

Compact moving least squares: an optimization framework for generating high-order compact meshless discretizations¹

Nathaniel Trask^a, Martin Maxey^a, Xiaozhe Hu^b

^a*Division of Applied Mathematics, Brown University, Providence, RI 02912*

^b*Department of Mathematics, Tufts University, Medford, Massachusetts 02155*

Abstract

A generalization of the optimization framework typically used in moving least squares is presented that provides high-order approximation while maintaining compact stencils and a consistent treatment of boundaries. The approach, which we refer to as compact moving least squares, resembles the capabilities of compact finite differences but requires no structure in the underlying set of nodes. An efficient collocation scheme is used to demonstrate the capabilities of the method to solve elliptic boundary value problems in strong form stably without the need for an expensive weak form. The flexibility of the approach is demonstrated by using the same framework to both solve a variety of elliptic problems and to generate implicit approximations to derivatives. Finally, an efficient preconditioner is presented for the steady Stokes equations, and the approach's efficiency and high order of accuracy is demonstrated for domains with curvi-linear boundaries.

Keywords: Compact moving least squares, CMLS, optimization, Compact finite difference, meshless method

1. Introduction

Meshless methods have long presented a promising Lagrangian framework for simulating flows with complex moving geometries and moving interfaces, but have failed to gain traction as a high-order method due to a lack of efficient quadrature rules for rational functions, the large support of high-order basis functions, and deficiencies in approximation near boundaries. The current work stems from an attempt to extend the flexibility of low-order meshless approaches to solve Lagrangian hydrodynamics [8] to high order using the the moving least squares (MLS) framework [7]. MLS provides a simple and rigorous approach to achieve high-order reconstruction, but the previous list of challenges prevent their stable and efficient application to standard pressure projection schemes. The modified framework that we present here, which we refer to as compact moving least squares (CMLS), remedies these issues while generalizing classical compact finite difference schemes.

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 [16] or by developing implicit formulas for derivatives[10]. We recall fundamental properties of these classes of schemes in Section 2. 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 analytically 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[12]. For lower order approximation, smooth particle hydrodynamics (SPH) has been established as the oldest meshless method for a variety of problems, despite technical challenges regarding numerical instabilities and a lack of even zeroth order consistency in some formulations of the method [8, 6].

The current approach will present an optimization framework that introduces a regularization to incorporate Hermite data into the MLS process, generalizing the CFDM method to arbitrary particle distributions. The stencils generated using this approach behave in a manner identical to CFDM: we will demonstrate that, provided a solution to the optimization problem can be stably computed, for problems with smooth solutions the discretization is consistent and convergence is obtained up to the order of polynomial used in the MLS reconstruction and easily preconditioned with standard techniques. In comparison to SPH methods, we demonstrate that for the support typically used to discretize the Laplacian when simulating viscous flow [37], the CMLS method is able to attain a sixth order discretization. This is relevant to a recent trend in which implicit projection methods are used to simulate Lagrangian hydrodynamics in the incompressible SPH (ISPH) methods (see e.g. [1, 2, 3, 4, 5]), where our new discretization can be used to achieve substantially higher order convergence while avoiding challenges associated with consistent enforcement of boundary conditions.

In the RBF community, generalizations of the finite difference radial basis function method (RBF-FD) employ a similar strategy using information gleaned from the underlying PDE to achieve compact discretizations (e.g. [39, 9, 38, 42]). While our approach is similar in broad strategy, posing the reconstruction as an ℓ_2 -optimization allows a flexible framework that can make use of the wealth of information regarding stable solution of least squares problems in the literature [41, 21] and a simplified analysis; assuming that for a given particle distribution a local polynomial reconstruction exists, the existence of an MLS reconstruction follows from standard convex optimization arguments[12, 13]. We will show that an optimization framework allows boundary conditions to be enforced locally via equality constraints without introducing global penalty parameters. 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 [16], to our knowledge the current approach marks the first method that allows implicit formulas for differential operators generalizing Lele-type schemes[10]. 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.

We begin by providing a brief review of compact finite difference methods in Section 2 and of classical moving least squares in Section 3 before introducing the CMLS method in Section 4. We demonstrate recovery of original Lele-type implicit derivatives in Section 5 and proceed in Section 6 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 novel boundary condition approach in performing Helmholtz decompositions as would be used in a projection method for solving fluid mechanics problems. We finally use the framework to solve the monolithic Stokes equations in Section 7 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 faster when compared to an explicit method.

2. Compact finite differences

In compact finite differences, there are two broad strategies which we will refer to as Weinan-type schemes[16] and Lele-type schemes[10]. 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

$$u_i'' \approx D_{xx}u = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = u_i'' + Ch^2u_i'''' + O(h^4) \quad (1)$$

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

$$Ch^2u_i'''' = Ch^2f_i'' \approx Ch^2D_{xx}f + O(h^4) \quad (2)$$

and after reorganizing, a fourth order expression for the second derivative is obtained with the same bandwidth as the centered difference.

$$D_{xx}^{compact}u_i = D_{xx}u_i - Ch^2D_{xx}f_i = u_i'' + O(h^4) \quad (3)$$

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

$$\sum_{j=-N}^N \alpha_j u_{i+j}' = \frac{\sum_{j=1}^M a_j (u_{i+j} - u_{i-j})}{h} + O(h^{Q+1}) \quad (4)$$

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.

3. Collocated MLS

3.1. Classical MLS

In this work we follow the framework presented in [12]. 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 [12], and for the sake of

simplicity in this paper we will informally assume a “well behaved pointset” in the sense that all particles are separated by a finite distance, maintain polynomial unisolvency (see [12]) and are characterized by the fill distance

$$h_{X_h, \Omega} = \sup_{x \in \Omega} \min_{1 \leq i \leq N} \|x - x_i\| \quad (5)$$

(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.

$$|u(x) - u_h(x)| \leq Ch_{X_h, \Omega}^{m+1} |u|_{C^{m+1}(\Omega)} \quad (6)$$

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

$$\min_{p \in \pi_m(\mathbb{R}^d)} \left\{ \sum_{j=1}^N [u(x_j) - p(x_j)]^2 W(\|x - x_j\|) \right\} \quad (7)$$

