approximations to derivatives. This becomes infeasible due to the implicit spatial dependence of the coefficient vector, e.g. for a gradient approximation:

$$\nabla s(\mathbf{x}) = (\nabla \mathbf{P})^{\mathsf{T}} \mathbf{c}^* + \mathbf{P}^{\mathsf{T}} (\nabla \mathbf{c}^*) \tag{4.8}$$

the first term is trivial to compute, but the second term requires the evaluation of the matrix inverse derivative

$$\nabla \mathbf{M}^{-1}(x) = -\mathbf{M}^{-1} (\nabla \mathbf{M}) \mathbf{M}^{-1} \tag{4.9}$$

and since $\mathbf{M}$ is dense, this results in three $O(N^3)$ operations to invert and multiply $\mathbf{M}$ which quickly increases computational cost. The extension to higher order derivatives is even more cumbersome, and for this reason MLS was underused in the literature for some time.

It is possible however to make the so called *diffuse derivative approximation* in which the spatial dependence of the coefficient vector is neglected. Adopting multi-index notation, we write the approximation to an arbitrary α^{th} order differential operator as

$$D_h^\alpha u(\mathbf{x}) = (D^\alpha u(\mathbf{x}))^\intercal \, \mathbf{c}^* \tag{4.10}$$

Although used in a variety of contexts beforehand, the work by Mirzaei [106] formalizes this assumption and develops the following approximation results.

Theorem 3 (Local polynomial reproduction of weak derivatives [106])**.** *Given the approximation to $D^\alpha u$ in Equation 4.10 using the m^{th} order MLS approximation satisfies for $p \in \pi_m(\mathbb{R}^d)$ satisfies*

$$D_h^\alpha p = D^\alpha p$$

As a corollary to this theorem the following approximation result holds:

Theorem 4 ([106])**.** *If m and p satisfy $Q > \frac{n}{p} - 1$, then the following interpolation estimate holds for each point i*

$$||D^\alpha u(x_i) - D_h^\alpha u(x_i)||_{L^p(\Omega)} \leq C(m)\epsilon^{m+1-|\alpha|}||u||_{W^{m+1,p}(\Omega)}$$

where $||\cdot||_{W^{m+1,p}(\Omega)}$ denotes the standard Sobelev norm.

We omit the proofs of these results, noting only that they mirror the proof of Theorem 1 and use polynomial consistency to eliminate the first m terms of the Taylor series for u.

4.5 Collocation schemes for MLS

The rational nature of the approximation to functions and derivatives in Equations 4.7 and 4.10 leads to difficulty when attempting to pose solutions of PDE in a weak form. Without exact quadrature rules for rational polynomials, methods utilizing weak forms require the evaluations of the approximant at many quadrature points, each of which requires an inversion of Equation 4.6. In order to avoid this, we seek to solve PDE in strong form as follows.

Given the linear boundary value problem

$$\begin{cases} \mathcal{L}_1 u = f, & \text{if } x \in \Omega \\ \mathcal{L}_2 u = g, & \text{if } x \in \partial\Omega \end{cases} \tag{4.11}$$

for some linear differential operators $\mathcal{L}_1, \mathcal{L}_2$, we discretize by distributing particles on the boundary $\partial\Omega$ and within the domain Ω. We then obtain a linear system of equations by applying the diffuse derivative approximation of the differential operators operators at each particle, i.e. for each particle i,

$$\begin{cases} \mathcal{L}_1^h u(x_i) = f(x_i), & \text{if } x_i \in \Omega \\ \mathcal{L}_2^h u(x_i) = g(x_i), & \text{if } x_i \in \partial\Omega \end{cases} \tag{4.12}$$

The resulting linear system is sparse but in general asymmetric due to anisotropy in the data site locations, independent of the PDE. We will demonstrate this by solving the Poisson problem ($\mathcal{L}_1 = \nabla^2$) with Neumann ($\mathcal{L}_2 = \partial_n$) and Dirchlet ($\mathcal{L}_2 = I$) boundary conditions, but first point out some implementation details about how to efficiently evaluate derivatives at data sites.

To perform the approximation stably and avoid ill-conditioning when inverting Equation 4.6, for each particle i we select the basis consisting of the ϵ-scaled Taylor monomials, i.e. in multi-index notation

$$\mathbf{P}(x) = \left\{ \frac{1}{\alpha!} \left(\frac{x - x_i}{\epsilon} \right)^{\alpha} \right\}_{|\alpha|<m} \tag{4.13}$$

Although for large m it is well-known that this gives rise to the poorly conditioned Hilbert matrix, for moderate m and using double precision it is not an issue to solve Equation 4.6 with LU-decomposition. Special care will only need to be taken when going to higher than sixth order, as considered in later chapters. This ill-conditioning can be handled with standard techniques (QR decomposition, singular value decomposition) or by selecting a better conditioned matrix. These options will be discussed when relevant later, but for now we use Taylor monomials because the basis has the following property when evaluated at $x = x_i$:

$$\mathbf{P}(x_i) = \{1, 0, 0, ..., 0\} \tag{4.14}$$

and when the derivative operator D^{β} is applied to it, only the element of the vector corresponding to the multi-index $\alpha = \beta$ is nonzero.

$$D_h^{\beta}\mathbf{P}(x_i) = \left\{ 0, ..., 0, \frac{1}{\epsilon^{|\beta|}}, 0, ..., 0 \right\} \tag{4.15}$$

Therefore, when evaluating Equation 4.10, rather than applying a full matrix vector product through the normal matrices, only a single inner product needs to be evaluated. We write this concisely as:

$$D_h^{\alpha}u(x_i) = \frac{1}{\epsilon^{\alpha}}\hat{e}_{\alpha}^{\intercal}\mathbf{M}_i^{-1}\left(\sum_{j \in N_i} \mathbf{P}^{\intercal}(x_j)W(x_i - x_j)u_j \right) \tag{4.16}$$

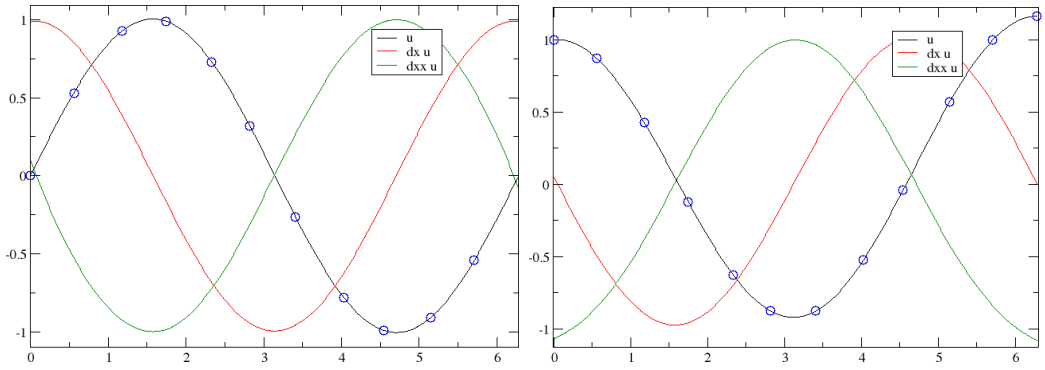


Figure 4.1: Solution to Poisson problem with Dirichlet (left) and Neumann (right).

m	2	3	4	5	6
order	2.014	2.016	3.984	4.028	5.967

Table 4.1: Convergence rates for 1D Dirichlet Poisson problem.

where $\hat{e}_\alpha$ denotes the cardinal unit vector that is non-zero in the α^{th} entry. We stress here that this expression can equivalently be viewed as a finite difference-like stencil:

$$D_h^\alpha u(x_i) = \sum_j \alpha_{ij} u_j \tag{4.17}$$

so that in effect, this collocation approach is equivalent to applying a finite difference method on arbitrary pointsets. In fact, we recover standard second-order, centered, finite difference approximations in the case where particles lie on a Cartesian (i, j) grid, the weighting function is taken to have unit value on the points $\{(i, j), (i + 1, j), (i, j + 1), (i - 1, j), (i, j - 1)\}$, and a linear polynomial space is used.

With an efficient means of assembling the matrices, we now present convergence results on the interval $\Omega = [0, 2\pi]$ solving the Poisson problem with increasingly high-order polynomial reconstructions. For this simple 1D case we can solve the resulting discrete system of equations using LU-factorization, and we present representative solutions and convergence rates in Figures 4.1 and 4.2.

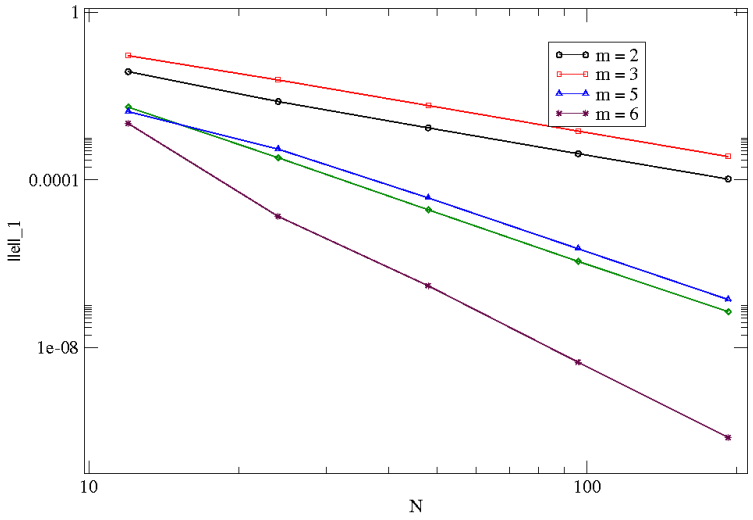


Figure 4.2: Solution to Poisson problem with Dirichlet (left) and Neumann (right).

4.6 An ALE scheme for finite Reynolds number problems

We consider now the MLS discretization of a unsteady Stokes problem discretized in time with a backward difference formula (BDF) scheme[75] as

$$\begin{cases} \dfrac{\gamma_0 \mathbf{u}^{n+1} - \sum_k \alpha_k \mathbf{u}^{n-k}}{\Delta t} = -\nabla p^{n+1} + \nu \nabla^2 \mathbf{u}^{n+1}, \\ \nabla \cdot \mathbf{u}^{n+1} = 0, \end{cases} \tag{4.18}$$

and a splitting of the Stokes operator via a velocity-correction method

$$\begin{cases} \dfrac{\gamma_0 \mathbf{u}^{*} - \sum_k \alpha_k \mathbf{u}^{n-k}}{\Delta t} = -\nabla p^{n+1} - \nu \nabla \times \nabla \times \mathbf{u}_H, \\ \nabla \cdot \mathbf{u}^{*} = 0, \end{cases} \tag{4.19}$$

$$\left\{ \frac{\gamma_0 \mathbf{u}^{n+1} - \gamma_0 \mathbf{u}^{*}}{\Delta t} = \nu \left(\nabla^2 \mathbf{u}^{n+1} + \nabla \times \nabla \times \mathbf{u}_H \right). \right. \tag{4.20}$$

where $\mathbf{u}_H := \sum_\beta \beta_k \mathbf{u}^{n-k}$ is an extrapolation of the velocity from values available at previous timesteps, and α_k, β_k, and γ_0 are coefficients chosen to ensure a scheme of a given order in time. The solution of Equations 4.19 amount to solving a Poisson equation to obtain the pressure p^{n+1}. This pressure is used to obtain a provisional divergence-free velocity $\mathbf{u}^*$, which is corrected by solving a Helmholtz equation to obtain $\mathbf{u}^{n+1}$. Details of these schemes can be found in the reference by Karniadakis [74].

Velocity-correction methods have a well-known stabilizing effect, as they relax the divergence free constraint and enforce instead that the divergence satisfy a heat equation [75, 94]. We demonstrate the stability of this scheme by expanding in terms of the normal mode problem ansatz

$$\mathbf{u}(x, y, t) = \exp(-\omega^2 t + ikx)\hat{\mathbf{u}}(y) \tag{4.21}$$

$$p(x, y, t) = \exp(-\omega^2 t + ikx)\hat{p}(y) \tag{4.22}$$

and calculating the generalized eigenvalues associated with a single mode $k = 1$, following the analysis in [94]. Figure 4.3 demonstrates the stability of these discretizations by calculating the amplification factor associated with these generalized eigenvalues, and Figure 4.4 demonstrates convergence in time corresponding to the order of the BDF discretization used.

Having demonstrated the stability of this approach we now present an ALE scheme for simulating Lagrangian hydrodynamics. When considering the implementation of the velocity-correction scheme Equation 4.19 in a Lagrangian framework we must determine a consistent way to control the distribution of particles. As with SPH, even if the flow field is calculated exactly, flows containing stagnation points

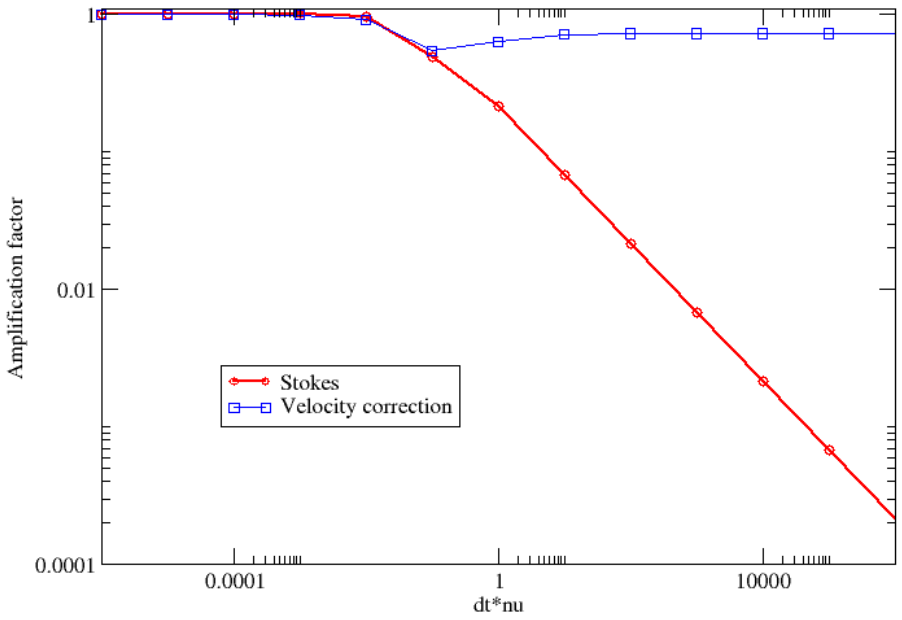


Figure 4.3: Amplification factors for third order BDF discretization of Stokes and velocity correction schemes demonstrating stability of collocated MLS discretization.

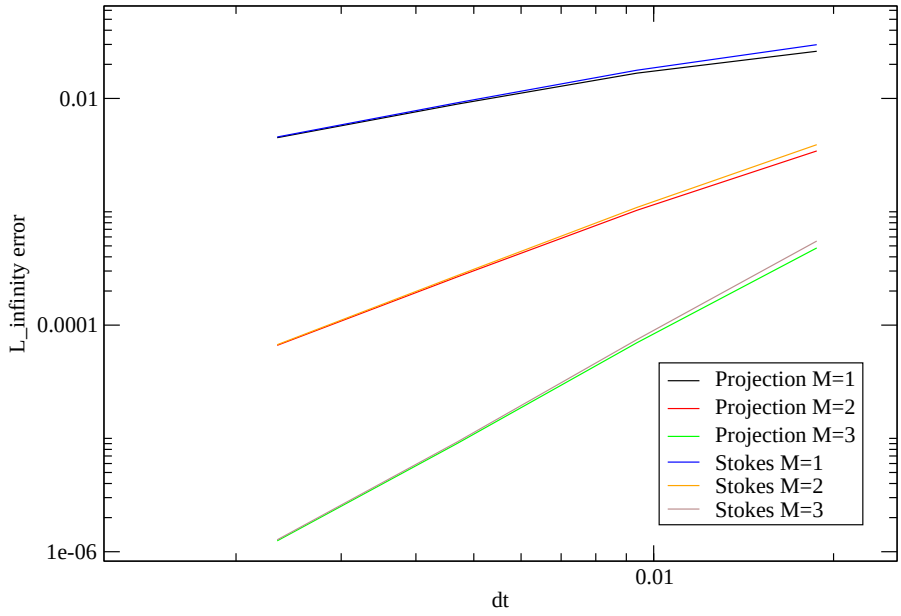


Figure 4.4: Optimal convergence for first, second, and third order BDF discretizations of Stokes and velocity correction schemes.

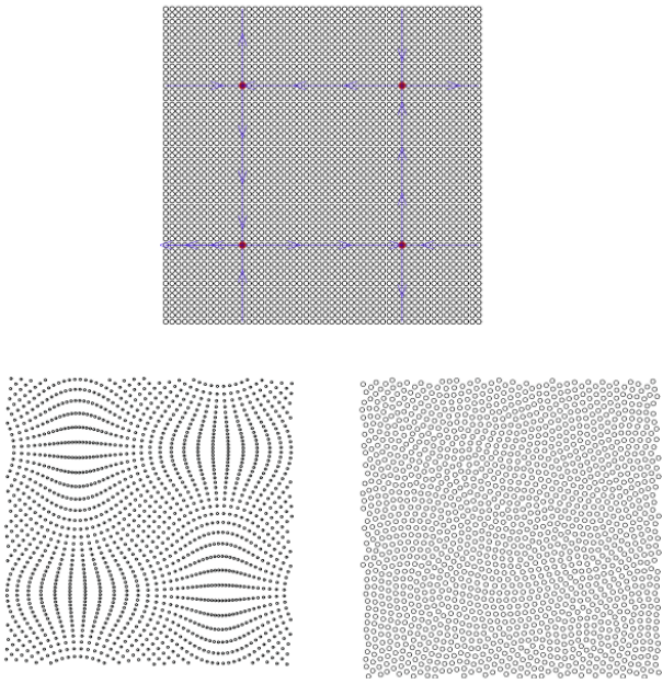


Figure 4.5: Particle "bunching" occuring near stagnation points of a Taylor-Green vortex simulation. An anisotropy correction is used to break particle symmetry, pushing particles off streamlines. (top) Streamlines and initial particle configuration. (bottom left) advected particle locations without correction, (bottom right) advected particle locations with correction (Figure taken from Xu et al.[132]).

are prone to particle "bunching", and at each timestep particles need to be perturbed by an anisotropy correction to avoid degenerate particle distributions. We follow the correction scheme used in the SPH literature, Equation 2.1, (See Figure 4.5). Note that in general any correction vector Δr can be applied to each particle, so that the particle position can be integrated with a BDF formula as

$$\frac{\gamma_0 x^{n+1} - \sum_k \alpha_k x^{n-k}}{\Delta t} = u^{n+1} + \frac{\gamma_0}{\Delta t}\Delta r(x^{n+1}) \tag{4.23}$$

where we have written the dependence of the correction on the set of particle locations at time n, x^n as $\Delta r(x^n)$. If we couple this together with Equation 4.19, we obtain for the particle motion, a non-linear coupled system of equations, since the differential operators in the Stokes operator are implicitly functions of particle positions. We can decouple the particle integration from the flow solver at each timestep by extrapolating the update velocity from previous timesteps to obtain a predictor of particle position as

$$\frac{\gamma_0 x^{*} - \sum_k \alpha_k x^{n-k}}{\Delta t} = \sum_k \alpha_k u^{n-k} \tag{4.24}$$

and evaluating the shift vector at this position $\mathbf{x}^*$. We may then correct this with

$$\frac{\gamma_0 x^{n+1} - \sum_k \alpha_k x^{n-k}}{\Delta t} = \sum_k \alpha_k u^{n-k} + \frac{\gamma_0}{\Delta t}\Delta r(x^{*}). \tag{4.25}$$

With the positions updated, we can solve the Stokes operator with an added advection term to account for the anisotropy correction

$$\begin{cases} \dfrac{\gamma_0 \mathbf{u}^{n+1} - \sum_k \alpha_k \mathbf{u}^{n-k}}{\Delta t} + \dfrac{\gamma_0}{\Delta t} \Delta r(x^*) \cdot \nabla \mathbf{u}^{n+1} = -\nabla p^{n+1} + \nu \nabla^2 \mathbf{u}^{n+1}, \\ \nabla \cdot \mathbf{u}^{n+1} = 0, \end{cases} \tag{4.26}$$

This Stokes operator can finally be split into a velocity correction scheme with appropriate rotational pressure boundary conditions. Although we have introduced an advection term, we avoid the nonlinearity of the full Navier-Stokes equations because the perturbation term is small. We recall from the previous chapter that Δr is of magnitude $CU_{max}\Delta t$, where $C \approx 0.0001$ is chosen just large enough to break symmetry. We define a scheme using m^{th} order BDF formula for the time derivative and n^{th} order extrapolation for the velocity as a (m, n) BDF scheme. This scheme therefore is fully implicit in velocity, stiffly stable, and linear. We demonstrate accuracy of the scheme by solving the manufactured solution

$$u = U_0 \cos x \cos y \sin t \tag{4.27}$$

$$v = U_0 \sin x \sin y \sin t \tag{4.28}$$

$$p = U_0 \sin x \sin y \sin t \tag{4.29}$$

and adding a forcing term such that the Navier-Stokes equations are satisfied. For a spatially well-resolved solution, we note that the RMS error scales optimally with the timestep size in Figure 4.6. By increasing U_0 while keeping $\Delta t/\Delta x$ fixed, we see in Figure 4.7 that the code is stable until reaching a Courant number of roughly 2.

We demonstrate in Figure 4.8 some example flows of cylinders in channels exhibiting recirculation zones. This scheme was implemented in both the serial FASP code and the parallel LAMMPS/Trilinos framework. However, for moderate Reynolds

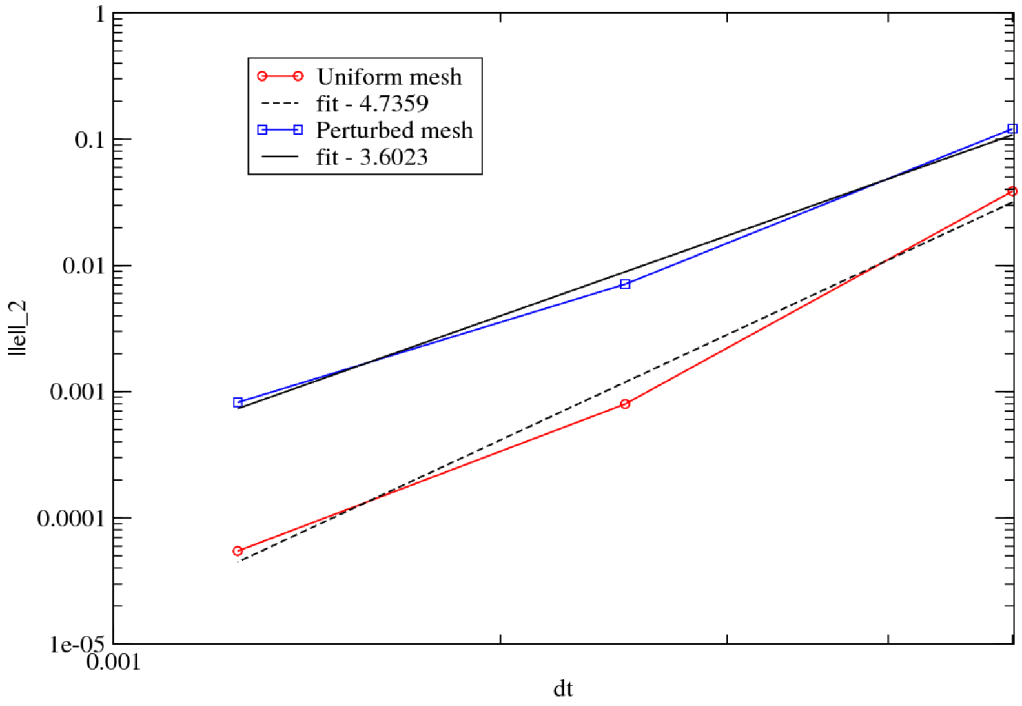


Figure 4.6: Convergence for a spatially well-resolved $(3,3)$ BDF scheme as timestep is refined, with $Co \approx 1$ and fourth order spatial reconstruction.

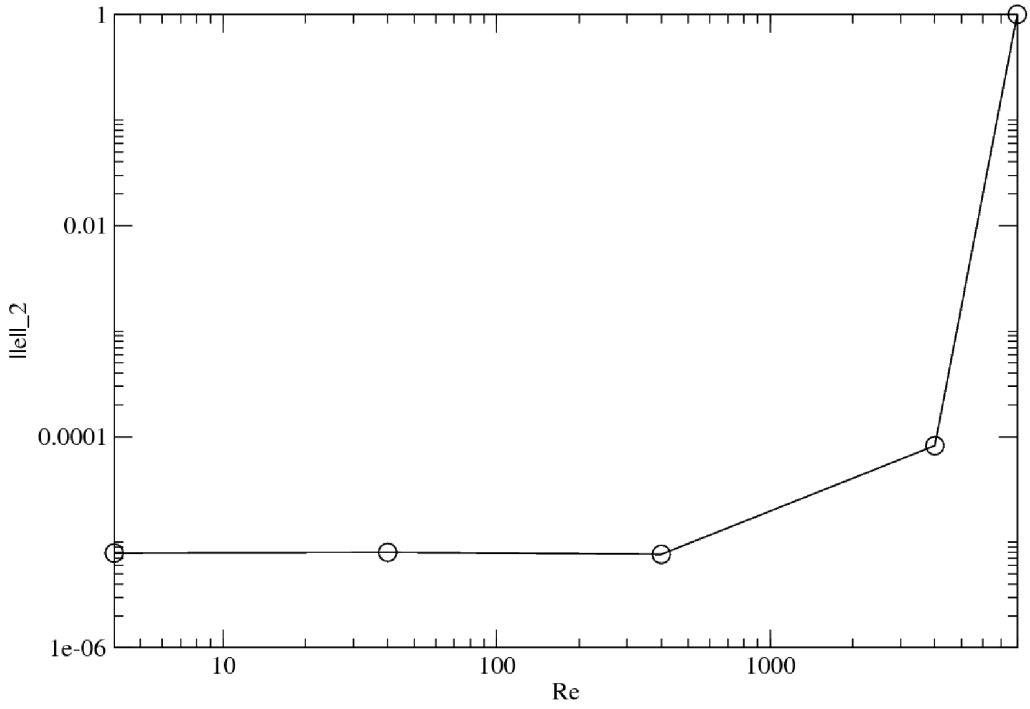


Figure 4.7: Range of Reynolds numbers for which the scheme is stable, for fixed $N = 64$, $dt/dx \approx 1$, and increasing force magnitude. For Reynolds numbers larger than 1000 the RMS error quickly grows, ultimately leading to a crash. This corresponds to the point at which $Co = 2$.

number a subtle error appears in the numerical results. Near boundaries, a large error in the divergence field appears (See Figure 4.9 for an example occuring near a rectangular cylinder in channel flow). The divergence error is so large that particles are artificially pushed away from the boundary. While a divergence error is always attributed to the splitting error in the velocity correction scheme [94, 75], it is generally confined to a boundary layer of magnitude $O((\nu\Delta t)^k)$, for a (k,k) BDF scheme. Therefore, by using a high-order scheme this effect can usually be controlled efficiently by taking a sufficiently small timestep. However, in this scheme, the divergence error was found to vary linearly, regardless of the order scheme used, leading to prohibitively small timestep sizes to control divergence errors. This is attributed to a lack of compatibility in the MLS discretization. A key step in the velocity-correction scheme is the approximation of the viscous term with an explicit $\nabla \times \nabla \times \mathbf{u}$ term, whose divergence vanishes on the interior due to the identity $\nabla \cdot \nabla \times \mathbf{v}$ for any $\mathbf{v}$. The MLS discretization, like most standard finite difference methods, fails to provide a mimetic approximation to this identity.

4.7 Preface to remainder of thesis

In the chapters thus far we have demonstrated a highly scalable meshless framework that is attractive for solving a range of problems. When attempting to extend this to higher order over a range of of Reynolds numbers, we have demonstrated in this chapter that a lack of compatibility leads to large divergence errors contaminating the convergence of solutions when we move beyond smooth manufactured solutions to physically relevant problems. The remainder of this thesis is focused on remedying this lack of compatibility. There are two main strategies that can be employed to approach this. We can either build differential operators with discrete compatibility,

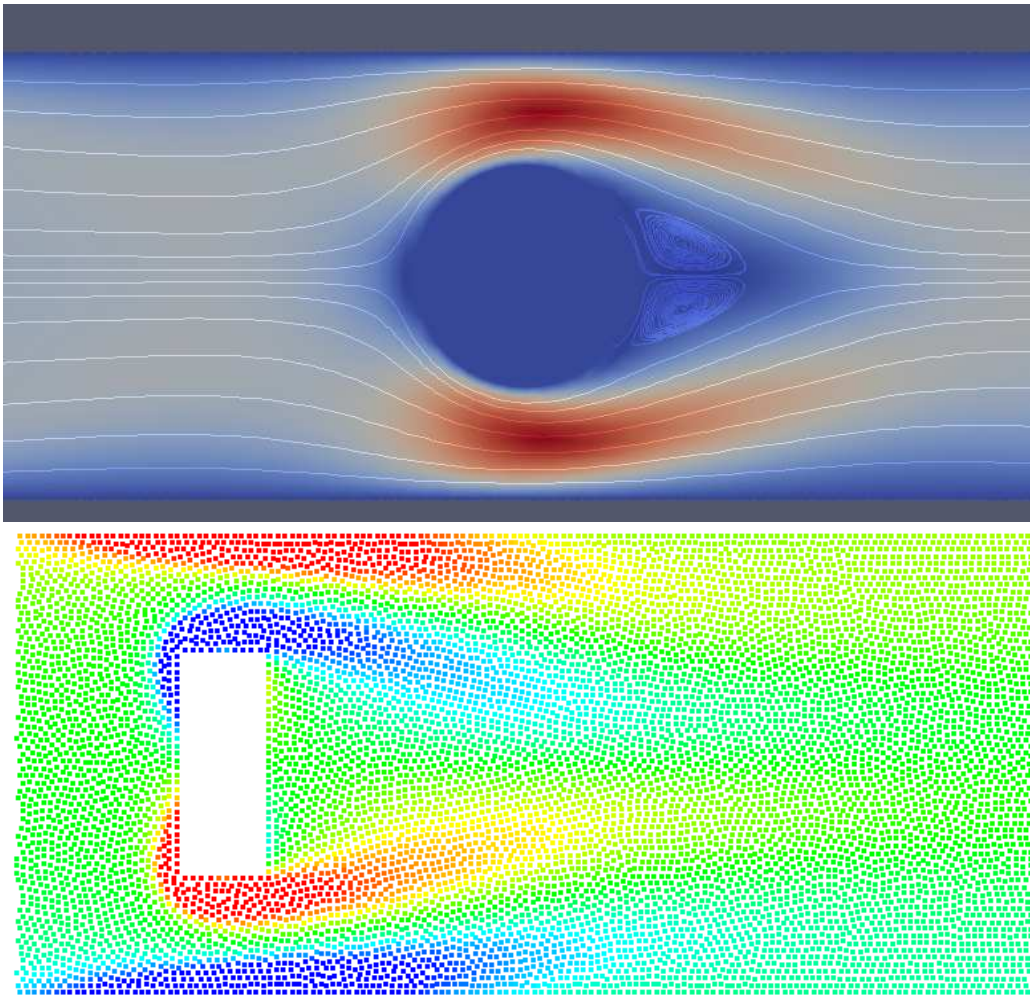


Figure 4.8: Some example flows of cylinders in channels exhibiting recirculation zones. (top) implementation in serial code (bottom) implementation in parallel LAMMPS/Trilinos library.

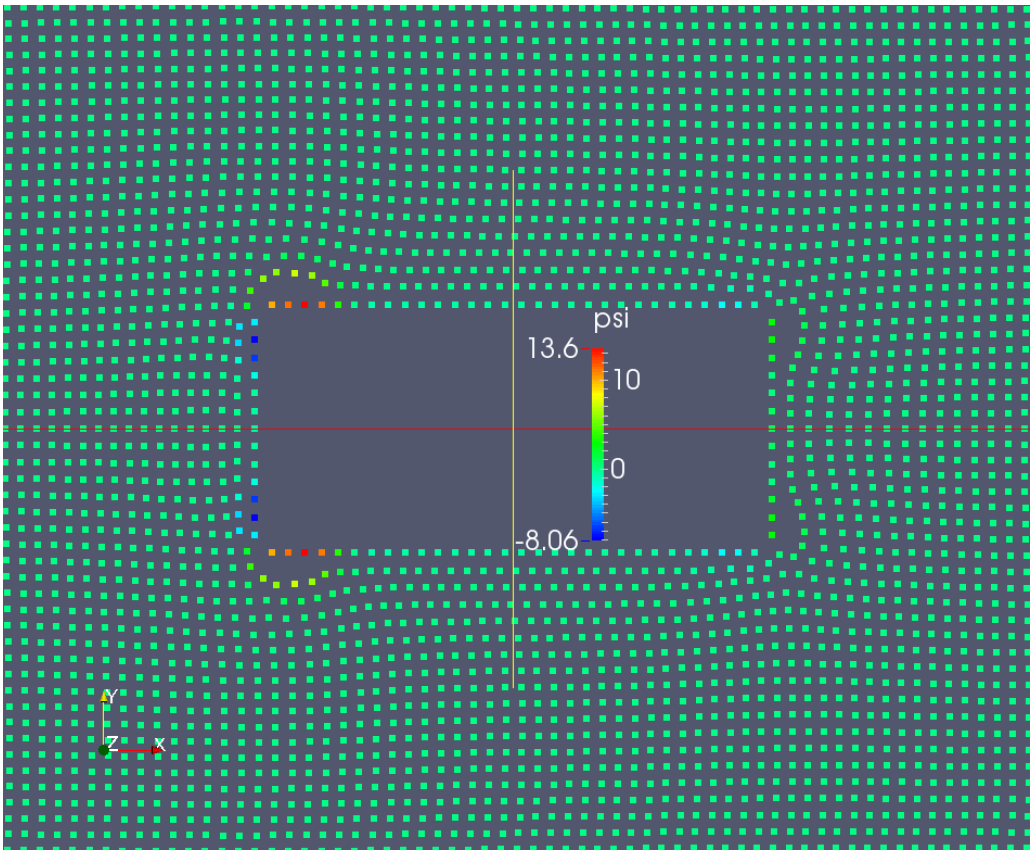


Figure 4.9: Divergence concetration near boundaries giving rise to erroneous particle voids.

or we can develop more accurate differential operators so that the difference between the differential and continuous operators is minimal.

The first, and perhaps more difficult, requires us to construct a compatible meshless discretization for the Stokes operator. We pursue this in the following three chapters by first studying the case in which a mesh is re-introduced to the problem by constructing Voronoi diagrams for each particle. This will illustrate a path to compatibility by mimicking the structure of compatible divergence and gradient operators. In the next chapter we pursue this analogy to develop a compatible meshless discretization for the model div-grad problem, and in the next chapter combine that approach to solve the full Stokes equations. In doing so, we will ultimately attain not only a remedy to the incompatibility driving the previous divergence errors, but also obtain enhanced stability and convergence behavior normally associated with compatible finite element schemes. In the second strategy, we attempt to develop more accurate approximation to achieve compatibility by more accurately approximating the continuous operators. This is achieved by using Hermite MLS approximation to develop compact, higher order schemes that generalize classical compact finite difference schemes.

In both approaches, we obtain stable highly accurate approximations that can be used to solve the Stokes equations.

Chapter Five

Conservative moving least squares: Meshless approximation for conservation laws using graphs

In this chapter we present two new schemes that present some form of conservation to remedy the lack of consistency in the collocation scheme. As discussed in the first chapter, conservation follows from Stokes theorem, and therefore requires a simplicial complex to pair with exterior derivatives. We begin this chapter by briefly reintroducing a mesh to the problem by generating the Voronoi diagram associated with a given point set. While we do not pursue this formulation since it is not truly meshless, it is instructive in examining the role that mesh topology will play and we will show how moving least squares can be used to generate high-order finite volume schemes. We then abandon the mesh and consider a case in which we can generate a graph that is able to function as an abstract simplicial complex and provide a discrete form of conservation, mirroring the process used in the Voronoi diagram case.

5.1 Voronoi diagrams: reintroducing a mesh

Given a set of points $\mathbf{X}_h = \{x_i\} \subset \mathbf{R}^n$, we associate with each point x_i the Voronoi cell

$$C_i = \left\{ x \in \mathbf{R}^d | d(x, x_i) < d(x, x_j) \text{ for all } x_j \in \mathbf{X}_h \right\} \tag{5.1}$$

Assuming that $\mathbf{X}_h$ approximates a compact domain Ω, we partition the points into boundary and interior $\mathbf{X}_h = \mathbf{X}_h^B \cup \mathbf{X}_h^I$, where $\mathbf{X}_h^B \subset \partial\Omega$. We associate with each boundary point $x_i \in \mathbf{X}_h^B$ an outward pointing normal $\vec{n}_i$, and truncate corresponding boundary cells

$$C_i^B = \left\{ x \in \mathbf{R}^d | d(x, x_i) < d(x, x_j) \text{ for all } x_j \in \mathbf{X}_h, (x - x_i) \cdot \hat{n}_i < 0 \right\} \tag{5.2}$$

In this way we obtain a mesh conforming to the boundary of Ω. We denote the faces of the cell F_{ij}. To generate this diagram, we use the Voro++ library [133]. Figure 5.1 presents the Voronoi mesh generated by the process for a unit square. To facilitate the calculation of integrals, each cell is partitioned into triangles so that integration can be performed by mapping to a reference element. We then calculate and store the moments associated with each cell and face (Figure 5.2). Consider a vector consisting of the elements of a basis for m^{th} order vector polynomials

$$\mathbf{P}(x) = \left\{\vec{\phi}_1(x), \ldots, \vec{\phi}_Q(x)\right\} \tag{5.3}$$

where $Q = dim(\pi_m(\mathbf{R}^d))$ and $\pi_m(\mathbf{R}^d) = span(\vec{\phi}_1, \ldots, \vec{\phi}_Q)$. We then associate with each cell the vector of volume moments

$$\mathbf{V}_i = \left\{\int_{C_i} \vec{\phi}_\alpha dV\right\}_{\alpha=1,\ldots,Q} \tag{5.4}$$

and with each face the vector of area moments

$$\mathbf{A}_{ij} = \left\{\int_{F_{ij}} \vec{\phi}_\alpha \cdot d\vec{A}\right\}_{\alpha=1,\ldots,Q}, \tag{5.5}$$

noting that $\mathbf{A}_{ij} = \mathbf{A}_{ji}$.

We can then write any polynomial $\vec{p} \in \pi_m(\mathbf{R}^d)$ in terms of our basis as $\vec{p} = \mathbf{c}^\intercal \mathbf{P}$ so that we obtain the following discrete integration by parts formula

$$\int_{C_i} \nabla \cdot \vec{p} dV = \mathbf{c}^\intercal \mathbf{D} \mathbf{V}_i = \mathbf{c}^\intercal \sum_{F_{ij} \in \partial C_i} \mathbf{A}_{ij} = \sum_{F_{ij} \in \partial C_i} \int_{F_{ij}} \vec{p} \cdot d\vec{A} \tag{5.6}$$

where $\mathbf{D}$ is a $Q \times Q$ matrix mapping the vector polynomials 5.3 to their derivatives. In order to couple this with a moving least squares reconstruction, we consider the

conservation law

$$\partial_t \vec{u} + \nabla \cdot \vec{\mathbf{F}}(u) = 0 \tag{5.7}$$

where $\vec{\mathbf{F}}$ is a linear flux. While in general for compressible gas dynamics it is necessary to consider the case of non-linear fluxes, we restrict ourselves to viscous flows for which the governing equations are completely linear.

