Mesh-free collocation for surface differential operators

Abhinav Singh[*] Alejandra Foggia[*] Pietro Incardona[*]
Ivo F. Sbalzarini[*†]

Abstract

We present a mesh-free collocation scheme to discretize intrinsic surface differential operators over surface point clouds with given normal vectors. The method is based on Discretization-Corrected Particle Strength Exchange (DC-PSE), which generalizes finite difference methods to mesh-free point clouds and moving Lagrangian particles. The resulting Surface DC-PSE method is derived from an embedding theorem, but we analytically reduce the operator kernels along the surface normals, resulting in an embedding-free, purely surface-intrinsic computational scheme. We benchmark the scheme by discretizing the Laplace-Beltrami operator on a circle and a sphere, and present convergence results for both explicit and implicit solvers. We then showcase the algorithm on the problem of computing mean curvature of an ellipsoid and of the Stanford Bunny by evaluating the surface divergence of the normal vector field with the proposed Surface DC-PSE method.

1 Introduction

Partial Differential Equations (PDEs) on curved surfaces and differentiable manifolds are an important tool in understanding and studying physical phenomena such as surface flows [27, 19] or active morphogenesis [15]. Analytically solving intrinsic PDEs in curved surfaces quickly becomes impossible for nonlinear PDEs or surfaces that are not spheres. Therefore, numerical methods for solving intrinsic PDEs on curved surfaces are important, and a wide variety of both embedded and embedding-free schemes have been developed to consistently discretize intrinsic differential operators on surfaces.

Embedding-free methods require a parametrization of the surface in order to discretize the differential operators via coordinate charts or a local basis of the manifold [28]. This includes methods based on moving frames [5], a concept originally

[*] Technische Universität Dresden, Faculty of Computer Science, Dresden, Germany.
& Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany.
& Center for Systems Biology Dresden, Dresden, Germany.
[†] Cluster of Excellence Physics of Life, TU Dresden, Dresden, Germany.

This work was funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) under grants FOR-3013 (“Vector- and tensor-valued surface PDEs”), GRK-1907 (“RoSI: role-based software infrastructures”), and SB-350008342 (“OpenPME”), and by the Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) under funding code 031L0160 (project “SPlaT-DM – computer simulation platform for topology-driven morphogenesis”).

developed in continuous group theory, where the surface geometry is locally represented by orthonormal vectors. The concept of moving frames has also been combined with discontinuous Galerkin discretization, e.g., to solve shallow-water equations on arbitrary rotating surfaces [6]. Other embedding-free Finite Element Methods (FEM) include intrinsic SurfaceFEM, which discretizes differential operators on a triangulation of the surface [1, 10] and methods based on Discrete Exterior Calculus (DEC) [18].

Embedding methods discretize the surface problem in the embedding space and use projections to restrict the differential operators computed in the embedding space to the surface. This includes methods that use explicit tracer points to represent the surface, but interpolate to an embedding mesh to evaluate differential operators [13], diffuse-interface methods based on phase-field representations of the surface [17], embedding FEM such as TraceFEM [20], narrow-band level-set methods based on orthogonal extension of the surface quantities [7, 3], level-set methods based on the closest-point transform [22, 14], and volume-of-fluid methods for surface PDE problems [12].

While each of these methods has its specific strengths, embedding methods usually generalize better to complex-shaped or arbitrary surfaces [22]. However, they tend to have higher computational cost, because computations are done in the higher-dimensional embedding space and additional extension (for level sets), right-hand-side evaluation (for phase fields), or interpolation (for closest-point transforms) steps are required, albeit specific optimizations are available, e.g., for level sets [21]. Embedding-free methods are generally more accurate because they avoid the interpolation and projection errors arising when the discretization of the embedding space does not trace the surface exactly, but they tend to be more difficult to implement and harder to generalize to complex-shaped surfaces.

Here, we present a mesh-free collocation method that combines elements from embedding and embedding-free approaches. The method is *algorithmically* embedding-free in the sense that surface quantities are represented on tracer points that are contained in the surface. This also discretizes and represents the surface itself. But the method is *mathematically* related to embedding approaches, since the stencils used to approximate differential operators at the surface points are computed in the embedding space by a reduction operation along the local normal vector, which needs to be known. The resulting method shares properties with moving frame approaches, such as the low dimensionality (and hence low computational cost) and the mesh-free character [5]. It combines these with properties of embedding methods, such as their flexibility in generalizing to complex surfaces [22]. Our method is based on the Discretization-Corrected Particle Strength Exchange (DC-PSE) collocation scheme, which generalizes finite differences to arbitrary point clouds. Given the local surface normal $\boldsymbol{n}$, we derive intrinsic discrete operators by first creating an embedding narrow band and placing collocation points along the normal from each surface point. We then determine the regular DC-PSE operator kernels in the embedding space. These kernels are then reduced under the condition of normal extension $\nabla f \cdot \boldsymbol{n} = 0$ for any (sufficiently) differentiable field f to derive intrinsic kernels at the surface points. This is possible due to the kernel nature of DC-PSE, and it preserves the information from the embedding space in a scheme that only requires computation on the surface points.