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 increasingly poorly conditioned. In the current work we take $W(r) = \left(1 - \left(\frac{r}{\epsilon}\right)^4\right)_+$. By defining a basis $\mathbf{P} = \{\phi_1, \dots, \phi_Q\}$ such that $\text{span}(\phi_1, \dots, \phi_Q) = \pi_m(\mathbb{R}^d)$, we can pose this as finding an optimal coefficient vector \mathbf{c} such that $p^* = \mathbf{c}^\top \mathbf{P}$. The optimal coefficient vector solving Equation 7 is given by

$$\mathbf{c}(x) = \mathbf{M}(x)^{-1} \sum_j \mathbf{P}_j^\top W(\|x - x_j\|) \mathbf{u}_j \quad (8)$$

where

$$\mathbf{M}(x) = \sum_k \mathbf{P}_k^\top W(\|x - x_k\|) \mathbf{P}_k \quad (9)$$

The MLS approximation is therefore given by $u_h(x) = \mathbf{P}^\top(x) \mathbf{c}$. 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.

$$D_h^\alpha u(x) = (D^\alpha \mathbf{P}(x))^\top \mathbf{c} \quad (10)$$

Mirzaei has shown [13] that this approximation yields the following error estimate

$$\|D^\alpha u - D_h^\alpha u\|_{L^\infty(\Omega)} \leq C(m) h_{X_h, \Omega}^{m+1-|\alpha|} \|u\|_{W^{m+1,p}(\Omega)} \quad (11)$$

which, in comparison to results analyzing the full derivative [14][15], 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)

$$\phi_\alpha(x) = \frac{1}{\alpha!} \left(\frac{x - x_i}{\epsilon} \right)^\alpha \quad (12)$$

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.

$$D^\beta \mathbf{P}_i = \hat{e}_\beta \epsilon^{-\beta} \quad (13)$$

where \hat{e}_β 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

$$\mathbf{M}_i = \sum_k \mathbf{P}_i(x_k)^\top W_{ik} \mathbf{P}_i(x_k) \quad (14)$$

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 :

$$D_h^\alpha u_i = \epsilon^{-\alpha} \hat{e}_\alpha^\top \mathbf{M}_i^{-1} \sum_j \mathbf{P}_i(x_j) W_{ij} u_j \quad (15)$$

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 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[21, 13]. 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.

$$\phi_{\alpha(i,j)}(x, y) = P_i\left(\frac{x - x_i}{\epsilon}\right) P_j\left(\frac{y - y_i}{\epsilon}\right) \quad (16)$$

where $\alpha = 1, \dots, m^2$ 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 6th order polynomial reconstruction and are used for efficiency reasons.

3.2. 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[30], 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[31], finite point method[32], and others[33, 34]).

For each particle, we construct, invert, and store the following correction matrix for each point x_i :

$$\mathbf{M}_i = \sum_k \mathbf{P}_i(x_k)^\top W_{ik} \mathbf{P}_i(x_k) \quad (17)$$

Derivatives are then reconstructed via

$$D_h^\alpha u_i = D_h^\alpha \mathbf{P}_i^\top \mathbf{M}_i^{-1} \sum_j \mathbf{P}_i(x_j) W_{ij} (u_j - u_i) \quad (18)$$

In order to use this to discretize a linear boundary value problem on Ω

$$\begin{cases} \mathcal{L}_1 u = f, & \text{if } x \in \Omega \\ \mathcal{L}_2 u = g, & \text{if } x \in \partial\Omega \end{cases} \quad (19)$$

we can simply distribute a “well-behaved” pointset X_Ω and solve the following equation for each particle x_i

$$\begin{cases} \mathcal{L}_1^h u(x_i) = f(x_i), & \text{if } x \in \Omega \\ \mathcal{L}_2^h u(x_i) = g(x_i), & \text{if } x \in \partial\Omega \end{cases} \quad (20)$$

where the linear operators \mathcal{L}_1^h and \mathcal{L}_2^h are discretized using Equation 18. 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 [21]. 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 7 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 $\nabla \cdot \nabla \times \mathbf{u} \neq 0$). Splitting schemes commonly used in Galerkin formulations (see e.g. [22]) 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)[24] 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 [23]. These challenges render an application of this method to solve the steady Stokes equations challenging.

4. Compact moving least squares

4.1. Optimization framework

Rather than take a larger stencil to achieve high-order, the strategy of the CMLS approach is to use Hermite data to obtain a high-order reconstruction while maintaining compact stencils. With the intention of solving the boundary value problem in Equation 19, we pose the following least squares problem for the optimal polynomial reconstruction at each node i :

$$\min_{p \in \pi_m(\mathbb{R}^d)} \sum_j \left\{ \left[(u_j - p_j)^2 + \epsilon_1 (f_j - \mathcal{L}_1 p_j)^2 + \mathbb{1}_{x_j \in \partial\Omega} \epsilon_2 (g_j - \mathcal{L}_2 p_j)^2 \right] W(\|x_i - x_j\|) \right\} \quad (21)$$

Here ϵ_1 and ϵ_2 are regularization parameters and $\mathbb{1}_{x_j \in \partial\Omega}$ is an indicator taking unit value for boundary particles and zero value elsewhere. The three competing 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 regularization 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,

$$\mathcal{L}_2 p_i = g_i \quad (22)$$

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 ,

$$\mathbf{M}_i = \sum_j \left(\mathbf{P}_j \mathbf{P}_j^\top + \epsilon_1 (\mathcal{L}_1 \mathbf{P}_j) (\mathcal{L}_1 \mathbf{P}_j^\top) + \mathbb{1}_{x_j \in \partial\Omega} \epsilon_2 (\mathcal{L}_2 \mathbf{P}_j) (\mathcal{L}_2 \mathbf{P}_j^\top) \right) W_{ij} \quad (23)$$

which can be used to compute the optimal coefficients,

$$\mathbf{c}_i = \mathbf{M}_i^{-1} \sum_j \left(\mathbf{P}_j^\top u_j + \epsilon_1 \mathcal{L}_1 \mathbf{P}_j^\top f_j + \mathbb{1}_{x_j \in \partial\Omega} \epsilon_2 \mathcal{L}_2 \mathbf{P}_j^\top g_j \right) W_{ij} \quad (24)$$

and estimate derivatives using the diffuse derivative concept as

$$D_h^\alpha u(x) = (D^\alpha \mathbf{P}(x))^\top \mathbf{c} \quad (25)$$

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

$$\hat{\mathbf{M}}_i = \begin{bmatrix} \mathbf{M}_i & \mathcal{L}_2 \mathbf{P}_i \\ \mathcal{L}_2 \mathbf{P}_i^\top & 0 \end{bmatrix} \quad (26)$$

$$\hat{\mathbf{c}}_i^\top = \begin{bmatrix} \mathbf{c} & \lambda \end{bmatrix} \quad (27)$$

$$\hat{\mathbf{b}}_i^\top = \begin{bmatrix} \sum_j \left(\mathbf{P}_j^\top u_j + C_1 \mathcal{L}_1 \mathbf{P}_j^\top f_j + C_2 \mathcal{L}_2 \mathbf{P}_j^\top g_j \right) W_{ij} & g_i \end{bmatrix} \quad (28)$$