For each cell, we define the following local reconstruction of the vector field

$$\vec{u}_h(x; C_i) = \mathbf{c}_u^\intercal \mathbf{P}(x) \tag{5.8}$$

where $\mathbf{c}_u$ is obtained as the solution to the standard MLS problem

$$\mathbf{c}_u = \arg\min_{c \in \mathbb{R}^Q} \sum_j (c^\intercal \mathbf{P}(x_j) - u(x_j))^2 W_{ij} \tag{5.9}$$

Finally, this allows us to approximate the conservation law in each cell in a standard finite volume approach, i.e. for each cell we obtain an equation

$$\frac{d}{dt} \int_{C_i} \vec{u}_h(x; C_i) dV = - \sum_{F_{ij} \in \partial C_i} \int_{F_{ij}} \nabla \cdot \vec{\mathbf{F}}_h(u_h) \tag{5.10}$$

where $\vec{\mathbf{F}}_h$ is a numerical flux evaluated by either centered difference (i.e. evaluating $\frac{\mathbf{c}_F^i + \mathbf{c}_F^j}{2}$ at the face F_{ij}) or by using the coefficients of the upwind cell. For the sake of brevity the details of this calculation are omitted, but we note that ultimately we arrive at an expression for each particle of the form

$$\frac{d}{dt} \mathbf{V}_i^\intercal \mathbf{c}_u = - \sum_{F_{ij} \in \partial C_i} \mathbf{A}^\intercal \mathbf{c}_F \tag{5.11}$$

As the coefficients have the form

$$\mathbf{u} = \sum_j \alpha_{ij} u_j \tag{5.12}$$

for some α_{ij} obtained as in the previous chapter, we see that Equation 5.11 gives finite difference-like formulae. We plot a representative stencil in Figure 5.3. A detailed investigation of the accuracy of this approach is omitted here, as we only discuss this scheme to motivate the next section. We do briefly demonstrate the high order convergence of the method by evaluating the average truncation error for the flux term when evaluated for the smooth solution $u = \sin x \sin y$. In Figure 5.4 we present the convergence of the truncation error using centered difference reconstruction of the polynomial basis at the face. We also compare the resulting truncation error if the following alternative polynomial reconstruction is performed, whereby we pose an additional moving least squares problem centered at the face centroid, x_{ij}.

$$\mathbf{c}_F = \underset{c \in \mathbb{R}^Q}{\arg\min} \sum_j (c^\intercal \mathbf{P}(x_j) - u(x_j))^2 W(||x_j - x_{ij}||) \tag{5.13}$$

The centered-difference approximation is substantially more efficient than solving this additional optimization problem at each face, and we see that for this problem, both approaches provide nearly identical behavior. For the remainder of this chapter we restrict ourselves to centered-difference flux reconstruction. We recall from Figure 4.2, that for the collocation scheme odd-ordered polynomial reconstruction does not provide an additional order of accuracy in convergence rate. In Figure 5.4 we observe similar behavior, but for even-ordered reconstruction.

Figure 5.1: Voronoi mesh generated for points on the unit square.

When we solve the Poisson problem with Dirichlet data and a smooth solution,

$$\nabla^2 p = -2 \sin x \sin y \tag{5.14}$$

$$p|_{\partial\Omega} = \sin x \sin y, \tag{5.15}$$

we obtain an extra order of convergence as compared to the collocation discretization outlined in the previous chapter (5.6). The finite volume scheme is clearly conservative due to the anti-symmetry of the flux terms, but for a given resolution and order of convergence, the collocation approach is able to obtain more accurate results.

5.2 Graphical finite volume MLS method

In this section, we seek to define a scheme for conservation laws using a graph instead of a mesh. While Voronoi diagrams are costly to compute, it is trivial to compute the ϵ-neighborhood graph of a point cloud. If we can construct a discrete summation

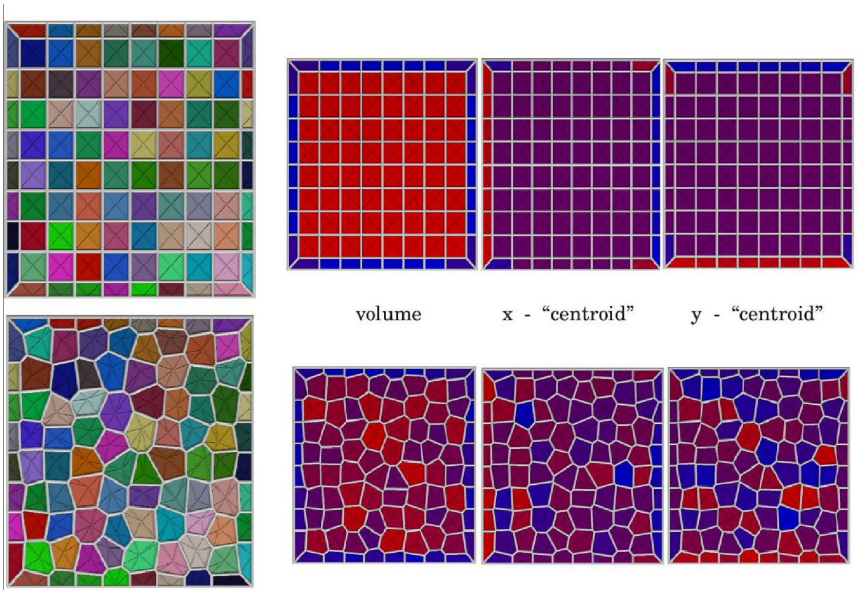


Figure 5.2: Calculation of first order moments for Cartesian (top) and randomly perturbed (bottom) particle arrangements.

by parts formula that replicates the Gauss divergence theorem, this will change the expensive geometric problem of dynamic mesh motion into an inexpensive problem of updating neighbor lists at every time step. To do this, we seek a meshless divergence operator mimicking the algebraic form of Equation 5.11. We first summarize a previous attempt to use optimization to define such an operator, and then present a scheme extending their approach and some simple 1D examples.

5.2.1 Previous work

The scheme presented in the following section is a generalization of the work by Chiu [30] and Katz [76]. We will briefly outline the key elements of their approach here before proceding to define a new approach.

In [30], Chiu et al. seek to define a discrete summation by parts operator by seeking second order stencils for general point clouds of the form

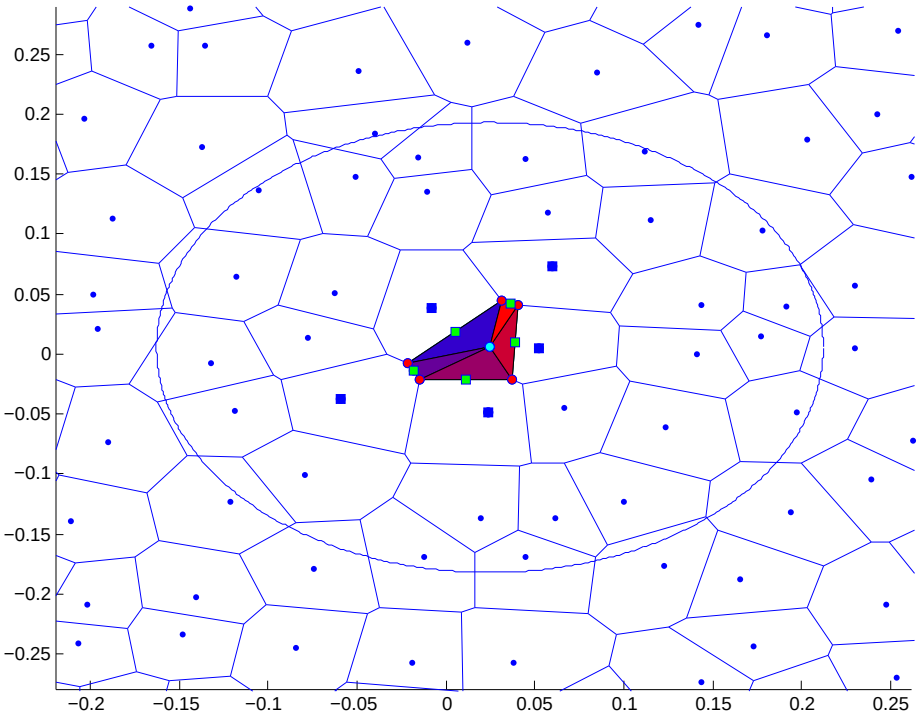


Figure 5.3: Stencil for a given Voronoi cell V_i. Reconstruction involves all points within ϵ-ball of x_i, but fluxes are only exchanged with neighboring cells.

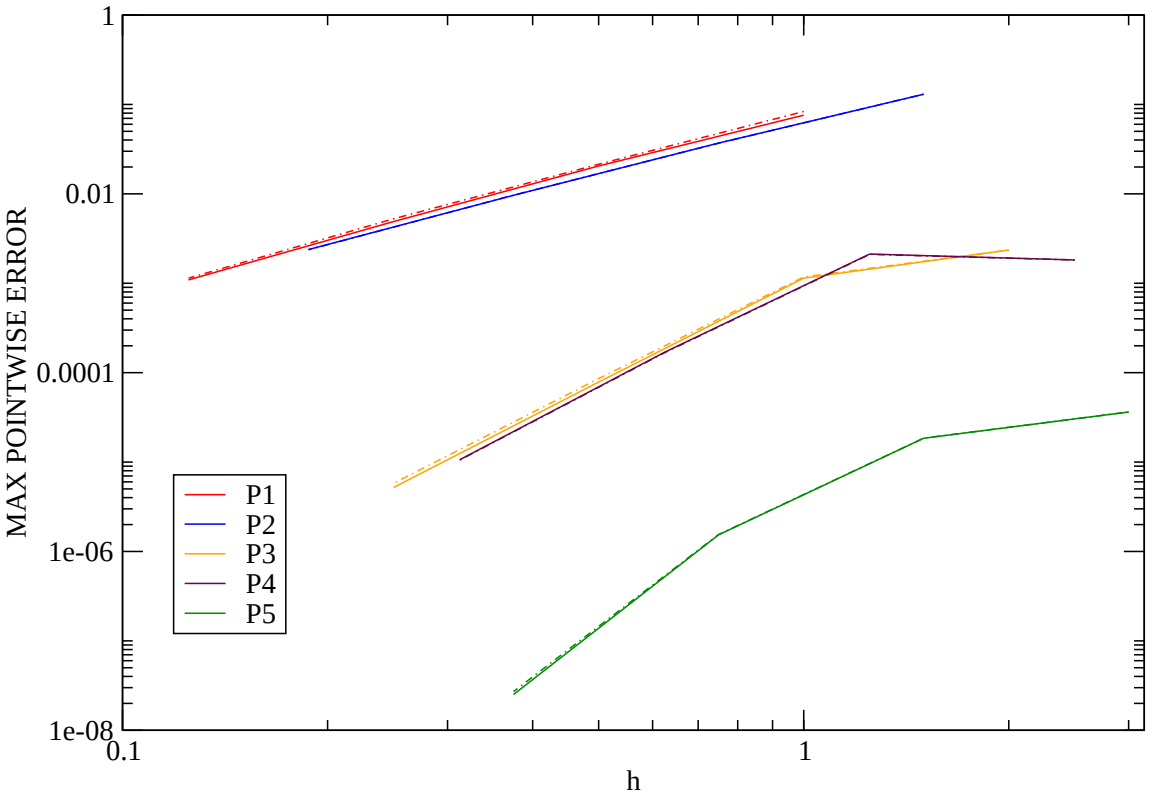


Figure 5.4: Convergence of reconstructed gradient fluxes $F = -\nabla\phi$ at cell faces for increasing order reconstruction. Fluxes calculated using centered-difference reconstruction (dashed lines) or an additional MLS reconstruction (solid lines) provide equivalent results.

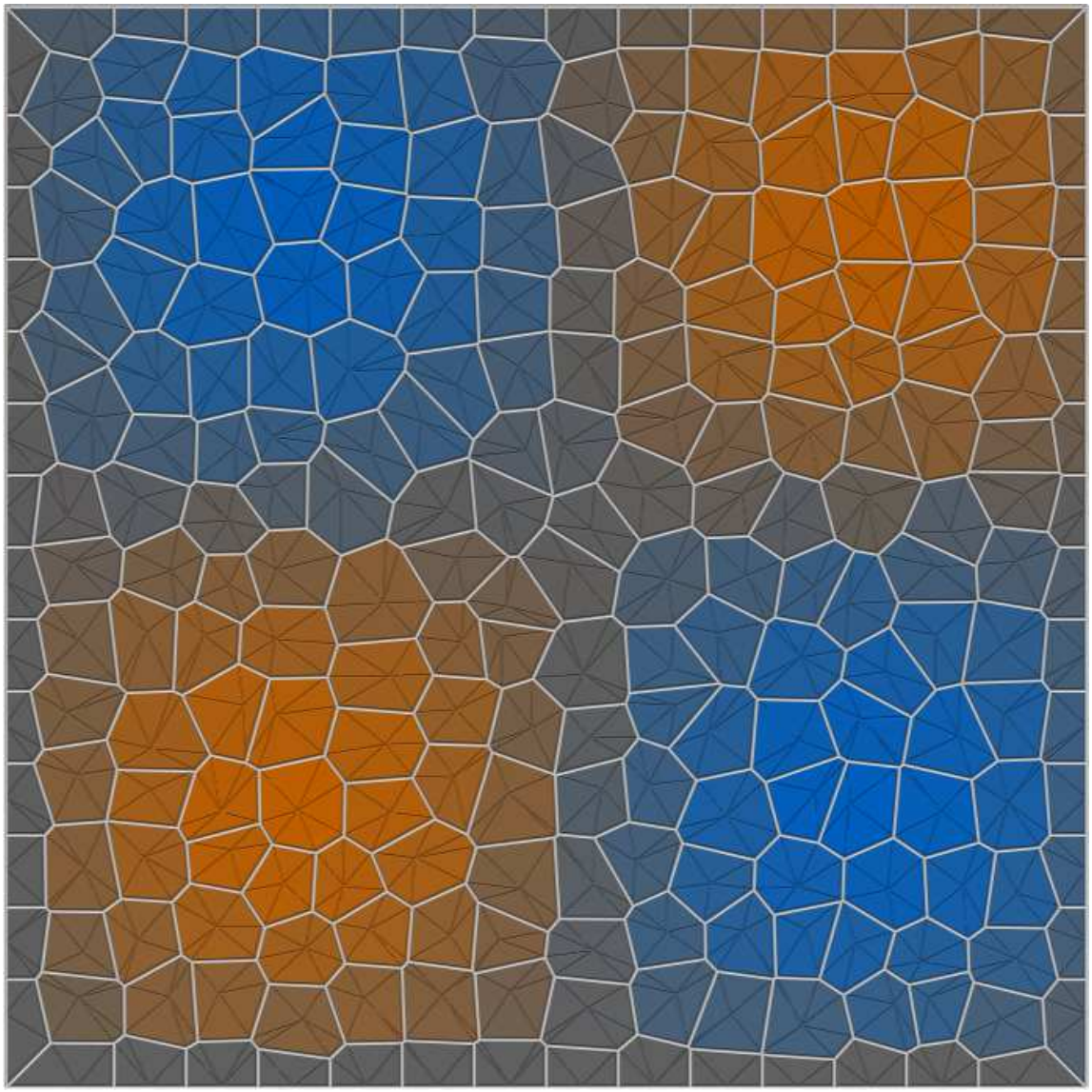


Figure 5.5: Numerical solution to Poisson problem (Equation 5.15). Refer to digital version for color plot.

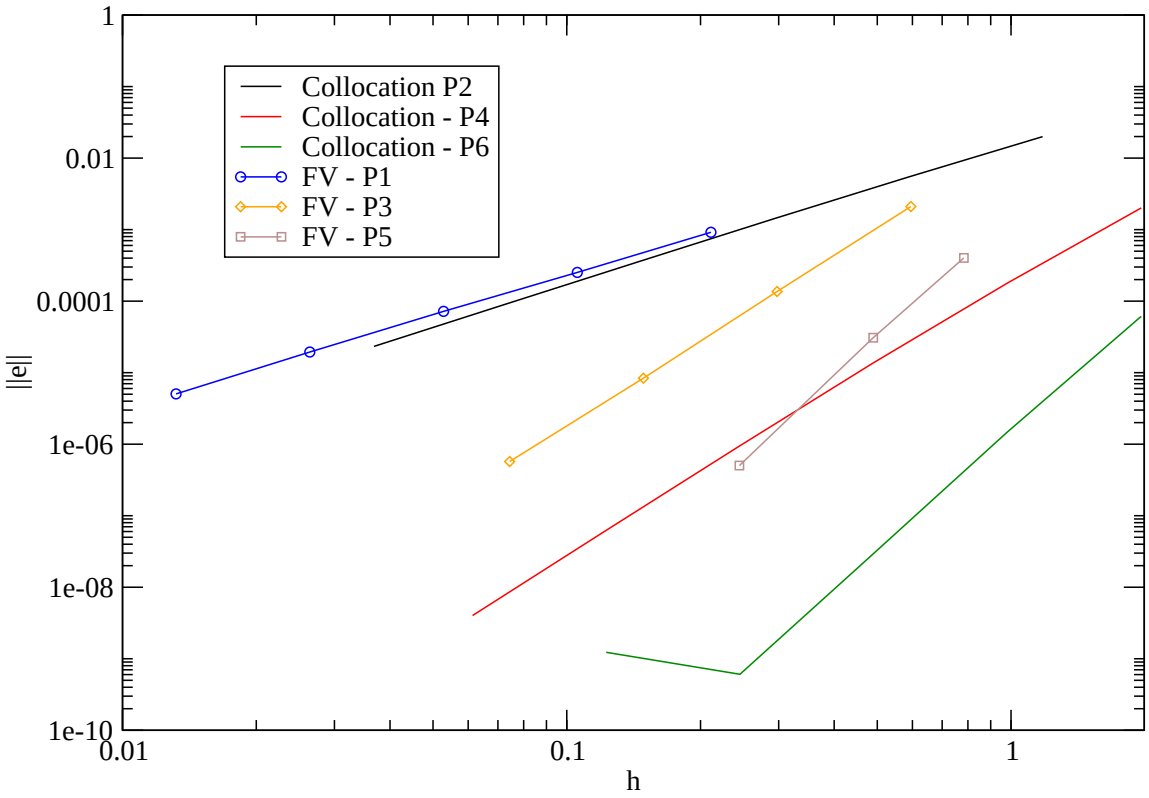


Figure 5.6: RMS error comparison for numerical solution of Equation 5.15 using finite volume scheme (Equation 5.11) and the collocation scheme from Section 4.5. Error saturation for sixth order reconstruction corresponds to tolerance in linear solver of 10^{-9}.

$$V_i(\nabla \cdot \vec{u})_i = \sum_j \vec{A}_{ij} \cdot \vec{u}_{ij} \tag{5.16}$$

where $V_i > 0$ and $A_{ij} = -A_{ji}$ are unknown weights, and $u_{ij} = u(\frac{x_i + x_j}{2})$. If such a formula holds, then the divergence operator is conservative in the diagonal norm $V = diag(V_1, \ldots, V_N)$, i.e.

$$\sum_{i=1}^{N} V_i \nabla \cdot \vec{u}_i = \sum_{i,j} \vec{A}_{ij} \cdot \vec{u}_{ij} \tag{5.17}$$

$$= \frac{1}{2} \left(\sum_{i,j} \vec{A}_{ij} \cdot \vec{u}_{ij} + \sum_{j,i} \vec{A}_{ji} \cdot \vec{u}_{ji} \right) \tag{5.18}$$

$$= \frac{1}{2} \left(\sum_{i,j} \vec{A}_{ij} \cdot \vec{u}_{ij} - \sum_{i,j} \vec{A}_{ij} \cdot \vec{u}_{ij} \right) \tag{5.19}$$

$$= 0 \tag{5.20}$$

In order to find such weights, they pose an ℓ_2 optimization problem

$$min \sum_{A_{ij}} |A_{ij}|^2 + V_i^2 \tag{5.21}$$

and use constraints to enforce symmetry, positivity of weights, and consistency of Equation 5.16 for linear polynomials. Unlike the ℓ_2 problem in moving least squares, Equation 5.21 is very large, involving unknowns and several constraints for every node and edge of the graph. This is because they attempt to simultaneously solve for both the graph topology and the polynomial reconstruction. We demonstrate in the following section that it is sufficient to solve for virtual areas only, which implicitly define a virtual volume through a Stokes theorem. MLS can then be used to reconstruct fluxes in as a separate inexpensive problem.

5.2.2 Construction of graph topology from edge weights

Given a set of points $\mathbf{X}_h = \{x_i\}$, we construct a set of edges

$$\mathbf{E}_{h,\epsilon} = \{(i,j), s.t. ||x_i - x_j|| < \epsilon\}\,. \tag{5.22}$$

We seek a set of vector valued edge weights A_{ij} associated with the midpoint of each edge (i.e. the point $x_{ij} = \frac{x_i + x_j}{2}$) satisfying the properties:

$$A_{ij} = -A_{ji} \tag{5.23}$$

and for fixed i,

$$\sum_j A_{ij} = 0 \tag{5.24}$$

We propose the ansatz that $A_{ij} = a_{ij} W_{ij} \frac{xij}{||x_{ij}||}$, where a_{ij} is an unknown scalar, W_{ij} is a compactly supported kernel function to smoothly weight contributions locally, and $\frac{x_{ij}}{||x_{ij}||}$ is the unit vector pointing from node i to the neighboring edge j. To determine the values of a_{ij} satisfying the above constraints, we solve the following optimization problem:

$$min \sum_{A_{ij} \in \mathbf{E}_{h,\epsilon}} |A_{ij}|^2 \tag{5.25}$$

under the constraints that for all i,

$$\sum_j A_{ij} = 0 \tag{5.26}$$

where the reciprocity constraint is implicitly satisfied by the fact that the ansatz for A_{ij} is anti-symmetric in i and j.

We associate with each x_i unknown virtual volumes V_i and barycenters $\bar{x}_i$. We will use the set of edge weights given by the solution to the optimization problem (5.25) and (5.26) to determine compatible values for these such that a discrete summation by parts formula will hold.

The particle barycenter is defined by analogy to the relation for a polygonal cell:

$$\bar{x}_i = \frac{\sum_j x_{ij}|A_{ij}|}{\sum_k |A_{ik}|} \tag{5.27}$$

and the volume associated with the cell is defined by analogy with the Gauss divergence theorem applied to the vector function $\vec{x}$:

$$V_i = \sum_j x_{ij} A_{ij} \tag{5.28}$$

We seek to follow the procedure in the classical finite volume method where cell-averages of functions are associated with the barycenter of each cell. To do this, we associate particle average degrees of freedom $\bar{u}_i$, and associate them with the barycenters $\bar{x}_i$.

We can then define the discrete divergence operator:

$$\mathbf{D}u_i = \frac{1}{V_i} \sum_j A_{ij} u(x_{ij}) \tag{5.29}$$

where $u(x_{ij})$ is determined through a standard reconstruction procedure. We again

follow the finite volume procedure, but use a moving least squares reconstruction from data at particle barycenters. For example, we first locally define an ansatz for the reconstruction associated with each particle i: $u_i(x) = \bar{u}_i + \nabla u_i \cdot (x - \bar{x}_i)$, where we seek an expression for ∇u_i that is consistent for linear polynomial functions. This can be determined through a MLS fit of surrounding $\bar{u}_j$ located at barycenters. Finally, at the point x_{ij} we evaluate the convex combination $u(x_{ij}) = \theta u_i(x_{ij}) + (1-\theta) u_j(x_{ij})$. Taking $\theta = \frac{1}{2}$ we obtain a midpoint flux limiter; alternatively for convection problems, θ can be chosed to be one in the upwind direction and zero downwind to obtain an upwind limiter. It is trivial to demonstrate that for a globally linear function this reconstruction procedure and divergence operator is exact. Further, due to the symmetry in the reconstruction at x_{ij} and the anti-symmetry in the edge weights, this divergence operator is conservative in the sense that $\sum_i \mathbf{D} u_i = 0$ for a periodic domain.

5.2.3 1D Results

To demonstrate the convergence properties and stability of the discretization, we solve the 1D advection diffusion equation

$$u_t + \nabla \cdot F(u) = 0 \tag{5.30}$$

$$F(u) = au - \nu \nabla u \tag{5.31}$$

$$u(-\pi) = u(\pi) \tag{5.32}$$

To discretize first in space, we calculate $F(u)$ at barycenters as

$$F(u)_i = a\bar{u}_i + \nu \nabla u_i \tag{5.33}$$

where ∇u_i is calculated using a standard MLS construction from neighboring barycenter data. The discrete divergence operator is then applied with a third order Adams-Bashforth (AB3) scheme to obtain:

$$\frac{\gamma_0 \bar{u}^{n+1} - \sum \alpha_k \bar{u}^{n-k}}{\Delta t} = -\sum_k \beta_k \mathbf{D} F(u)_i^{n-k} \tag{5.34}$$

where $\gamma_0, \{\alpha_k\}$, and $\{\beta_k\}$ are the coefficients of the AB3 scheme.

The following plots demonstrate the solution for the above equation for the initial condition $u(x,0) = sin(x)$, using the collocation method for MLS using a quadratic basis (Figure 5.7) and the conservative graph based method using a linear basis and midpoint fluxes (Figure 5.8).

For both uniform and perturbed node distributions, second order convergence is obtained for the smooth solution (Figure 5.9). For discontinuous initial data and zero viscosity, we see that upwinding stabilizes the solution in the standard way in Figure 5.10.

5.3 Conclusions

In this chapter we have presented the necessary components to achieve conservation statements from a meshless scheme, demonstrating the need to generate discrete summation-by-parts formulae with volume and area weights that maintain polynomial consistency and antisymmetry. In the first section we demonstrate that this

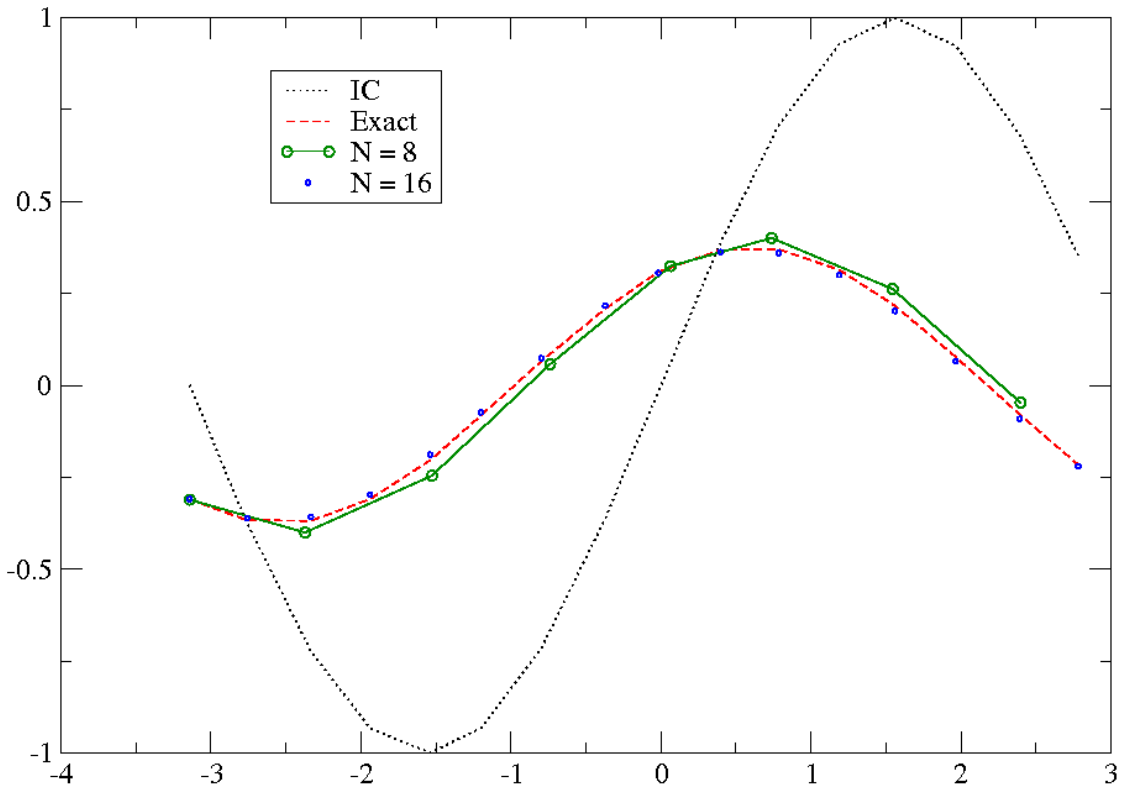


Figure 5.7: Smooth solution to 1D advection-diffusion equation using collocated MLS.

can be achieved by calculating the Voronoi diagram associated with a point cloud to derive weights directly from moments of cells and faces. We then demonstrate that this can alternatively be achieved abstractly on a graph by identifying nodes and edges of an ϵ-graph with the cells and faces of the Voronoi diagram, respectively. In fact, for certain particle arrangements, these edges coincide with the edges of the Delaunay triangulation dual to the Voronoi mesh, suggesting a possible connection for these schemes via the Hodge star operator [18]. For the purposes of this work however, both approaches presented in this section are too slow. For the Voronoi cell case, although $O(N)$ in implementation, the cost associated with the generation of the Voronoi cells was several times larger than the time to solve the resulting linear system of equations. For the graphical case, even after reducing the problem to only solving for area weights, the optimization program is global and requires

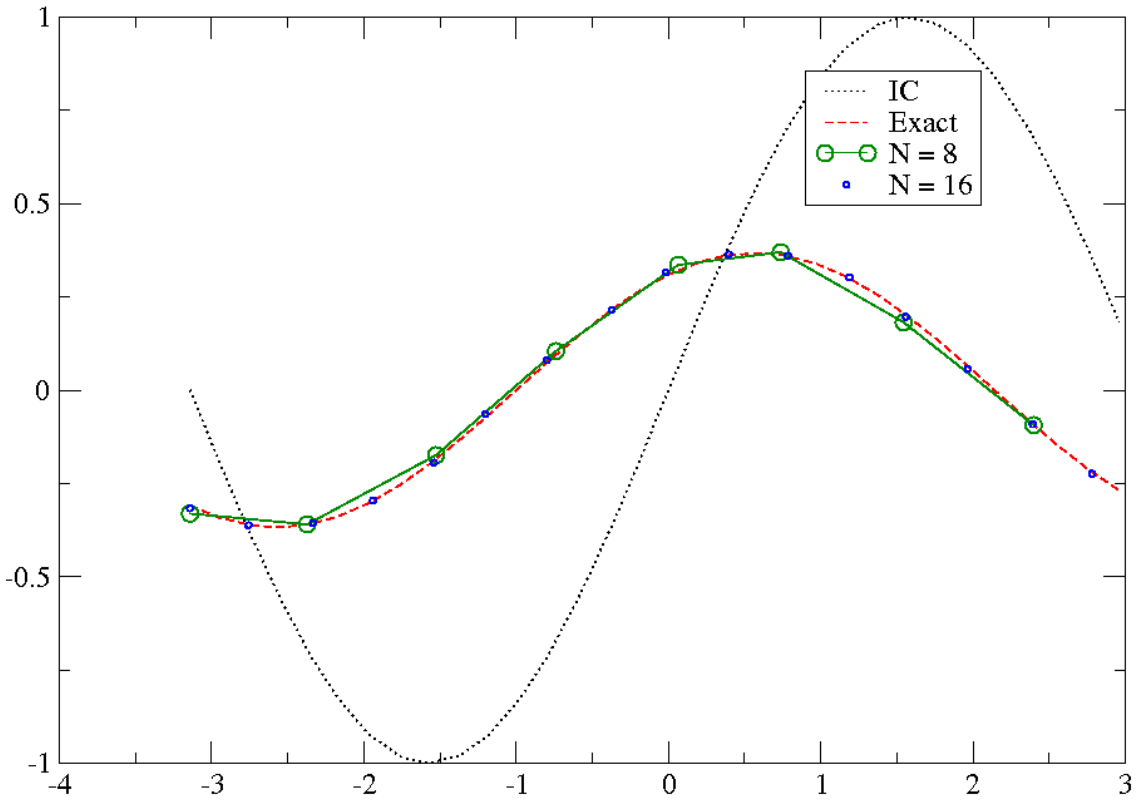


Figure 5.8: Smooth solution to 1D advection-diffusion equation using graphical finite volume method.

$O(N^2)$ methods for solving. Unless a local method can be posed to construct edge weights, it is unlikely that these approaches will lead to a tractible method. In the next chapter however, we pursue the same graph/mesh analogy to construct a new scheme mimicking primal/dual discretizations on staggered grids. In this case, similar to marker and cell finite difference schemes, additional stability can be achieved by mimicking the domain and range of finite volume schemes but without explicitly calculating cell volumes or areas. This will allow us to bypass the question of how to efficiently come up with compatible area weights efficiently.

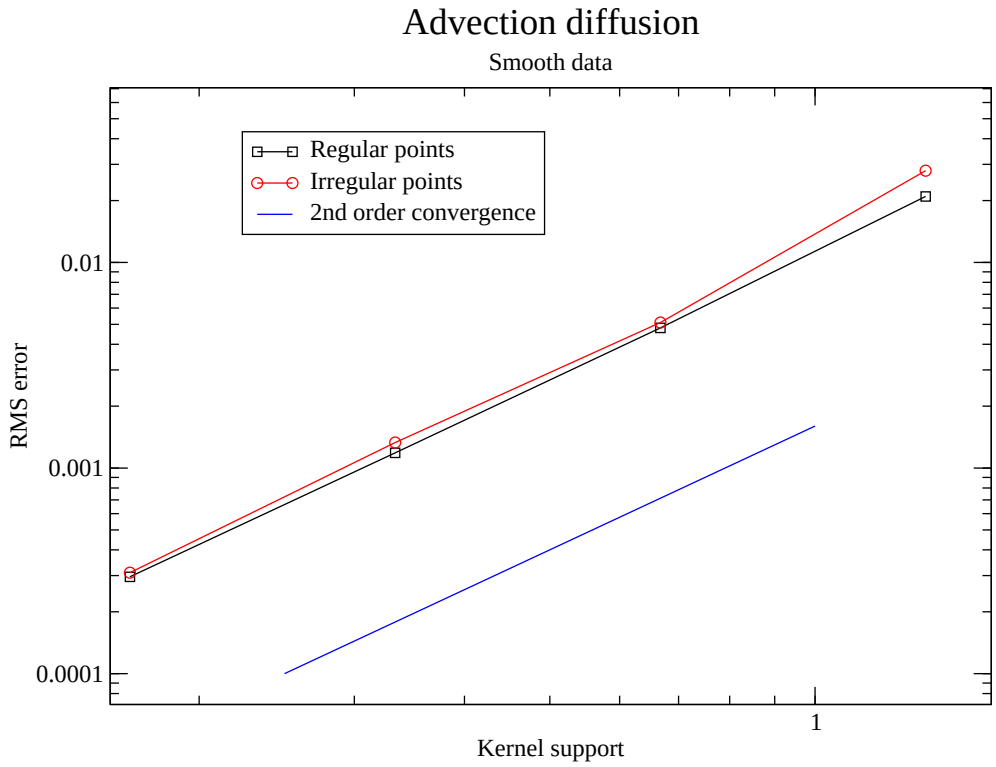


Figure 5.9: Second order convergence of graphical finite volume method for smooth solutions

.

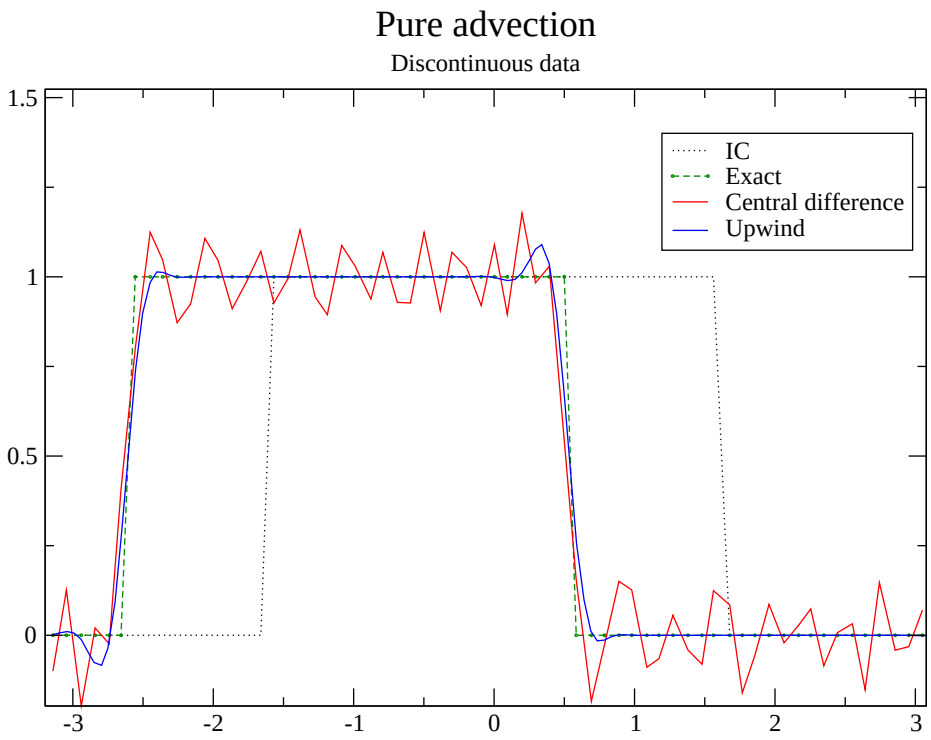


Figure 5.10: For discontinuous solutions, graphical finite volume approach can be stabilized in standard way with upwinding.

Chapter Six

Staggered moving least squares: Generalizing marker-and-cell schemes to meshless methods

In this chapter, we pursue the graphical analogue to conservation laws presented in the previous chapter to develop a new scheme for the model div-grad problem that generalizes marker-and-cell (MAC) type finite difference methods from Cartesian geometries to general pointsets. Like MAC methods, this scheme will rely heavily on the exterior calculus structures of the div-grad problem but will not require the explicit calculation of volume and area weights that were shown to be prohibitively expensive in the previous chapter. Similar to compatible methods for finite elements (e.g. Nedelec elements), we will prove that this scheme provides equal high-order convergence in both an ℓ_2 and a discrete H_1 norm. In addition to this enhanced accuracy, we will demonstrate that this approach also inherits the compatibility typical of $H(div)$ conforming elements by studying a Darcy flow problem with discontinuous coefficients. We then use this discretization to solve electrostatics problems for dielectric materials with jumps in coefficients of several orders of magnitude, and finally demonstrate that this scheme resolves the issues with divergence errors presented in Chapter 4. In the chapter that follows, we then use this stable discretization of the saddle-point system as the foundation for a new pseudo-divergence free, mixed discretization for solving the Stokes equations.

6.1 Model div/grad problem and existing approaches

We consider a new compatible meshless discretization for the model elliptic boundary value problem

$$\begin{aligned} -\nabla \cdot \kappa \nabla \phi &= f \quad \text{in } \Omega \\ \phi &= 0 \quad \text{on } \Gamma \end{aligned} \tag{6.1}$$

and the associated first-order system

$$\begin{aligned} \nabla \cdot \boldsymbol{u} &= f \quad \text{in } \Omega \\ \boldsymbol{u} + \kappa \nabla \phi &= 0 \quad \text{in } \Omega \\ \phi &= 0 \quad \text{on } \Gamma \end{aligned} \tag{6.2}$$

In (6.1) Ω is a bounded region in $\mathbb{R}^d$, $d = 2, 3$ with a piecewise Lipschitz continuous boundary $\Gamma = \partial\Omega$, κ is a symmetric positive definite tensor describing a material property and f is a given source term.

In important applications such as porous media flow[167], heat transfer [146] and semiconductor devices; see, e.g. [23], the flux $\boldsymbol{u} = -\kappa\nabla\phi$ is the variable of primary interest. Predictive simulations of such problems depend on accurate and locally conservative approximation of this variable, which requires divergence-compatible discretizations, i.e., methods that satisfy some form of a discrete Gauss divergence theorem.