At this point the PDE can be discretized at all points, including the boundary, as

$$\mathcal{L}_1^h u(x_i) = f(x_i), \forall x_i \in \Omega \cup \partial\Omega \quad (29)$$

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 20.

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 21 alleviates the one-sidedness of

the approximation near boundaries, while the constraint ensures compatibility between the PDE and the boundary condition along the boundary.

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 23 and 24 compared to Equations 17 and 18, 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 6.

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 2. 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 [19][20] 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 [17] [18].

4.2. 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.

$$\nabla^2 p = f \tag{30}$$

$$\partial_n p = g \tag{31}$$

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 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 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 2 demonstrates the expected algebraic convergence for MLS and CMLS. Although both methods demonstrate the same convergence rate for a

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 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.

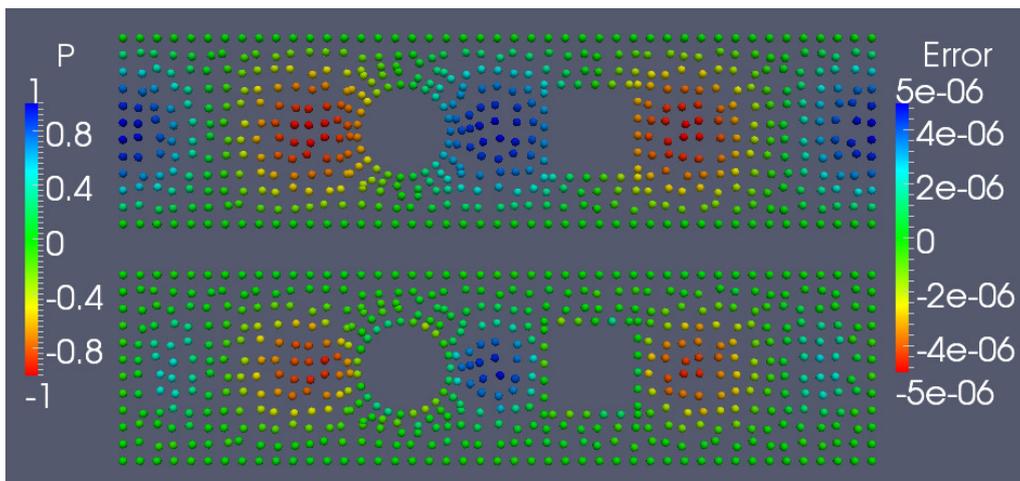


Figure 1: p (top) and truncation error $\|\nabla^2 p - \nabla_h^2 p\|$ (bottom) using 7th order polynomial space. $H \times L$ geometry discretized with initially uniform Cartesian particles $H/dx = 12$ for $\chi = 0.2dx$.

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 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 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 3) in the sense that the error decreases exponentially with 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 4th order MLS with 7th 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 provides better parallel scaling as compared with the domain decomposition approaches that would be necessary in h-refinement.

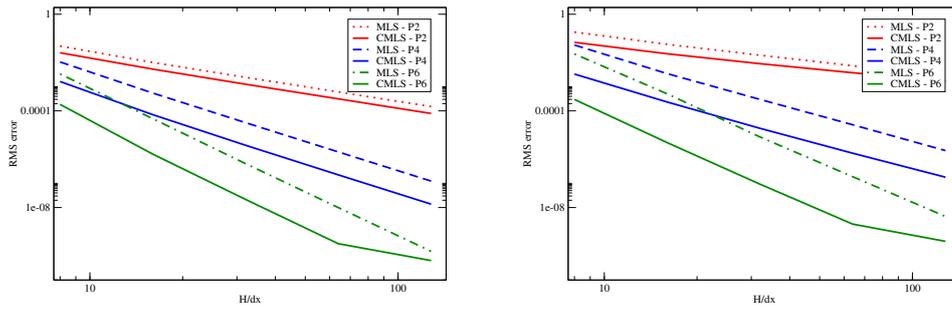


Figure 2: Truncation error for gradient operators (left) and Laplacian operators (right) for standard MLS (dashed lines) and CMLS (solid lines).

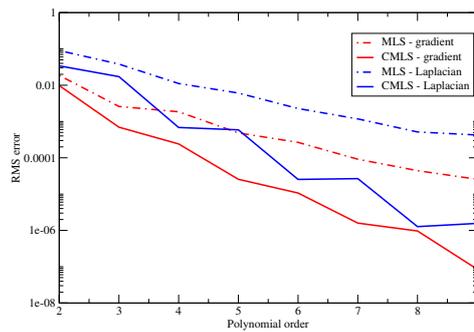


Figure 3: Spectral-like convergence for fixed resolution $H/dx = 12$ and increasing polynomial order.