Generally speaking, such *compatible discretizations* of (6.1), resp. (6.2) fall into one of the following two categories. *Single grid* methods such as mixed finite elements [22], virtual elements [10] or mimetic finite differences [89][139] construct an approximation of one of the two operators (typically the divergence) on the given grid and then use its adjoint[1] as a proxy for the other operator (the gradient). In so doing these methods satisfy a discrete version of Green's theorem.

In contrast, *primal-dual grid* approaches, such as box integration [80], covolume [113, 114], or finite volume [107] schemes use the two grids in the primal-dual complex

[1]Integration by parts in the second equation of (6.2) accomplishes this construction in mixed finite elements. Other methods, such as mimetic finite differences define the gradient as the algebraic adjoint of the discrete divergence.

to approximate the divergence and the gradient independently. When the two grids are topologically dual, such as in the case of Voronoi-Delauney triangulations or rectilinear[2] primal grids, these methods assume a particularly simple and elegant form.

The abundance of mesh-based methods that provide accurate and locally conservative flux approximations stands in stark contrast with the almost complete absence of divergence-compatible meshless methods, i.e., meshless methods that satisfy some discrete version of the Gauss Divergence theorem. Few notable exceptions are the uncertain grid method [36] and the conceptually similar meshless volume schemes [76, 30] for conservation laws. In the previous chapter we have shown this is due to the necessity of solving a global problem for virtual edge areas. As shown in the first few chapters of this thesis, SPH inherits a notion of div-grad compatibility similar to mimetic finite difference methods, but at the expense of a lack of zeroth order consistency.

In what follows, we pursue an analogy to primal-dual grid methods where divergence and gradient operators map between the ϵ-ball graph of point cloud neighbor connectivity and its dual. Our approach is truly meshless because it only requires this easily computable graph of nearby neighbor connectivity. The vertices and the edges of this graph define a local primal grid for every point. With this grid we associate a virtual, local dual grid comprising of a virtual cell dual to $\boldsymbol{x}_i$ and a set of virtual faces dual to the edges of the connectivity graph. The dual grid is virtual in the sense that the method does not require its physical construction and specification of the dual cell volumes or the dual face areas is implicit in the method's construction. As in primal-dual grid methods we use our local grid complex to define

[2]In this case the dual grid is a translation of the primal grid by a half cell size. This creates the appearance of all variables living on the same grid but at different locations and explains the term "staggered grid methods" that is often used in reference to such schemes.

independent discretizations of the gradient and divergence operators as follows. A node-to-edge incidence matrix scaled by the primal edge length defines a gradient operator on the local primal mesh. A MLS reconstruction of the normal component of a vector field from data on the virtual dual faces yields a notion of a divergence operator on the virtual dual grid. In so doing we obtain discrete divergence and gradient operators which provably reproduce polynomials of arbitrary orders and can be combined to obtain a meshless approximation of the div-grad operator. Because of the conceptual similarity of our new method with primal-dual grid methods we call it a *staggered MLS* discretization of (6.1).

In the discussion that follows, we omit a discussion of standard MLS, as it has been included in previous chapters of this thesis.

6.2 An approximation result for staggered reconstruction

In standard finite volume methods, Rhie-Chow interpolation [48] is often applied to non-staggered meshes to approximate flux variables at faces while storing pressure and velocity at cell centers. As a first attempt at pursuing the mesh/graph analogy from the previous chapter, we identify the midpoints of the edges of the ϵ-graph with face barycenters, and study the reconstruction of polynomials at nodes from these midpoint values. We demonstrate now an approximation result that will allow us to develop a primal/dual graphical discretization while maintaining high-order polynomial reconstruction and a collocated set of primal variables.

Consider a point $\mathbf{x}_i$ and set of neighbors $N_j = \{\mathbf{x}_j, |\mathbf{x_i} - \mathbf{x_j}| < \epsilon\}$ and define

$\mathbf{x}_{ij} = \frac{\mathbf{x}_i + \mathbf{x}_j}{2}$. For a function $u \in C^m(\Omega)$, if the function values were available at the points x_{ij}, then the polynomial reconstruction $p^* \in \pi_m(\mathbf{R}^d)$ solving

$$\min_{p \in \pi_m(\mathbb{R}^d)} \left\{ \sum_{j=1}^{N} \left[u_{ij} - p_{ij} \right]^2 W(||x_i - x_{ij}||) \right\} \tag{6.3}$$

will exactly reproduce polynomials, as will the corresponding diffuse derivative. If the function value u_{ij} is not available however, we can reconstruct this value with a simple average $u_{ij} \approx \frac{u_i + u_j}{2}$ while preserving high-order convergence.

Theorem 5. *For P^m-unisolvent $\{x_{ij}\}$ and $u \in \pi_m(\mathbb{R}^d)$, the reconstruction $p^* \in \pi_m(\mathbb{R}^d)$ solving*

$$\min_{p \in \pi_m(\mathbb{R}^d)} \left\{ \sum_{j=1}^{N} \left[\frac{u_i + u_j}{2} - p_{ij} \right]^2 W(||x_i - x_{ij}||) \right\}$$

exactly reproduces u at the point x_i, i.e.

$$p_i = u_i$$

and for $|\alpha| = 1$,

$$D^\alpha p_i = D^\alpha u_i$$

Proof. Any $u \in \pi_m(\mathbb{R}^d)$ has a Taylor series expansion about x_i given by

$$u(x) = u(x_i) + \sum_{|\alpha|=1}^{M} D^\alpha u(x_i) \phi_\alpha(x)$$

Using the scaling property

$$\phi_\alpha(x_{ij}) = \left(\frac{x_j - x_i}{2} \right)^\alpha = 2^{-|\alpha|} (x_j - x_i)^\alpha = 2^{-|\alpha|} \phi_\alpha(x_j) \tag{6.4}$$

of the Taylor monomials it follows that

$$\begin{aligned}\frac{u(x_i)+u(x_j)}{2} &= u(x_i)+\frac{1}{2}\sum_{|\alpha|=1}^{M} D^{\alpha}u(x_i)\phi_{\alpha}(x_j)\\ &= u(x_i)+\sum_{|\alpha|=1}^{M} 2^{|\alpha|-1}D^{\alpha}u(x_i)\phi_{\alpha}(x_{ij}).\end{aligned}$$

As a result, the weighted least-squares problem

$$\min_{p\in\pi_m(\mathbb{R}^d)}\left\{\sum_{j=1}^{N}\left[\frac{u_i+u_j}{2}-p_{ij}\right]^2 W(||x_i-x_{ij}||)\right\}$$

assumes the form

$$\min_{p\in\pi_m(\mathbb{R}^d)}\left\{\sum_{j=1}^{N}\left[u(x_i)+\sum_{|\alpha|=1}^{M}2^{|\alpha|-1}D^{\alpha}u(x_i)\phi_{\alpha}(x_{ij})-p_{ij}\right]^2 W(||x_i-x_{ij}||)\right\}.$$

By the polynomial reproduction property of Equation 6.3, this problem has a solution

$$p^*(x)=u(x_i)+\sum_{|\alpha|=1}^{M}2^{|\alpha|-1}D^{\alpha}u(x_i)\phi_{\alpha}(x).$$

Because the Taylor monomials vanish at x_i it follows that $p^*(x_i)=u(x_i)$. On the other hand, since

$$D^{\beta}\phi_{\alpha}(x_i)=\delta_{\alpha\beta}|\alpha|! \tag{6.5}$$

where δ_{ij} is the Kronecker delta, we have that

$$D^{\beta}p^*(x_i):=D^{\beta}p^*(x)\big|_{x_i}=\sum_{|\alpha|=1}^{M}2^{|\alpha|-1}D^{\alpha}u(x_i)D^{\beta}\phi_{\alpha}(x)\big|_{x_i}=2^{|\beta|-1}|\beta|!D^{\beta}u(x_i)$$

Therefore, $D^{\beta}p^*(x_i)=D^{\beta}u(x_i)$ for $|\beta|=1$.

This result is unexpected in the sense that the use of an average to reconstruct

the midpoint values is only accurate to second order, but if this reconstruction is evaluated at the nodes high order approximation is maintained. This will allow a high-order staggered arrangement of operators where fluxes are reconstructed at faces from nodal velocities through simple averages.

Remark 1. *Approximating u in $\mathbf{x}_{ij}$ with the average value of $u(\mathbf{x}_{ij}) \approx \frac{1}{2}u_j + \frac{1}{2}u_i$ is equivalent to evaluating the auxiliary function*

$$v^i := \frac{1}{2}u(2\mathbf{x} - \mathbf{x}_i) + \frac{1}{2}u(\mathbf{x}_i)$$

in $\mathbf{x}_{ij}$, in fact $v^i(\mathbf{x}_{ij}) = \frac{1}{2}u_j + \frac{1}{2}u_i$. This means that at each i the polynomial p^* of 6.3 is an approximation of v^i, rather than of u. When u is a linear function, $v^i \equiv u$, however, for general functions v^i can be significantly different than u. For example, if $u = |\mathbf{x}|^2$ and $\mathbf{x}_i = \mathbf{0}$, then $v^i = 2|\mathbf{x}|^2$. The reason why the method works is that u and v^i have the same value and first derivatives at $\mathbf{x}_i$:

$$v^i(\mathbf{x}_i) = \frac{1}{2}u(\mathbf{x}_i) + \frac{1}{2}u(\mathbf{x}_i) = u(\mathbf{x}_i), \qquad \nabla v^i\big|_{\mathbf{x}=\mathbf{x}_i} = \frac{1}{2}2\,\nabla u|_{\mathbf{x}=\mathbf{x}_i} = \nabla u|_{\mathbf{x}=\mathbf{x}_i}$$

Higher-order derivatives of u and v^i are not in general equal in x_i.

However, one can reverse the relation between v^i and u and get

$$u(\mathbf{x}) = 2v^i\left(\frac{1}{2}\mathbf{x} + \frac{1}{2}\mathbf{x}_i\right) - u(\mathbf{x}_i) \approx 2p^*\left(\frac{1}{2}\mathbf{x} + \frac{1}{2}\mathbf{x}_i\right) - p^*(\mathbf{x}_i),$$

and

$$D^\beta u(\mathbf{x}) \approx 2^{(1-|\beta|)}\, D_h^\beta p^*\left(\frac{1}{2}\mathbf{x} + \frac{1}{2}\mathbf{x}_i\right).$$

Guided by this approximation result, we see that flux quantities can be reconstructed at edge midpoints without contaminating high-order convergence. In the

following section, we pursue this strategy to develop compatible divergence and gradient operators.

6.3 Staggered moving least squares method

Given a set of nodes $\mathbf{N} = \{x_i\}_{i=1,\ldots,N} \subset \Omega$, we build a set of edges

$$\mathbf{E} = \{e_{ij} s.t. ||x_i - x_j|| < \epsilon\}. \tag{6.6}$$

We construct an analogy to a mesh of Ω by associating the graph $\Omega_h = G(\mathbf{N}, \mathbf{E})$ with corresponding components of the simplicial complex. We associate with each node $\mathbf{N_i}$ and point x_i a virtual cell $\mathbf{C}_i$ and with each edge and point x_{ij} the virtual face $\mathbf{F}_{ij}$, with associated normal $\hat{n}_{ij} = \frac{x_j - x_i}{||x_j - x_i||}$. As illustrated in Figure 6.1, we see that $\Omega_h = \cup C_i$.

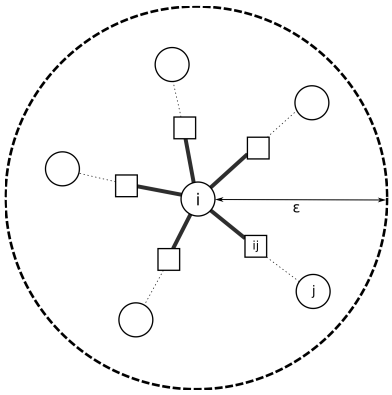


Figure 6.1: A graphical cell $\mathbf{C}_i$ associated with node x_i and the set of edge midpoints lying within a ball of radius $\epsilon/2$.

Mimicking the structure of $H(div)$-compatible discretizations, we seek discrete divergence and gradient operators that map

$$div_h : \mathbf{F} \to \mathbf{C}$$

and

$$grad_h : \mathbf{C} \to \mathbf{F}.$$

We define flux variables $f_{ij} = \vec{u} \cdot \hat{n}_{ij}$ located at faces and build a divergence operator by first using the MLS approach to reconstruct a vector polynomial $\vec{p}^*$ from flux variables

$$\min_{\vec{p} \in (\pi_m(\mathbb{R}^d))^d} \left\{ \sum_{j=1}^{N} \left[f_{ij} - \vec{p}_{ij} \cdot \hat{n}_{ij} \right]^2 W(||x_i - x_{ij}||) \right\} \tag{6.7}$$

and utilizing the diffuse derivative concept to take derivatives, we approximate the divergence at $\mathbf{C}_i$ as $div_h(\vec{u})_i = \nabla \cdot \vec{p}^*$.

To evaluate gradients at cell faces, we apply a simple centered difference along the edges of the graph to obtain directional derivatives

$$grad_h(p)_{ij} \cdot \hat{n}_{ij} = \frac{p_j - p_i}{||x_j - x_i||} \tag{6.8}$$

A mixed approximation to the Laplacian can then be formed by solving

$$-div_h(u) = f \tag{6.9}$$

$$u + \kappa grad_h(p) = 0 \tag{6.10}$$

By taking the composition of these operators at each node, we will ultimately obtain a single equation for the mixed Laplacian at each node that resembles staggered

finite difference methods.

$$div_h \circ \kappa grad_h p(x_i) = f(x_i) \tag{6.11}$$

Although approximating the gradient operator with a centered difference along the edge is only second order accurate at the point x_{ij}, we will demonstrate by an extension of Theorem 5 that this approach maintains high order polynomial consistency for constant κ. In the following, we first prove the accuracy of this approach and then provide details of its implementation.

We define also for convenience the following reconstruction operators when face variables are needed at cells and vice-versa

$$I_c : \mathbf{F} \to \mathbf{C}$$

and

$$I_f : \mathbf{C} \to \mathbf{F}$$

where $I_c \vec{u}_i := \vec{p}_i^*$ is reconstructed from fluxes

$$\min_{\vec{p} \in (\pi_m(\mathbb{R}^d))^d} \left\{ \sum_{j=1}^{N} \left[\frac{\vec{u}_i + \vec{u}_j}{2} \cdot \hat{n}_{ij} - \vec{p}_{ij} \cdot \hat{n}_{ij} \right]^2 W(||x_i - x_{ij}||) \right\} \tag{6.12}$$

and

$$I_f p_{ij} = \frac{p_i + p_j}{2} \hat{n}_{ij}. \tag{6.13}$$

6.3.1 Polynomial consistency of staggered discretization

So as to prove the consistency of this approach and some mimetic properties, we note that when reconstructing gradient fluxes $f_{ij} = \frac{\phi_j - \phi_i}{||x_j - x_i||}$ in Equation 6.7, we can absorb, for analysis purposes, the r_{ij} term into a rescaled kernel by noting that

$$\min_{\vec{p}\in(\pi_m(\mathbb{R}^d))^d} \left\{ \sum_{j=1}^{N} \left[\frac{(\phi_j - \phi_i)}{||x_j - x_i||} - \vec{p}_{ij} \cdot \frac{x_j - x_i}{||x_j - x_i||} \right]^2 W(||x_i - x_{ij}||) \right\} \tag{6.14}$$

is equivalent to

$$\min_{\vec{p}\in(\pi_m(\mathbb{R}^d))^d} \left\{ \sum_{j=1}^{N} \left[(\phi_j - \phi_i) - \vec{p}_{ij} \cdot (x_j - x_i) \right]^2 \widetilde{W}(||x_i - x_{ij}||) \right\} \tag{6.15}$$

where $\widetilde{W}(||x_i - x_{ij}||) := \frac{W(||x_i - x_{ij}||)}{||x_j - x_i||^2}$. This reformulation allows a simple analysis in terms of polynomial rather than rational functions. This rescaled kernel maintains all of the necessary properties to not affect the optimization problem: symmetry in i and j, preserves positivity, and remains finite provided $x_i \neq x_j$. In what follows we drop the superscript $\widetilde{W} = W$ and assume that all points are separated by a minimum finite distance.

Theorem 6. *For P^{m+1}-unisolvent $\{x_{ij}\}$, and $f_{ij} = \phi(\mathbf{x}_j) - \phi(\mathbf{x}_i)$, with $\phi \in \pi_{m+1}(\mathbb{R}^d)$, the non-unique reconstruction $\mathbf{p}^* \in \left[\pi_m(\mathbb{R}^d)\right]^d$ solving*

$$\min_{\mathbf{p}\in\left[\pi_m(\mathbb{R}^d)\right]^d} \left\{ \sum_{j=1}^{N} \left[f_{ij} - \mathbf{p}(\mathbf{x}_{ij}) \cdot (\mathbf{x}_j - \mathbf{x}_i) \right]^2 W(|\mathbf{x}_i - \mathbf{x}_{ij}|) \right\} \tag{6.16}$$

satisfies $\mathbf{p}^*(\mathbf{x}_i) = \nabla\phi(\mathbf{x}_i)$, $\nabla \times \mathbf{p}^*(\mathbf{x}_i) = \mathbf{0}$ and $\operatorname{div} \mathbf{p}_i^*(\mathbf{x}_i) = \Delta\phi(\mathbf{x}_i)$.

Proof. Let $\Pi_i^m := \left\{ \mathbf{p}_i(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{x}_i),\, \mathbf{p}_i(\mathbf{x}) \in \left[\pi_m(\mathbb{R}^d)\right]^d \right\}$. Clearly $\Pi_i^m \subset \pi_{m+1}(\mathbb{R}^d)$, and every polynomial in Π_i^m can be uniquely reconstructed knowing its values at $\{\mathbf{x}_{ij}\}$. Following an argument analogous to that used in Remark 1, we define an auxiliary function

$$g_i(\mathbf{x}) = \phi(2\mathbf{x} - \mathbf{x}_i) - \phi(\mathbf{x}_i)$$

and note that $g_i(\mathbf{x}_{ij}) = \phi(\mathbf{x}_j) - \phi(\mathbf{x}_i) = f_{ij}$. We express g_i using a Taylor series at $\mathbf{x} = \mathbf{x}_i$

$$g_i(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_i)^T \nabla g_i(\mathbf{x}_i) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T D^2 g_i(\mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i) + \ldots.. \tag{6.17}$$

and observe that $g_i \in \Pi_i^m$.

We look for $p_i^* \in \Pi_i^m$ as the solution of

$$\min_{p \in \Pi_i^m} \left\{ \sum_{j=1}^{N} \left[g_i(\mathbf{x}_{ij}) - 2\, p(\mathbf{x}_{ij}) \right]^2 W(|\mathbf{x}_i - \mathbf{x}_{ij}|) \right\}.$$

Using standard arguments of optimization theory (following e.g. the proof of theorem 4.3 in [159]) it can be shown that there exists a unique $p_i^* \in \Pi_i^m$. Also, because $g_i \in \Pi_i^m$ we have $p_i^* = \frac{1}{2} g_i$.

Any vector polynomial $\mathbf{p}_i^*$, such that $\mathbf{p}_i^*(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{x}_i) = p_i^*(\mathbf{x})$ is a argmin of (6.16), in fact

$$\mathbf{p}_i^*(\mathbf{x}_{ij}) \cdot (\mathbf{x}_j - \mathbf{x}_i) = 2\, \mathbf{p}_i^*(\mathbf{x}_{ij}) \cdot (\mathbf{x}_{ij} - \mathbf{x}_i) = 2\, p_i^*(\mathbf{x}_{ij}) = g_i(\mathbf{x}_{ij}) = f_{ij}.$$

We note that $\mathbf{p}_i^*$ is not unique. In particular, if $\mathbf{p}_i^*$ is a argmin of (6.16) all the other argmins of (6.16) are given by $\mathbf{p}_i^* + \mathbf{p}^0$, with $\mathbf{p}_i^0 \in \left[\pi_m(\mathbb{R}^d)\right]^d$ such that $\mathbf{p}_i^0(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{x}_i) = 0$. We notice, however, that the value of $\mathbf{p}_i^*$ at $\mathbf{x}_i$ is independent of the particular

choice of $\mathbf{p}_i^0(\mathbf{x})$ as $\mathbf{p}_i^0(\mathbf{x}_i) = 0$. This can be seen by writing (using Taylor expansion) $\mathbf{p}_i^0(\mathbf{x}) = \mathbf{p}_i^0(\mathbf{x}_i) + O(|\mathbf{x} - \mathbf{x}_i|)$, hence $\mathbf{p}_i^0(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{x}_i) = \mathbf{p}_i^0(\mathbf{x}_i) \cdot (\mathbf{x} - \mathbf{x}_i) + O(|\mathbf{x} - \mathbf{x}_i|^2)$. Since $\mathbf{p}_i^0(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{x}_i)$ must be identically zero, then $\mathbf{p}_i^0(\mathbf{x}_i) = 0$.

We have $p_i^* = g_i$ and considering the expression for g_i in (6.17) we can pick

$$\mathbf{p}_i^*(\mathbf{x}) = \frac{1}{2}\nabla g_i(\mathbf{x}_i) + O(\mathbf{x} - \mathbf{x}_i) = \nabla\phi(\mathbf{x}_i) + O(|\mathbf{x} - \mathbf{x}_i|).$$

It follows that $\mathbf{p}_i^*(\mathbf{x}_i) = \nabla\phi(\mathbf{x}_i)$. Also, taking the curl of $\mathbf{p}_i^*$ we get

$$\nabla \times \mathbf{p}_i^*(\mathbf{x}) = \nabla \times \left(\frac{1}{4}(\mathbf{x} - \mathbf{x}_i)^T D^2 g_i(\mathbf{x}_i)\right) + O(|\mathbf{x} - \mathbf{x}_i|) = O(|\mathbf{x} - \mathbf{x}_i|).$$

The last equality follows from the fact that the Hessian $D^2 g_i(\mathbf{x}_i)$ is symmetric and hence $\nabla \times \left((\mathbf{x} - \mathbf{x}_i)^T D^2 g_i(\mathbf{x}_i)\right) = \mathbf{0}$. Therefore $\nabla \times \mathbf{p}_i^*(\mathbf{x}_i) = \mathbf{0}$. Similarly, we show that the divergence of $\mathbf{p}$ is the Laplacian of ϕ at point $\mathbf{x}_i$:

$$\operatorname{div} \mathbf{p}_i^*(\mathbf{x}) = \operatorname{div}\left(\frac{1}{4}(\mathbf{x} - \mathbf{x}_i)^T D^2 g_i(\mathbf{x}_i)\right) + O(|\mathbf{x} - \mathbf{x}_i|) = \operatorname{trace}\left(\frac{1}{4} D^2 g_i(\mathbf{x}_i)\right) + O(|\mathbf{x} - \mathbf{x}_i|).$$

Therefore $\operatorname{div} \mathbf{p}_i^*(\mathbf{x}_i) = \operatorname{trace}\left(\frac{1}{4} D^2 g_i(\mathbf{x}_i)\right) = \operatorname{trace}\left(D^2\phi(\mathbf{x}_i)\right) = \Delta\phi(\mathbf{x}_i)$.

This result is significant because it suggests that in some sense, the polynomial reconstruction preserves the mimetic property that $\nabla \times \nabla\phi = 0$ for any ϕ. We note also that Equation 6.15 can be viewed as a discretization of

$$\min_{\vec{p} \in (\pi_m(\mathbb{R}^d))^d} \left\{ \sum_{j=1}^{N} \left[\int_{e_{ij}} (\nabla\phi - \vec{p}) \cdot d\mathbf{s} \right]^2 \widetilde{W}(||x_i - x_{ij}||) \right\} \tag{6.18}$$

where the integral of the first term is evaluated exactly via the fundamental theorem of calculus

$$\int_{e_{ij}} \nabla\phi \cdot d\mathbf{s} = \phi_j - \phi_i \tag{6.19}$$

and the second is discretized with midpoint quadrature

$$\int_{e_{ij}} \vec{p} \cdot d\mathbf{s} \approx \vec{p}(x_{ij}) \cdot \frac{x_j - x_i}{||x_j - x_i||}|e_{ij}| = \vec{p}(x_{ij}) \cdot (x_j - x_i)$$

This interpretation of the method will provides useful intuition when reconciling the high-order convergence of the method with the expected second order convergence of midpoint quadrature. Although in its discretized form the method appears to be approximating gradients using a centered difference, in this interpretation we see that the gradient is actually treated exactly in an integral sense.

6.4 Implementation details

Since the reconstruction process is posed in terms of an ℓ_2-optimization, the polynomial reconstruction in Equation 6.7 has an explicit solution. While some work has been done achieving more compact positive stencils by switching to an ℓ_1 formulation [135, 137], we use the least squares setting here for ease of implementation. The ℓ_2 setting also allows us to ensure reciprocity - two nodes sharing an edge will both have contributions to their stencils, while in ℓ_1 solutions tend to be sparser.

We rephrase the problem by seeking an optimal coefficient vector $\mathbf{c}^*$ such that the optimal polynomial can be expressed as $\vec{\mathbf{p}} = \mathbf{c}^\intercal\vec{\phi}$, where $span(\vec{\phi}) = \left(\pi_m(\mathbb{R}^d)^d\right)$, i.e. poses a basis for vector valued polynomials of order m. The value of $\mathbf{c}^*$ reconstructed from flux components is

$$\mathbf{c}^* = \mathbf{M}_i^\dagger \sum_j \mathbf{P}_{ij} \cdot \hat{n}_{ij} W_{ij} f_{ij} \tag{6.20}$$

where $\mathbf{P}_{ij}$ is a vector whose components are the basis functions evaluated at x_{ij} and $\dagger$ denotes the pseudo-inverse[151] of the weighted normal equation matrix.

$$\mathbf{M}_i = \sum_k \mathbf{P}_{ik} \cdot \hat{n}_{ik} W_{ik} \mathbf{P}_{ik} \cdot \hat{n}_{ik}. \tag{6.21}$$

As shown in the proof of theorem 6, the reconstruction from normal components is not unique, therefore matrix $\mathbf{M}_i$ is rank deficient. In this work we choose to overcome this issue by using the pseudo-inverse. A more efficient approach would be to use only the basis functions $\vec{\phi}$ that are independent in the quotient space $(\pi_m(\mathbb{R}^d)^d \; N_i^d$, $N_i^d := \{\mathbf{p} \in \pi_m(\mathbb{R}^d)^d, \mathbf{p} \cdot (\mathbf{x} - \mathbf{x}_i) = 0\}$. For example, considering the 2D basis

$$\vec{\phi} \in \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x - x_i \\ 0 \end{pmatrix}, \begin{pmatrix} y - y_i \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x - x_i \end{pmatrix}, \begin{pmatrix} 0 \\ y - y_i \end{pmatrix} \right\} \tag{6.22}$$

the vector $\begin{pmatrix} 0 \\ x - x_i \end{pmatrix}$ can be removed from the basis set because is linearly dependent on $\begin{pmatrix} y - y_i \\ 0 \end{pmatrix}$. In fact $\begin{pmatrix} 0 \\ x - x_i \end{pmatrix} = \begin{pmatrix} y - y_i \\ 0 \end{pmatrix} + \begin{pmatrix} -(y - y_i) \\ x - x_i \end{pmatrix}$ and $\begin{pmatrix} -(y - y_i) \\ x - x_i \end{pmatrix} \in N_i^d$.

For the reconstruction to have a solution, there must be enough neighbors within the support of the kernel W to maintain polynomial unisolvency. For the results in this work, we select the support size $\epsilon := supp(W) = (m + 1)\Delta x$ where Δx is the characteristic polynomial spacing. For reconstruction near boundaries where the stencil becomes one-sided this provides sufficient neighbors to maintain unisolvency and results in an overly conservative number of neighbors in the interior; this can easily be remedied by implementing an adaptive kernel radius. In the following

chapter, we will pursue an adaptive kernel radius to attain a multiresolution scheme.

After precomputing the inverses of the weighted normal matrices at each particle, the divergence operator is given by

$$div_h(x_i) = (\nabla \cdot \mathbf{P})_i^{\intercal} \mathbf{c}_i^* \tag{6.23}$$

which, after combining operations gives an expression of the form

$$div_h(x_i) = \sum_j \alpha_{ij} f_{ij} \tag{6.24}$$

for some coefficients a_{ij}. To discretize for example the mixed Poisson problem

$$\nabla \cdot \nabla \phi = f, \tag{6.25}$$

we take the composition of the divergence and gradient operators to assemble the linear system

$$\sum_j \alpha_{ij}(\phi_j - \phi_i) = f_i. \tag{6.26}$$

Boundary conditions are imposed by constraining the polynomial in the least squares reconstruction (Equation 6.7) to exactly satisfy the boundary condition at boundary particles. For an arbitrary boundary condition, we therefore add the constraint that

$$\mathcal{L} p_i = g_i. \tag{6.27}$$

In implementation, this amounts to augmenting the weighted normal matrices to handle an additional degree of freedom for a Lagrange multiplier. The optimal

coefficients are then extracted from the solution of the system $\hat{\mathbf{M}}_i \hat{\mathbf{c}^*}_i = \hat{\mathbf{b}}_i$, where

$$\hat{\mathbf{M}}_i = \begin{bmatrix} \mathbf{M}_i & \mathcal{L}\mathbf{P}_i \\ \mathcal{L}\mathbf{P}_i^\intercal & 0 \end{bmatrix}, \tag{6.28}$$

$$\hat{\mathbf{c}}_i^\intercal = \left[\begin{array}{c|c} \mathbf{c} & \lambda \end{array} \right] \tag{6.29}$$

and

$$\hat{\mathbf{b}}_i^\intercal = \left[\begin{array}{c|c} \sum_j \mathbf{P}_{ij} \cdot \hat{n}_{ij} W_{ij} f_{ij} & g_i \end{array} \right] \tag{6.30}$$

So for example, selecting $\mathcal{L} = \partial_n$ and $g_i = 0$ amounts to imposing homogeneous Neumann boundary conditions. By constraining the reconstruction in this way, the discretization naturally satisfies boundary conditions, and for the example of the mixed Laplacian the discretized problem takes the following form for boundary particles,

$$\sum_j \alpha_{ij}(\phi_j - \phi_i) = f_i + \beta_i g_i. \tag{6.31}$$

Due to the collocation style of discretization of the PDE and the lack of symmetry in particle arrangement, the global matrix that arises when assembling Equations 6.26 and 6.31 is in general not symmetric. Special care must be taken when solving problems with a non-trivial null-space (e.g. the Poisson problem with Neumann boundary conditions). This can be remedied in various ways by: artificially pinning a degree of freedom, adding a global Lagrange multiplier to remove the singularity, or projecting out the kernel if a Krylov subspace method is used to solve the resulting equations. We refer to an in-depth discussion of these issues in previous works for

details[150, 16].

In exact arithmetic, the choice of the polynomial basis is arbitrary and will provide optimal accuracy as long as the basis is complete. In practice however there are several possible choices for stability and performance reasons. Because the Taylor basis has the Kronecker delta property (Equation 6.5), they allow a more efficient evaluation of the divergence since the matrix-vector product $(\nabla \cdot \mathbf{P})_i^\intercal \mathbf{M}_i^\dagger$ occurring in Equation 6.23 can be reduced to a single inner product. This causes the weighted normal matrix to take the form of a Hilbert matrix with unstable condition number when going to high order. To stabilize this a Legendre tensor product basis can be used instead for an increased computational cost. For the results in this work, the Taylor basis was found to be sufficiently stable in double precision to achieve up to a fifth order polynomial space.

6.5 Numerical examples

6.5.1 Case setup

To discretize the domain for all problems in this work, we first discretize the boundary of the domain with a length scale dx and then discretize the interior of the domain by covering the interior with a Cartesian lattice with spacing dx, perturbing each particle by a uniformly distributed random variable of magnitude η, and deleting all particles that lie outside of the domain. The variable η is used to demonstrate insensitivity of the results to particle anisotropy, and for all results presented is set to $\eta = 0.1dx$. We characterize levels of discretization by number of particles in each direction of the lattice $N = \frac{1}{dx}$.

Algebraic multigrid is used to efficiently solve the discretized asymmetric system of equations. While an in-depth discussion of the performance of this preconditioner is left for a future work, we briefly comment that a combination of GMRes and auxilliary space preconditioning [162] provides an efficient means of solution that converges nearly linearly in the degrees of freedom.

6.5.2 Elliptic problems with smooth coefficients

We first demonstrate the convergence rate of the method for smooth solutions by solving the mixed Poisson problem on a annular 2D domain with inner radius $R_o = \frac{\pi}{2}$ and inner radius $R_i = \frac{\pi}{4}$ (Figure 6.2)

$$div_h \cdot grad_h \phi_i = f_i \tag{6.32}$$

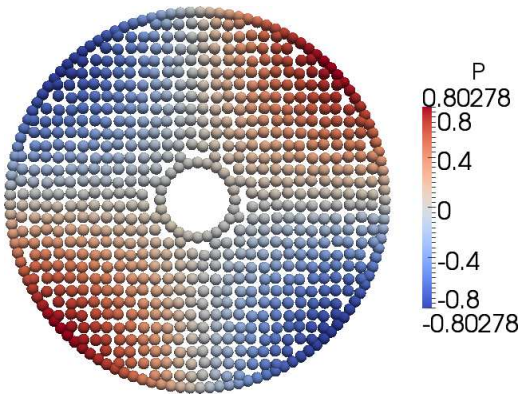


Figure 6.2: Typical particle distribution with $N = 32$ for perforated disk geometry.

We manufacture the right hand side to obtain a smooth exact solution ϕ_{ex}.

$$\phi_{ex} = \sin x \sin y \tag{6.33}$$

i.e.

$$f_i = -2 \sin x_i \sin y_i \tag{6.34}$$

We investigate both Dirichlet and Neumann boundary condition cases and set the boundary condtions to be compatible with the exact solution. As no mesh is available to quantify the error, we measure convergence in a root mean square sense by defining the following norm

$$||f||_{l_2} = \sqrt{\frac{\sum_i f_i^2}{N}} \tag{6.35}$$

and semi-norm

$$||f||_{h^1} = \sqrt{\frac{\sum_i (I_c grad_h f_i)^2}{N}} \tag{6.36}$$

We compare these results against the standard diffuse MLS discretization described in Chapter 4, when used with the Taylor basis and identical kernels. Figure 6.3 demonstrates the convergence in the l_2-norm of the two approaches for increasing refinement and increasing order reconstruction. While the standard MLS discretization converges similarly to classical finite difference methods, the staggered scheme converges with order $m+1$ for an m^{th}-order reconstruction and consistently more accurately than the corresponding standard MLS result of degree $m+1$. This suggests that although it is necessary to do some extra work to construct a vector polynomial basis in the staggered scheme (actually d times as much work for approximation in $\mathbb{R}^d$), the ability to use a lower order reconstruction will amortize that cost.

What is more remarkable is the fact that the h^1-seminorm error scales with the same order of convergence as the l_2-norm error, suggesting empirically that an error bound of the form

$$||u - u_{ex}||_{l^2} + ||u - u_{ex}||_{h^1} \leq Ch^{m+1} \tag{6.37}$$

may hold for some constant C independent of the discretization. This type of bound is typical of mimetic methods or compatible finite element spaces, and suggests that that staggered graph arrangement is in some sense faithful to the exterior calculus of the operators.

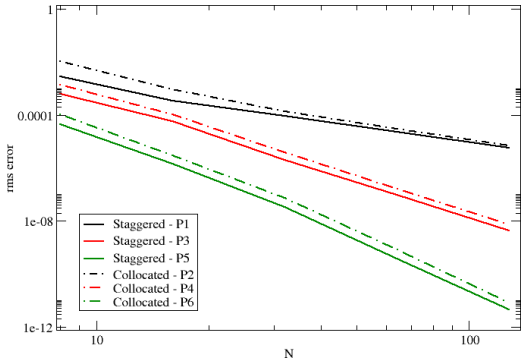


Figure 6.3: Convergence in discrete l_2-norm for manufactured solution with Dirichlet boundary conditions using second, fourth and sixth order standard MLS (dashed lines) and of first, third and fifth order staggered MLS (solid lines). Although using one order lower reconstruction, the staggered results are consistently more accurate.

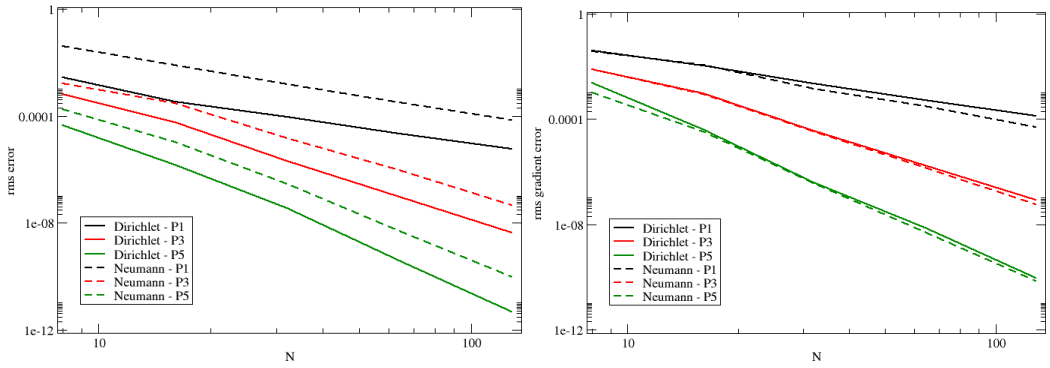


Figure 6.4: Convergence in discrete l_2-norm (left) and h_1-seminorm (right) for manufactured solution with Dirichlet (solid lines) and Neumann (dashed lines) boundary conditions, suggesting a convergence result of $||u - u_{ex}||_{l_2} + ||u - u_{ex}||_{h^1} \leq Ch^{m+1}$ for the staggered scheme.

6.5.3 Elliptic problems with rough coefficients

Motivated by the analogues to other mimetic discretizations, we seek to demonstrate that the current approach is applicable to the class of problems where mimetic methods have also found success. A standard benchmark used to demonstrate compatibility is to solve the equations for Darcy flow

$$\nabla \cdot \mathbf{u} = f \tag{6.38}$$

$$\mathbf{u} + \mu \nabla \phi = 0 \tag{6.39}$$

for discontinuous values of μ. In our staggered framework, we discretize this as

$$\nabla \cdot \mathbf{u}_{ij} = f_i \tag{6.40}$$

$$\mathbf{u}_{ij} + \mu_{ij} \nabla \phi_i = 0 \tag{6.41}$$

where we assume the permeability μ is known at particle locations and we therefore reconstruct $\mu_{ij} = I_f \mu$. We discretize the unit square into strips

$$\Omega_i = \{(x, y) \mid 0.2(i-1) \leq y \leq 0.2i; 0 \leq x \leq 1\} \tag{6.42}$$

and we assign each strip a piecewise constant permeability: $\mu_1 = 16$, $\mu_2 = 6$, $\mu_3 = 1$, $\mu_4 = 10$, and $\mu_5 = 2$. By employing Neumann boundary conditions such that a linear solution is obtained ($\phi_i = 1 - x_i$), we have then a piecewise constant exact solution for the flux $u = \mu$.

This test case has been used to demonstrate incompatibility in the context of

nodal finite elements versus a stabilized discontinuous Galerkin scheme[102] and in the context of nodal least squares finite element methods versus compatible least squares finite element methods [17]. In both cases elements were generated that conformed to the strips and the incompatible discretizations exhibited either spurious oscillations throughout the domain[102] or incompatibility with Neumann boundary conditions [17], while the compatible discretization could reproduce the solution exactly. As illustrated in Figure 6.5, we obtain similar compatible results, but do not require that the particle conform to the boundaries of Ω_i. When the simulation is discretized identically but with the standard collocated version of MLS, spurious oscillations are obtained in the fluxes analagous to [102] (Figure 6.6).

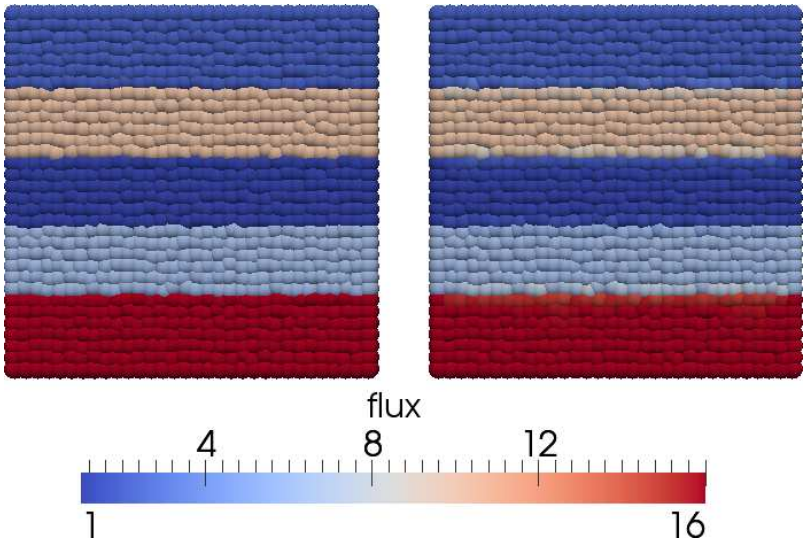


Figure 6.5: Transverse fluxes for the Darcy flow strips case for non-conforming particle arrangement for exact solution (left), staggered (right).

The div-grad system of equations 6.40 can be recast to solve problems in electrostatics by taking $f = 0$ and identifying μ instead as the electric permitivity (usually denoted ϵ). In electrostatics problems involving complex geometries typically the only available tools to handle deforming domains with discontinuous material properties involve either costly arbitrary Lagrangian-Eulerian methods or some diffuse interfacial treatment [98]. For the current approach, particles can be identified as

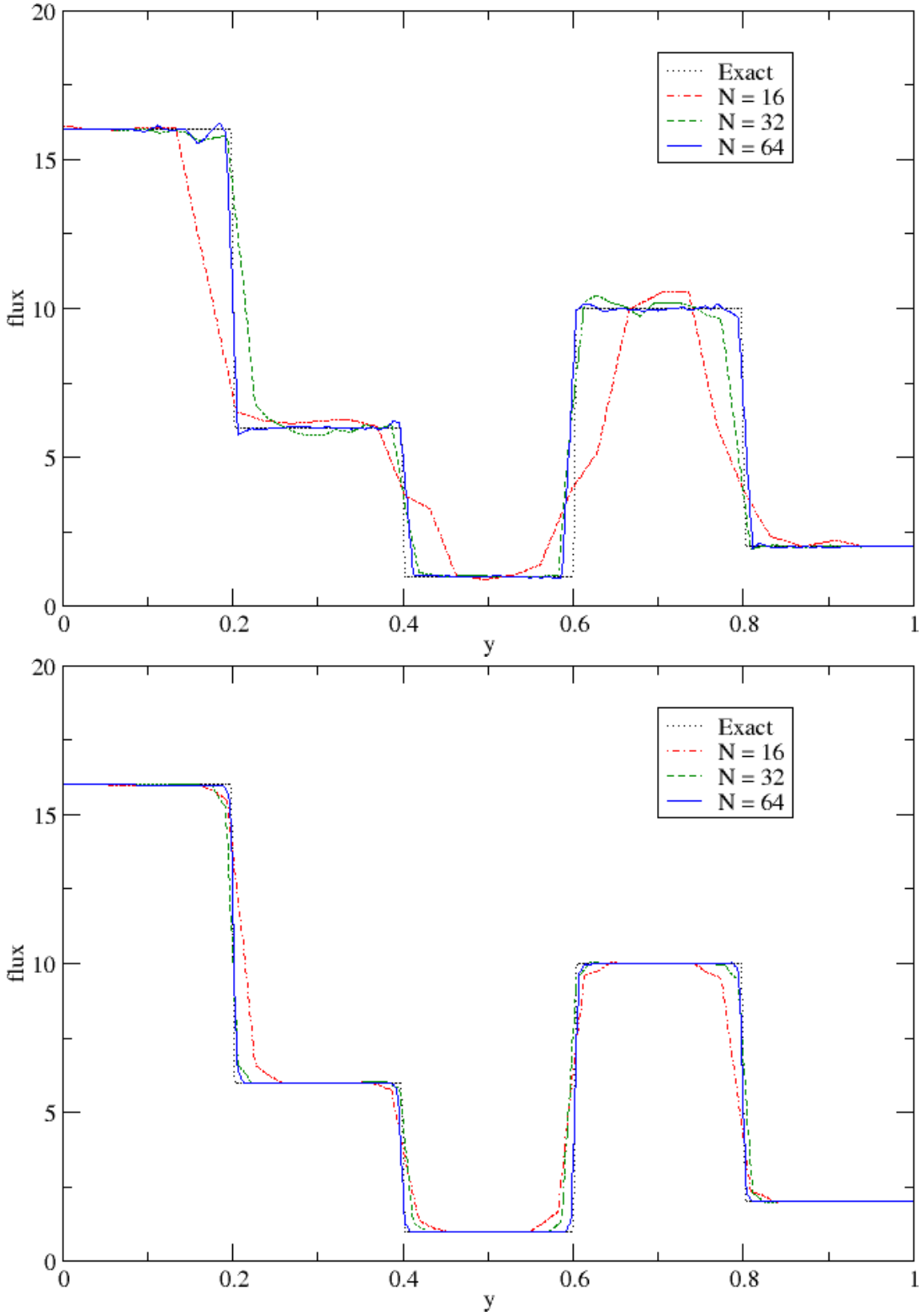


Figure 6.6: Transverse fluxes along the line $x = 0.5$ for the collocated (top) and staggered (bottom) schemes with increasing resolution.

belonging to one phase or the other and μ assigned appropriately. As an example, we simulate a cylinder of radius $R = 1/2$ in a medium with permitivity $\mu/\mu_\infty = 2$ exposed to a uniform electric field $\nabla\phi = \langle 1, 0\rangle$ for which an analytic solution can easily be calculated in an infinite domain. To solve this numerically, we impose the analytic solution as a Dirichlet boundary condition on a unit square and demonstrate convergence in Figure 6.7. This discretization is stable for a ratio of permitivities ranging from $1 - 1000$ (Figure 6.8).

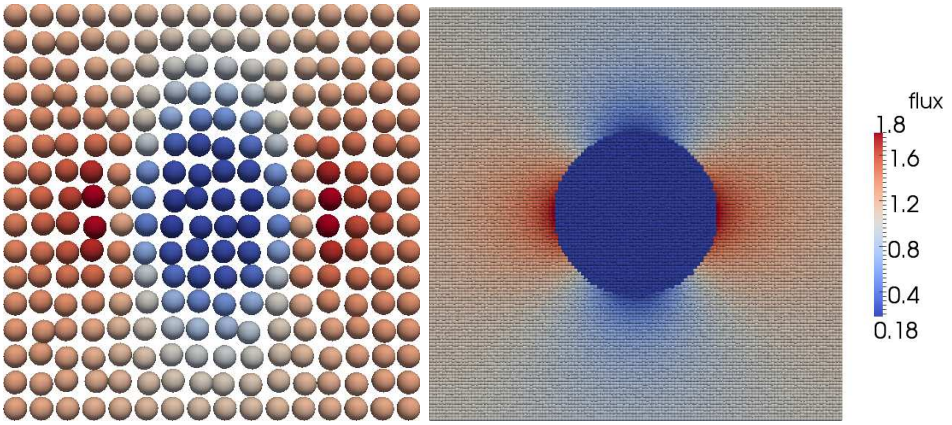


Figure 6.7: Plot of electric flux $\nabla\phi$ for coarsest ($N = 16$) and finest ($N = 256$) resolution.

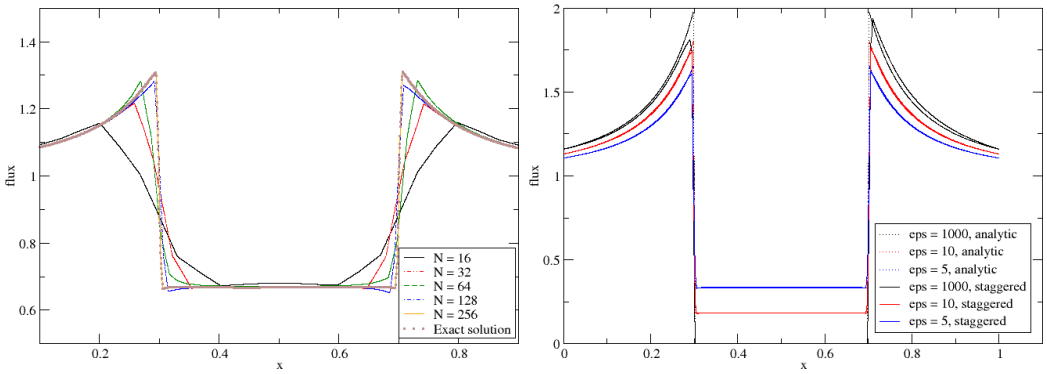


Figure 6.8: Electric flux along center line for increasing resolution (left) and for a range of permitivities at a fixed resolution of $N = 128$ (right).

As an example of the flexibility of the method for complex geometries, we now solve Equation 6.40 where μ is taken by processing an image of the author's cat (Figure 6.9) and assigning permitivities to regions of similar color. Particles are assigned values for μ_i by sampling pixels from the picture, and an approximately

uniform electric field is imposed by setting Dirichlet boundary conditions.

$$\phi_i|_{\partial\Omega} = 1 - x_i \tag{6.43}$$

The robustness of the method in the presence of fluctuations in the permitivity below the resolution lengthscale is shown in the refinement study presented in Figure 6.10. Examining the y-component of the resulting "electric field" for resolution ranging from 32^2 to 1024^2, the results remain consistent and non-oscillatory. Note that using AMG the resulting matrix solve took only a few seconds on a desktop computer. While this is a slightly whimsical demonstration of the capabilities of the method, we hope to motivate that this approach could easily be adopted to any problem in which voxel data is available of the underlying diffusivity (e.g. applications in MRI imaging, geological imaging, etc).

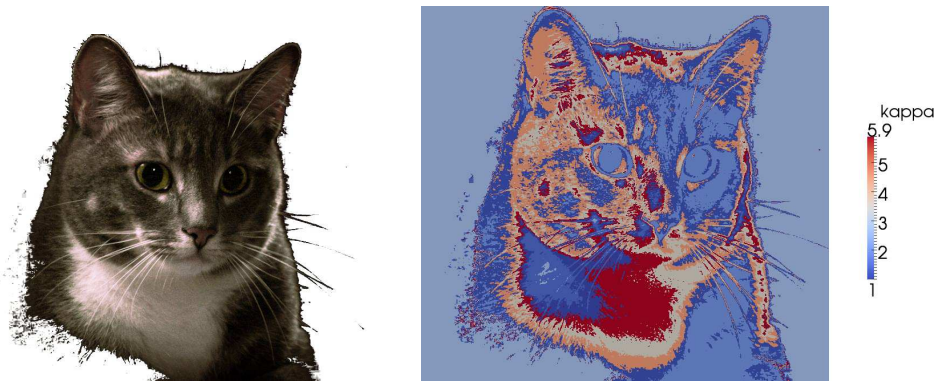


Figure 6.9: Conductivity for the underresolved stability test case.

6.6 Navier-Stokes equations

While we have focused primarily on the solution of the model div-grad problem in this chapter, we briefly demonstrate here that this discretization provides stability when solving the Navier-Stokes equations with a projection method as well. We refer

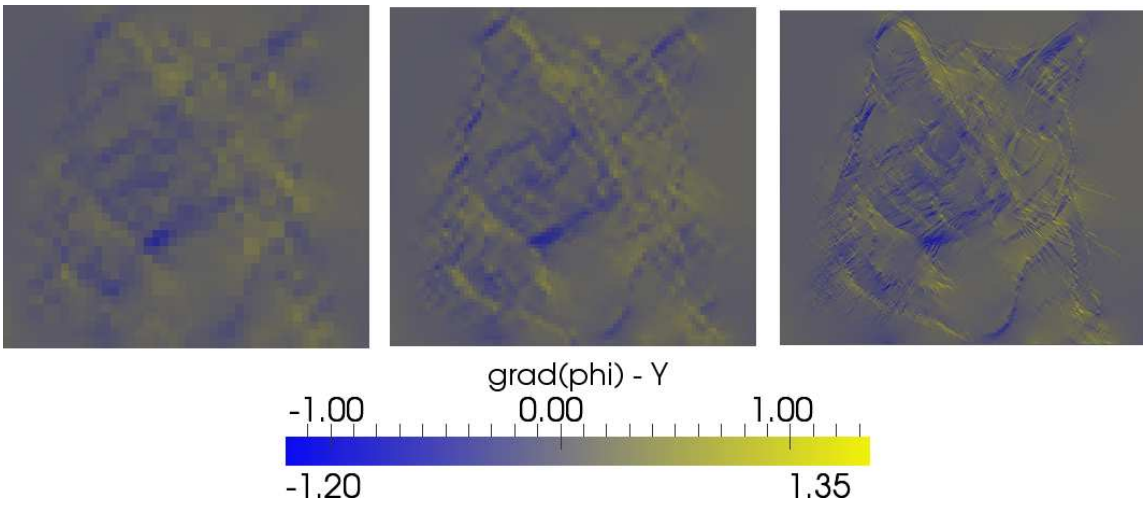


Figure 6.10: Test results to demonstrate stability even with poor resolution. Values of $\partial_y \phi$ for resolutions of $N = 64$ (left), $N = 128$ (center), and $N = 256$ (right). Underresolved cases provide similar results despite highly oscillatory conductivity below resolved lengthscale.

readers unfamiliar with the topic to the review article for governing equations and details [58]. In so-called pressure correction schemes, a provisional velocity field is made divergence-free by performing a Helmholtz-Hodge decomposition through the solution of a Poisson equation. For finite-element spaces that fail to meet an inf-sup condition [20, 94], this scheme is unstable and admits spurious modes in the pressure field that contaminate high-order convergence of the method. In collocated finite-differences[142] and the incompressible smooth particle hydrodynamics method[32], a similar phenomenon occurs in which spurious oscillations appear that must be stabilized either with additional elliptic terms[142] or by performing the projection in an approximate sense[32, 11]. In finite-differences, the introduction of a staggered variable arrangement with the marker-and-cell (MAC) method [103] avoids this and allows for the simulation of high Reynolds number flows. We will briefly demonstrate that the discretization introduced in this paper generalizes the stability of the MAC scheme to arbitrary particle arrangements. This resolves the complications presented in Chapter 4 and provides a promising new avenue for high-order meshless Lagrangian hydrodynamics for high-Reynolds number flows.

We implement the rotational incremental pressure correction scheme developed by Timmermans, Minev and Van de Vosse [148] and discussed in [58, 94]. The unsteady incompressible Navier-Stokes equations:

$$\partial_t \mathbf{u} + \nabla \cdot \mathbf{u}\mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} \tag{6.44}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{6.45}$$

are discretized with a backward difference formula in time and an operator splitting is applied to first solve for a provisional velocity $\tilde{\mathbf{u}}$ satisfying no-slip boundary conditions as

$$\frac{1}{2\Delta t}(3\tilde{\mathbf{u}} - 4\mathbf{u}^k + \mathbf{u}^{k-1}) = -\nu \nabla^2 \tilde{\mathbf{u}} + \nabla p^k = \mathbf{f}^{k+1/2} \tag{6.46}$$

$$\tilde{\mathbf{u}}|_{\partial\Omega} = 0. \tag{6.47}$$

The nonlinear term has been lumped into a body force on the right hand side and is treated explicitly. From this provisional velocity, the divergence-free velocity $\mathbf{u}^{n+1}$ is then calculated as

$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}}{2\Delta t} = -\nabla q \tag{6.48}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \tag{6.49}$$

$$\mathbf{u}^{n+1} \cdot \hat{n}|_{\partial\Omega} = 0 \tag{6.50}$$

Taking the divergence of this equation provides the following Poisson problem with

homogeneous Neumann boundary conditions to enforce zero velocity flux at walls.

$$\nabla^2 q = \frac{\nabla \cdot \tilde{\mathbf{u}}}{2\Delta t} \tag{6.51}$$

$$\partial_n q|_{\partial\Omega} = 0 \tag{6.52}$$

After using this value of q to update the velocity, the pressure at the next timestep is given by

$$p^{k+1} = p^k + q - \nu \nabla \cdot \tilde{\mathbf{u}} \tag{6.53}$$

In order to discretize these equations with the staggered MLS scheme, there are several options depending upon whether the fluxes or the velocities are taken as the primary variables. To minimize the size of the resulting discretized system, we discretize all equations at particle locations as

$$\frac{1}{2\Delta t}(3\tilde{\mathbf{u}}_i - 4\mathbf{u}_i^k + \mathbf{u}_i^{k-1}) = -\nu div_h(grad_h(\tilde{\mathbf{u}}_i)) + I_c grad_h(p_i^k) = \mathbf{f}_i^{k+1/2} \tag{6.54}$$

$$\tilde{\mathbf{u}}_i|_{\partial\Omega} = 0 \tag{6.55}$$

and

$$div_h(grad_h(q))_i = \frac{div_h(\tilde{\mathbf{u}})_i}{2\Delta t}. \tag{6.56}$$

We enforce homogenous Neumann boundary conditions for q. For simplicity, the nonlinear term is treated semi-implicitly as

$$\mathbf{f} = -\mathbf{u}_i^n \cdot I_f grad_h(\mathbf{u}^{n+1})_i. \tag{6.57}$$

More involved conservative or skew-symmetric discretizations are also possible and have been shown to provide better stability[75].

While there are many different schemes for discretizing the Navier-Stokes equations, we have chosen this one as it has been identified in [94] as being sensitive to a lack of compatibility in the pressure and velocity approximation and has no terms for artificially stabilizing the scheme.

To demonstrate the stability of the discretization over a range of Reynolds numbers, we match the benchmark of steady flow in a lid-driven cavity studied by Ghia and Ghia [52] for a range of Reynolds numbers. For this case flow in a unit square is driven by imposing unit velocity along the upper boundary and enforcing no-slip boundary conditions at the other walls. This case has been a standard benchmark for a number of years due to several challenging features of the flow: at all Reynolds numbers there is a singularity in the pressure field in the upper corners, and as the Reynolds number increases the flow is driven by an increasingly thin viscous boundary layer that is difficult to resolve. At high Reynolds number there are several secondary and tertiary vortices that appear in the flow that are difficult to resolve with low-order methods, which introduce artificial numerical dissipation into the scheme. We leave a rigorous investigation of the discretization for a future work, and present the result of running a 128^2 particle discretization with third-order reconstruction at $Re = 100 - 10,000$ to demonstrate the stability of the approach. For $Re = 100$ and $Re = 1000$ we ran the simulation to steady state and present the velocity profiles orthogonal to the horizontal and vertical centerlines of the box in Figure 6.11. These show excellent agreement to the Ghia benchmark. We ran the $Re = 10,000$ case for long enough time that the main vortex is fully developed. Note that for $Re > 7000$ the solution is naturally unsteady and a comparison to the Ghia benchmark is not meaningful. Figure 6.12 demonstrates that the high order resolution is able to resolve secondary Moffat eddies in the bottom left and right corners of the cavity.

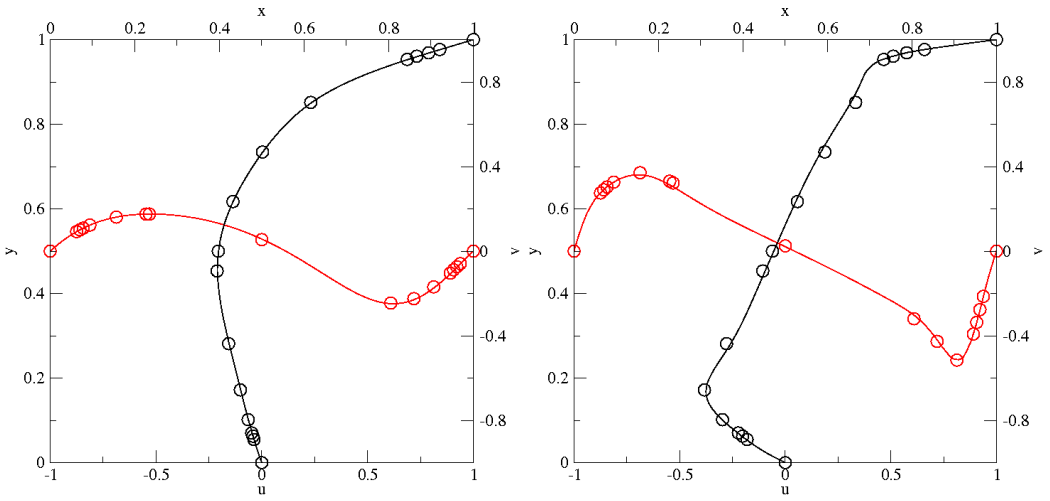


Figure 6.11: Velocity plot along cavity centerlines for Reynolds number 100 (left) and 1000 (right) with comparison to benchmark[52].

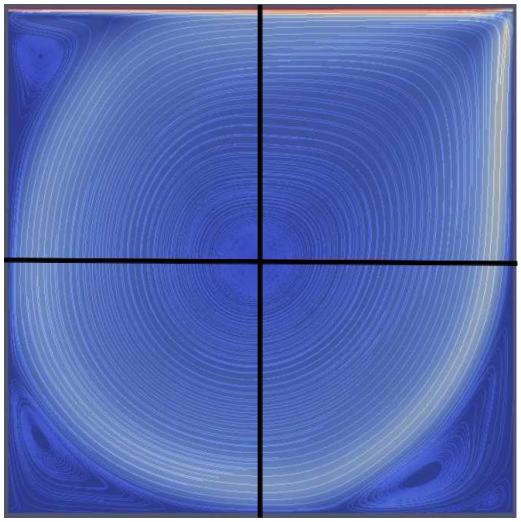


Figure 6.12: Streamlines demonstrating high resolution of secondary eddies for Reynolds number 10000. Black lines denote the location of velocity profiles taken in Figure 6.11.

6.7 Conclusion

In this chapter we have presented a new discretization generalizing the staggered finite difference method to general pointsets. We have demonstrated enhanced accuracy as compared to the previous standard collocation scheme with equal order convergence in both the ℓ_2 and a discrete H^1 norm. Numerical results have confirmed this high-order convergence and shown empirically that the method inherits

the stability of other compatible discretizations for discontinuous elliptic problems. Finally, we have demonstrated how this model problem can be used in a projection scheme for incompressible flow, resolving the compatibility issues noted in Chapter 4. In the next chapter, we will present a mixed method for the Stokes equations in which staggered MLS is used to handle pressure terms and a divergence-free reconstruction is used to approximate viscous terms. The model div-grad problem studied in this chapter will provide the foundation for the stability and accuracy of this mixed approach.

Chapter Seven

Mixed moving least squares: Compatible meshless discretization for Stokes equations with applications to electrophoretic suspensions

With the compatible discretization for the model div-grad problem defined in the previous chapter, we now present a robust framework for stable numerical simulation of Stokes flows. We present a mixed scheme in which we combine a staggered reconstruction for the pressure with a divergence-free reconstruction for the velocity field. We will see that this discretization provides equal order convergence for both the velocity and the pressure, surpassing the typical P^n/P^{n-2} inf-sup conditions that limit accuracy for most discretizations. In mesh based methods this type of convergence is unusual; for meshless methods there are no techniques that achieve stable $O(N)$ efficient approximation of the Stokes equations. In this chapter we will use this enhanced pressure accuracy to present a direct numerical simulation capability for suspension flows that employs a monolithic fully implicit discretization for both the hydrodynamics and the colloid equations of motion. At zero Reynolds number and large volume fractions, these flows are driven by extremely stiff lubrication forces that occur in narrow gaps between colloids. In the scheme that follows we pursue an adaptive resolution scheme that allows resolution of these lubrication forces, and uses a completely implicit formulation so that the increased resolution does not impose stiff stability restrictions. These problems are currently only tractable using one of the following: methods relying on simple particle/domain geometry and the use of some form of Green's function [96, 166, 19, 127]; methods employing a diffuse treatment of colloid boundaries that are restricted to second-order [81, 99, 123, 55]; low-order meshless methods that easily handle complex geometry but use either artificial compressibility approaches or have accuracy limitations[60, 14, 112, 2]; or ALE-type methods with a mesh directly conforming to colloid boundaries but requiring expensive remeshing and small timesteps [66, 65]. Since the lubrication forces scale as the inverse of the lubrication gap radius in three dimensions, most of these methods employ some form of sub-grid scale correction for the lubrication forces so as to avoid computation of these singular forces. We present some cases where there

is an analytic solution and demonstrate this scheme's ability to completely resolve the relevant physics to high order. We then present some flows to demonstrate that this approach extends easily to arbitrary colloid and domain geometry.

We conclude by presenting a modification to this scheme in which electrokinetic effects are incorporated by solving the nonlinear Poisson-Boltzmann equation to model electrostatic interaction between colloids and the domain boundary. The addition of these effects can still be incorporated in a completely implicit scheme, and we present some preliminary results where the flow dynamics are driven by an external electric field.

7.1 Suspension flow governing equations

We consider a domain Ω with piecewise continuous boundary $\partial\Omega$. The domain contains N_c colloidal particles each with boundary $\partial\mathbf{\Omega}_i$. The boundary of the domain can therefore be partitioned $\partial\Omega = \partial\Omega_D \cup (\cup_i \partial\Omega_i)$, where a flow $\mathbf{w}$ will be prescribed along the portion of the boundary $\partial\Omega_D$. Each colloid has position $\mathbf{X}_i$, orientation $\mathbf{\Theta}_i$ and undergoes a solid body rotation characterized by translational and rotational velocities $\mathbf{V}_i$ and $\mathbf{\Theta_i}$. Given a set of particle positions, the flow in Ω is completely specified by the particle positions and assuming steady Stokes flow

$$\begin{cases} -\nu\nabla^2\mathbf{u} + \nabla p = f & \text{if } x \in \Omega \\ \nabla \cdot \mathbf{u} = 0 & \text{if } x \in \Omega \\ \mathbf{u} = \mathbf{w} & \text{if } x \in \partial\Omega_D \\ \mathbf{u} = \mathbf{V}_i + \mathbf{\Omega}_i \times (x - \mathbf{X}_i) & \text{if } x \in \partial\Omega_i, \text{for all } i \end{cases} \tag{7.1}$$

If the velocity is chosen from an appropriate space of solenoidal vector fields, this is equivalent to solving the following pair of equations coupled only through an inhomogeneous Neumann pressure boundary condition.

$$\begin{cases} \nu\nabla\times\nabla\times\mathbf{u}+\nabla p=f & \text{if } x\in\Omega \\ \mathbf{u}=\mathbf{w} & \text{if } x\in\partial\Omega_D \\ \mathbf{u}=\mathbf{V}_i+\boldsymbol{\Omega}_i\times(x-\mathbf{X}_i) & \text{if } x\in\partial\Omega_i, \text{for all } i \end{cases} \tag{7.2}$$

$$\begin{cases} \nabla^2 p=\nabla\cdot\mathbf{f} & \text{if } x\in\Omega \\ \partial_n p+\nu\hat{n}\cdot\nabla\times\nabla\times\mathbf{u}=\hat{n}\cdot f & \text{if } x\in\partial\Omega \end{cases} \tag{7.3}$$

By recasting the Stokes operator in this equivalent form, after a spatial discretization we will ultimately obtain a more computationally tractable block diagonally-dominant system of equations.

The translational and angular velocities for each particle can then either be imposed explicitly or determined by imposing a force/torque-free condition

$$\begin{cases} \int_{\partial\Omega_i}\sigma_V\cdot d\mathbf{A}=0 & \text{for all } i \\ \int_{\partial\Omega_i}(x-\mathbf{X}_i)\times\sigma_V\cdot d\mathbf{A}=0 & \text{for all } i \end{cases} \tag{7.4}$$

where $\sigma_V=-p\mathbf{I}+\frac{\nu}{2}\left(\nabla\mathbf{u}+\nabla\mathbf{u}^\intercal\right)$ is the viscous stress tensor. This force-free and torque-free condition has been selected for this work to match cases in which semi-analytical solutions are available. In practice however, particle inertia can also be

taken into account by instead solving

$$\begin{cases} m_i \ddot{\mathbf{X}}_i = \int_{\partial\Omega_i} \sigma_V \cdot d\mathbf{A} & \text{for all } i \\ I_i \ddot{\mathbf{\Theta}}_i = \int_{\partial\Omega_i} (x - \mathbf{X}_i) \times \sigma_V \cdot d\mathbf{A} & \text{for all } i. \end{cases} \tag{7.5}$$

where m_i and I_i are the mass and moment of inertia, respectively, of particle i.

For the purposes of this work, we will initially specify colloid translations and rotations to demonstrate the high-order convergence of the discretization for some analytic solutions to Stokes equations. We will then solve Equations 7.2,7.3, and 7.4 concurrently as a monolithic system to study the dynamics of colloidal suspensions under the assumption that the flow is completely in equilibrium. Therefore, given a set of colloid states $(\mathbf{X}_i, \mathbf{\Theta}_i)_{i=1,\ldots,N_C}$, the coupled solution of these equations provides the complete dynamics of the problem. If we write the set of colloid degrees of freedom as $\vec{\mathbf{X}} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_c}, \mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_{N_c}\}$, we can concisely describe the dynamics of the colloid motion as

$$\dot{\vec{\mathbf{X}}} = F(\vec{\mathbf{X}}, t) \tag{7.6}$$

where F denotes the solution of the given Stokes system with boundary prescribed by $\vec{\mathbf{X}}$ and driven by an external flow $\mathbf{w}$. Equation 7.6 can then be integrated in time to evolve the particle locations and calculate particle trajectories. To perform this integration, we apply a second-order Adams-Bashforth/Adams-Moulton predictor-corrector scheme.

$$\frac{\vec{\mathbf{X}}^* - \vec{\mathbf{X}}^n}{\Delta t} = \frac{3F(\vec{\mathbf{X}}^n, t^n) - 3F(\vec{\mathbf{X}}^{n-1}, t^{n-1})}{2} \tag{7.7}$$

$$\frac{\vec{\mathbf{X}}^{n+1} - \vec{\mathbf{X}}^n}{\Delta t} = \frac{5F(\vec{\mathbf{X}}^*, t^{n+1}) + 8F(\vec{\mathbf{X}}^n, t^n) - F(\vec{\mathbf{X}}^{n-1}, t^{n-1})}{12} \tag{7.8}$$

7.2 A mixed discretization for Stokes equations

We present here a new mixed discretization utilizing a pseudo-divergence free reconstruction for the velocity and a staggered compatible meshless reconstruction for the pressure. The domain is discretized by distributing a set of points $\mathbf{N} := \{x_i\}_{i=1,\dots,N_p}$ throughout the domain and along the boundary. We will discuss in Section 7.3 the details of this particle distribution, but for now state loosely that the particles are quasi-uniform in the sense of [158]. We seek a set of discrete differential operators taking values at each point. To do this, we associate with each particle x_i a lengthscale ϵ_i, and construct the ϵ_i-graph of particles near x_i.

$$\mathbf{E}_i = \{(i,j) \text{ such that } ||x_i - x_j|| < \epsilon_i\} \tag{7.9}$$

The construction of this neighbor connectivity can be easily performed in $O(N)$ time using standard binning algorithms. The discretization used here is an extension of the standard MLS approach, in which a differential operator is constructed by finding a local polynomial reconstruction of function values $u_j := u(x_j)$ as the solution to the weighted ℓ_2-optimization

$$p^* = \underset{p \in \pi_m}{\arg\min} \left\{ \sum_{j=1}^{N} \left[u_j - p_j\right]^2 W_{ij} \right\} \tag{7.10}$$

where π_m denotes the set of m^{th}-order polynomials, and W_{ij} denotes a positive weight depending on the relative distances of x_i and x_j, satisfying $W_{ij} = W_{ji}$. To approximate an operator D^α evaluated at x_i, the operator is simply applied to the reconstructed polynomial

$$D^\alpha u_i \approx D^\alpha_h u_i := D^\alpha p^*(x_i) \tag{7.11}$$

By choosing W_{ij} with compact support, this efficiently generates finite difference like stencils that can be applied to complex geometries. A detailed discussion of these problems and their solution can be found in [158, 106] and Chapter 4.7 of this thesis. We note that since ℓ_2 minimization has an analytic solution, Equation 7.11 provides a means of deriving a finite-difference style stencil of the form

$$D_h^{\alpha} u_i = \sum_{j \in supp(W_{ij})} \alpha_{ij} u_j \tag{7.12}$$

where α_{ij} are coefficients obtained from the solution to Equation 7.10. Unlike standard finite difference methods, these coefficients do not inherit the symmetry of the corresponding continuous operators (e.g. when approximating a gradient, $\alpha_{ij} \neq -\alpha_{ji}$ except for special cases of uniform particle arrangements).

In the staggered moving least squares method, for each ϵ_i-graph $\mathbf{E_i}$, we loosely identify edges as virtual faces and nodes as virtual cells and seek a meshless discretization of divergence and gradient operators mimicking primal/dual methods. Defining $\mathbf{E} := \cup_i \mathbf{E}_I$, we seek staggered discretizations of the form

$$div_h(\mathbf{u}) : \mathbf{E} \rightarrow \mathbf{N} \tag{7.13}$$

$$grad_h(\phi) : \mathbf{N} \rightarrow \mathbf{E}. \tag{7.14}$$

By associating integral degrees of freedom with each edge, a discrete combinatorial gradient is defined

$$grad_h(\phi)_{ij} = \int_{e_{ij}} \nabla \phi \cdot d\mathbf{s} = \phi_j - \phi_i \tag{7.15}$$

that is exact via the fundamental theorem of calculus. A divergence operator can

then be defined via the MLS procedure by reconstructing a vector polynomial from edge integrals

$$\mathbf{p}^* = \underset{\mathbf{p}\in\pi_m}{\arg\min} \left\{ \sum_{j=1}^{N} \left[\int_{e_{ij}} (\mathbf{u} - \mathbf{p}) \cdot d\mathbf{s} \right]^2 W_{ij} \right\} \tag{7.16}$$

and defining the divergence by evaluating the reconstructed polynomial at the nodes.

$$\nabla \cdot \mathbf{u}(x_i) \approx div_h(\mathbf{u})_i := \nabla \cdot \mathbf{p}^*(x_i) \tag{7.17}$$

This leads to a compatible discretization of the Poisson operator as

$$\nabla^2 \phi \approx \nabla_h^2 \phi := div_h(grad_h(\phi)) \tag{7.18}$$

Inhomogeneous Neumann boundary conditions are enforced at boundary particles by adding a constraint to Equation 7.16 that the reconstructed polynomial satisfy the boundary condition exactly at the point x_i. Details of the implementation of these methods can be found in Chapter 6, where it is demonstrated analytically and numerically that for an m^{th} order reconstruction, Equation 7.18 provides $m+1$ order truncations error, and when used to discretize the Poisson equation with Dirichlet/Neumann boundary conditions equal order optimal convergence is obtained in both an ℓ_2 norm and a discrete energy norm. To discretize the Stokes equations, we will also require a gradient operator evaluated at nodes. We define this by evaluating the reconstruction of the combinatorial gradient at x_i via

$$\nabla \phi(x_i) \approx \mathbf{R}_{grad}(\nabla_h p)_i := \mathbf{p}^*(x_i) \tag{7.19}$$

where $\mathbf{p}^*$ is the solution to Equation 7.16 for $\mathbf{u} = grad_h(p)_{ij}$.

With the divergence and gradient operators discretized, we finally seek a discretization of the viscous operator compatible with the solenoidal constraint. In

order to do this, we define the set of divergence-free vector polynomials of order m as π_m^{div}. We note that a basis is easily reconstructed for this set, for example in two dimensions for the case $m = 1$, for any $\phi \in \pi_1^{div}$

$$\phi \in span\left\{\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}y\\0\end{pmatrix}, \begin{pmatrix}0\\x\end{pmatrix}, \begin{pmatrix}x\\-y\end{pmatrix}\right\} \tag{7.20}$$

Therefore we can procede with standard MLS reconstruction, albeit using a modified basis.

$$\mathbf{p}^* = \underset{p\in\pi_m^{div}}{\arg\min}\left\{\sum_{j=1}^{N}\left[\mathbf{u}_j - \mathbf{p}_j\right]^2 W_{ij}\right\} \tag{7.21}$$

The viscous operator can then be discretized with

$$\nabla\times\nabla\times\mathbf{u}(x_i) \approx (\nabla\times\nabla\times)_h\mathbf{u}_i := \nabla\times\nabla\times\mathbf{p}^*(x_i) \tag{7.22}$$

As mentioned in the introduction, the lack of a mesh or symmetry in particle arrangement makes it difficult to pose a rigorous analysis of the sense in which this reconstruction is actually faithful to the solenoidal constraint. The importance of discrete mass conservation for finite element methods has been carefully considered in the literature, see for example [44, 87, 27, 115]. It is unclear whether a discrete mass conservation statement can be shown here. In Huerta et al.'s work [70], however, a similar reconstruction was used but for a weak formulation, allowing them to demonstrate that the reconstruction passes a numerical inf-sup test. We will demonstrate in Section 7.6.1 that the use of this divergence-free reconstruction similarly provides stable approximation for this strong form discretization while maintaining high order convergence for both pressure and velocity.

Similar to Equation 7.12, Equations 7.16 and 7.21 provide the following set of

finite-difference like stencils

$$\nabla_h^2 p_i = \sum_{j \in supp(W_{ij})} \alpha_{ij}^1 p_j \tag{7.23}$$

$$\nabla_h p_i = \sum_{j \in supp(W_{ij})} \alpha_{ij}^2 p_j \tag{7.24}$$

$$\nabla \times \nabla \times_h \mathbf{u}_i = \sum_{j \in supp(W_{ij})} \alpha_{ij}^3 \mathbf{u}_j \tag{7.25}$$

$$\tag{7.26}$$

where $\alpha^1, \alpha^2, \alpha^3$ are coefficients obtained from the solution of each optimization problem. These stencils can then be used to discretize the Stokes equations in a manner identical to standard finite-difference methods: a global matrix is assembled by discretizing Equations 7.2 and 7.3 using Equation 7.26 at each point in the interior of the domain. For points lying on the boundary, the Dirichlet boundary condition in Equation 7.2 is enforced by adding a one to the diagonal of the global matrix and the imposed value to the right-hand side. The Neumann boundary conditions for the pressure in Equation 7.3 are enforced implicitly via the constraint in 7.16. As in any method discretizing the Poisson problem with Neumann boundary conditions, the discrete Poisson operator contains a constant vector in its right null space that must be handled to obtain a unique solution. In this work we do this by adding a single Lagrange multiplier to the Poisson problem to enforce a zero mean pressure (see, for example, [142], for details).

$$\sum_{i=1,\ldots,N_p} p_i = 0 \tag{7.27}$$

7.3 Multiresolution approach: kernel adaptivity and particle distribution

So as to simulate the suspension flow described in Equation 7.1 from first principles, it is necessary to resolve both the singular pressure occuring in the narrow lubrication gap and the larger length scales of the particle/boundary interactions that drive the flow. To achieve this we adaptively select the optimization weights W_{ij} to match the surrounding particle refinement. In standard MLS-type approaches, the weights are chosen by selecting a positive symmetric kernel with support ϵ.

$$W_{ij} = W_\epsilon(||x_i - x_j||) \tag{7.28}$$

In this work, we will consider kernels of the form $W_\epsilon(r) = (1 - \frac{|r|}{\epsilon})^p$, with $p = 4$. For a distribution of particles characterized by a quasi-uniform lengthscale Δx, the support can be scaled as $\epsilon = C \cdot \Delta x$, where C is selected to ensure that enough neighbors lie in the support of W_{ij} that Equation 7.10 has a unique solution. Assuming that the points maintain polynomial unisolvency and avoid degenerate arrangements (e.g. particles are not co-linear and maintain a finite separation distance), this can loosely be considered as requiring that the number of neighbors be larger than the dimension of the polynomial space

$$\#\{j \in supp(W_{ij})\} > dim(\pi_m) \tag{7.29}$$

For example, taking $C = m + 1$, where m is the order of the MLS reconstruction, provides a slightly conservative interaction radius that is appropriate near corners and boundaries of the domain where one-sided stencils are generated.