5. Lele-type derivatives

While the approach in the previous section more closely resembles the Weinan-type scheme discussed in Section 2, 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 the following expression for the reconstruction

$$u_h(x; x_i) = \mathbf{P}^\top(x) \sum_j \left(\epsilon_1 D^\alpha \mathbf{P}(x_j) W_{ij} D^\alpha u_j + \mathbf{P}(x_j) W_{ij} u_j \right). \quad (32)$$

If we then require that the diffuse derivative matches the nodal data, we obtain a stencil of the form

$$D^\alpha u_i = D^\alpha u_h(x_i; x_i) = \sum_j \left(\alpha_{ij} D^\alpha u_j + a_{ij} u_j \right) \quad (33)$$

where for simplicity we have lumped the terms in Equation 24 into the coefficients α_{ij} and a_{ij} to obtain a stencil of the same form as Equation 4. Further, since the optimization problem Equation 21 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 $W = 1$.

$$u'_{i+1} + 4u'_i + u'_{i-1} = 3 \left(\frac{u_{i+1} - u_{i-1}}{h} \right) + O(h^4) \quad (34)$$

$$6u'_{i+2} + 96u'_{i+1} + 216u'_i + 96u'_{i-1} + 6u'_{i-2} = \frac{25(u_{i+2} - u_{i-2}) + 160(u_{i+1} - u_{i-1})}{h} + O(h^8) \quad (35)$$

		CMLS			MLS		
		4	6	8	2	4	6
\mathbf{m}		2.5	4.5	6.5	2.5	4.5	6.5
$\epsilon/d\mathbf{x}$		2.5	4.5	6.5	2.5	4.5	6.5
$\frac{H}{dx}$	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 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 [10] for tridiagonal and pentadiagonal 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 21. Because the coefficients of the stencil are generated as the solution of an optimization problem however and require no a

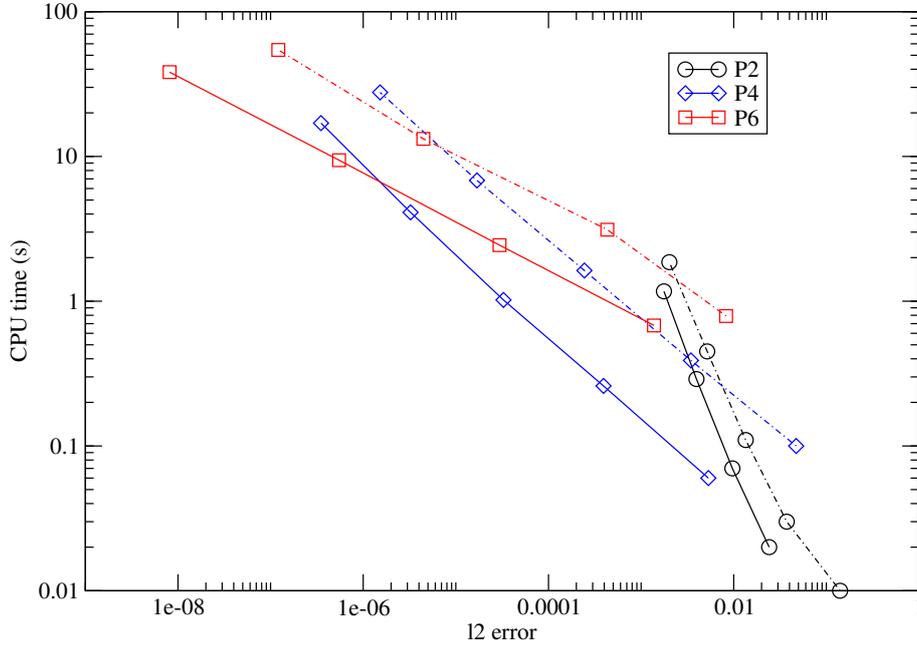


Figure 4: Necessary CPU time to obtain a given truncation error for varying order reconstruction. Standard MLS given by dashed lines while compact MLS given by solid lines.

priori calculation, the current approach is applicable to one-sided stencils near boundaries and on general pointsets. We revisit the case discussed previously (Figure 1) and present the resulting truncation error when Equation 33 is used to calculate the gradient of the function $u = \sin x \sin y$ in Table 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 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[21], for a given order reconstruction the above results all converged to a fixed solver tolerance in a roughly constant number of iterations independent 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 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.

6. 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.

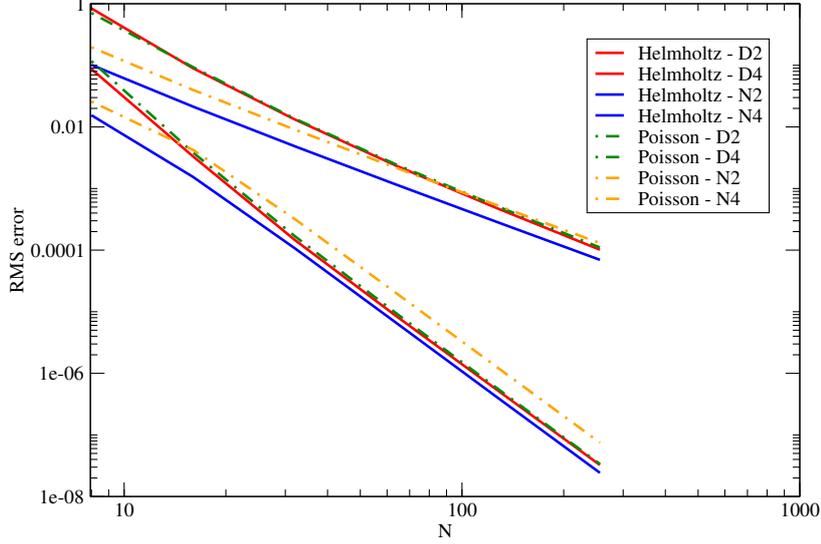


Figure 5: Second and fourth order convergence for Poisson and Helmholtz problems with Dirichlet (D), or Neumann (N) BCs and $m = 2$ or 4 .

6.1. 1D example

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 $[0, 2\pi]$,

$$\partial_{xx}u = -2 \sin x \quad (36)$$

we take for the first penalty operator $\mathcal{L}_1 = \partial_{xx}$ and $f = -2\sin(x)$ 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

$$u - \partial_{xx}u = -\sin(x) \quad (37)$$

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 1 are used. 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 5).

6.2. Helmholtz decomposition

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 field \mathbf{u} into a

divergence-free part \mathbf{u}^* and a gradient of a potential scalar field.

$$\mathbf{u} = \mathbf{u}^* + \nabla p \quad (38)$$

Either taking the divergence or taking the dot product with the normal provides a Poisson problem or Neumann boundary conditions, respectively.

$$\nabla \cdot \nabla p = \nabla \cdot \mathbf{u} \quad (39)$$

$$\partial_n p = \hat{n} \cdot (\mathbf{u} - \mathbf{u}^*) \quad (40)$$

Discretizing this by taking the composition of the discrete divergence and gradient operators will provide a vector field with discrete divergence equal to zero. In practice, compatibility is required between the vector and potential reconstruction to avoid spurious solutions. Approximating Equation 39 with the discrete Laplacian circumvents this issue, but the resulting vector field is only divergence-free up to the truncation error of the discretization, since $\nabla_h^2 \neq \nabla_h \cdot \nabla_h$ 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 a classical MLS reconstruction and plot the resulting concentration of error in Figure 6, taking as an initial divergence-free velocity

$$\mathbf{u}^* = \langle \sin x \cos y, -\cos x \sin y \rangle \quad (41)$$

and

$$\mathbf{u} = \mathbf{u}^* + \langle \sin x, 0 \rangle. \quad (42)$$

Although the pressure converges optimally, we present in Table 3 the convergence of the RMS errors for the velocity and the divergence. The error concentrated near boundaries (See Figure 6) 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 3: Convergence of Helmholtz projection for classical MLS. Results for the sixth order case are unavailable because the iterative solver failed to converge.