For a multiresolution discretization, the support of the kernel is chosen adaptively as a function of position, i.e. $\epsilon_i := \epsilon(x_i)$. An initial lengthscale characterizing the maximal discretization lengthscale is calculated

$$\Delta x^{\infty} = sup_i \min_{j \neq i} ||x_i - x_j|| \tag{7.30}$$

and a preliminary support radius is calculated as $\epsilon_i^{prelim} = (m+1)\Delta x^{\infty}$. Neighbor lists are then assembled and sorted in order of increasing distance. In light of Equation 7.29, the distance to neighbor number $dim(\pi_m)$ is calculated. This distance r_{ij}^{min} denotes the minimum distance to obtain enough neighbors to reconstruct a polynomial under the assumption that all neighbors are π_m-unisolvent. To account for the possibility of degenerate particle arrangements, the particle support is conservatively chosen as $\epsilon_i = 1.5 r_{ij}^{min}$. The weights used in ℓ_2 optimization are then defined as

$$W_{ij} = \frac{W_{\epsilon_i}(||x_i - x_j||) + W_{\epsilon_j}(||x_i - x_j||)}{2} \tag{7.31}$$

These weights are again positive and symmetric in i and j, but scale appropriately between regions of high and low resolution. The neighbor lists are then recalculated to match the support of W_{ij}, and the global matrix can be assembled.

Before weights can be calculated however, particles must be distributed throughout the domain. For simplicity, we pursue the same refinement strategy for all cases in this work. The discretization is characterized by a lengthscale Δx^{∞}, the number of refinement levels N_r and the number of refinement layers N_l. Each refinement level L_i is characterized by a lengthscale $\Delta x_i = \Delta x^{\infty} 2^{i - N_r}$. Particles are first placed along colloid boundaries with separation Δx_0. Then for $i = 1, \ldots, N_r$, N_l layers of thickness and arclength Δx_i are placed marching outward from the boundary. If any of these particles intersect a more highly refined layer belonging to another particle,

they are not added. After these nested layers are generated, a Cartesian grid of particles with spacing Δx^{∞} is placed covering the remainder of the domain, and finally the outter boundary is discretized by placing particles around the perimeter with spacing Δx^{∞}. For the results presented later, each discretization will be described by the triplet $(\Delta x^{\infty}, N_r, N_l)$. While this refinement strategy is adopted here for simplicity, more sophisticated refinement strategies can be easily adapted, provided they ensure sufficient neighbors to perform the MLS reconstruction.

7.4 Evaluation of viscous stresses at colloid boundaries

To couple the motion of the colloids to the flow solver via Equation 7.4, it is necessary to derive an approximation to the integral of the stress tensor in Equation 7.4. We describe here the process for the force balance, but the process for the torque balance is evaluated in an identical manner. We identify each boundary particle $x_j \in \partial\Omega_i$ with a portion of arclength $d\theta_j$ and a unit normal vector pointing outward from the colloid $\vec{\mathbf{n}}_j$. The integral in Equation 7.4 can then be partitioned

$$\int_{\partial\Omega_i} \sigma_V \cdot d\mathbf{A} = \sum_{j \in \partial\Omega_i} \int_{d\theta_j} \sigma_V \cdot d\mathbf{A} \tag{7.32}$$

A high order quadrature rule can then be derived by posing the quadrature via a standard MLS reconstruction

$$\int_{d\theta_j} \sigma_V \cdot d\mathbf{A} \approx \int_{d\theta_j} \sigma^* \cdot d\mathbf{A} \tag{7.33}$$

where σ^* is calculated by taking the reconstructions of pressure and velocity

$$p^* = \arg\min_{\phi \in \pi_m} \left\{ \sum_{j=1}^{N} \left[p_j - \phi_j \right]^2 W_{ij} \right\} \tag{7.34}$$

$$\mathbf{u}^* = \arg\min_{\mathbf{p} \in \pi_m^{div}} \left\{ \sum_{j=1}^{N} \left[\mathbf{u}_j - \mathbf{p}_j \right]^2 W_{ij} \right\} \tag{7.35}$$

$$\tag{7.36}$$

and calculating

$$\sigma_V = -p^* \mathbf{I} + \frac{\nu}{2} \left(\nabla \mathbf{u}^* + \nabla \mathbf{u}^{*\intercal} \right). \tag{7.37}$$

This can then be used to evaluate the integral in Equation 7.32. Although accurate to high order, for the cases considered in this work, the use of the simpler midpoint quadrature rule

$$\mathbf{F}_d \approx \sum_{j \in \partial\Omega_i} \sigma^*(x_j) \cdot \vec{\mathbf{n}}_j R_i d\theta \tag{7.38}$$

provided no discernable difference, and is used for the results presented.

7.5 Efficient solution of the monolithic system

If the colloid motion is coupled together with the flow equations through Equations 7.2,7.3 and 7.4 and the discretization described in Sections 7.2 and 7.4 is applied, a monolithic block structured matrix is obtained coupling together velocity, pressure and the 6-DOF describing the rigid body rotation of each particle.

7.6 Stokes solver: Validation and convergence study

To validate the mixed scheme, we begin by studying cases where a single colloid has a known position, velocity, and angular velocity. This will allow a study of the convergence of the Stokes solver and force/torque calculation before coupling together the colloid dynamics.

7.6.1 Wannier flow

Wannier flow consists of two cylinders of radii R_1 and R_2 with eccentricity e rotating with fixed angular velocity ω_1 and ω_2. Originally studied as a model for lubrication forces in journal bearings, the flow's analytic solution for velocity, pressure, and the force on the inner cylinder have made this a standard bechmark in validating high-order methods in the presence of curvilinear geometry[74]. The analytic solution for this problem can be found in [156].

For this case, we first consider a family of discretizations characterized by the particle distribution $(N^{-1}, 1, 1)$, meaning that no adaptive refinement is applied. The parameter N represents the number of particles distributed in each direction, and N^{-1} provides the grid spacing between particles. We begin by considering cylinders of radii $R_1 = \pi/2$, $R_2 = \pi/10$, with eccentricity $\pi/5$, and impose angular velocities of $\omega_1 = 10/\pi$ and $\omega_2 = 1/\pi$. We demonstrate that, for an $(m, m-1)$-order reconstruction of the velocity and pressure, equal order optimal m^{th} order convergence is obtained for both the velocity and the pressure in Figure 7.1. The results for all of these simulations take less than a minute to run on a desktop computer with no hardware acceleration.

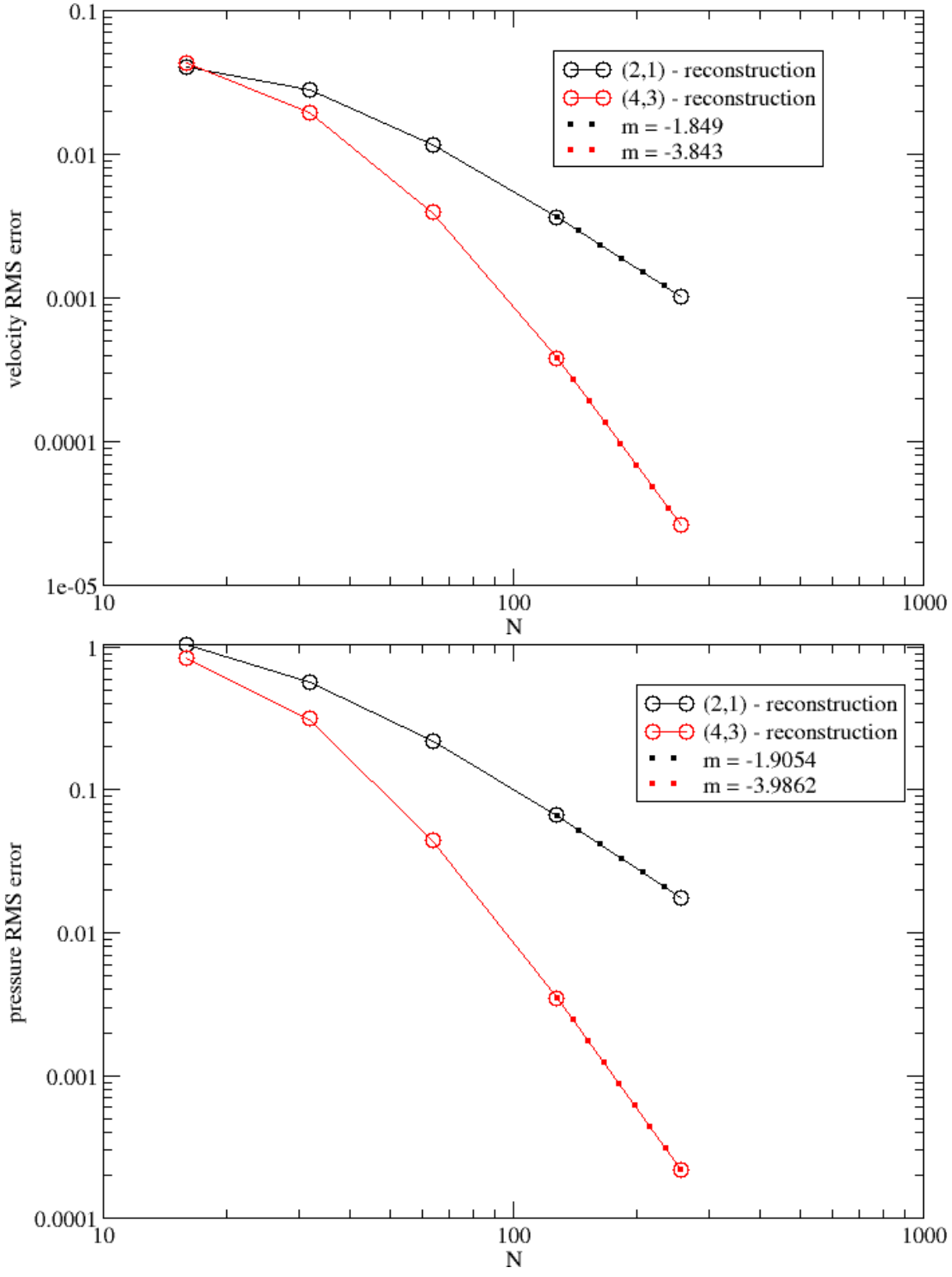


Figure 7.1: Optimal equal order convergence for velocity (left) and pressure (right).

We then systematically study the singular lubrication limit as the cylinder separation approaches zero. To do this, we use a $(4, 3)$-order velocity and pressure reconstruction and apply a mild refinement around each cylinder using the family of discretizations $(N^{-1}, 2, 2)$ and choose N sufficiently large so that there are always at least four points spanning the gap separating the two cylinders. This allows just enough resolution that the stencils within the gap maintain polynomial unisolvency. The resulting geometry and pressure distribution are shown in Figure 7.2, and we demonstrate excellent agreement with analytic solution in Figure 7.3. While more aggresive adaptivity is certainly possible, we adopt this strategy to demonstrate heuristically that roughly five particle per smallest lengthscale is adequate to obtain accurate results.

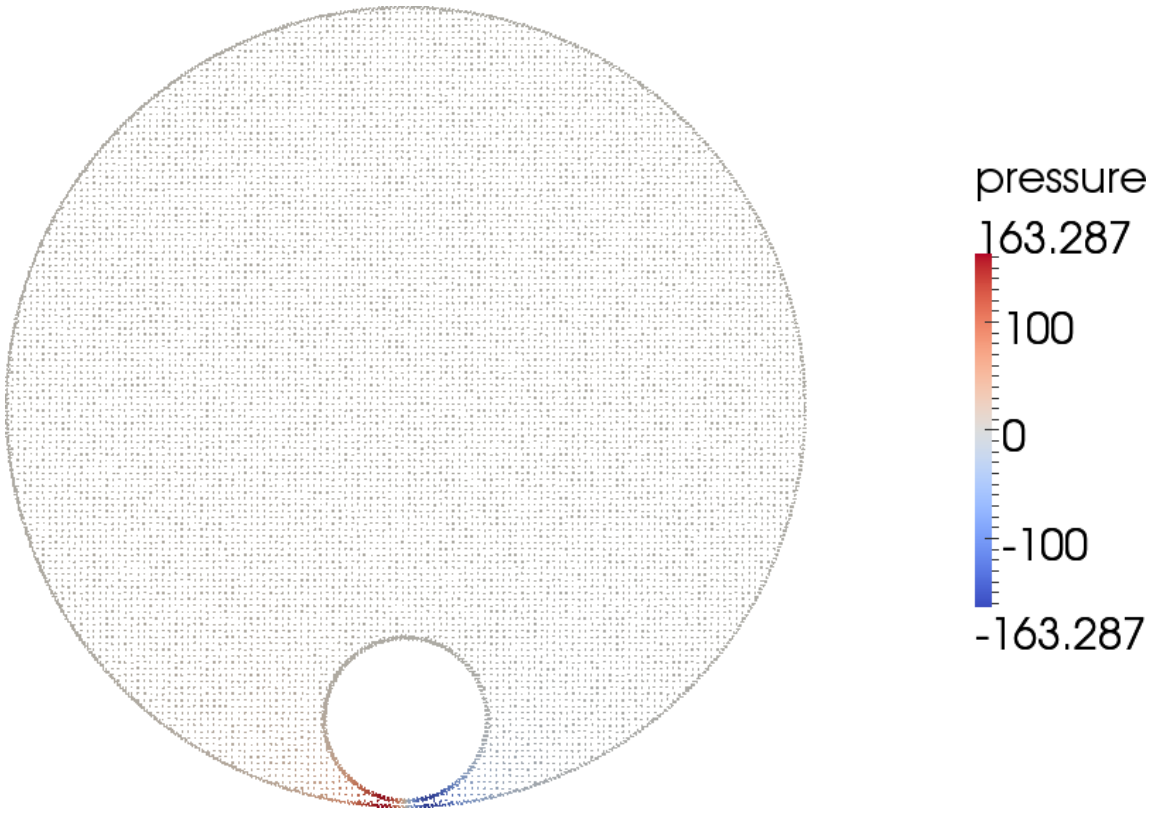


Figure 7.2: Nearly singular pressure distribution for gap width $a_1/32$.

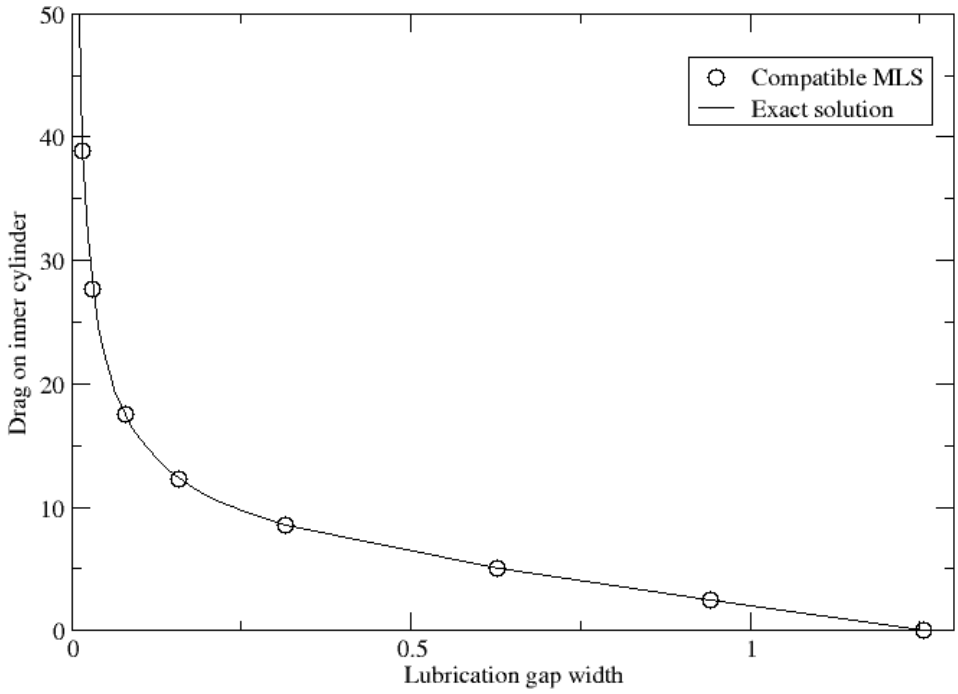


Figure 7.3: Accurate resolution of singular lubrication force as gap width is reduced. Gap width and forces are given in non-dimensional form.

7.7 Flow past cylinder in channel

To validate the accuracy of the discretization when the colloid dynamics are coupled to the Stokes solver, we consider flow past a single cylinder of radius a in a channel of length L and height H (See Figures 7.4,7.5). We consider two cases: one in which zero flow is imposed at the boundary and the particle moves with a constant velocity V, and another in which the particle is fixed and a Pouseille flow of magnitude U is imposed at the inlet and outlet. In the limit $L \to \infty$ an exact solution exists for the drag exerted on the particle as a function of blockage ratio a/H. By linearity, Jeong and Yoon[72] use this relation to deduce the drift velocity of a particle that is free to advect under the flow by balancing the flow induced drag with the drag

induced by the particle motion. We will calculate this by solving the monolithic system of Equations 7.2,7.3 and 7.4. For this and the remainder of this work we use a $(4, 3)$-order velocity and pressure reconstruction. We adopt a $(N^{-1}, 3, 1)$ family of refinement for a given blockage ratio, and then refine until approximately five points span the gap between the particle and the wall. We consider the choice of parameters $H = 2$, $L = 6$, and vary the particle radius a. We demonstrate in Figure 7.6 that the corresponding drag forces are very well reproduced, even as the blockage ratio approaches 1 and lubrication effects become dominant. Finally, having validated that drag forces are accurately resolved, we can couple the colloid dynamics together with the Stokes solver via Equation 7.4 to match Jeong and Yoon's relation for drift velocity (Figure 7.7) [72].

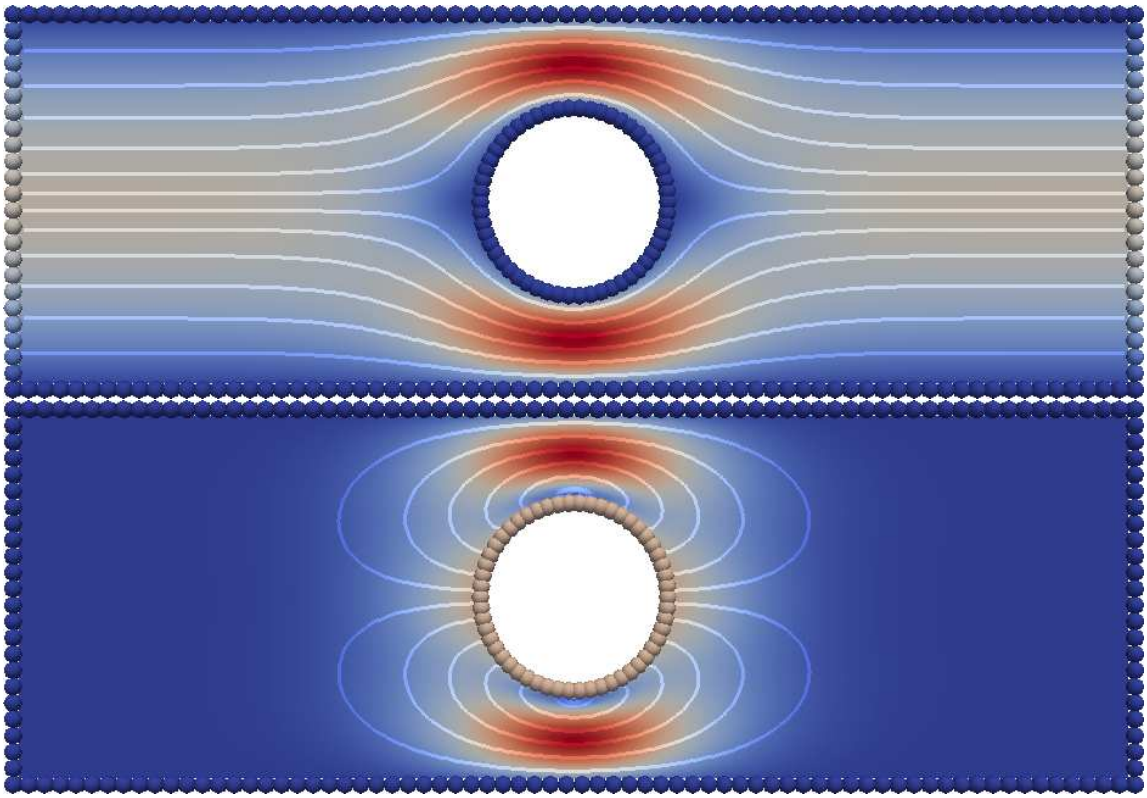


Figure 7.4: Velocity magnitude and streamlines for the cases: $V = 0$, $U = 1$, $a = 0.5$ (top) and $V = 1$, $U = 0$, $a = 0.5$ (bottom).

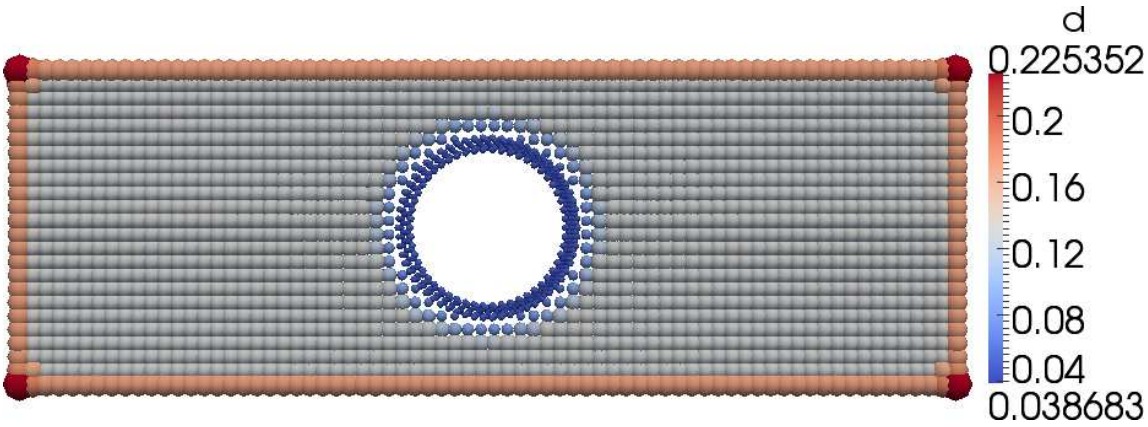


Figure 7.5: Particle adaptivity visualized by rendering spheres at each point with diameter proportional to ϵ_i.

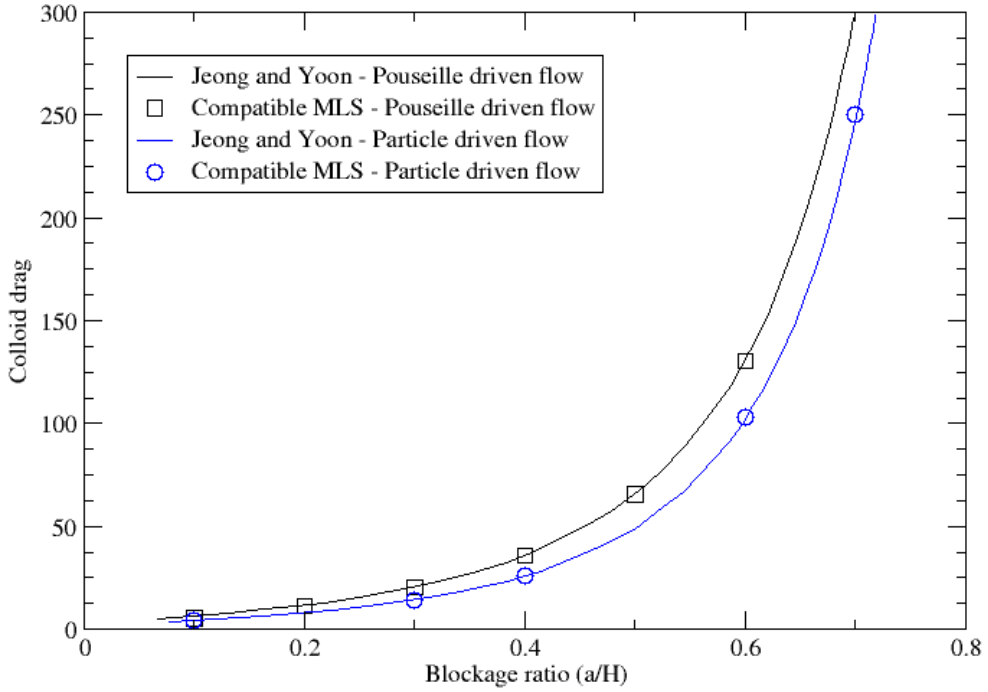


Figure 7.6: Force for flow driven by Poiseuille flow and by colloid with comparison to exact solution in non-dimensional units.

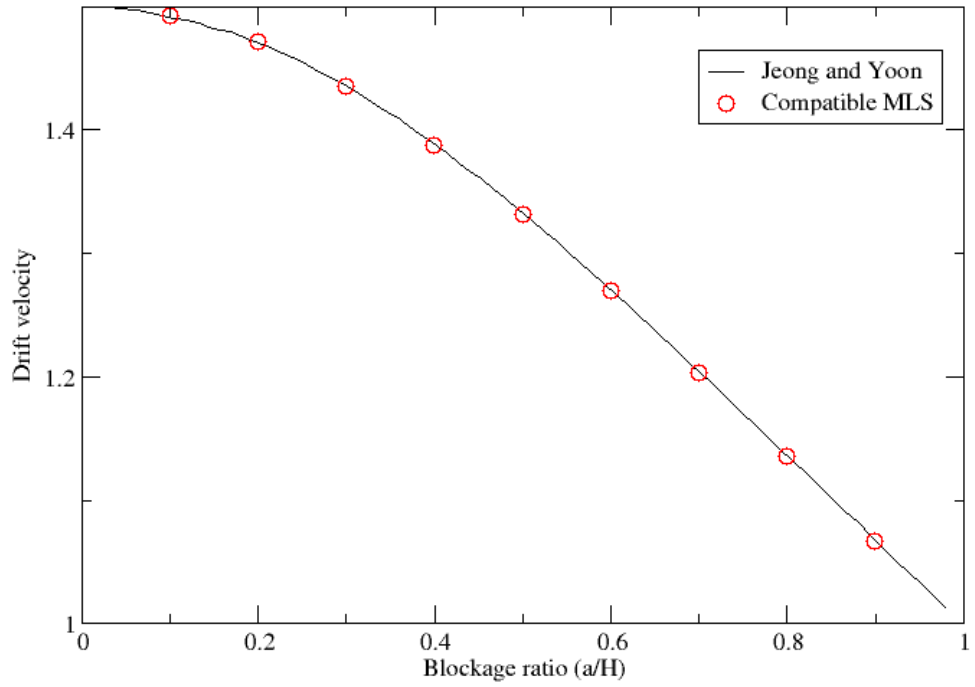


Figure 7.7: Drift velocity when zero force/torque condition is used to couple colloid motion to flow

7.8 Deforming geometry: cylindrical colloids in shear flow

With the Stokes solver and its coupling to colloid dynamics validated, we finally have a means of evaluating the right hand side of Equation 7.8 so that the hydrodynamic forces can be integrated numerically to calculate colloid trajectories. We consider the case where two cylindrical particles of radius a are placed in a channel of dimension $L \times H$ with initial positions $\mathbf{X}_1 = (-L_h, L_v)$ and $\mathbf{X}_2 = (L_h, -L_v)$ and flow is driven by a shear rate γ_0 (Figure 7.8). For the case where $\frac{L+H}{a} \rightarrow \infty$, the trajectory of the particles is governed by a non-linear ODE that can be integrated numerically

to provide a highly accurate benchmark solution [7]. This case has been used as to validate lower order methods for simulating colloidal suspensions in the past[14], however in this case we again stress that no lubrication models are used here, and an accurate solution will be obtained directly from solution of the Stokes equations using adaptivity to resolve the narrow lubrication layer between particles. We consider $a = 1$,$L_h = 1.5$ and the family of trajectories with initial condition given by $L_v \in \{2, 1, 0.5, 0.25\}$ and also the configuration $\mathbf{X}_1 = (-1.2, 0)$,$\mathbf{X}_2 = (1.2, 0)$. The exact solution for this problem demonstrates that the first set of initial configurations will give rise to open trajectories, while the second will lead to closed orbits in which the two particles will circle each other indefinitely[7]. The shear rate $\dot{\gamma}$ is imposed by enforcing a Couette flow on Ω_d

$$\mathbf{w} = \dot{\gamma} y \tag{7.39}$$

and for this work we take $\dot{\gamma} = 1$. To reproduce the exact solution, it is necessary to take a sufficiently large domain that wall effects do not impact the particle trajectories; we have selected $L = H = 40a$. From experiments and the reference solution however, it can be shown that, for the given choice of colloid configuarations, as the particles pass each other the lubrication gap can be as small as $a/50$. That presents a ratio of relevant length scales of ~ 2000, and following the previously defined heuristic of five particles per lengthscale would lead to a simulation with 10^{10} particles if a uniform discretization lengthscale were used. We will address this by using relatively aggresive adaptivity, using a $(32^{-1}, N_r, 5)$ family of $(4, 3)$-order discretizations and choose N_r so that there are approximately five points across the lubrication gap. This leads to simulations of roughly ten thousand particles with very fine resolution in the gap, however because the monolithic Stokes/colloid system is completely implicit no stability condition is imposed, and $\Delta t = 0.1$ provides an accurate resolution of the particle trajectories when compared to the benchmark solution (Figure 7.10). One of the key benefits of this meshless approach is that arbitrary colloid and

boundary geometry can easily be handled. We highlight this capability in Figures 7.11 and 7.12, where we demonstrate results for the same problem but using square colloids of unit size, and in Figure 7.13 where we demonstrate the trajectory of a pressure-driven flow in which a square colloid passes through a notched microchannel. For these cases, the complex geometry makes validation against an analytic solution impossible and we have left the validation of these results for a future work. We include them here however because they highlight this scheme's ability to tackle a class of problems for which no high-order methods currently exist.

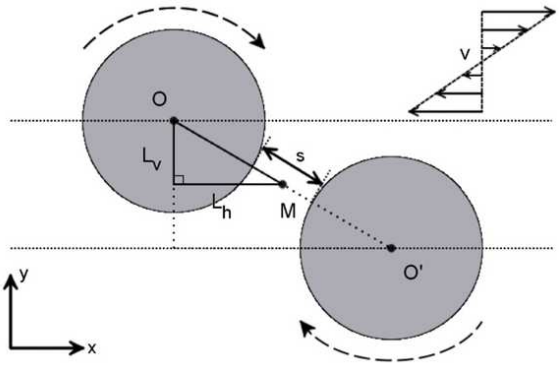


Figure 7.8: Geometry description for shear flow case [14].

7.9 Electrophoresis of colloids in microchannels

With a robust means of computing fluid structure interaction, we demonstrate now a new scheme coupling the hydrodynamics of colloidal particles with electrostatic interaction. For suspensions in an electrolyte solution, colloids with a given surface charge density will attract a cloud of counter-charged ions. Electrophoresis occurs when these ions are exposed to an external electric field to induce a flow in the bulk of the solution. This effect is predominant for flows with a large surface to volume ratio typical of many micro/nano-fluidic applications occurring at length scales ranging from nanometers to microns. An in depth discussion of the fundamentals of these

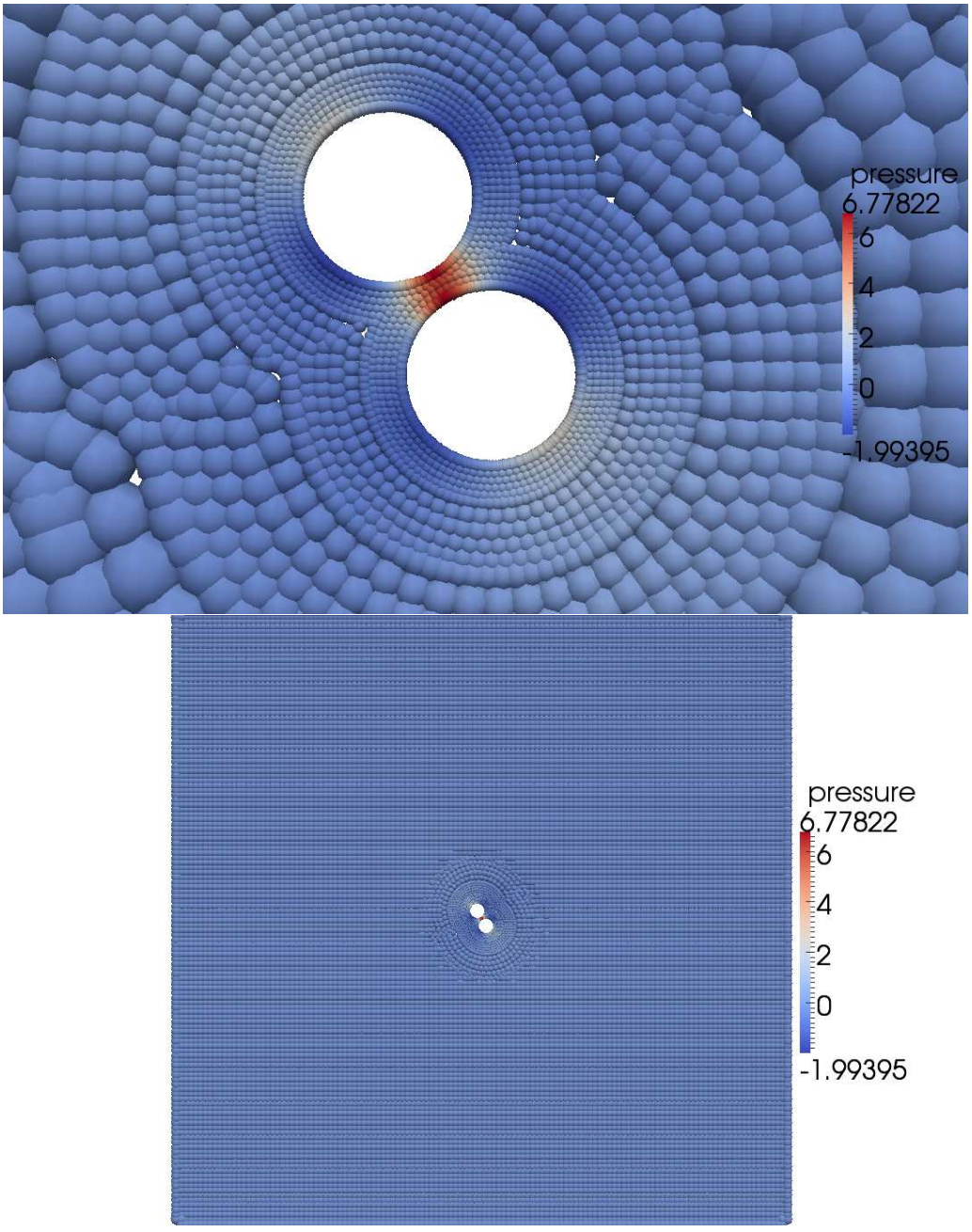


Figure 7.9: Point adaptivity for colloids interacting under shear flow. Particle adaptivity visualized by rendering spheres at each point with diameter proportional to ϵ_i. First few layers of refinement in vicinity of lubrication layer (top), and complete extant of domain (bottom).

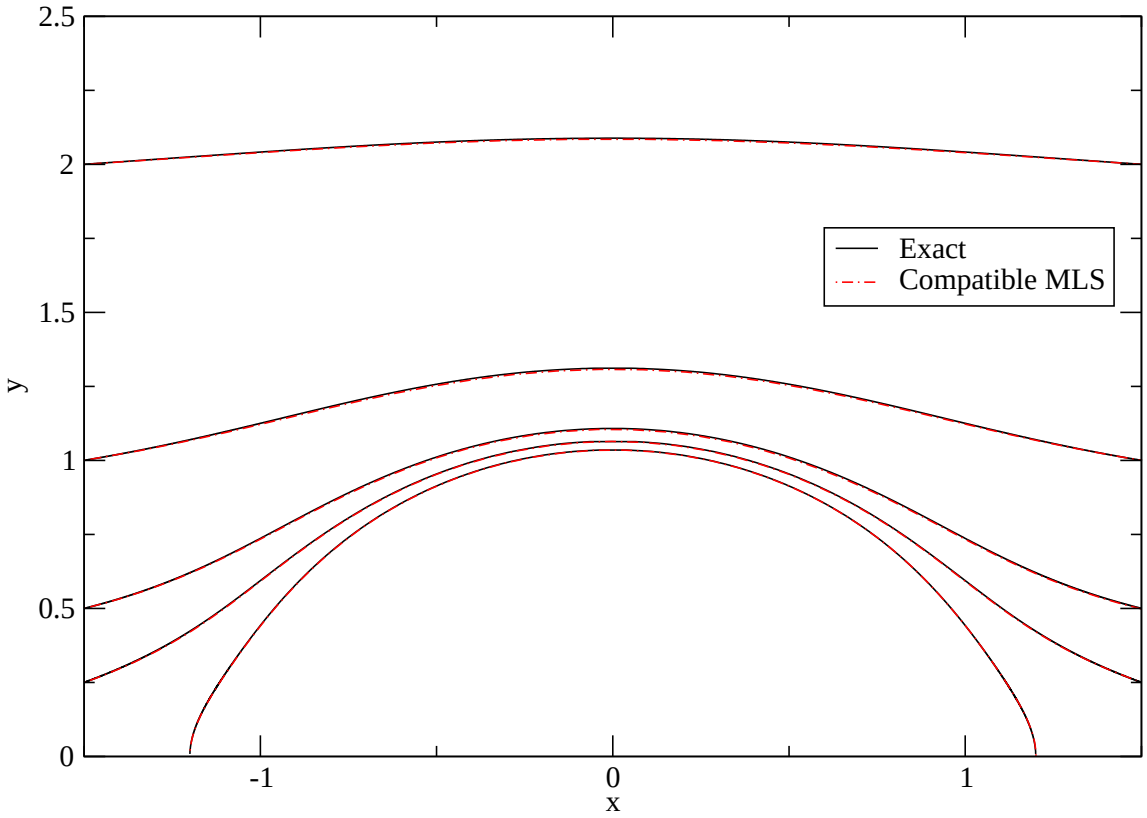


Figure 7.10: Trajectory of top left particle for varying choice of initial colloid configurarion. For closed orbit, separation is driven by lubrication forces occuring in gap width of size $a/50$.

flows is beyond the scope of this thesis, but the topic is mature and details of a number of models describing these flows can be found in textbooks[79, 101, 24].