	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 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 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 6: Convergence of Helmholtz projection for CMLS with $\epsilon_1 = 0$ and $\epsilon_2 = 0$.

If instead CMLS is used, the convergence is regular with no concentration of divergence in the projected velocity field. Figure 7 demonstrates the uniform convergence 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 divergence, although in an actual projection method the

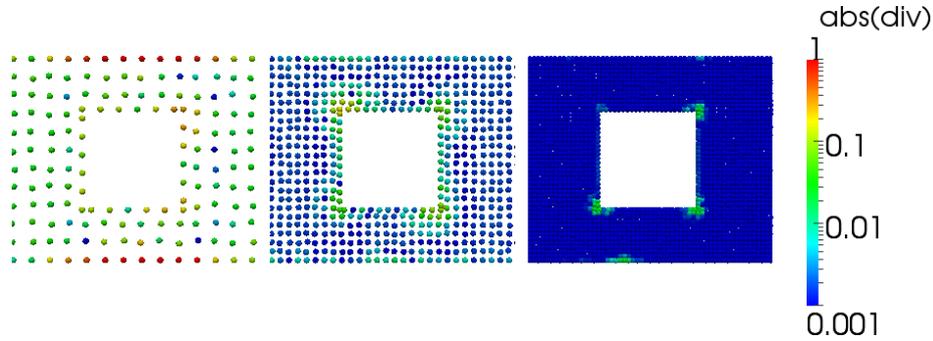


Figure 6: Concentration of error in divergence for MLS near sharp corners for $H/dx = 12, 24, 48$ and $m = 4$. Even with high-order polynomial space, error in divergence is first order.

momentum equation could be used to generate compact velocity operators. The convergence of the velocity and divergence errors using CMLS is presented in Table 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 6. In Table 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 26-28) so that the PDE can be enforced at the boundary point rather than enforcing the boundary condition.

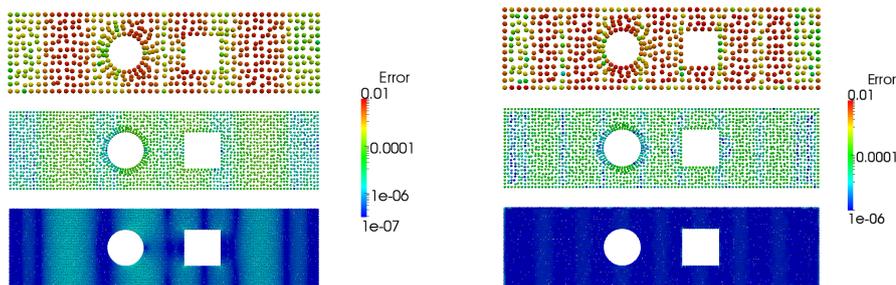


Figure 7: Pointwise error for projected velocity magnitude (left) and divergence (right) for CMLS discretization. In comparison to Figure 6, there is no concentration of divergence error near boundaries and uniform convergence is obtained across domain.

7. Compact moving least squares for vector PDE - Unsteady Stokes equations

7.1. Discretization

To solve vector valued PDEs, the optimization problem in Equation 21 can be extended for the approximation of the vector $\mathbf{y} \in \mathbb{R}^n$ as

$$\min_{\mathbf{p} \in (\pi_m(\mathbb{R}^d))} \sum_j \left\{ \left[(\mathbf{y}_j - \mathbf{p}_j)^\top (\mathbf{y}_j - \mathbf{p}_j) + (\mathbf{f}_j - \mathcal{L}_1 \mathbf{p}_j)^\top \epsilon_1 (\mathbf{f}_j - \mathcal{L}_1 \mathbf{p}_j) + \mathbb{1}_{x_j \in \partial\Omega} (\mathbf{g}_j - \mathcal{L}_2 \mathbf{p}_j)^\top \epsilon_2 (\mathbf{g}_j - \mathcal{L}_2 \mathbf{p}_j) \right] W(\|x_i - x_j\|) \right\} \quad (43)$$

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.

$$\begin{cases} -\nu \nabla^2 \mathbf{u} + \nabla p = f, & \text{if } x \in \Omega \\ \nabla \cdot \mathbf{u} = 0, & \text{if } x \in \Omega \\ \mathbf{u} = \mathbf{w}, & \text{if } x \in \partial\Omega \\ \partial_n p - \nu \hat{n} \cdot \nabla^2 \mathbf{u} = \hat{n} \cdot f, & \text{if } x \in \partial\Omega \end{cases} \quad (44)$$

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 [23].

These equations can be put into the form of Equation 43 by taking $\mathbf{y} = \langle \mathbf{u}, p \rangle^\top$, $\mathcal{L}_1 = \begin{bmatrix} -\nu \nabla_h^2 & \nabla_h \\ \nabla_h^\top & 0 \end{bmatrix}$, $\mathcal{L}_2 = \begin{bmatrix} \mathbf{I} & 0 \\ -\nu \hat{n} \cdot \nabla_h^2 & \partial_n \end{bmatrix}$, $\mathbf{f} = \langle f, 0 \rangle^\top$, and $\mathbf{g} = \langle \mathbf{w}, \hat{n} \cdot f \rangle^\top$, and an approximation at each point is constructed in the usual way:

$$\mathbf{M}_i = \sum_j \left(\mathbf{P}_j \mathbf{P}_j^\top + \epsilon_1 (\mathcal{L}_1 \mathbf{P}_j) (\mathcal{L}_1 \mathbf{P}_j^\top) + \mathbb{1}_{x_j \in \partial\Omega} \epsilon_2 (\mathcal{L}_2 \mathbf{P}_j) (\mathcal{L}_2 \mathbf{P}_j^\top) \right) W(\|x_i - x_j\|) \quad (45)$$

where now $\mathbf{P} \in \mathbb{R}^{n \times Q} = \left\{ \begin{pmatrix} p_1 \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} p_Q \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ p_1 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ p_Q \end{pmatrix} \right\}$ is a vector valued polynomial basis for $(\pi_m(\mathbb{R}^d))^n$. Solution of this least squares problem gives an optimal coefficient

$$\mathbf{c}_i = \mathbf{M}_i^{-1} \sum_j \left(\mathbf{P}_j^\top \mathbf{y}_j + \epsilon_1 \mathcal{L}_1 \mathbf{P}_j^\top \mathbf{f}_j + \mathbb{1}_{x_j \in \partial\Omega} \epsilon_2 \mathcal{L}_2 \mathbf{P}_j^\top \mathbf{g}_j \right) W(\|x_i - x_j\|) \quad (46)$$

which can be used to approximate differential operators using the diffuse derivative.

$$D_h^\alpha \mathbf{y}_i = (D^\alpha \mathbf{P}_i)^\top \mathbf{c}_i \quad (47)$$

When used to discretize Equation 44, 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

$$\nabla_h^2 \mathbf{u}_i = \sum_j a_{ij} \cdot \mathbf{u}_j + b_{ij} p_j + c_{ij} \mathbf{f}_j \quad (48)$$

where for simplicity we have compactly expressed Equation 47 in terms of coefficients a_{ij}, b_{ij} and c_{ij} . As a result the fully discretized form of Equation 44 assumes the following 2×2 block matrix form.

$$\begin{bmatrix} \mathbf{L} & \mathbf{G} \\ \mathbf{D} & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ g \end{bmatrix}. \quad (49)$$

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:

$$\begin{bmatrix} \mathbf{P} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \quad (50)$$

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 (49). It is well-known that an efficient and robust preconditioner is crucial for the performance of the GMRes method for saddle-point problems [26]. Here we adopt the lower triangular block preconditioner based on the following block factorization of the 2×2 block system (49).

$$\begin{bmatrix} \mathbf{L} & \mathbf{G} \\ \mathbf{D} & \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{D} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{L}^{-1}\mathbf{G} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \quad (51)$$

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 [35]. 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 $\tilde{\mathbf{S}} = \mathbf{P} - \mathbf{D} \text{diag}(\mathbf{L})^{-1}\mathbf{G}$ and the lower triangular block preconditioner is given by

$$\begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{D} & \tilde{\mathbf{S}} \end{bmatrix}.$$

To apply the preconditioner, we still need to invert \mathbf{L} and $\tilde{\mathbf{S}}$, which could be difficult and expensive. In our implementation, we use the AMG method to efficiently solve them both[36].

We use this preconditioner to simulate the so-called Wannier flow of two eccentric rotating cylinders. For this setup an analytic solution exists[27] and provides a standard benchmark for evaluating high-order discretizations of curvilinear geometries with strong viscous boundary layers[28]. 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 demonstrates representative particle distributions and the streamlines of the resulting solution, while Figure 9 demonstrates

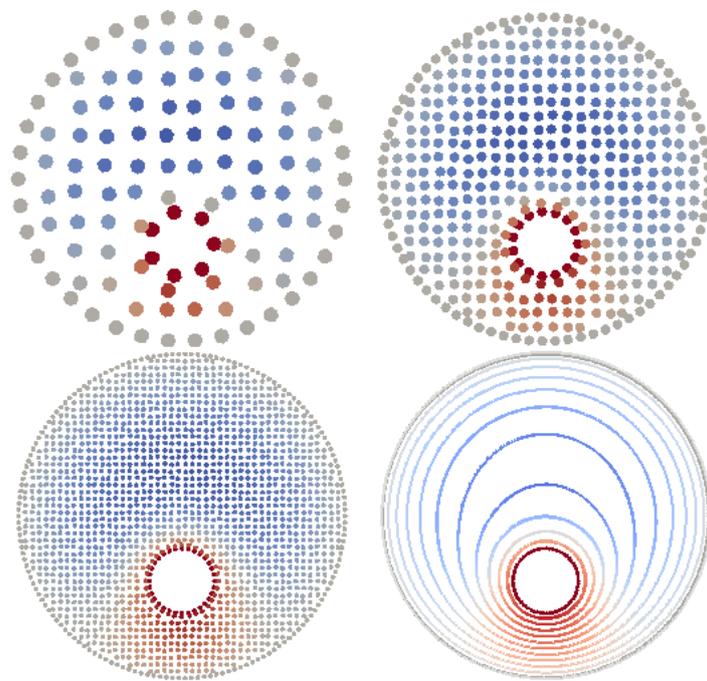


Figure 8: Particle distributions for resolutions of $D/\Delta x \in \{12, 24, 48\}$ and streamlines of solution colored by velocity magnitude.

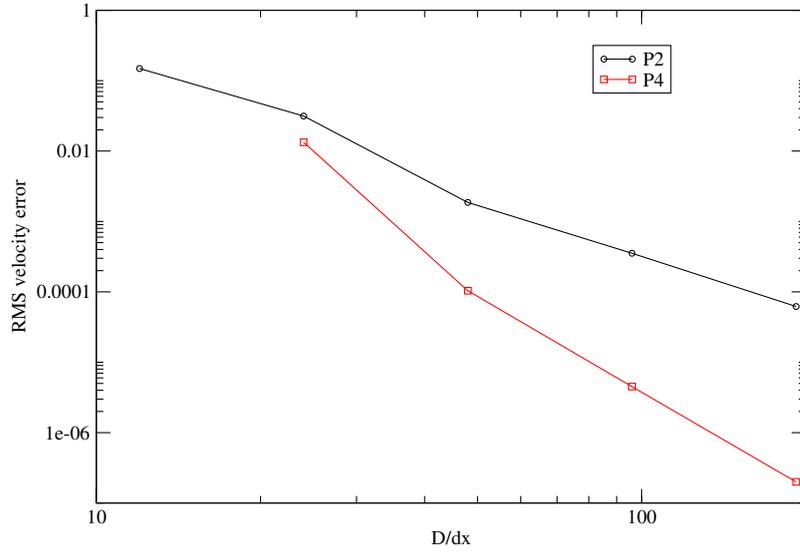


Figure 9: Convergence for the Wannier flow case with quadratic and quartic polynomial reconstruction.