We consider here the Poisson-Boltzmann model to describe the electrostatic interactions between solvent and colloids. This model is valid in the limit of vanishing Peclet number, but has the attractive feature of only requiring one way coupling between the electrostatics model and the flow solver. This separation of electrostatic and flow timescales will lead to a negligible increase in computational effort when coupling these physics together, despite the fact that the Poisson-Boltzmann model will require the solution of a non-linear elliptic equation at every timestep. In the case of flows where this assumption of timescale separation is inappropriate, it is possible to then solve the Poisson-Nernst-Planck (PNP) equations[101]. In the

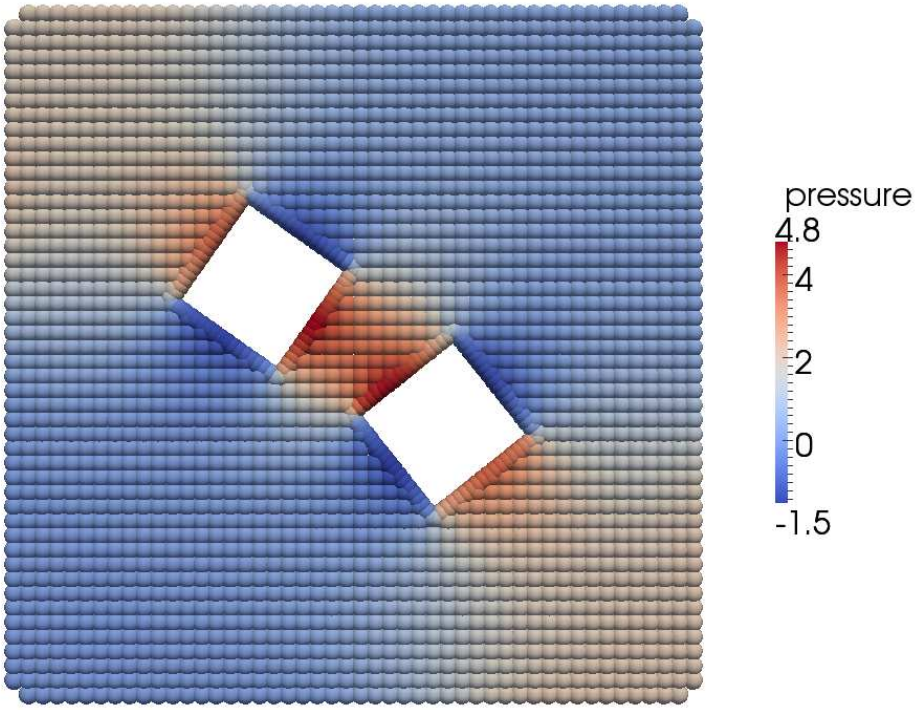


Figure 7.11: (Left) Initial configuration and pressure distribution for square particles of unit width in shear flow. (Right) Trajectory of square particles colored by initial position (red) to final position (blue).

Poisson-Boltzmann case, the ion distribution is postulated to satisfy a Boltzmann distribution caused by the balance of Brownian motion and electrostatic interaction, while in the PNP equations an additional PDE for the ion species transport is solved. This transport equation adds non-linear coupling resulting from the advection of ions by the flows. While we restrict ourselves to the Poisson-Boltzmann case in what follows for simplicity, we note that the PNP case could indeed be solved, but would require several additional Newton iterations per timestep.

We consider the same problem setup as the previous section, but now assume each particle maintains a possibly non-uniform surface charge density $q(x)$. For the case in which the solution consists of a symmetric electrolyte, the electric field $\mathbf{E} = -\nabla\phi$ can be characterized by the following non-dimensional equation for the electric potential ϕ.

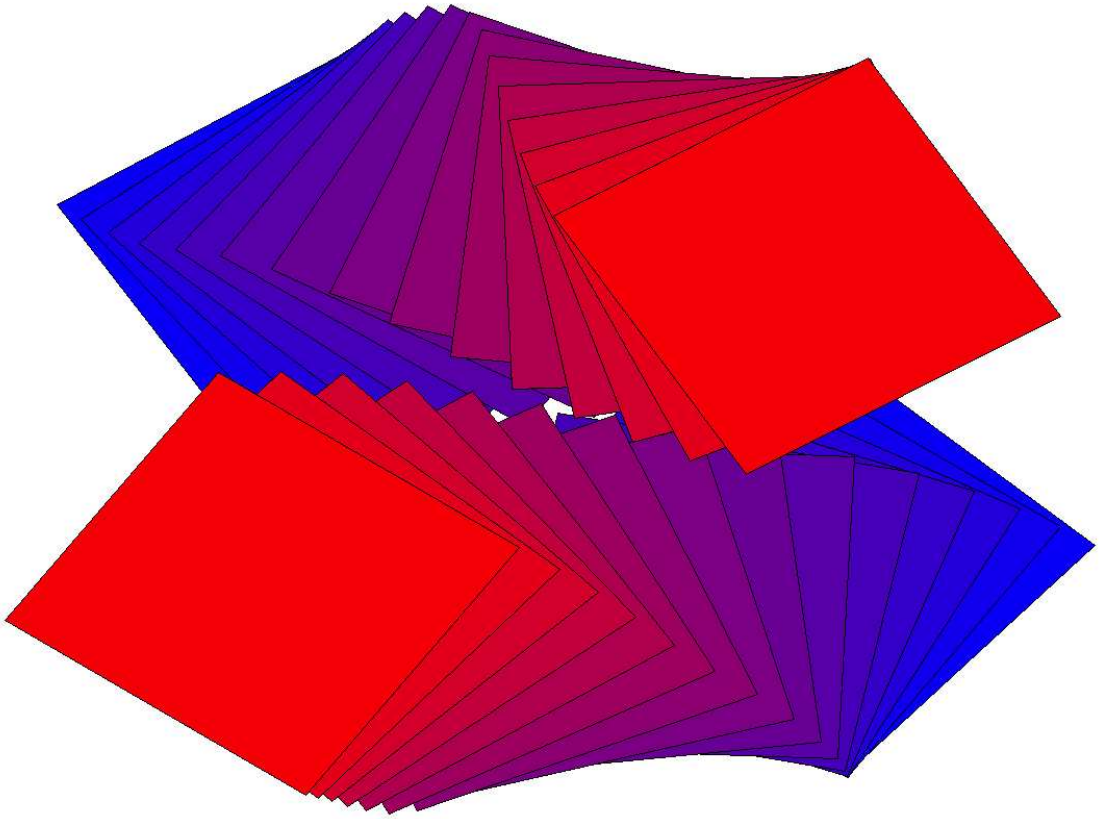


Figure 7.12: (Left) Initial configuration and pressure distribution for square particles of unit width in shear flow. (Right) Trajectory of square particles colored by initial position (blue) to final position (red).

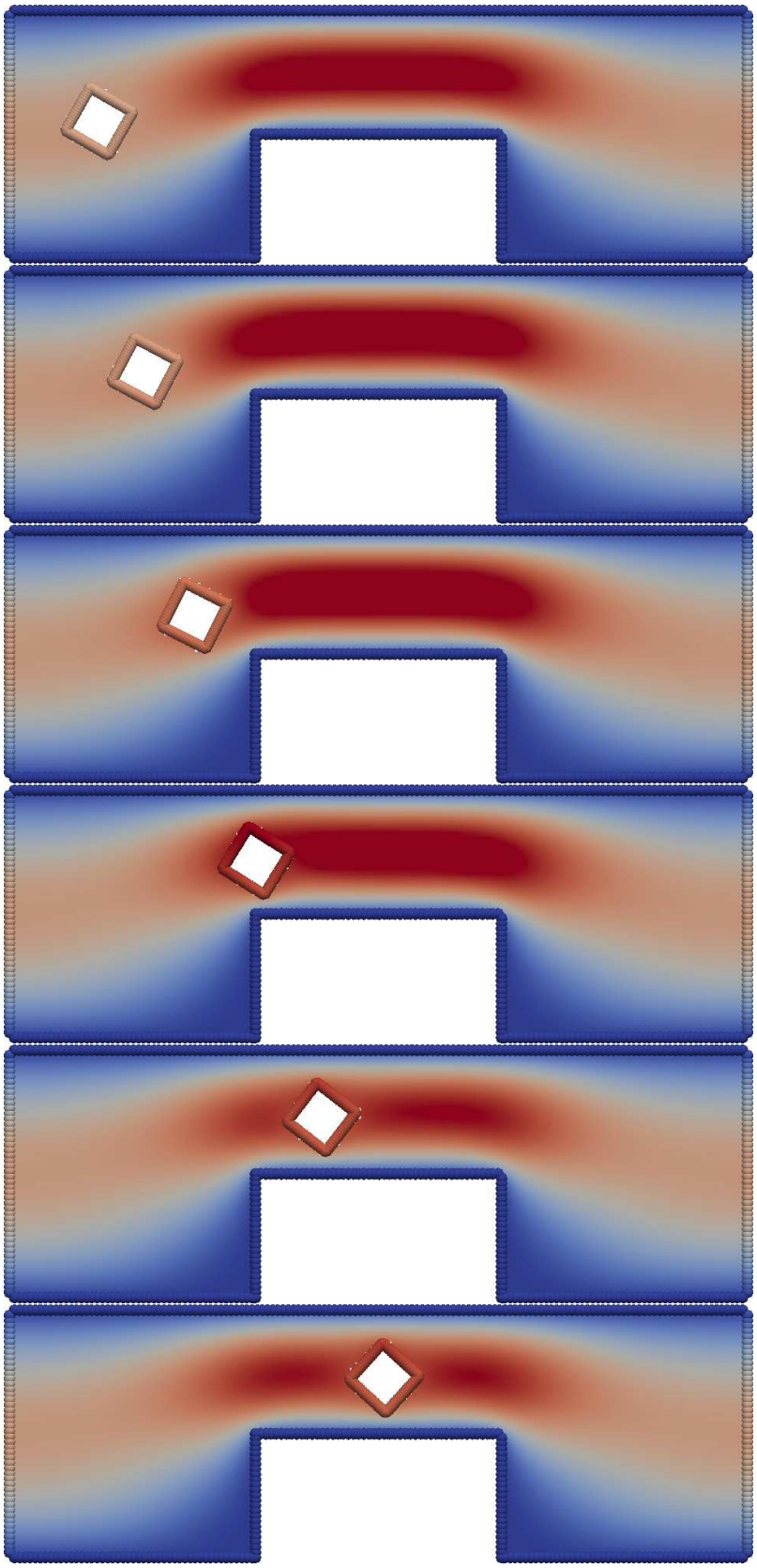


Figure 7.13: Trajectory of a square colloid passing through a notched microchannel.

$$\nabla^2\phi = \kappa^2 \sinh\phi \tag{7.40}$$

where κ is the non-dimensional inverse Debye length, scaling with the inverse thickness of the ion double layer adsorbed to the colloid surface. An inhomogeneous Neumann boundary condition can then be employed to enforce the surface charge density

$$\partial_n\phi = -\kappa h I \tag{7.41}$$

where h is a characteristic length scale of the problem, and I is a non-dimensional surface charge density. Alternatively a Dirichlet boundary condition can be used to enforce a zeta-potential at the colloid surface. The electostatics then couple to the flow solver in the bulk through the addition an electrostatic body force to Equation 7.1

$$\begin{cases} -\nu\nabla^2\mathbf{u} + \nabla p = -\rho_e\nabla\phi & \text{if } x \in \Omega \\ \nabla\cdot\mathbf{u} = 0 & \text{if } x \in \Omega \\ \mathbf{u} = \mathbf{w} & \text{if } x \in \partial\Omega_D \\ \mathbf{u} = \mathbf{V}_i + (x - \mathbf{X}_i)\times\mathbf{\Omega}_i & \text{if } x \in \partial\Omega_i, \text{for all } i \end{cases} \tag{7.42}$$

and where the force/torque balance in Equation 7.4 now balances both the viscous stress tensor σ_V and the Maxwell stress tensor

$$\sigma_M = -\epsilon_0\left(\mathbf{E}\otimes\mathbf{E} + E^2\mathbf{I}\right) \tag{7.43}$$

The ion number density is calculated from the Boltzmann distribution

$$\rho_e = -\kappa^2 \sinh \phi. \tag{7.44}$$

The simultaneous solution of Equations 7.40 and 7.42 can be calculated in a similar framework to the previous section. The entire state of the system is again governed by the vector of colloid positions and orientations $\vec{\mathbf{X}}$. Given this vector, a mesh is generated and the Poisson-Boltzmann equation is solved by iteratively defining the following expression for the residual

$$g(\phi) = \nabla^2 \phi - \kappa^2 \sinh \phi \tag{7.45}$$

The iteration involves the following Newton's method for successive approximations $\{\phi^i\}_{i=1,\ldots}$,

$$\nabla^2 \phi^{i+1} - \kappa^2 \sinh(\phi^i) - \kappa^2 \cosh(\phi^i) \left(\phi^{i+1} - \phi^i\right) = 0 \tag{7.46}$$

For the cases considered here, this Newton scheme converges to machine precision in five iterations. Because each iteration is equivalent to a single linear Poisson solve, the work involved in this step is negligible in comparison to the Stokes solver. With the electric potential calculated, we can then adapt Equation 7.42 into the mixed discretization by solving the equivalent coupled pair of equations

$$\begin{cases} \nu\nabla\times\nabla\times\mathbf{u}+\nabla p=-\rho_e\nabla\phi & \text{if } x\in\Omega \\ \mathbf{u}=\mathbf{w} & \text{if } x\in\partial\Omega_D \\ \mathbf{u}=\mathbf{V}_i+(x-\mathbf{X}_i)\times\mathbf{\Omega}_i & \text{if } x\in\partial\Omega_i, \text{for all } i \end{cases} \tag{7.47}$$

$$\begin{cases} \nabla^2 p=-\nabla\cdot\rho_e\nabla\phi \\ \partial_n p+\nu\hat{n}\cdot\nabla\times\nabla\times\mathbf{u}=-\rho_e\partial_n\phi & \text{if } x\in\partial\Omega \end{cases} \tag{7.48}$$

and after solving a monolithic system of equations we are left with a fully implicit set of colloid translational and angular velocities corresponding to equilibrium between hydrodynamics and electrostatic stresses. Similar to Equation 7.6, this can be written concisely as

$$\dot{\vec{\mathbf{X}}}=F_{PB}(\vec{\mathbf{X}},t) \tag{7.49}$$

and integrated with a predictor corrector scheme. We validate the case of an electrical double layer around a cylinder studied using the the smooth profile method (SPM) by Luo [98], but consider the more challenging case in which a Neumann surface charge density boundary condition is imposed instead of a Dirichlet zeta potential. Due to the small thickness of the EDL for this problem, SPM has difficulty resolving these boundary conditions. We present in Figure 7.14 results for the electric potential profile in the vicinity of a charged cylinder corresponding to Section 3.1.2 of [98]. For this case we consider a cylinder of radius $a=11$ placed in a 252×252 box, with parameters $\kappa=1$, $I=1.47$. We see that for $2-4$ particles across the gap excellent agreement is achieved within the EDL.

While further validation of this electrokinetic formulation is the subject of on-

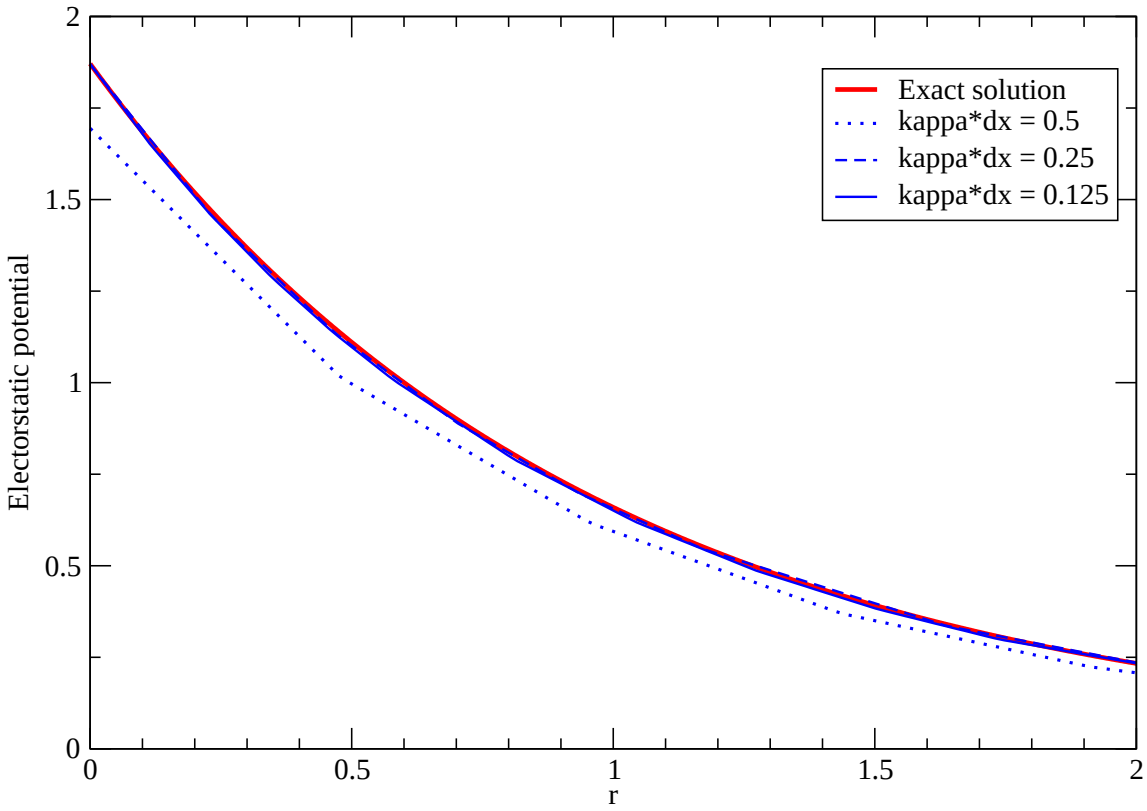


Figure 7.14: Dimensionless electric potential profile in EDL for fourth order reconstruction with comparison to exact solution. For sufficient resolution such that at least two MLS span the EDL, the profile is matched to less than 1% error.

going work, we include results from an un-validated simulation of charged particles coupled with the Stokes solver contained within a closed, insulated box in Figure 7.15. .

7.10 Conclusions

In this chapter we have demonstrated the capabilities of a new mixed formulation built on the model div-grad system from the previous chapter. Numerical evidence for curvilinear geometry suggests that this new discretization provides equal m^{th}-order ℓ_2 convergence for both pressure and velocity if a $(m, m-1)$-reconstruction is used for the velocity and pressure, respectively. While analysis for this problem is

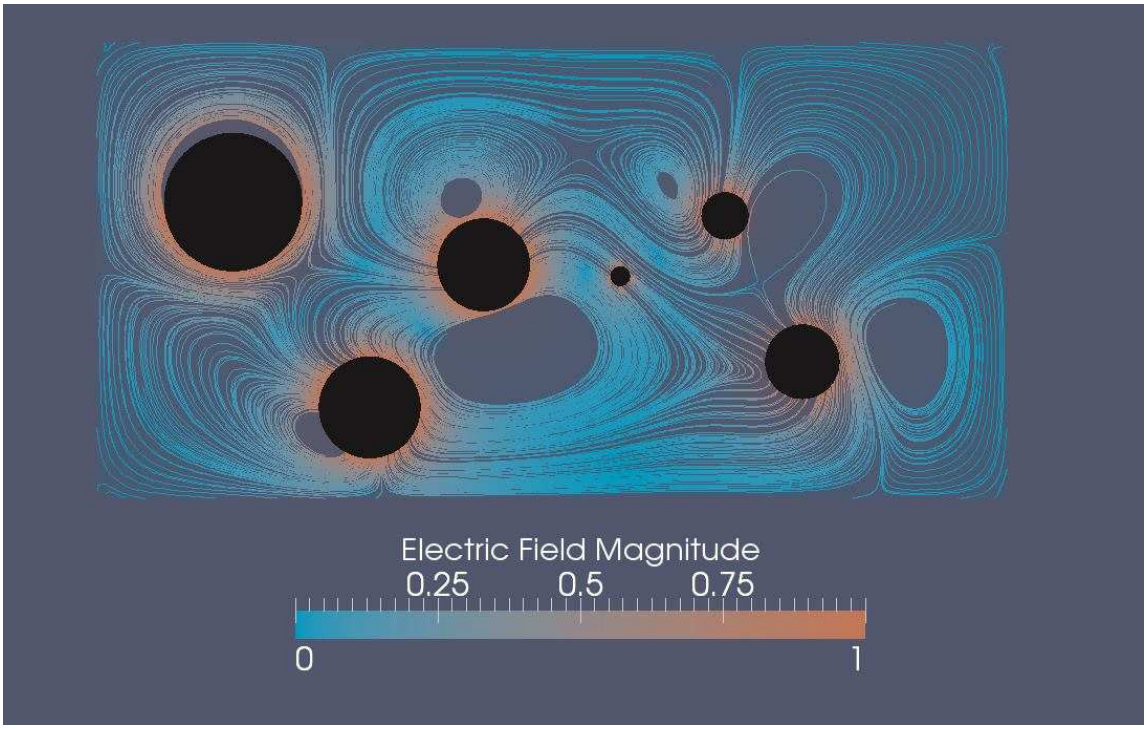


Figure 7.15: Representative solution demonstrating coupling between Stokes solver and non-linear Poisson Boltzmann solver for a collection of repulsive particles with equal surface charge density and varying size.

challenging due to a lack of a mesh, we have instead opted to demonstrate the capabilities of the method by exploring a range of problems related to viscous colloidal suspension flows. For these cases, we are able to obtain a monolithic discretization of the full Stokes equation coupled with fluid-structure interaction between boundaries and colloids. These results are extremely high fidelity and are able to resolve geometrically stiff problems for which lubrication forces occurring in narrow gaps between colloids introduce a range of length scales spreading over several orders of magnitude. We then demonstrated the promise this approach holds for simulating multiphysics problems in which the colloid motion is driven by electrokinetic effects.

Chapter Eight

Compact moving least squares: Generalizing compact finite difference schemes to meshless methods

Having outlined in the past few chapters an approach to recover compatible meshless discretization of the Stokes operator, in this chapter we present a separate strategy that was also successful. While so far we have focused on recovering a compatible discretization for the Stokes operator by mimicking staggered finite difference schemes, here we will mimick compact finite difference schemes in what we will refer to as the *compact moving least squares* method.

Classical compact finite difference methods (CFDM) achieve accuracy competitive with spectral/hp-element methods by exploiting symmetry in the stencil and knowledge of derivatives of the underlying function, obtained either by tailoring the reconstruction to incorporate information from the PDE [157] or by developing implicit formulas for derivatives[84]. We recall fundamental properties of these classes of schemes in Section 8.1. By exploiting this additional information, high-order polynomial reproduction can be obtained using a small number of neighbors per particle, but these schemes require equispaced neighbors to derive stencils from Taylor series. For a general distribution of nodes, multivariate interpolation of nodal data is in general not possible and moving least squares (MLS) and radial basis functions (RBF) have emerged as the two leading methods for robust high order meshless approximation on general datasets[158].

The current approach will present an optimization framework that uses an MLS reconstruction of Hermite data to generalize the CFDM method to arbitrary particle distributions. The stencils generated using this approach behave in a manner identical to CFDM. In comparison to SPH methods, we demonstrate that for the support typically used to discretize the Laplacian when simulating viscous flow [111], the CMLS method is able to attain a sixth order discretization.

In the RBF community, generalizations of the finite difference radial basis func-

tion method (RBF-FD) employ a similar strategy using information gleaned from the underlying PDE to achieve compact discretizations (e.g. [46, 160, 141, 29]). While our approach is similar in broad strategy, posing the reconstruction as an $\ell - 2$ optimization allows a more flexible framework that can make use of the wealth of information regarding stable solution of least squares problems in the literature [83, 152] and a simplified analysis. Assuming for a given particle distribution a local polynomial reproduction exists, the existence of an MLS reconstruction follows from standard convex optimization arguments and does not require the reproducing kernel Hilbert space machinery typically used in RBF methods [158, 106]. An optimization framework allows boundary conditions to be enforced easily via constraints and, while for simplicity in this work we present a polynomial reconstruction, in principle the reconstruction space can be enriched with any test functions (possibly singular). Additionally, although RBF type approaches have successfully generalized CFDM schemes resembling [157], to our knowledge the current approach marks the first method that allows implicit formulae for differential operators generalizing Lele-type schemes[84]. To this end, we claim that this approach is therefore more flexible, and we provide evidence that we are able to obtain $O(N)$ results for both implicit approximation of derivatives and for the monolithic solution of the steady Stokes equations. While both approaches require the adjustment of parameters (the shape parameter in RBF methods or dimensional scaling parameters in the current approach), all of the results presented in this work are presented using a single set of scaling parameters suggesting that the method is in fact robust.

We begin by providing a brief review of compact finite difference methods in Section 8.1 and of classical moving least squares in Section 8.2 before introducing the CMLS method in Section 8.3. We demonstrate recovery of the original Lele-type implicit derivatives in Section 8.4 and proceed in Section 8.5 by presenting simple

examples to demonstrate how the method can be used to solve Poisson and Helmholtz problems in 1D and 2D. We then demonstrate the necessity of our boundary condition approach in performing Helmholtz decompositions as would be used in a projection method for solving fluid mechanics problems and how implicit derivatives yield a compact projection scheme. We finally use the framework to solve the monolithic Stokes equations in Section 8.6 and present a block preconditioner that is able to solve the resulting system with $O(N)$ complexity while recovering optimal convergence. We provide a comparison to results obtained using SPH to demonstrate that despite the overhead of block preconditioning, the accuracy gains of the current approach lead to better answers more quickly as compared to an explicit method.

8.1 Compact finite differences

In compact finite differences, there are two broad strategies which we will refer to as Weinan-type schemes[157] and Lele-type schemes[84]. We will later show that the CMLS schemes generalize both of these approaches for unstructured stencils and arbitrary order.

In the Weinan-type schemes, the PDE is used together with exact expressions for truncation error to achieve fourth-order compact stencils. Consider the solution of the Poisson problem $u'' = f$ on a periodic domain in 1D. To discretize the second derivative, a standard centered difference is used

$$u_i'' \approx D_{xx}u = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = u_i'' + Ch^2 u_i'''' + O(h^4) \tag{8.1}$$

where C is a constant that can be calculated exactly using Taylor series. The PDE

can be used to eliminate the second order term in the truncation error

$$Ch^2 u_i'''' = Ch^2 f_i'' \approx Ch^2 D_{xx} f + O(h^4) \tag{8.2}$$

and after reorganizing, a fourth order expression for the second derivative is obtained with the same bandwidth as the centered difference.

$$D_{xx}^{compact} u_i = D_{xx} u_i - Ch^2 D_{xx} f_i = u_i'' + O(h^4) \tag{8.3}$$

This approach requires that C be calculable, which is trivial for uniform grids. In Lele-type schemes, implicit expressions for derivatives are sought of the form

$$\sum_{j=-N}^{N} \alpha_j u'_{i+j} = \frac{\sum_{j=1}^{M} a_j \left(u_{i+j} - u_{i-j}\right)}{h} + O(h^{Q+1}) \tag{8.4}$$

The coefficients $\{\alpha_j, a_j\}$ are obtained by enforcing that the expression is exact for polynomials of order Q. Calculating these coefficients is again trivial on a uniform mesh using Taylor series, and once they are available, derivatives may be calculated by solving a well-conditioned global matrix for $\{u_i'\}$ given known values of $\{u_i\}$ on the right hand side. For three point stencils ($M = N = 1$) a fourth order discretization is obtained similar to the Weinan-scheme.

8.2 Collocated MLS

The details of using classical MLS in a collocation scheme has already been discussed in detail earlier in this thesis, but the CMLS scheme will require the use of a Legendre basis to avoid ill-conditioning. We include here discussion from a paper discussing these details for the classical case, but Section 8.2.1 can safely be skipped as it is redundant with earlier chapters.

8.2.1 Classical MLS

In this work we follow the framework presented in [158]. Given a set of particles in a compact domain $\mathbf{X}_h = \{x_i\}_{i=1,\dots,N} \subseteq \Omega \subseteq \mathbb{R}^d$, we seek an approximate reconstruction of a function $u \in C^\infty(\Omega)$ from its values $u_i = u(x_i)$ over the domain. This approximation $u_h(x) = \sum_j f_j(x) u_j$ is called a local polynomial reproduction of order m if for some family of functions $\{f_j\} < \infty$ with compact support ϵ, the approximation is exact for all polynomials of order m, i.e. $\forall p \in \pi_m(\mathbb{R}^d), p_h = p$. The question of what conditions are necessary on $\mathbf{X}_h$ such that a local polynomial reproduction exists is technical and discussed in [158], and for the sake of simplicity in this chapter we will informally assume a "well behaved pointset" in the sense that all particles are separated by a finite distance, maintain polynomial unisolvency (see [158]) and are characterized by the fill distance

$$h_{X_h,\Omega} = \sup_{x \in \Omega} \min_{1 \leq i \leq N} ||x - x_j|| \tag{8.5}$$

(i.e. the radius of the largest ball that can fit between data sites). If such an approximation exists then the following pointwise approximation result holds:

$$|u(x) - u_h(x)| \leq C h_{X_h,\Omega}^{m+1} |u|_{C^{m+1}(\Omega)} \tag{8.6}$$

where the seminorm on the right hand side $|u|_{C^{m+1}(\Omega)} = \max_{|\alpha|=m+1} ||D^\alpha u||_{L^\infty}$. For the purposes of this work we will drop the dependence of the fill distance on the pointset unless relevant ($h = h_{X_h,\Omega}$).

The basis for the moving least squares method is to seek such an approximation as the solution of an optimization problem $u_h(x) = p^*(x)$, where p^* is the minimizer of

$$\min_{p \in \pi_m(\mathbb{R}^d)} \left\{ \sum_{j=1}^{N} \left[u(x_j) - p(x_j) \right]^2 W(||x - x_j||) \right\} \tag{8.7}$$

for a fixed point x, and where $W(r)$ is any positive radially symmetric kernel with support ϵ. The convergence rate of the discretization is independent of choice of W; however if in the limit as ϵ approaches zero W approximates a Dirac delta function, the approximation will interpolate the nodal data while becoming increasing poorly conditioned. In the current work we take $W(r) = \left(1 - \left(\frac{r}{\epsilon}\right)^4\right)_+$. By defining a basis $\mathbf{P} = \{\phi_1, \ldots, \phi_Q\}$ such that $span(\phi_1, \ldots, \phi_Q) = \pi_m(\mathbb{R}^d)$, we can pose this as finding an optimal coefficient vector $\mathbf{c}$ such that $p^* = \mathbf{c}^\intercal \mathbf{P}$. Assuming that a local polynomial reproduction exists, the optimal coefficient vector is given by

$$\mathbf{c}(x) = \mathbf{M}(x)^{-1} \sum_j \mathbf{P}_j^\intercal W(||x - x_j||) \mathbf{u}_j \tag{8.8}$$

where

$$\mathbf{M}(x) = \sum_k \mathbf{P}_k^\intercal W(||x - x_k||) \mathbf{P}_k \tag{8.9}$$

The MLS approximation is therefore given by $u_h(x) = \mathbf{P}^\intercal(x)\mathbf{c}$. To use this approximation to attain estimates of derivatives of u, a direct application of $D^\alpha u(x) \approx D^\alpha u_h(x)$ requires taking matrix derivatives of $\mathbf{M}^{-1}$, due to the implicit spatial dependence $\mathbf{c}$ coming from the $W(||x - x_j||)$ terms. For high order derivatives this is prohibitively expensive.

In the diffuse derivative approximation, the spatial dependence of $\mathbf{c}$ is neglected and derivatives are applied directly to the polynomial basis, i.e.

$$D_h^\alpha u(x) = (D^\alpha \mathbf{P}(x))^\intercal \mathbf{c} \tag{8.10}$$

Mirzaei has shown [106] that this approximation yields the following error estimate

$$||D^\alpha u - D_h^\alpha u||_{L^\infty(\Omega)} \leq C(m) h_{X_h,\Omega}^{m+1-|\alpha|} ||u||_{W^{m+1,p}(\Omega)} \tag{8.11}$$

which, in comparison to results analyzing the full derivative [3][170], demonstrates that the diffuse derivative assumption preserves the optimal $m+1-|\alpha|$ convergence rate.

If one only evaluates these derivatives at the particle locations, then this provides an efficient means to estimate derivatives for all $x_i \in \mathbf{X}_h$. Selecting at node i as a local polynomial basis $\mathbf{P}_i(x)$ the Taylor monomials scaled by the kernel support ϵ for conditioning purposes (using multi-index notation)

$$\phi_\alpha(x) = \frac{1}{\alpha!}\left(\frac{x - x_i}{\epsilon}\right)^\alpha \tag{8.12}$$

We note that the derivative operator is only nonzero when the multi-index of the derivative is the same as that of the monomial, i.e.

$$D^{\beta}\mathbf{P}_i = \hat{e}_{\beta}\epsilon^{-\beta} \tag{8.13}$$

where $\hat{e}_{\beta}$ is the canonical basis vector consisting of zeros with a one in the entry corresponding to multi-index β. Finally, we see that after constructing and inverting the correction matrix

$$\mathbf{M}_i = \sum_k \mathbf{P}_i(x_k)^{\intercal} W_{ik} \mathbf{P}_i(x_k) \tag{8.14}$$

all of the diffuse derivatives become available at node i by performing a single inner product between the row corresponding to α and $\mathbf{P}_i$:

$$D_h^{\alpha} u_i = \epsilon^{-\alpha} \hat{e}_{\alpha}^{\intercal} \mathbf{M}_i^{-1} \sum_j \mathbf{P}_i(x_j) W_{ij} u_j \tag{8.15}$$

where $W_{ij} = W(||x_i - x_j||)$.

The use of the Taylor monomials as a basis is efficient but the spectrum of the resulting normal equations matrix (Equation 8.14) resembles the well-known poorly conditioned Hilbert matrix, which typically requires special care to invert. To handle this ill-conditioning, QR or SVD decomposition can be used to invert this matrix stably or the Moore-Penrose pseudo-inverse may be used to stabilize nearly singular values[152, 106]. For high order polynomial reconstruction this ill-conditioning can be avoided by selecting instead the tensor product of the Legendre basis scaled by the kernel support length.

$$\phi_{\alpha(i,j)}(x, y) = P_i(\frac{x - x_i}{\epsilon}) P_j(\frac{y - y_i}{\epsilon}) \tag{8.16}$$

where $\alpha = 1, \cdots, m^2$ and $P_i(x)$ denotes the i^{th} order 1D Legendre polynomial. This

stabilizes the correction matrices but requires more neighbors relative to the Taylor basis. For the remainder of this work unless otherwise noted, the Taylor monomials have been found to be sufficiently stable for up to 6^{th} order polynomial reconstruction and are used for efficiency reasons.

8.2.2 Collocation

Because the MLS process generates rational basis functions, a lack of efficient quadrature rules make Galerkin-type schemes, such as the element-free Galerkin method[12], expensive and motivate the use of a collocation framework. Variations of the moving least squares approach used in this setting can be found in the literature under various names (e.g. generalized finite differences[90], finite point method[116], and others[105, 77]). In this work, we alter the optimization problem slightly to enforce interpolation at the nodes by removing the constant vector from the polynomial basis, i.e. for each i solve

$$\min_{p\in\pi_m(\mathbb{R}^d)\setminus\vec{1}} \left\{ \sum_{j=1}^{N} \left[(u_j - u_i) + p(x_j)\right]^2 W_{ij} \right\} \tag{8.17}$$

This allows a simple implementation of Dirichlet boundary conditions and removes the need for mass matrices. So for each particle, we sweep through every point, construct and invert the following correction matrix

$$\mathbf{M}_i = \sum_k \mathbf{P}_i(x_k)^\intercal W_{ik} \mathbf{P}_i(x_k) \tag{8.18}$$

where now $\mathbf{P}$ contains the $Q-1$ basis functions of order m excluding the constant vector. Derivatives are then reconstructed via

$$D_h^\alpha u_i = D_h^\alpha \mathbf{P}_i^\intercal \mathbf{M}_i^{-1} \sum_j \mathbf{P}_i(x_j) W_{ij} \left(u_j - u_i\right) \tag{8.19}$$

In order to use this to discretize a linear boundary value problem on Ω

$$\begin{cases} \mathcal{L}_1 u = f, & \text{if } x \in \Omega \\ \mathcal{L}_2 u = g, & \text{if } x \in \partial\Omega \end{cases} \tag{8.20}$$

we can simply distribute a "well-behaved" pointset X_Ω and solve the following equation for each particle x_i

$$\begin{cases} \mathcal{L}_1^h u(x_i) = f(x_i), & \text{if } x \in \Omega \\ \mathcal{L}_2^h u(x_i) = g(x_i), & \text{if } x \in \partial\Omega \end{cases} \tag{8.21}$$

where the linear operators $\mathcal{L}_1^h$ and $\mathcal{L}_2^h$ are discretized using Equation 8.19. In general the resulting linear system will be sparse but asymmetric and can be solved with standard Krylov subspace methods such as BiCGSTAB or GMRes [152]. While this approach is efficient and applicable to general pointsets and complicated geometries, it suffers several drawbacks common to meshless methods. First, maintaining invertible correction matrices will require having a sufficiently large number of neighbors that Equation 8.17 has a solution. This necessitates the use of large stencils, increasing the cost of iterative methods and in practice limiting the feasibility of this approach to second or fourth order. Second, as with standard finite difference methods, near the boundaries the approximation becomes one-sided. Although this does not affect the convergence rate, the magnitude of error is concentrated at boundaries. A final and more subtle drawback is that this approach, like most finite difference-type methods, does not maintain any mimetic properties of the continuous PDE (for example $\nabla \cdot \nabla \times \cdot \mathbf{u} \neq 0$). Splitting schemes commonly used in Galerkin for-

mulations (see e.g. [58]) often rely on these properties and when implemented in this context suffer in terms of accuracy. It is therefore desirable to either develop mimetic discretizations (which often requires reintroducing a mesh)[88] or to use a high order discretization to better approximate the continuous operators. Further, near boundaries Galerkin methods provide a compatibility condition between the governing PDE and the boundary conditions via integration by parts. For finite difference methods these relations generally only have an interpretation in terms of linear algebra [142].

8.3 Compact moving least squares

8.3.1 Optimization framework

Rather than take a larger stencil to achieve high order, the strategy of the CMLS approach is to tailor the approximation to the PDE to be discretized. With the intention of solving the boundary value problem in Equation 8.20, we pose the following least squares problem for the optimal polynomial reconstruction at each node i:

$$\begin{aligned}\min_{p\in\pi_m(\mathbb{R}^d)\setminus\vec{1}} \sum_j \Big\{ \big[\left((u_j - u_i) - p_j\right)^2 + \epsilon_1 \left(f_j - \mathcal{L}_1 p_j\right)^2 \\ + \mathbb{1}_{x_j\in\partial\Omega}\epsilon_2 \left(g_j - \mathcal{L}_2 p_j\right)^2 \big] W(||x_i - x_j||) \Big\}\end{aligned} \tag{8.22}$$

Here ϵ_1 and ϵ_2 are penalty parameters and $\mathbb{1}_{x_j\in\partial\Omega}$ is an indicator taking unit value for boundary particles and zero value elsewhere. The three penalty terms reflect (from left to right): how close the reconstruction is to an interpolant, how faithful it

is to the PDE, and how faithful it is to the boundary conditions for nearby points. The choice of the penalty parameters ϵ_1 and ϵ_2 provide control over the relative importance of these three competing objectives, and must scale with ϵ such that each term is of comparable magnitude as resolution is increased. For particles lying on the boundary, we add the constraint that the boundary condition is satisfied exactly,

$$\mathcal{L}_2 p_i = g_i \tag{8.23}$$

so that the boundary condition is satisfied exactly at the point i and approximately at the points j through the third penalty term.