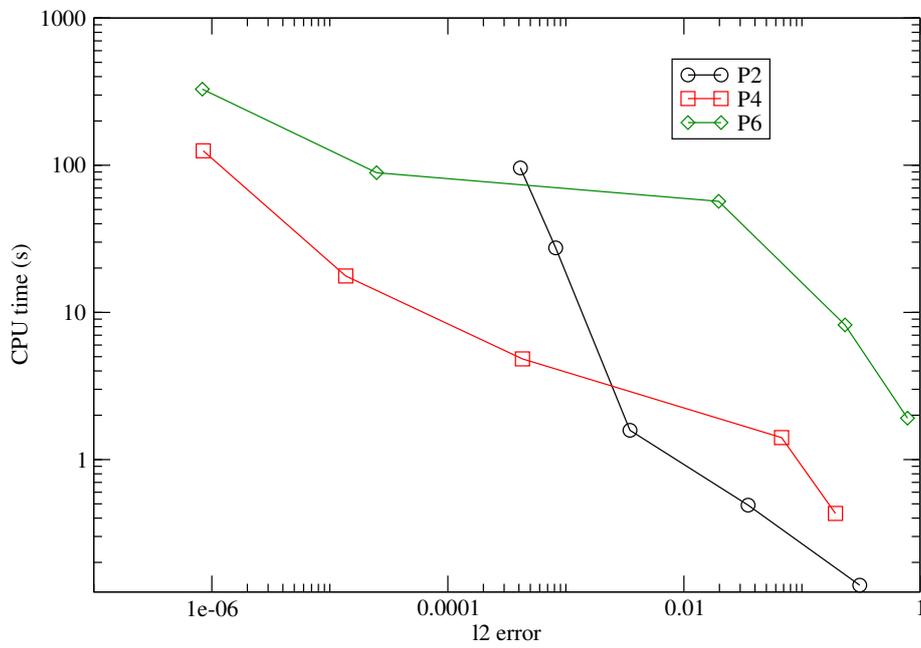


Figure 10: Necessary CPU time to obtain a given RMS velocity error for varying order reconstruction.

the convergence of the RMS velocity error. After the curvature of the boundary is adequately resolved, quadratic and quartic reconstruction achieve 2nd and 4th 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 10. 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, because the cost associated with the large dimension of the polynomial basis when inverting the mass matrices becomes prohibitive.

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 7: Preconditioner performance for stationary Stokes problem with quartic reconstruction (GMRes with stopping criterion that relative residual is less than 10^{-6})

Preliminary benchmarking results for the preconditioner performance for the Wannier flow problem are presented in Table 7. Although the number of iterations to obtain 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. Conclusions and future work

A flexible optimization framework for generating high-order compact collocation schemes has been presented that generalizes compact finite difference scheme to general pointsets. While the RBF-FD method maintains several similarities to the current approach, the use of an optimization framework rather than RBF interpolation allows several distinct advantages. We have demonstrated that boundary conditions may be handled naturally via equality constraints, which allows a degree of compatibility when considering the Helmholtz problem - a key component of many splitting schemes for the Navier-Stokes equations. We have shown that the method extends to vector PDE by solving the steady Stokes equations. To our knowledge, this marks the first time that a fully meshless method for the Stokes problem has been developed that is able to achieve $O(N)$ computational complexity while maintaining high-order convergence.

In another work, one of the authors has developed a separate scheme pursuing a similar optimization strategy to generalize staggered finite difference methods to general pointsets[43]. Preliminary results have shown that the two methods can be combined to obtain a meshless, staggered, compact finite-difference approach, similar to standard finite difference methods[44]. The ability to obtain a highly stable meshless method while simultaneously achieving efficient high-order convergence is very promising.

Although we have presented a completely meshfree formulation, the optimization framework can just as easily be applied to attain compact stencils in the finite volume method. Cueto-Felgueroso et al. demonstrated how standard MLS can be applied to obtain high-order convergence for unstructured problems with curvilinear boundaries[29] and this approach can easily be modified to incorporate the compact optimization framework in Equation 21. As mentioned previously, the optimization framework is flexible in the sense that additional constraints can be added to achieve more theoretical properties (see e.g. [19, 20]). The method is currently being extended to solve incompressible Navier-Stokes equations in a parallel Lagrangian framework where the compactness of the operators will lead to improved scalability for problems requiring high-order resolution.

9. Acknowledgments

We acknowledge advice from George Karniadakis regarding the choice of benchmarks used in this work. 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 Collaboratory on Mathematics for Mesoscopic Modeling of Materials (CM4), under Award Number DE-SC0009247.

References

- [1] Cummins, Sharen J., and Murray Rudman. "An SPH projection method." *Journal of computational physics* 152.2 (1999): 584-607.
- [2] Skillen, Alex, et al. "Incompressible smoothed particle hydrodynamics (SPH) with reduced temporal noise and generalised Fickian smoothing applied to bodywater slam and efficient wavebody interaction." *Computer Methods in Applied Mechanics and Engineering* 265 (2013): 163-173.
- [3] Ihmsen, Markus, et al. "Implicit incompressible SPH." *Visualization and Computer Graphics, IEEE Transactions on* 20.3 (2014): 426-435.
- [4] Shadloo, Mostafa Safdari, et al. "A robust weakly compressible SPH method and its comparison with an incompressible SPH." *International Journal for Numerical Methods in Engineering* 89.8 (2012): 939-956.
- [5] Hu, X. Y., and N. A. Adams. "A SPH model for incompressible turbulence." *arXiv preprint arXiv:1204.5097* (2012).
- [6] Monaghan, J. J. "Smoothed particle hydrodynamics and its diverse applications." *Annual Review of Fluid Mechanics* 44 (2012): 323-346.
- [7] Lancaster, Peter, and Kes Salkauskas. "Surfaces generated by moving least squares methods." *Mathematics of computation* 37.155 (1981): 141-158.
- [8] Trask, Nathaniel, et al. "A scalable consistent second-order SPH solver for unsteady low Reynolds number flows." *Computer Methods in Applied Mechanics and Engineering* (2015).
- [9] Wright, Grady B., and Bengt Fornberg. "Scattered node compact finite difference-type formulas generated from radial basis functions." *Journal of Computational Physics* 212.1 (2006): 99-123.
- [10] Lele, Sanjiva K. "Compact finite difference schemes with spectral-like resolution." *Journal of computational physics* 103.1 (1992): 16-42.
- [11] Monaghan, J. J. "Smoothed particle hydrodynamics." *Reports on progress in physics* 68.8 (2005): 1703.
- [12] Wendland, Holger. *Scattered data approximation*. Vol. 17. Cambridge: Cambridge University Press, 2005.
- [13] Mirzaei, Davoud, Robert Schaback, and Mehdi Dehghan. "On generalized moving least squares and diffuse derivatives." *IMA Journal of Numerical Analysis* (2011): drr030.
- [14] Armentano, Mara G., and Ricardo G. Durn. "Error estimates for moving least square approximations." *Applied Numerical Mathematics* 37.3 (2001): 397-416.
- [15] Zuppa, Carlos. "Error estimates for moving least square approximations." *Bulletin of the Brazilian Mathematical Society* 34.2 (2003): 231-249.
- [16] E, Weinan; Liu, Jian-Guo Essentially compact schemes for unsteady viscous incompressible flows. *J. Comput. Phys.* 126 (1996), no. 1, 122138.