As this is still a least squares problem, the resulting algorithm closely mirrors the process from the previous section; first construct and invert a single correction matrix for each particle i,

$$\mathbf{M}_i = \sum_j \left(\mathbf{P}_j \mathbf{P}_j^\intercal + \epsilon_1 (\mathcal{L}_1 \mathbf{P}_j)(\mathcal{L}_1 \mathbf{P}_j^\intercal) + \mathbb{1}_{x_j \in \partial\Omega} \epsilon_2 (\mathcal{L}_2 \mathbf{P}_j)(\mathcal{L}_2 \mathbf{P}_j^\intercal) \right) W(||x_i - x_j||) \tag{8.24}$$

which can be used to compute the optimal coefficients,

$$\mathbf{c}_i = \mathbf{M}_i^{-1} \sum_j \left(\mathbf{P}_j^\intercal (u_j - u_i) + \epsilon_1 \mathcal{L}_1 \mathbf{P}_j^\intercal f_j + \mathbb{1}_{x_j \in \partial\Omega} \epsilon_2 \mathcal{L}_2 \mathbf{P}_j^\intercal g_j \right) W(||x_i - x_j||) \tag{8.25}$$

Then estimate derivatives using the diffuse derivative concept as

$$D_h^\alpha u(x) = (D^\alpha \mathbf{P}(x))^\intercal \, \mathbf{c} \tag{8.26}$$

So as to enforce the boundary condition constraint, the correction matrices are augmented to handle an additional degree of freedom for a Lagrange multiplier and the

optimal coefficients are extracted from the solution of the system $\hat{\mathbf{M}}_i \hat{\mathbf{c}}_i = \hat{\mathbf{b}}_i$, where

$$\hat{\mathbf{M}}_i = \begin{bmatrix} \mathbf{M}_i & \mathcal{L}_2 \mathbf{P}_i \\ \mathcal{L}_2 \mathbf{P}_i^\intercal & 0 \end{bmatrix} \tag{8.27}$$

$$\hat{\mathbf{c}}_i^\intercal = \left[\begin{array}{c|c} \mathbf{c} & \lambda \end{array} \right] \tag{8.28}$$

$$\hat{\mathbf{b}}_i^\intercal = \left[\begin{array}{c|c} \sum_j \left(\mathbf{P}_j^\intercal (u_j - u_i) + C_1 \mathcal{L}_1 \mathbf{P}_j^\intercal f_j + C_2 \mathcal{L}_2 \mathbf{P}_j^\intercal g_j \right) W(x_i - x_j) & g_i \end{array} \right] \tag{8.29}$$

At this point the PDE can be discretized at all points, including the boundary, as

$$\mathcal{L}_1^h u(x_i) = f(x_i), \forall x_i \in \Omega \cup \partial\Omega \tag{8.30}$$

since the boundary conditions have already been enforced naturally when building the reconstruction and do not need to be enforced as an additional equation in contrast to Equation 8.21.

By incorporating the PDE into the approximation process, the amount of information per node at each point is substantially increased. For interior points, the correction matrices are invertible with roughly half as many neighbors, allowing high order reconstruction with a compact stencil and small bandwidth in the resulting discretized linear system. For boundary points, the extra information coming from the third penalty term in Equation 8.22 alleviates the one-sidedness of the approximation near boundaries, while the constraint ensures compatibility between the PDE and the boundary condition along the boundary. The importance of this compat-

ibility is subtle but will be later demonstrated to be important in fluid mechanics problems with respect to mass conservation.

In comparison to standard MLS, the current compact approach asymptotically requires no additional work, and for a given order polynomial reconstruction will be shown to actually be faster. The compact scheme requires only the addition of two additional terms in Equations 8.24 and 8.25 compared to Equations 8.18 and 8.19, but the more compact support leads to fewer terms in the sums of each equation. Therefore, at worst the construction of stencils for each point is perhaps two times more expensive, but the process is entirely local. Meanwhile, the more compact stencil leads to a sparser global linear system and ultimately a faster method in addition to being more accurate. For parallel implementations, this is ideal; the increased ratio of global to local work with less communication due to sparsity leads to a more scalable algorithm. We present performance results highlighting this in Section 8.5.

The idea of incorporating the PDE into the reconstruction process has been utilized previously in a finite difference context in the two schemes discussed in Section 8.1. Both of these approaches require a uniform mesh or exact estimates of the truncation error which are impractical to obtain for general particle distributions. For the current approach, no information is necessary other than the underlying PDE. This new framework is also flexible in the sense that additional properties can be built into the scheme by applying additional constraints. For example, Seibold [136][138] was able to achieve an M-matrix structure in the Poisson problem by applying inequality constraints and adopting an L1 minimization. The approach can be extended to give even more compact stencils by using Hermite nodal data following [85] [144].

8.3.2 Truncation error comparison

To demonstrate the behavior of the truncation error in this process, we artificially generate a discretization that would correspond to solving a pressure Poisson equation with Neumann data.

$$\nabla^2 p = f \tag{8.31}$$

$$\partial_n p = g \tag{8.32}$$

This corresponds to taking $\mathcal{L}_1 = \nabla^2$, $\mathcal{L}_2 = \partial_n$, $p = sin(x)cos(y)$, $f = \nabla^2 p$, and $g = \partial_n p$. To ensure equal magnitude across the three penalty terms in the optimization problem we select penalty parameters to match the dimensions of the Poisson and first derivative operators squared, i.e. $\epsilon_1 = C\epsilon^4$ and $\epsilon_2 = C\epsilon^2$ for C = 0.001. We discretize a rectangle of width W and height H with a circle and square of radius $H/2$ removed from its interior to demonstrate error behavior near curved and corner geometries (See Figure 8.1). For a spacing of $dx = H/N$, a Cartesian grid of particles are generated within the rectangle and particles lying outside of the domain or on the boundary are removed. Boundary particles are then introduced with spacing dx. The interior particles are then perturbed by uniform random variable $\sim [-\chi, \chi] \times [-\chi, \chi]$ to demonstrate a lack of sensitivity to particle anisotropy. The kernel cutoff radii used in both methods for increasing polynomial order are presented in Table 8.1. Because we will use a uniform kernel size for all particles, the results are overly conservative; only particles near boundaries where a one-sided approximation is formed need this large cutoff. In the interior where the approximation resembles a centered difference with twice as many available particles, a cutoff of approximately half this size could be used.

By increasing the number of particles N, Figure 8.2 demonstrates the expected

m	2	3	4	5	6	7	8	9
ϵ/dx (MLS)	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
ϵ/dx (CMLS)	1.5	2.5	2.5	3.5	3.5	4.5	4.5	5.5

Table 8.1: Cutoff radii for MLS and CMLS for increasing order of polynomial space $\pi_m(\mathbb{R}^D)$. For a given order, MLS requires roughly twice the support of CMLS.

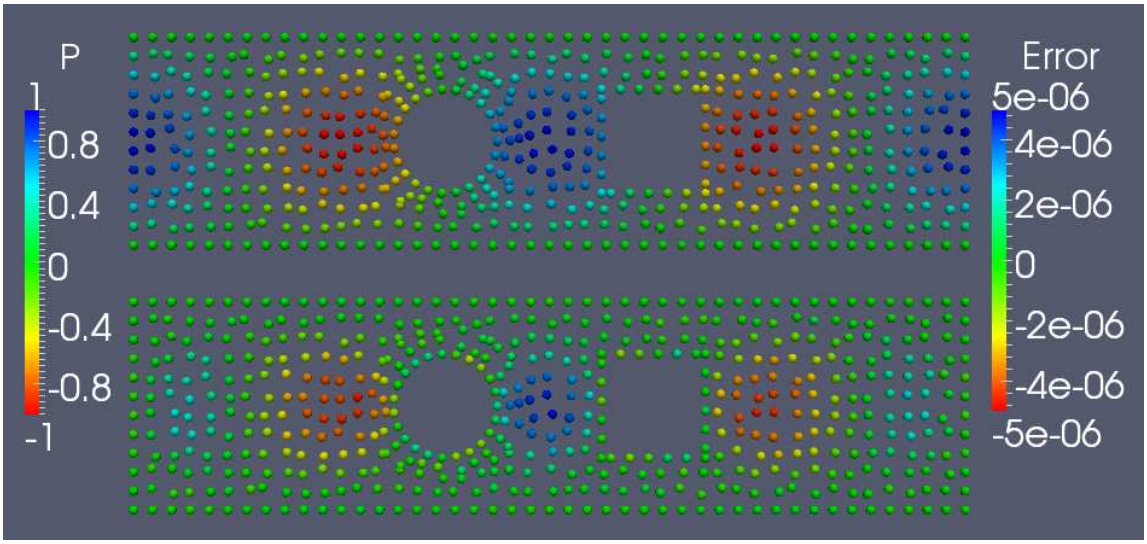


Figure 8.1: p (top) and truncation error $||\nabla^2 p - \nabla_h^2 p||$ (bottom) using 7^{th} order polynomial space. $H \times L$ geometry discretized with initially uniform Cartesian particles $H/dx = 12$ for $\chi = 0.2dx$.

algebraic convergence for MLS and CMLS. Although both methods demonstrate the same convergence rate for a given polynomial order, CMLS achieves accuracy several orders of magnitude smaller with half the stencil size until saturating near machine precision. Taking values of $\chi \in [0 * dx, 0.4 * dx]$ give nearly identical results. Plots of the truncation error in Figure 8.1 show that the error is concentrated near gradients in the underlying function and not near boundaries or corners.

Alternatively, we may fix the number of particles $N = 12$ and increase the order of the polynomial reconstruction. For this relatively coarse discretization (Figure 8.1) there are only a couple particles across the gap between the circle and square and the outer boundary. We see spectral-like convergence for both the gradient and Laplacian (Figure 8.3) in the sense that the error decreases exponentially with

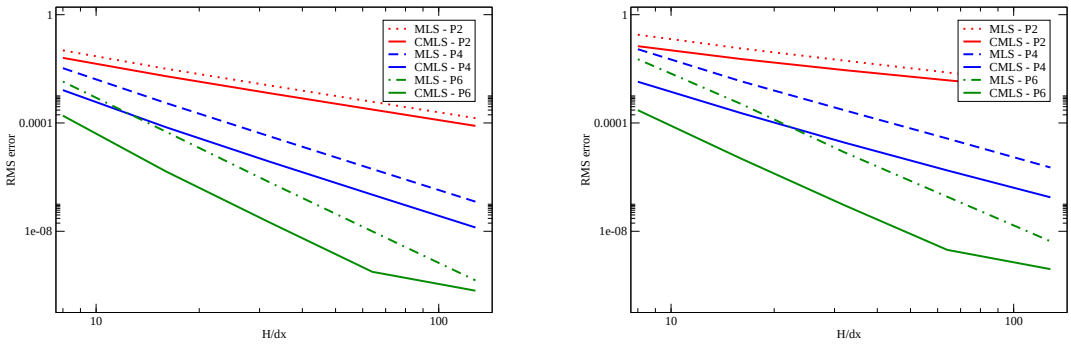


Figure 8.2: Truncation error for gradient operators (left) and Laplacian operators (right) for standard MLS (dashed lines) and CMLS (solid lines).

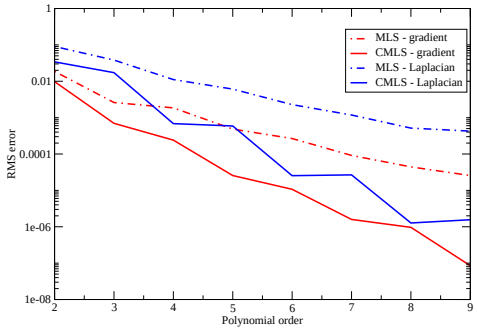


Figure 8.3: Spectral-like convergence for fixed resolution $H/dx = 12$ and increasing polynomial order.

polynomial order, although the support increases with m and is therefore not exactly spectral. For a fair comparison between MLS and CMLS, we compare accuracy for a given stencil size. For example, for $\epsilon = 4.5dx$ we compare 4^{th} order MLS with 7^{th} order CMLS to see an increased accuracy of three orders of magnitude. Computationally, this p-refinement approach is preferable to the h-refinement, in the sense that this increased accuracy is achieved only by inverting larger matrices at every particle. This process is entirely local and will provide better parallel scaling as compared with the domain decomposition approaches that would be necessary in h-refinement.

8.4 Lele-type derivatives

While the approach in the previous section more closely resembles the Weinan-type scheme discussed in Section 8.1, there are many scenarios in which a derivative needs to be computed but a PDE is unavailable. For these problems, a Lele-type approach is preferable, where the derivatives are determined by solving a global problem to relate nodal function values to derivatives.

To calculate a derivative $D^\alpha u$, we select $\mathcal{L}_1 = D^\alpha, \mathcal{L}_2 = 0, \epsilon_1 = C\epsilon^{2\alpha}, \epsilon_2 = 0, \mathbf{f}_i = D^\alpha u_i$ and generate a local reconstruction in the same manner as the previous sections. This gives an expression for the reconstruction of the form

$$u_h(x; x_i) = \mathbf{P}^\intercal(x) \sum_j \left(\alpha_{ij} D^\alpha u_j + a_{ij}(u_j - u_i)\right) \tag{8.33}$$

where for simplicity we have lumped the terms in Equation 8.25 into the coefficients α_{ij} and a_{ij}. If we then require that the diffuse derivative matches the nodal data

$$D^\alpha u_i = D^\alpha \mathbf{P}_i^\intercal \sum_j \left(\alpha_{ij} D^\alpha u_j + a_{ij}(u_j - u_i)\right) \tag{8.34}$$

we obtain a relationship between the derivatives and the nodal data of the same form as Equation 8.4. Further, since the optimization problem Equation 8.22 exactly reproduces polynomials, we have obtained a scheme of order $m + 1$.

To highlight the similarity between this and Lele's approach, we use a symbolic computation library to solve for the coefficients α_{ij} and a_{ij} analytically for a simple 1D case. We consider either three or five point stencils with uniform spacing h and obtain the following result for a choice of $C = 1$ and $W = 1$.

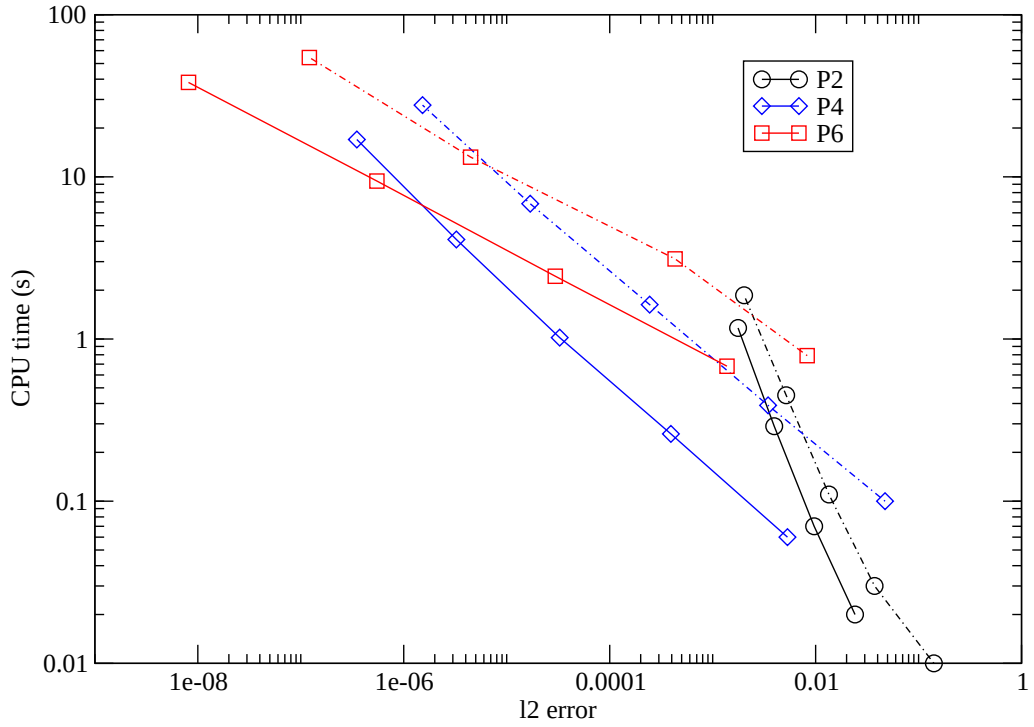


Figure 8.4: Necessary CPU time to obtain a given truncation error for varying order reconstruction. Standard MLS given by dashed lines while compact MLS given by solid lines.

$$u'_{i+1} + 4u'_i + u'_{i-1} = 3\left(\frac{u_{i+1} - u_{i-1}}{h}\right) + O(h^4) \tag{8.35}$$

$$6u'_{i+2} + 96u'_{i+1} + 216u'_i + 96u'_{i-1} + 6u'_{i-2} = \frac{25\,(u_{i+2} - u_{i-2}) + 160\,(u_{i+1} - u_{i-1})}{h} + O(h^8) \tag{8.36}$$

		CMLS			**MLS**		
	m	4	6	8	2	4	6
	$\epsilon/\mathbf{dx}$	2.5	4.5	6.5	2.5	4.5	6.5
$\frac{\mathbf{H}}{\mathbf{dx}}$	8	3.71E-2	4.99E-3	1.38E-3	8.40E-2	2.93E-2	1.06E-2
	16	8.75E-4	4.58E-5	3.91E-6	2.13E-2	1.75E-3	1.95E-4
	32	3.34E-5	6.59E-7	1.70E-8	5.44E-3	1.10E-4	2.76E-6
	64	1.53E-6	8.11E-9	2.99E-9	1.37E-3	6.80E-6	4.33E-8
Conv. rate		-4.442	-6.344	-8.464	-1.972	-3.996	-5.996

Table 8.2: Truncation error for $grad_h \sin x \sin y$ for Lele-type CMLS and standard MLS with identical support. Convergence rate for 8^{th} order reconstruction is calculated before error saturates around 10^{-7}.

For this choice of weights, Lele's original formulas [84] for tridiagonal and penta-

diagonal schemes for the first derivative are exactly recovered, and different choices of C and W will yield a family of solutions that are optimal in the sense of Equation 8.22. Because the coefficients of the stencil are generated as the solution of an optimization problem however and require no a priori calculation, the current approach is applicable to one-sided stencils near boundaries and on general pointsets. We revisit the case discussed previously (Figure 8.1) and present the resulting truncation error when Equation 8.34 is used to calculate the gradient of the function $u = \sin x \sin y$ in Table 8.2 and compare to the truncation error using standard MLS.

Because no special treatment is used near the boundary (i.e. $\epsilon_2 = 0$) a slightly larger stencil size is required than in the previous section. Also this approach was found to be more sensitive to the poor conditioning of Equation 8.14 and the Legendre basis was used to achieve high order reconstruction. The resulting global system of equations were found to be extremely well-conditioned. When using GMRes with diagonal preconditioning to solve the problem[152], for a given order reconstruction the above results all converged to a fixed solver tolerance in a roughly constant number of iterations indepedent of the degrees of freedom in the global system. Therefore, while for standard compact finite differences the banded structure of the discretized system allows application of direct solvers in $O(N)$ time, the current approach is able to maintain the same algorithmic complexity using an iterative solver. Figure 8.4 demonstrates that even with the additional global solve, the compact formulation is faster than standard MLS since both methods spend more time computing the correction factors than solving the global system; for all results presented here less than 1% of the time was spent solving the global system.

8.5 Compact moving least squares for scalar PDE

To actually solve PDEs using this discretization we first solve simple 1D elliptic problems. A Helmholtz decomposition problem is then used to illustrate how the compatibility of the discretization provides better answers for fluid mechanics problems. Finally a pressure correction splitting scheme is implemented to demonstrate results for a simple lid-driven cavity flow.

8.5.1 1D examples

We begin by demonstrating the convergence of the CMLS method for solving the Poisson and Helmholtz problems with either Dirichlet or Neumann boundary conditions in a one-dimensional context. To solve the Poisson equation on the interval $[0, 2\pi]$,

$$\partial_{xx} u = -2 \sin x \tag{8.37}$$

we take for the first penalty operator $\mathcal{L}_1 = \partial_{xx}$ and $f = -2sin(x)$ with $\epsilon_1 = C\epsilon^4$ and $C = 0.001$. For Neumann or Dirichlet boundary conditions we take either: $\mathcal{L}_2 = \mathbf{I}$, $g = sin(x)$, and $\epsilon_2 = 1$ or $\mathcal{L}_2 = \partial_n$, $g = \hat{n} \cdot cos(x)$, and $\epsilon_2 = C \cdot h^2$, respectively. To solve the Helmholtz equation

$$u - \partial_{xx} u = -sin(x) \tag{8.38}$$

we lump the first term on the left hand side into f as follows: $\mathcal{L}_1 = -\partial_{xx}$ and $f = -sin(x) - u$ with $\epsilon_1 = C\epsilon^4$ and $C = 0.001$ and handle the boundary conditions identically as with the Poisson case. Both equations have as an exact solution $u_{ex} = sin(x)$. For all problems the kernel radii from Table 8.1 are used. A representative

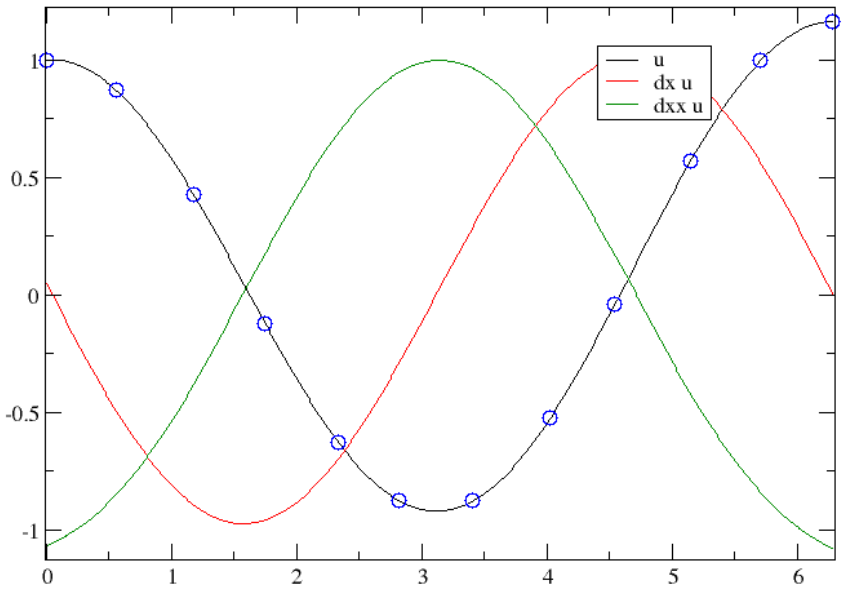


Figure 8.5: Representative post-processed solution of Poisson problem with Neumann boundary conditions for $m = 4$.

solution for the Poisson problem with Neumann boundary conditions using a quartic polynomial space is provided in Figure 8.5.

For this simple 1D problem, LU decomposition with partial pivoting can be used to solve the global system of equations directly. All combinations of the Poisson/Helmholtz and Neumann/Dirichlet problems provide the expected algebraic convergence (Figure 8.6).

8.5.2 Pressure projection

In all projection methods for solving the Stokes or Navier-Stokes equations, at some point in the scheme a Helmholtz decomposition is performed to decompose a velocity

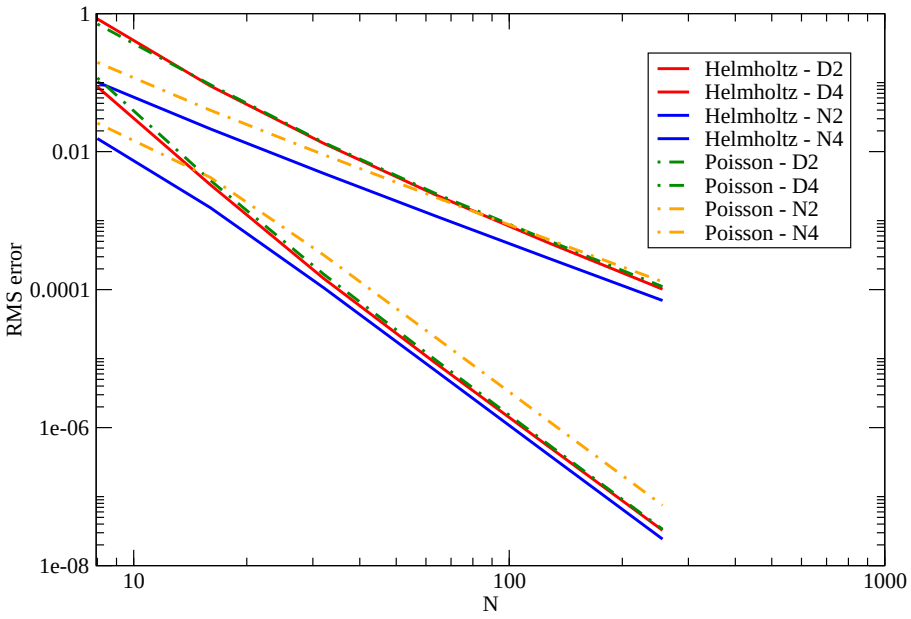


Figure 8.6: Second and fourth order convergence for Poisson and Helmholtz problems with Dirichlet (D), or Neumann (N) BCs and m = 2 or 4.

field $\mathbf{u}$ into a divergence-free part $\mathbf{u}^*$ and a gradient of a scalar field that is identified as the pressure.

$$\mathbf{u} = \mathbf{u}^* + \nabla p \tag{8.39}$$

Either taking the divergence or taking the dot product with the normal provides a Poisson problem or Neumann boundary conditions, respectively.

$$\nabla \cdot \nabla p = \nabla \cdot \mathbf{u} \tag{8.40}$$

$$\partial_n p = \hat{n} \cdot (\mathbf{u} - \mathbf{u}^*) \tag{8.41}$$

Discretizing this by taking the composition of the discrete divergence and gradient operators will provide a velocity with discrete divergence equal to zero. In practice compatibility is required between the pressure and velocity operators to avoid spurious solutions. Approximating this with the discrete Laplacian circumvents this issue, but the resulting velocity is only divergence free up to the truncation error of the discretization because $\nabla_h^2 \neq \nabla_h \cdot \nabla_h$ for finite difference type schemes. These two approaches are referred to as exact and approximate projections in the literature. Although classical MLS can be used to solve the Poisson problem accurately, the resulting error in the divergence from performing the approximate projection introduces large errors in the divergence near the boundary. To illustrate this, we discretize the circle and square channel geometry from the previous section with MLS and plot the resulting concentration of error in Figure 8.7, taking as an initial

divergence-free velocity

$$\mathbf{u}^* = \langle \sin x \cos y, -\cos x \sin y \rangle \tag{8.42}$$

and

$$\mathbf{u} = \mathbf{u}^* + \langle \sin x, 0 \rangle \tag{8.43}$$

.

The discrete system is solved using algebraic multigrid and Lagrange multipliers to remove the singularity associated with the fact that the solution is specified up to an arbitrary constant. Although the pressure converges optimally, we present in Table 8.3 the convergence of the RMS errors for the velocity and the divergence. The error concentrated near boundaries (See Figure 8.7) contaminates the overall convergence rate and suboptimal convergence is obtained. For high order polynomial spaces, the problem is poorly conditioned and doesn't converge to a meaningful solution.

	m	2	4	6
For Laplacian	h/dx - MLS	2.5	4.5	6.5
For divergence calculation	h/dx - MLS	2.5	4.5	6.5
velocity RMS error	level 1	0.459968	0.0603222	crash
	level 2	0.0295923	0.00273707	crash
	level 3	0.0102574	0.000132552	crash
Order convergence		-1.53	-4.35	n/a
divergence RMS error	level 1	1.13786	0.353851	crash
	level 2	0.144321	0.0116065	crash
	level 3	0.0973838	0.00117144	crash
Order convergence		-0.57	-3.32	n/a

Table 8.3: Convergence of Helmholtz projection for classical MLS. Results for the sixth order case are unavailable because the iterative solver failed to converge.

If instead CMLS is used, the convergence is regular with no concentration of divergence in the projected velocity field. Figure 8.8 demonstrates the uniform con-

	m	2	4	6
For Laplacian	h/dx - CMLS	1.5	2.5	3.5
For divergence calculation	h/dx - MLS	2.5	4.5	6.5
velocity RMS error	level 1	0.0724882	0.0118253	0.00327502
	level 2	0.0164781	0.000652156	4.95633e-05
	level 3	0.00392025	3.88768e-05	7.12517e-07
Order convergence		-2.03	-4.07	-6.11
divergence RMS error	level 1	0.101949	0.0177057	0.00485005
	level 2	0.0265572	0.00106521	6.95958e-05
	level 3	0.00672531	6.53083e-05	1.05734e-06
Order convergence		-1.97	-4.02	-6.05

Table 8.4: Convergence of Helmholtz projection for CMLS with $\epsilon_1 = 0.001h^4$ and $\epsilon_2 = 0.001h^2$.

	m	2	4	6
For Laplacian	h/dx - CMLS	1.5	2.5	4.5
For divergence calculation	h/dx - CMLS2	1.5	2.5	4.5
velocity RMS error	level 1	0.0675608	0.00441102	0.0035661
	level 2	0.0153701	0.000241528	6.04593e-05
	level 3	0.0036496	1.40698e-05	7.84013e-07
Order convergence		-2.03	-4.10	-6.26
divergence RMS error	level 1	0.0980297	0.00956967	0.00560728
	level 2	0.0249954	0.000460916	8.98542e-05
	level 3	0.00628819	2.90169e-05	1.263e-06
Order convergence		-1.97	-3.99	-6.15

Table 8.5: Convergence of Helmholtz projection for CMLS with $\epsilon_1 = 0.001h^4$ and $\epsilon_2 = 0.001h^2$. Divergence is calculated with compact Lele-type MLS discretization using $C = 0.01$.

	m	2	4	6
For Laplacian	h/dx - MLS	2.5	4.5	6.5
For divergence calculation	h/dx - MLS	2.5	4.5	6.5
velocity RMS error	level 1	0.0936412	0.0196564	0.408108
	level 2	0.0284358	0.000917299	7.29934e-05
	level 3	0.0328767	4.80272e-05	1.27732e-06
Order convergence		-1.18	-4.25	-5.84
divergence RMS error	level 1	0.141367	0.0310297	1.91815
	level 2	0.0174174	0.00149609	0.00030176
	level 3	0.00752521	0.000102156	6.0867e-06
Order convergence		-1.18	-3.86	-6.13

Table 8.6: Convergence of Helmholtz projection for CMLS with $\epsilon_1 = 0$ and $\epsilon_2 = 0$.

vergence of both the projected velocity and its divergence. For this model problem when forming the divergence on the right hand side we use the standard MLS di-

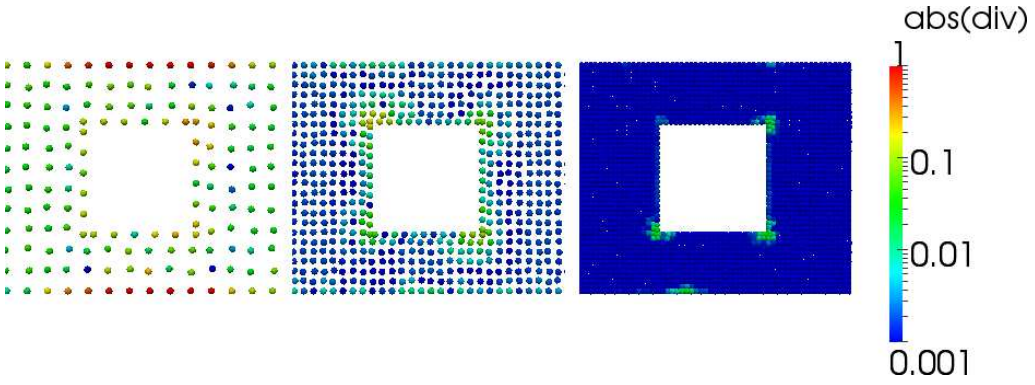


Figure 8.7: Concentration of error in divergence for MLS near sharp corners for H/dx = 12,24,48 and m = 4. Even with high order polynomial space, error in divergence is first order.

vergence, although in an actual projection method the momentum equation may be used to generate compact velocity operators. The convergence of the velocity and divergence errors using CMLS is presented in Table 8.4 and demonstrates optimal convergence. The concentration of error in the MLS results can be attributed to both the one-sidedness of the non-compact operators and the fact that at the boundary, the boundary conditions are imposed rather than the divergence free constraint. In finite element methods, the PDE and boundary conditions are related through Stokes theorems, but for finite difference type methods there is no compatibility and the divergence error appears to scale as $O(h)$ in Figure 8.7. In Table 8.6, we solve the system using the CMLS framework but set the penalty terms to zero. This is equivalent to using the standard MLS operators, but enforcing the boundary conditions via the Lagrange multiplier (Equations 8.27-8.29) so that the PDE can be enforced at the boundary point rather than enforcing the boundary condition.

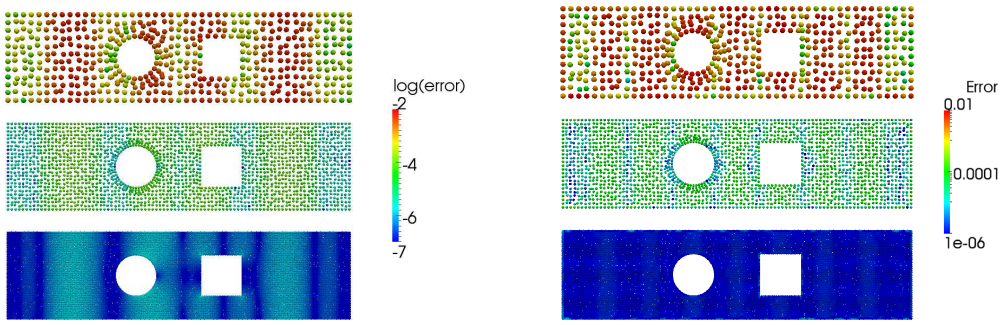


Figure 8.8: Pointwise error for projected velocity magnitude (left) and divergence (right) for CMLS discretization. In comparison to Figure 8.7, there is no concentration of divergence error near boundaries and uniform convergence is obtained across domain.

8.6 Compact moving least squares for vector PDE

8.6.1 Discretization

To solve vector valued PDEs, the optimization problem in Equation 8.22 can be extended for the approximation of the vector $\mathbf{y} \in \mathbb{R}^n$ as

$$\min_{\mathbf{p}\in(\pi_m(\mathbb{R}^d)\setminus\vec{1})^n} \sum_j \{[((\mathbf{y}_j-\mathbf{y}_i)-\mathbf{p}_j)^\intercal((\mathbf{y}_j-\mathbf{y}_i)-\mathbf{p}_j)+(\mathbf{f}_j-\mathcal{L}_1\mathbf{p}_j)^\intercal\epsilon_1(\mathbf{f}_j-\mathcal{L}_1\mathbf{p}_j) \\ +\mathbb{1}_{x_j\in\partial\Omega}(\mathbf{g}_j-\mathcal{L}_2\mathbf{p}_j)^\intercal\epsilon_2(\mathbf{g}_j-\mathcal{L}_2\mathbf{p}_j)]\,W(||x_i-x_j||)\} \tag{8.44}$$

where now the penalty terms ϵ_1 and ϵ_2 are diagonal matrices whose entries are used to scale the three penalty terms appropriately. As an example we consider solutions of the steady Stokes equations with Dirichlet velocity boundary conditions and pressure boundary condition consistent with the PDE.

$$\begin{cases} -\nu\nabla^2\mathbf{u} + \nabla p = f, & \text{if } x \in \Omega \\ \nabla \cdot \mathbf{u} = 0, & \text{if } x \in \Omega \\ \mathbf{u} = \mathbf{w}, & \text{if } x \in \partial\Omega \\ \partial_n p - \nu\hat{n} \cdot \nabla^2\mathbf{u} = \hat{n} \cdot f, & \text{if } x \in \partial\Omega \end{cases} \tag{8.45}$$

We have introduced a pressure boundary condition here because without a mesh and integration by parts, there is no way of efficiently ensuring that the divergence at the boundary is consistent with the velocity boundary conditions. Instead we ensure that the pressure at the boundary is compatible with the velocity boundary condition and the momentum equation. A discussion of these issues in a standard finite difference context can be found in [142].

These equations can be put into the form of Equation 8.44 by taking $\mathbf{y} = \langle \mathbf{u}, p \rangle^\intercal$, $\mathcal{L}_1 = \begin{bmatrix} -\nu\nabla_h^2 & \nabla_h \\ \nabla_h \cdot & 0 \end{bmatrix}$, $\mathcal{L}_2 = \begin{bmatrix} \mathbf{I} & 0 \\ -\nu\hat{n} \cdot \nabla_h^2 & \partial_n \end{bmatrix}$, $\mathbf{f} = \langle f, 0 \rangle^\intercal$, and $\mathbf{g} = \langle \mathbf{w}, \hat{n} \cdot f \rangle^\intercal$, and an approximation at each point is constructed in the usual way. We first calculate the correction matrix as

$$\mathbf{M}_i = \sum_j \left(\mathbf{P}_j \mathbf{P}_j^\intercal + \epsilon_1 (\mathcal{L}_1 \mathbf{P}_j)(\mathcal{L}_1 \mathbf{P}_j^\intercal) + \mathbb{1}_{x_j \in \partial\Omega} \epsilon_2 (\mathcal{L}_2 \mathbf{P}_j)(\mathcal{L}_2 \mathbf{P}_j^\intercal) \right) W(||x_i - x_j||) \tag{8.46}$$

where now $\mathbf{P} \in \mathbb{R}^{n \times Q} = \left\{ \begin{pmatrix} p_1 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} p_Q \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ p_1 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ p_Q \end{pmatrix} \right\}$ is a vector valued polynomial basis for $\left(\pi_m(\mathbb{R}^d)\right)^n$. Solution of this least squares problem gives an optimal coefficient vector

$$\mathbf{c}_i = \mathbf{M}_i^{-1} \sum_j \left(\mathbf{P}_j^\intercal (\mathbf{y}_j - \mathbf{y}_i) + \epsilon_1 \mathcal{L}_1 \mathbf{P}_j^\intercal \mathbf{f}_j + \mathbb{1}_{x_j \in \partial\Omega} \epsilon_2 \mathcal{L}_2 \mathbf{P}_j^\intercal \mathbf{g}_j \right) W(||x_i - x_j||) \quad (8.47)$$

which can be used to approximate differential operators using the diffuse derivative.

$$D_h^\alpha \mathbf{y}_i = (D^\alpha \mathbf{P}_i)^\intercal \mathbf{c}_i \quad (8.48)$$

When used to discretize Equation 8.45, the penalty term couples the approximation of velocity and pressure together; for example the approximation to the Laplacian of the velocity gives an expression of the form

$$\nabla_h^2 \mathbf{u}_i = \sum_j a_{ij} \cdot \mathbf{u}_j + b_{ij} p_j + c_{ij} \mathbf{f}_j \quad (8.49)$$

where for simplicity we've bundled the resulting terms into coefficients a_{ij},b_{ij} and c_{ij}. As a result the fully discretized form of Equation 8.45 assumes the following 2×2 block matrix form.

$$\begin{bmatrix} \mathbf{L} & \mathbf{G} \\ \mathbf{D} & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ g \end{bmatrix}. \quad (8.50)$$

As when solving the scalar Poisson problem, the pressure in this system is only specified up to an arbitrary constant. This is resolved by adding a single Lagrange multiplier and requiring that the pressure have zero mean. This amounts to modifying the block $\mathbf{P}$ by padding with an additional row and column:

$$\begin{bmatrix} \mathbf{P} & \mathbf{1} \\ \mathbf{1}^\intercal & 0 \end{bmatrix} \quad (8.51)$$

where $\mathbf{1}$ denotes a vector of ones and the other blocks are padded with zeros.

To solve the coupled linear system, we use the preconditioned GMRes method. Here, we adopt the lower triangular block preconditioners based on the following block factorization of (8.50). It is well-known that an efficient and robust preconditioner is crucial for the performance of the GMRes method for saddle-point problems [20]. Here we adopt the lower triangular block preconditioner based on the following block factorization of the $2x2$ block system (8.50).

$$\begin{bmatrix} \mathbf{L} & \mathbf{G} \\ \mathbf{D} & \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{D} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{L^{-1}G} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{8.52}$$

where $\mathbf{S} = \mathbf{P} - \mathbf{D}\mathbf{L}^{-1}\mathbf{G}$ is the Schur complement. It is well-known that if we use

$$\begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{D} & \mathbf{S} \end{bmatrix}$$

as the preconditioner, the GMRes method converges in three iterations [13]. This requires the explicit assembly of the Schur complement $\mathbf{S}$ however, which involves inverting $\mathbf{L}$ explicitly. This is expensive, particularly for large-scale problems, as the resulting $\mathbf{S}$ is dense, making the preconditioner prohibitively expensive. Therefore, we approximate the Schur complement by $\widetilde{\mathbf{S}} = \mathbf{P} - \mathbf{D}\ \mathrm{diag}(\mathbf{L})^{-1}\mathbf{G}$ and the lower triangular block preconditioner is given by

$$\begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{D} & \widetilde{\mathbf{S}} \end{bmatrix}.$$

To apply the preconditioner, we still need to invert $\mathbf{L}$ and $\widetilde{\mathbf{S}}$, which could be difficult and expensive. In our implementation, we use the algebraic multigrid (AMG) method

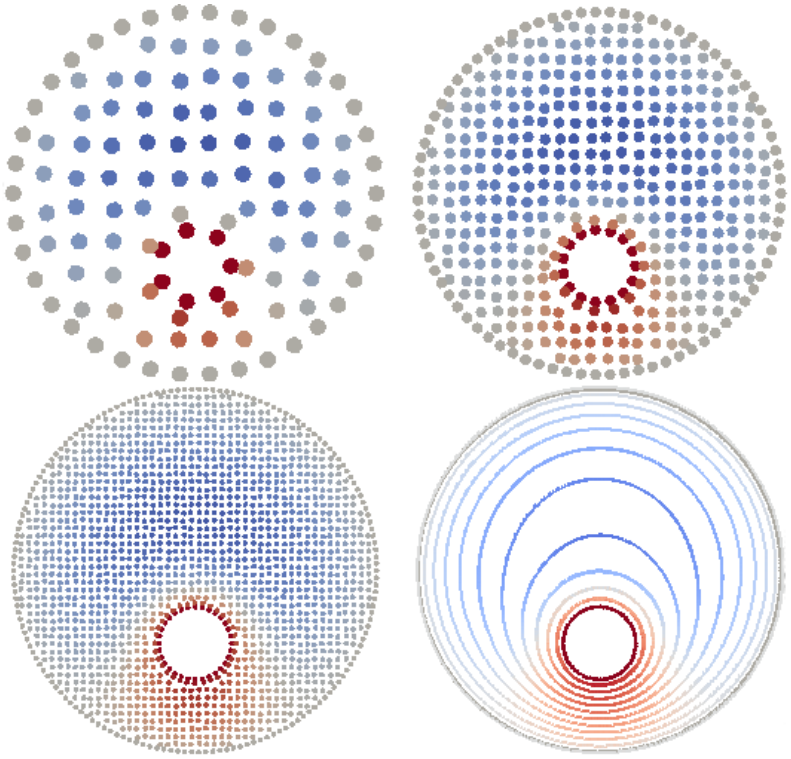
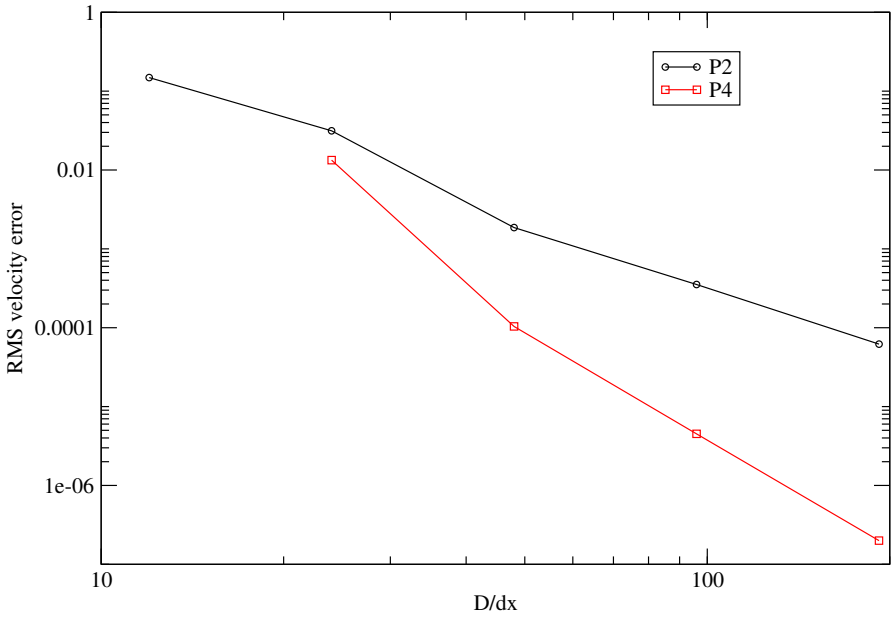


Figure 8.9: Particle distributions for resolutions of $D/\Delta x \in \{12, 24, 48\}$ and streamlines of solution colored by velocity magnitude.

to efficiently solve them both[161].

We use this preconditioner to simulate the so-called Wannier flow of two eccentric rotating cylinders. For this setup an analytic solution exists[156] and provides a standard benchmark for evaluating high-order discretizations of curvilinear geometries with strong viscous boundary layers[74]. For this problem we select as geometric parameters cylinder radii of $R_{inner} = \pi/10$ and $R_{outer} = \pi/2$ and an eccentricity of $e = \pi/5$. The cylinders are set to rotate with an angular velocity of $\Omega_{inner} = 1$ and $\Omega_{outer} = 1/2$. Particles are distributed on an initially uniform lattice with $D/\Delta x$ particles per domain diameter, removed if they fall outside of the domain, and perturbed randomly to remove any misleading accuracy gains from symmetries. Particles are then distributed along the boundary with spacing Δx. Figure 8.9 demonstrates representative particle distributions and the streamlines of the resulting solution, while



F
re

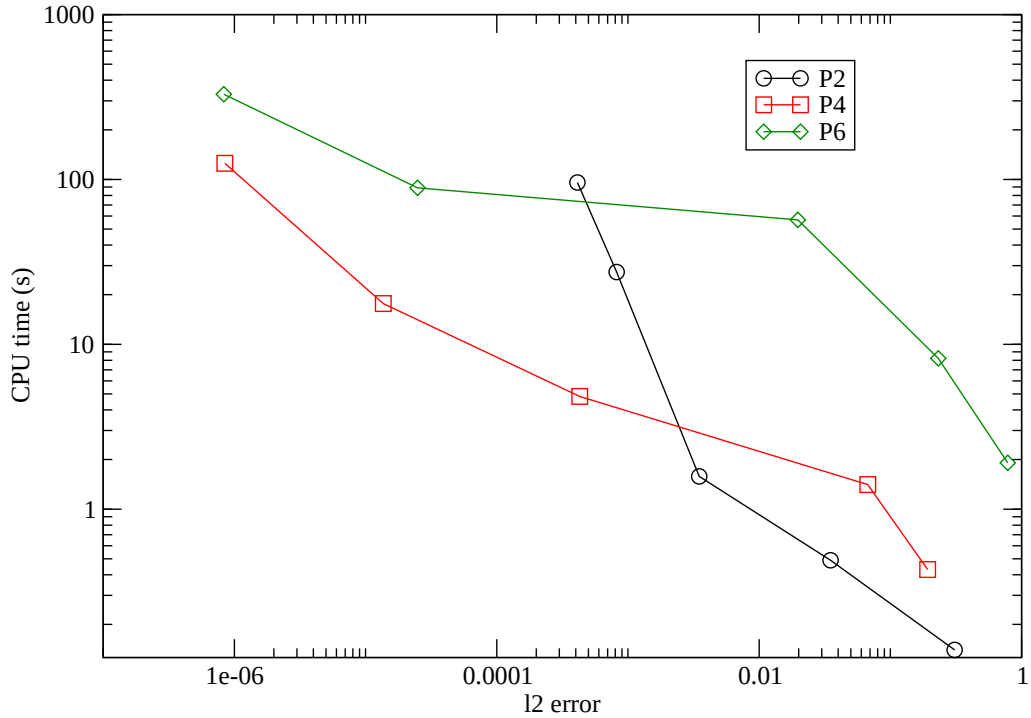


Figure 8.11: Necessary CPU time to obtain a given RMS velocity error for varying order reconstruction.

DOF	Number of Iterations	Setup Time (s)	Solve Time (s)	Total Time (s)
576	36	0.026	0.474	0.500
1,296	52	0.033	0.837	0.870
2,304	51	0.060	1.458	1.518
5,184	67	0.104	5.071	5.175
9,216	75	0.182	9.022	9.204

Table 8.7: Preconditioner performance for stationary Stokes problem with quartic reconstruction (GMRes with stopping criterion that relative residual is less than 10^{-6})

Figure 8.10 demonstrate the convergence of the RMS velocity error. After the curvature of the boundary is adequately resolved, quadratic and quartic reconstruction achieve 2^{nd} and 4^{th} order convergence, respectively. To investigate how best to choose the order of polynomial reconstruction, the required CPU time to achieve a given error tolerance is given in Figure 8.11. For qualitatively accurate solutions of $O(1\%)$ error the quadratic reconstruction provides the most efficient solution in under one second. To obtain several digits of accuracy however the quartic reconstruction is most efficient. Higher order reconstruction appears to provide diminishing returns however where the cost associated with the large dimension of the polynomial basis when inverting the mass matrices becomes prohibitive.

Preliminary performance results for the preconditioner performance for the Wannier flow problem are presented in table 8.7. Although the number of iterations to convergence grows mildly with the total number of particles, the total CPU time grows nearly linearly, demonstrating the viability of the preconditioner. Development of robust and efficient preconditioners for saddle-point systems is in general a difficult task, and we leave a more in depth discussion of performance for future work.

8.7 Conclusions and future work

A flexible optimization framework for generating high-order compact collocation schemes has been presented that generalizes approaches that previously could only be used for uniform geometries. While the intent of the current work has been to present the viability and flexibility of the new approach for a variety of problems, there are a number of questions remaining to develop this approach into a mature method. The results in this work are stable for the choice of penalty parameters presented, choices of larger or smaller penalty parameters degrade the quality of the results. A rigorous analysis of how to optimally select ϵ_1 and ϵ_2 is necessary, particularly for multi-resolution problems in which a uniform scaling of the form $\epsilon = Ch^p$ for all particles may not be viable.

Chapter Nine

Conclusion

In this thesis, we have presented a number of schemes building toward a compatible high-order meshless framework for simulating suspension flows. In the process of doing this, we have developed new discretizations for the div-grad system and for the Stokes equations that provide accuracy and stability competitive with mesh-based methods. In addition to these compatible schemes, we have developed a generalization of compact finite differences that uses regularization terms to develop implicit stencils for differential operators. Preliminary results have shown that these methods can actually be merged: a compact staggered scheme can easily be constructed.

These new discretizations arose out of initial work developing a massive-scale SPH implementation. While the work pursued in this thesis moved away from SPH in the pursuit of higher-order methods with more stability, this SPH code is useful in its own right. The introduction of linear consistency in a scalable framework has provided a useful framework that can be used to solve the Poisson problem and diffusion problems efficiently. While this discretization was not directly applicable to solve Stokes equations, the approach can be used to solve a variety of other applications: our collaborators are currently using this code as a means to study problems in electrokinetics, transport phenomena, and material science.

The discretizations presented mimic exterior calculus structures to obtain enhanced stability and accuracy. While an analysis of truncation error has been possible using local polynomial reproduction, the lack of an appropriate inner product space has made a stability analysis ellusive thus far. We are currently pursuing an extension of traditional exterior calculus to operate on graph topologies. While combinatorial graph theory has found many applications in computer science (see e.g. [73]), they are only being used to analyze the topology of data via Hodge decompositions and have not been used to develop high order schemes for numerical PDE. The encouraging results presented in this thesis suggest that the same com-

bination of finite elements and exterior calculus on simplicial complexes that led to the finite element exterior calculus could be developed for a graphical abstract simplicial complex and moving least squares. By analogy to how we pursued a div-grad model problem to solve the Stokes equation, we could by extension hope to study the curl-curl model problem as a path for the stable numerical solution of the Maxwell equations. Such an extension would also allow a stability analysis of the method and a more rigorous framework to explain the success of these schemes. We are currently in the process of developing an extension of this method in which we have obtained a curl operator with the property that the divergence of the curl is zero.

A key component of any meshless method is the ability to easily introduce a scheme for adaptive refinement. In the present work, we have exploited the fact that for dense colloidal suspensions, we know a priori that the region of the flow requiring extra resolution will occur in the small gap as colloidal particles approach each other. This has allowed us to introduce adaptive resolution by simply adding extra resolution as a numerical boundary layer conforming to each colloid. While this is sufficient for the purposes of demonstrating how adaptivity can be incorporated to this new discretization, the true benefit of a meshless method will become apparent after coupling this approach with the a posteriori error estimators that are standard in adaptive mesh refinement (AMR) codes. In the future we hope to pursue more sophisticated adaptivity.

In a more practical sense, the mixed discretization for the Stokes equations demonstrated here presents an exciting new framework for studying suspension flows with general geometry. While the current approach has been restricted to two dimensions, we are currently developing a parallel block matrix solver so that this approach can be pursued in three dimensions. The ability to study dense colloidal suspensions of arbitrary colloid shape in complex geometry without the need for

sub-grid scale lubrication models is a fundamentally new capability. The additional ability to study interactions between colloids and electromagnetic fields provides a wealth of potential applications.

Bibliography

[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 (2010), pp. 5011–5021.

[2] C. K. Aidun and J. R. Clausen, *Lattice-boltzmann method for complex flows*, Annual review of fluid mechanics, 42 (2010), pp. 439–472.

[3] M. G. Armentano and R. G. Durán, *Error estimates for moving least square approximations*, Applied Numerical Mathematics, 37 (2001), pp. 397–416.

[4] D. Arnold, R. Falk, and R. Winther, *Finite element exterior calculus: from hodge theory to numerical stability*, Bulletin of the American mathematical society, 47 (2010), pp. 281–354.

[5] D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, and M. Shashkov, *Compatible spatial discretizations*, vol. 142, Springer Science & Business Media, 2007.

[6] D. N. Arnold, R. S. Falk, and R. Winther, *Finite element exterior calculus, homological techniques, and applications*, Acta numerica, 15 (2006), pp. 1–155.

[7] P. Arp and S. Mason, *The kinetics of flowing dispersions: Viii. doublets of rigid spheres (theoretical)*, Journal of colloid and interface science, 61 (1977), pp. 21–43.

[8] M. Arroyo and M. Ortiz, *Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods*, International journal for numerical methods in engineering, 65 (2006), pp. 2167–2202.

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

[10] L. BEIRÃO DA VEIGA, F. BREZZI, A. CANGIANI, G. MANZINI, L. D. MARINI, AND A. RUSSO, *Basic principles of virtual element methods*, Mathematical Models and Methods in Applied Sciences, 23 (2013), pp. 199–214.

[11] J. B. BELL, P. COLELLA, AND H. M. GLAZ, *A second-order projection method for the incompressible navier-stokes equations*, Journal of Computational Physics, 85 (1989), pp. 257–283.

[12] T. BELYTSCHKO, Y. Y. LU, L. GU, ET AL., *Element? free galerkin methods*, International journal for numerical methods in engineering, 37 (1994), pp. 229–256.

[13] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta numerica, 14 (2005), pp. 1–137.

[14] X. BIAN AND M. ELLERO, *A splitting integration scheme for the sph simulation of concentrated particle suspensions*, Computer Physics Communications, 185 (2014), pp. 53–62.

[15] X. BIAN, S. LITVINOV, R. QIAN, M. ELLERO, AND N. A. ADAMS, *Multiscale modeling of particle in suspension with smoothed dissipative particle dynamics*, Physics of Fluids (1994-present), 24 (2012), p. 012002.

[16] P. BOCHEV AND R. B. LEHOUCQ, *On the finite element solution of the pure neumann problem*, SIAM review, 47 (2005), pp. 50–66.

[17] P. B. BOCHEV AND M. D. GUNZBURGER, *Least-squares finite element methods*, vol. 166, Springer Science & Business Media, 2009.

[18] P. B. BOCHEV AND J. M. HYMAN, *Principles of mimetic discretizations of differential operators*, in Compatible spatial discretizations, Springer, 2006, pp. 89–119.

[19] J. F. BRADY AND G. BOSSIS, *Stokesian dynamics*, Annual review of fluid mechanics, 20 (1988), pp. 111–157.

[20] D. BRAESS, *Finite elements: Theory, fast solvers, and applications in solid mechanics*, Cambridge University Press, 2007.

[21] T. BREINLINGER, P. POLFER, A. HASHIBON, AND T. KRAFT, *Surface tension and wetting effects with smoothed particle hydrodynamics*, Journal of Computational Physics, 243 (2013), pp. 14–27.

[22] F. BREZZI AND M. FORTIN, *Mixed and Hybrid Finite Element Methods*, Springer, Berlin, 1991.

[23] F. BREZZI, L. D. MARINI, AND P. PIETRA, *Numerical simulation of semiconductor devices*, Computer Methods in Applied Mechanics and Engineering, 75 (1989), pp. 493 – 514.

[24] H. BRUUS, *Theoretical microfluidics. 2008.*

[25] T. Q. Bui, T. N. Nguyen, and H. Nguyen-Dang, *A moving kriging interpolation-based meshless method for numerical simulation of kirchhoff plate problems*, International journal for numerical methods in engineering, 77 (2009), p. 1371.

[26] R. Campbell and E. Paterson, *Fluid–structure interaction analysis of flexible turbomachinery*, Journal of Fluids and Structures, 27 (2011), pp. 1376–1391.

[27] M. A. Case, V. J. Ervin, A. Linke, and L. G. Rebholz, *A connection between scott-vogelius and grad-div stabilized taylor-hood fe approximations of the navier-stokes equations*, SIAM Journal on Numerical Analysis, 49 (2011), pp. 1461–1481.

[28] L.-Q. Chen, *Phase-field models for microstructure evolution*, Annual review of materials research, 32 (2002), pp. 113–140.

[29] W. Chen, Z.-J. Fu, and C.-S. Chen, *Recent advances in radial basis function collocation methods*, Springer, 2014.

[30] E. K.-y. Chiu, Q. Wang, and A. Jameson, *A conservative meshless scheme: general order formulation and application to euler equations*, in AIAA 2011-651 49th Aerospace Sciences Meeting, 2011.

[31] S.-H. Chou, D. Y. Kwak, and P. S. Vassilevski, *Mixed covolume methods for elliptic problems on triangular grids*, SIAM journal on numerical analysis, 35 (1998), pp. 1850–1861.

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

[33] C. J. Cyron, M. Arroyo, and M. Ortiz, *Smooth, second order, non-negative meshfree approximants selected by maximum entropy*, International Journal for Numerical Methods in Engineering, 79 (2009), pp. 1605–1632.

[34] K. Devine, E. Boman, R. Heaphy, B. Hendrickson, and C. Vaughan, *Zoltan data management services for parallel dynamic applications*, Computing in Science & Engineering, 4 (2002), pp. 90–96.

[35] K. Dieter-Kissling, H. Marschall, and D. Bothe, *Numerical method for coupled interfacial surfactant transport on dynamic surface meshes of general topology*, Computers & Fluids, 109 (2015), pp. 168–184.

[36] O. V. Diyankov, *Uncertain grid method for numerical solution of pde*, tech. rep., NeurOK Software, 2008.

[37] 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 (2013), pp. 617–627.

[38] 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 (2013), pp. 1848–1860.

[39] S. Dong and J. Shen, *A time-stepping scheme involving constant coefficient matrices for phase-field simulations of two-phase incompressible flows with large density ratios*, Journal of Computational Physics, 231 (2012), pp. 5788–5804.

[40] P. C. Duffell and A. I. MacFadyen, *Tess: a relativistic hydrodynamics code on a moving voronoi mesh*, The Astrophysical Journal Supplement Series, 197 (2011), p. 15.

[41] V. Dyadechko and M. Shashkov, *Moment-of-fluid interface reconstruction*, Los Alamos National Laborator Report LA-UR-05-7571, (2005).

[42] R. F. Engberg and E. Y. Kenig, *Numerical simulation of rising droplets in liquid–liquid systems: A comparison of continuous and sharp interfacial force models*, International Journal of Heat and Fluid Flow, 50 (2014), pp. 16–26.

[43] P. Espanol and M. Revenga, *Smoothed dissipative particle dynamics*, Physical Review E, 67 (2003), p. 026705.

[44] R. S. Falk and M. Neilan, *Stokes complexes and the construction of stable finite elements with pointwise mass conservation*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 1308–1326.

[45] J. Fang and A. Parriaux, *A regularized lagrangian finite point method for the simulation of incompressible viscous flows*, Journal of Computational physics, 227 (2008), pp. 8894–8908.

[46] G. E. Fasshauer, *Solving partial differential equations by collocation with radial basis functions*, in Proceedings of Chamonix, vol. 1997, Citeseer, 1996, pp. 1–8.

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

[48] J. H. Ferziger and M. Peric, *Computational methods for fluid dynamics*, Springer Science & Business Media, 2012.

[49] S. Ganesan and L. Tobiska, *Arbitrary lagrangian–eulerian finite-element method for computation of two-phase flows with soluble surfactants*, Journal of Computational Physics, 231 (2012), pp. 3685–3702.

[50] L. Gavete, M. Gavete, and J. Benito, *Improvements of generalized finite difference method and comparison with other meshless method*, Applied Mathematical Modelling, 27 (2003), pp. 831–847.

[51] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala, *Ml 5.0 smoothed aggregation users guide*, tech. rep., Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

[52] U. Ghia, K. N. Ghia, and C. Shin, *High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method*, Journal of computational physics, 48 (1982), pp. 387–411.

[53] R. Gingold and J. Monaghan, *Kernel estimates as a basis for general particle methods in hydrodynamics*, Journal of Computational Physics, 46 (1982), pp. 429–453.

[54] R. A. Gingold and J. J. Monaghan, *Smoothed particle hydrodynamics: theory and application to non-spherical stars*, Monthly notices of the royal astronomical society, 181 (1977), pp. 375–389.

[55] R. Glowinski, T. Pan, T. Hesla, D. Joseph, and J. Periaux, *A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow*, Journal of Computational Physics, 169 (2001), pp. 363–426.

[56] M. Gorokhovski and M. Herrmann, *Modeling primary atomization*, Annu. Rev. Fluid Mech., 40 (2008), pp. 343–366.

[57] Y. Gu, Q. Wang, and K. Lam, *A meshless local kriging method for large deformation analyses*, Computer Methods in Applied Mechanics and Engineering, 196 (2007), pp. 1673–1684.

[58] J. Guermond, P. Minev, and J. Shen, *An overview of projection methods for incompressible flows*, Computer methods in applied mechanics and engineering, 195 (2006), pp. 6011–6045.

[59] J. Guzman, M. A. Sanchez, and M. Sarkis, *A finite element method for high-contrast interface problems with error estimates independent of contrast*, arXiv preprint arXiv:1507.03873, (2015).

[60] M. Hashemi, R. Fatehi, and M. Manzari, *Sph simulation of interacting solid bodies suspended in a shear flow of an oldroyd-b fluid*, Journal of Non-Newtonian Fluid Mechanics, 166 (2011), pp. 1239–1252.

[61] M. Heroux, *Epetra performance optimization guide*, Sandia National Laboratories, Tech. Rep. SAND2005-1668, (2005).

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

[63] C. W. Hirt and B. D. Nichols, *Volume of fluid (vof) method for the dynamics of free boundaries*, Journal of computational physics, 39 (1981), pp. 201–225.

[64] 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 (2011), pp. 419–437.

[65] H. H. Hu, *Direct simulation of flows of solid-liquid mixtures*, International Journal of Multiphase Flow, 22 (1996), pp. 335–352.

[66] H. H. Hu, N. A. Patankar, and M. Zhu, *Direct numerical simulations of fluid–solid systems using the arbitrary lagrangian–eulerian technique*, Journal of Computational Physics, 169 (2001), pp. 427–462.

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

[68] X. Hu and N. Adams, *A sph model for incompressible turbulence*, arXiv preprint arXiv:1204.5097, (2012).

[69] X. Hu, P. S. Vassilevski, and J. Xu, *Comparative convergence analysis of nonlinear amli-cycle multigrid*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 1349–1369.

[70] A. Huerta, Y. Vidal, and P. Villon, *Pseudo-divergence-free element free galerkin method for incompressible fluid flow*, Computer Methods in Applied Mechanics and Engineering, 193 (2004), pp. 1119–1136.

[71] S. Idelsohn, E. Onate, and F. Del Pin, *A lagrangian meshless finite element method applied to fluid–structure interaction problems*, Computers & structures, 81 (2003), pp. 655–671.

[72] J.-T. Jeong and S.-H. Yoon, *Two-dimensional stokes flow around a circular cylinder in a microchannel*, Journal of Mechanical Science and Technology, 28 (2014), pp. 573–579.

[73] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye, *Statistical ranking and combinatorial hodge theory*, Mathematical Programming, 127 (2011), pp. 203–244.

[74] G. Karniadakis and S. Sherwin, *Spectral/hp element methods for computational fluid dynamics*, Oxford University Press, 2013.

[75] G. E. Karniadakis, M. Israeli, and S. A. Orszag, *High-order splitting methods for the incompressible navier-stokes equations*, Journal of computational physics, 97 (1991), pp. 414–443.

[76] A. Katz and A. Jameson, *A meshless volume scheme*, AIAA Paper, 3534 (2009), pp. 1–24.

[77] D. W. Kim and Y. Kim, *Point collocation methods using the fast moving least-square reproducing kernel approximation*, International Journal for Numerical Methods in Engineering, 56 (2003), pp. 1445–1464.

[78] H. Kim, J. Xu, and L. Zikatanov, *A multigrid method based on graph matching for convection–diffusion equations*, Numerical linear algebra with applications, 10 (2003), pp. 181–195.

[79] S. Kim and S. J. Karrila, *Microhydrodynamics: principles and selected applications*, Courier Corporation, 2013.

[80] K. Kramer and W. N. G. Hitchon, *Semiconductor Devices: A Simulation Approach*, Prentice Hall, Upper Saddle River, NJ 07458, 1997.

[81] M.-C. Lai and C. S. Peskin, *An immersed boundary method with formal second-order accuracy and reduced numerical viscosity*, Journal of Computational Physics, 160 (2000), pp. 705–719.

[82] P. Lancaster and K. Salkauskas, *Surfaces generated by moving least squares methods*, Mathematics of computation, 37 (1981), pp. 141–158.

[83] C. L. Lawson and R. J. Hanson, *Solving least squares problems*, vol. 161, SIAM, 1974.

[84] S. K. Lele, *Compact finite difference schemes with spectral-like resolution*, Journal of computational physics, 103 (1992), pp. 16–42.

[85] H. Li, J. Cheng, T. Ng, J. Chen, and K. Lam, *A meshless hermite-cloud method for nonlinear fluid-structure analysis of near-bed submarine pipelines under current*, Engineering structures, 26 (2004), pp. 531–542.

[86] H. Li, Q. Wang, and K. Lam, *Development of a novel meshless local kriging (lokriging) method for structural dynamic analysis*, Computer Methods in Applied Mechanics and Engineering, 193 (2004), pp. 2599–2619.

[87] A. Linke, *Collision in a cross-shaped domain–a steady 2d navier–stokes example demonstrating the importance of mass conservation in cfd*, Computer Methods in Applied Mechanics and Engineering, 198 (2009), pp. 3278–3286.

[88] K. Lipnikov, G. Manzini, and M. Shashkov, *Mimetic finite difference method*, Journal of Computational Physics, 257 (2014), pp. 1163–1227.

[89] K. Lipnikov, M. Shashkov, and D. Svyatskiy, *The mimetic finite difference discretization of diffusion problem on unstructured polyhedral meshes*, J. Comput. Phys., 211 (2006), pp. 473–491.

[90] T. Liszka, *An interpolation method for an irregular net of nodes*, International Journal for Numerical Methods in Engineering, 20 (1984), pp. 1599–1612.

[91] T. Liszka and J. Orkisz, *The finite difference method at arbitrary irregular grids and its application in applied mechanics*, Computers & Structures, 11 (1980), pp. 83–95.

[92] S. Litvinov, M. Ellero, X. Hu, and N. A. Adams, *Smoothed dissipative particle dynamics model for polymer molecules in suspension*, Physical Review E, 77 (2008), p. 066703.

[93] G.-R. Liu and M. B. Liu, *Smoothed particle hydrodynamics: a meshfree particle method*, World Scientific, 2003.

[94] J.-G. Liu, J. Liu, and R. L. Pego, *Stability and convergence of efficient navier-stokes solvers via a commutator estimate*, Communications on pure and applied mathematics, 60 (2007), pp. 1443–1487.

[95] M. Liu and G. Liu, *Smoothed particle hydrodynamics (sph): an overview and recent developments*, Archives of computational methods in engineering, 17 (2010), pp. 25–76.

[96] S. Lomholt and M. R. Maxey, *Force-coupling method for particulate two-phase flow: Stokes flow*, Journal of Computational Physics, 184 (2003), pp. 381–405.

[97] L. B. Lucy, *A numerical approach to the testing of the fission hypothesis*, The astronomical journal, 82 (1977), pp. 1013–1024.

[98] X. Luo, A. Beskok, and G. E. Karniadakis, *Modeling electrokinetic flows by the smoothed profile method*, Journal of computational physics, 229 (2010), pp. 3828–3847.

[99] X. Luo, M. R. Maxey, and G. E. Karniadakis, *Smoothed profile method for particulate flows: Error analysis and simulations*, Journal of Computational Physics, 228 (2009), pp. 1750–1769.

[100] F. Maciá, M. Antuono, L. M. González, and A. Colagrossi, *Theoretical analysis of the no-slip boundary condition enforcement in sph methods*, Progress of theoretical physics, 125 (2011), pp. 1091–1121.

[101] J. H. Masliyah and S. Bhattacharjee, *Electrokinetic and colloid transport phenomena*, John Wiley & Sons, 2006.

[102] A. Masud and T. J. Hughes, *A stabilized mixed finite element method for darcy flow*, Computer Methods in Applied Mechanics and Engineering, 191 (2002), pp. 4341–4370.

[103] S. McKee, M. Tomé, V. Ferreira, J. Cuminato, A. Castelo, F. Sousa, and N. Mangiavacchi, *The mac method*, Computers & Fluids, 37 (2008), pp. 907–930.

[104] S. Menon, K. G. Mooney, K. Stapf, and D. P. Schmidt, *Parallel adaptive simplical re-meshing for deforming domain cfd computations*, Journal of Computational Physics, 298 (2015), pp. 62–78.

[105] D. Mirzaei and R. Schaback, *Direct meshless local petrov–galerkin (dmlpg) method: a generalized mls approximation*, Applied Numerical Mathematics, 68 (2013), pp. 73–82.

[106] D. Mirzaei, R. Schaback, and M. Dehghan, *On generalized moving least squares and diffuse derivatives*, IMA Journal of Numerical Analysis, (2011), p. drr030.

[107] I. D. Mishev and Q.-Y. Chen, *A mixed finite volume method for elliptic problems*, Numerical Methods for Partial Differential Equations, 23 (2007), pp. 1122–1138.

[108] R. Mittal and G. Iaccarino, *Immersed boundary methods*, Annu. Rev. Fluid Mech., 37 (2005), pp. 239–261.

[109] J. Monaghan, *Smoothed particle hydrodynamics and its diverse applications*, Annual Review of Fluid Mechanics, 44 (2012), pp. 323–346.

[110] J. Monaghan and R. Gingold, *Shock simulation by the particle method sph*, Journal of computational physics, 52 (1983), pp. 374–389.

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

[112] N.-Q. Nguyen and A. Ladd, *Lubrication corrections for lattice-boltzmann simulations of particle suspensions*, Physical Review E, 66 (2002), p. 046708.

[113] R. Nicolaides and K. Trapp, *Covolume discretizations of differential forms*, in Compatible Discretizations, Proceedings of IMA Hot Topics workshop on Compatible Discretizations, D. N. Arnold, P. Bochev, R. Lehoucq, R. Nicolaides, and M. Shashkov, eds., vol. IMA 142, Springer Verlag, 2006, pp. 161–172.

[114] R. A. Nicolaides and X. Wu, *Covolume solutions of three-dimensional div-curl equations*, SIAM Journal on Numerical Analysis, 34 (1997), pp. 2195–2203.

[115] M. Olshanskii and A. Reusken, *Grad-div stablilization for stokes equations*, Mathematics of Computation, 73 (2004), pp. 1699–1718.

[116] E. Onate and S. Idelsohn, *A mesh-free finite point method for advective-diffusive transport and fluid flow problems*, Computational Mechanics, 21 (1998), pp. 283–292.

[117] E. Onate, S. Idelsohn, O. Zienkiewicz, and R. Taylor, *A finite point method in computational mechanics. applications to convective transport and fluid flow*, International journal for numerical methods in engineering, 39 (1996), pp. 3839–3866.

[118] E. Onate, S. Idelsohn, O. Zienkiewicz, R. Taylor, and C. Sacco, *A stabilized finite point method for analysis of fluid mechanics problems*, Computer Methods in Applied Mechanics and Engineering, 139 (1996), pp. 315–346.

[119] A. Ortiz, M. Puso, and N. Sukumar, *Maximum-entropy meshfree method for compressible and near-incompressible elasticity*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 1859–1871.

[120] ——, *Maximum-entropy meshfree method for incompressible media problems*, Finite Elements in Analysis and Design, 47 (2011), pp. 572–585.

[121] S. Osher and R. P. Fedkiw, *Level set methods: an overview and some recent results*, Journal of Computational physics, 169 (2001), pp. 463–502.

[122] M. L. Parks, R. B. Lehoucq, S. J. Plimpton, and S. A. Silling, *Implementing peridynamics within a molecular dynamics code*, Computer Physics Communications, 179 (2008), pp. 777–783.

[123] N. A. Patankar, P. Singh, D. D. Joseph, R. Glowinski, and T.-W. Pan, *A new formulation of the distributed lagrange multiplier/fictitious domain method for particulate flows*, International Journal of Multiphase Flow, 26 (2000), pp. 1509–1524.

[124] J. B. Perot, *Discrete conservation properties of unstructured mesh schemes*, Annual review of fluid mechanics, 43 (2011), pp. 299–318.

[125] C. S. Peskin, *The immersed boundary method*, Acta numerica, 11 (2002), pp. 479–517.

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

[127] C. Pozrikidis, *A spectral-element method for particulate stokes flow*, Journal of Computational Physics, 156 (1999), pp. 360–381.

[128] D. J. Price, *Modelling discontinuities and kelvin–helmholtz instabilities in sph*, Journal of Computational Physics, 227 (2008), pp. 10040–10057.

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

[130] B. Rao and S. Rahman, *An efficient meshless method for fracture analysis of cracks*, Computational mechanics, 26 (2000), pp. 398–408.

[131] J. Ruge and K. Stüben, *Algebraic multigrid*, Multigrid methods, 3 (1987), pp. 73–130.

[132] 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).

[133] C. Rycroft, *Voro++: A three-dimensional voronoi cell library in c++*, Lawrence Berkeley National Laboratory, (2009).

[134] Y. Saad, *Iterative methods for sparse linear systems*, Siam, 2003.

[135] B. Seibold, *Minimal positive stencils in meshfree finite difference methods for the poisson equation*, Computer Methods in Applied Mechanics and Engineering, 198 (2008), pp. 592–601.

[136] ——, *Minimal positive stencils in meshfree finite difference methods for the poisson equation*, Computer Methods in Applied Mechanics and Engineering, 198 (2008), pp. 592–601.

[137] ——, *Performance of algebraic multigrid methods for non-symmetric matrices arising in particle methods*, Numerical Linear Algebra with Applications, 17 (2010), pp. 433–451.

[138] ——, *Performance of algebraic multigrid methods for non-symmetric matrices arising in particle methods*, Numerical Linear Algebra with Applications, 17 (2010), pp. 433–451.

[139] M. Shashkov, *Conservative Finite Difference Methods on General Grids*, CRC Press, Boca Raton, FL, December 05 1995.

[140] 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 (2013), pp. 163–173.

[141] D. Stevens, H. Power, M. Lees, and H. Morvan, *The use of pde centres in the local rbf hermitian method for 3d convective-diffusion problems*, Journal of Computational Physics, 228 (2009), pp. 4606–4624.

[142] J. C. Strikwerda, *Finite difference methods for the stokes and navier-stokes equations*, SIAM Journal on Scientific and Statistical Computing, 5 (1984), pp. 56–68.

[143] N. Sukumar and R. Wright, *Overview and construction of meshfree basis functions: from moving least squares to entropy approximants*, International Journal for Numerical Methods in Engineering, 70 (2007), pp. 181–205.

[144] D. S. Sundar and K. Yeo, *A high order meshless method with compact support*, Journal of Computational Physics, 272 (2014), pp. 70–87.

[145] M. Sussman, E. Fatemi, P. Smereka, and S. Osher, *An improved level set method for incompressible two-phase flows*, Computers & Fluids, 27 (1998), pp. 663–680.

[146] C. R. Swaminathan, V. R. Voller, and S. V. Patankar, *A streamline upwind control volume finite element method for modeling fluid flow and heat transfer problems*, Finite Elements in Analysis and Design, 13 (1993), pp. 169 – 184.

[147] A. Tartakovsky, N. Trask, K. Pan, B. Jones, W. Pan, and J. Williams, *Smoothed particle hydrodynamics and its applications for multiphase flow and reactive transport in porous media*, Computational Geosciences, (2015), pp. 1–28.

[148] L. Timmermans, P. Minev, and F. Van De Vosse, *An approximate projection scheme for incompressible flow using spectral elements*, International journal for numerical methods in fluids, 22 (1996), pp. 673–688.

[149] 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, 289 (2015), pp. 155–178.

[150] ——, *A scalable consistent second-order sph solver for unsteady low reynolds number flows*, Computer Methods in Applied Mechanics and Engineering, 289 (2015), pp. 155–178.

[151] L. N. Trefethen and D. Bau III, *Numerical linear algebra*, vol. 50, Siam, 1997.

[152] ——, *Numerical linear algebra*, vol. 50, Siam, 1997.

[153] U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid*, Academic press, 2000.

[154] Ž. Tuković and H. Jasak, *A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow*, Computers & fluids, 55 (2012), pp. 70–84.

[155] P. S. Vassilevski, *Multilevel block factorization preconditioners: Matrix-based analysis and algorithms for solving finite element equations*, Springer Science & Business Media, 2008.

[156] G. H. Wannier, *A contribution to the hydrodynamics of lubrication*, Quarterly of Applied Mathematics, 8 (1950), pp. 1–32.

[157] E. Weinan and J.-G. Liu, *Essentially compact schemes for unsteady viscous incompressible flows*, Journal of Computational Physics, 126 (1996), pp. 122–138.

[158] H. Wendland, *Scattered data approximation*, vol. 17, Cambridge university press, 2004.

[159] ——, *Scattered data approximation*, vol. 17, Cambridge university press, 2004.

[160] G. B. Wright and B. Fornberg, *Scattered node compact finite difference-type formulas generated from radial basis functions*, Journal of Computational Physics, 212 (2006), pp. 99–123.

[161] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM review, 34 (1992), pp. 581–613.

[162] ——, *Iterative methods by space decomposition and subspace correction*, SIAM review, 34 (1992), pp. 581–613.

[163] J. Xu, *Fasp linear algebra package*, 2015. [Online; accessed 8-November-2015].

[164] R. Xu, P. Stansby, and D. Laurence, *Accuracy and stability in incompressible sph (isph) based on the projection method and a new approach*, Journal of Computational Physics, 228 (2009), pp. 6703–6725.

[165] 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 (2013), pp. 228–241.

[166] K. Yeo and M. R. Maxey, *Simulation of concentrated suspensions using the force-coupling method*, Journal of computational physics, 229 (2010), pp. 2401–2421.

[167] I. Yotov, *Mixed finite element methods for flow in porous media*, Tech. Rep. TR96-09, Dept. Comp. Appl. Math., Rice University, 1996.

[168] L. Zhang, P. Zhu, and K. Liew, *Thermal buckling of functionally graded plates using a local kriging meshless method*, Composite Structures, 108 (2014), pp. 472–492.

[169] P. Zhu and K. M. Liew, *Free vibration analysis of moderately thick functionally graded plates by local kriging meshless method*, Composite Structures, 93 (2011), pp. 2925–2944.

[170] C. Zuppa, *Error estimates for moving least square approximations*, Bulletin of the Brazilian Mathematical Society, 34 (2003), pp. 231–249.