- [17] Li, Hua, et al. "A meshless Hermite-Cloud method for nonlinear fluid-structure analysis of near-bed submarine pipelines under current." *Engineering structures* 26.4 (2004): 531-542.
- [18] Sundar, D. Shyam, and K. S. Yeo. "A high order meshless method with compact support." *Journal of Computational Physics* 272 (2014): 70-87.
- [19] Seibold, Benjamin. "Minimal positive stencils in meshfree finite difference methods for the Poisson equation." *Computer Methods in Applied Mechanics and Engineering* 198.3 (2008): 592-601.
- [20] Seibold, Benjamin. "Performance of algebraic multigrid methods for nonsymmetric matrices arising in particle methods." *Numerical Linear Algebra with Applications* 17.23 (2010): 433-451.
- [21] Trefethen, Lloyd N., and David Bau III. *Numerical linear algebra*. Vol. 50. Siam, 1997.
- [22] Guermond, J. L., Peter Mineev, and Jie Shen. "An overview of projection methods for incompressible flows." *Computer methods in applied mechanics and engineering* 195.44 (2006): 6011-6045.
- [23] Strikwerda, John C. "Finite difference methods for the Stokes and Navier-Stokes equations." *SIAM Journal on Scientific and Statistical Computing* 5.1 (1984): 56-68.
- [24] Lipnikov, Konstantin, Gianmarco Manzini, and Mikhail Shashkov. "Mimetic finite difference method." *Journal of Computational Physics* 257 (2014): 1163-1227.
- [25] Ghia, U. K. N. G., Kirti N. Ghia, and C. T. Shin. "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method." *Journal of computational physics* 48.3 (1982): 387-411.
- [26] Braess, Dietrich. *Finite elements: Theory, fast solvers, and applications in solid mechanics*. Cambridge University Press, 2007.
- [27] Wannier, Gregory H. "A contribution to the hydrodynamics of lubrication." *Quarterly of Applied Mathematics* 8.1 (1950): 1-32.
- [28] Karniadakis, George, and Spencer Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.
- [29] Cueto-Felgueroso, Luis, et al. "Finite volume solvers and Moving Least-Squares approximations for the compressible NavierStokes equations on unstructured grids." *Computer Methods in Applied Mechanics and Engineering* 196.45 (2007): 4712-4736.
- [30] Belytschko, Ted, Yun Yun Lu, and Lei Gu. "Elementfree Galerkin methods." *International journal for numerical methods in engineering* 37.2 (1994): 229-256.
- [31] Liszka, Tadeusz. "An interpolation method for an irregular net of nodes." *International Journal for Numerical Methods in Engineering* 20.9 (1984): 1599-1612.
- [32] Onate, E., and S. Idelsohn. "A mesh-free finite point method for advective-diffusive transport and fluid flow problems." *Computational Mechanics* 21.4-5 (1998): 283-292.
- [33] Mirzaei, Davoud, and Robert Schaback. "Direct meshless local PetrovGalerkin (DMLPG) method: a generalized MLS approximation." *Applied Numerical Mathematics* 68 (2013): 73-82.
- [34] Kim, Do Wan, and Yongsik Kim. "Point collocation methods using the fast moving leastsquare reproducing kernel approximation." *International Journal for Numerical Methods in Engineering* 56.10 (2003): 1445-1464.
- [35] Benzi, Michele, Golub, Gene, and Liesen, Jörg. "Numerical solution of saddle point problems." *Acta Numer.* 14 (2005): 1-137.
- [36] Xu, Jinchao. "Iterative methods by space decomposition and subspace correction." *SIAM review* 34.4 (1992): 581-613.
- [37] Morris, Joseph P., Patrick J. Fox, and Yi Zhu. "Modeling low Reynolds number incompressible flows using SPH." *Journal of computational physics* 136.1 (1997): 214-226.
- [38] Stevens, David, et al. "The use of PDE centres in the local RBF Hermitian method for 3D convective-diffusion problems." *Journal of Computational Physics* 228.12 (2009): 4606-4624.
- [39] Fasshauer, Gregory E. "Solving partial differential equations by collocation with radial basis functions." *Proceedings of Chamonix*. Vol. 1997. Vanderbilt University Press Nashville, TN, 1996.
- [40] Komargodski, Z., and D. Levin. "Hermite type moving-least-squares approximations." *Computers & Mathematics with Applications* 51.8 (2006): 1223-1232.
- [41] Lawson, Charles L., and Richard J. Hanson. *Solving least squares problems*. Vol. 161. Englewood Cliffs, NJ: Prentice-hall, 1974.
- [42] Chen, Wen, Zhuo-Jia Fu, and Ching-Shyang Chen. *Recent advances in radial basis function collocation methods*. Springer, 2014.
- [43] Trask, N., Perego, M., Bochev, P. "A high-order staggered meshless method for elliptic problems" Preprint available at <http://www.dam.brown.edu/people/ntrask/stagmls.pdf>
- [44] Boersma, Bendiks Jan. "A staggered compact finite difference formulation for the compressible NavierStokes equations." *Journal of Computational Physics* 208.2 (2005): 675-690.