This paper is organized as follows: Section 2 recollects the DC-PSE method for convenience and introduces the notation. In Section 3, we describe the Surface DC-PSE scheme for numerically consistent discretization of surface differential operators. We present validation and convergence result in Section 4 and conclude in Section 5.

2 Discretization-Corrected Particle Strength Exchange (DC-PSE)

DC-PSE is a numerical method for discretizing differential operators on irregular distributions of collocation points [24]. The method was originally derived as an improvement over the classic Particle Strength Exchange (PSE) [8] scheme, reducing its quadrature error on irregularly distributed collocation points, but mathematically amounts to a generalization of finite differences [24]. The PSE/DC-PSE class of collocation methods uses mollification with a symmetric smoothing kernel $\eta_\epsilon(\cdot)$ to approximate (sufficiently smooth) continuous functions $f(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega \subseteq \mathbb{R}^d$,

$$f(\boldsymbol{x}_p) \approx f_\epsilon(\boldsymbol{x}_p) = \int_\Omega f(\boldsymbol{x})\, \eta_\epsilon(\boldsymbol{x}_p - \boldsymbol{x})\, \mathrm{d}\boldsymbol{x}, \tag{1}$$

where $f_\epsilon(\boldsymbol{x}_p)$ is a regularized approximation of the function f at location $\boldsymbol{x}_p \in \Omega$ of collocation point p. The scalar ϵ is the smoothing length (or the kernel width) of the mollification. Linear differential operators in $\mathbb{R}^d$,

$$\boldsymbol{D}^{\boldsymbol{\alpha}} = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_d^{\alpha_d}}\,, \tag{2}$$

defined by the multi-index $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{Z}^d$ with $|\boldsymbol{\alpha}| = \sum_{i=1}^{d} \alpha_i$ are approximated by Taylor series expansion to find a discrete operator

$$\boldsymbol{Q}^{\boldsymbol{\alpha}} f(\boldsymbol{x}_p) = \boldsymbol{D}^{\boldsymbol{\alpha}} f(\boldsymbol{x}_p) + O\big(h(\boldsymbol{x}_p)^r\big) \tag{3}$$

at collocation point $\boldsymbol{x}_p$. The order of approximation r depends on the kernel η_ϵ used in Eq. (1), and $h(\boldsymbol{x}_p)$ is the average distance between collocation point p and its neighbors. The Taylor expansion yields integral constraints (also known as *continuous moment conditions*), which the kernel η_ϵ needs to fulfill in order to reach a certain order r [8].

DC-PSE uses different kernels $\eta_\epsilon^{\boldsymbol{\alpha}}(\cdot,\cdot)$ for different differential operators and directly acts on a given quadrature of Eq. (1) with collocation points $\boldsymbol{x}_q \in \Omega$, resulting in the discrete operator:

$$\boldsymbol{Q}_h^{\boldsymbol{\alpha}} f(\boldsymbol{x}_p) = \frac{1}{\epsilon(\boldsymbol{x}_p)^{|\boldsymbol{\alpha}|}} \sum_{\boldsymbol{x}_q \in \mathcal{N}(\boldsymbol{x}_p)} \big(f(\boldsymbol{x}_q) \pm f(\boldsymbol{x}_p)\big)\, \eta_\epsilon^{\boldsymbol{\alpha}}(\boldsymbol{x}_p, \boldsymbol{x}_q), \tag{4}$$

where $\mathcal{N}(\boldsymbol{x}_p)$ are all collocation points in the neighborhood (of a certain radius r_c defined by the kernel width) around point $\boldsymbol{x}_p$, as illustrated in Fig. 1a. The positive sign in the parenthesis is used for odd $|\boldsymbol{\alpha}|$, the negative sign for even $|\boldsymbol{\alpha}|$. This renders the operator conservative on symmetric collocation point distributions, i.e., when

$\eta^{\boldsymbol{\alpha}}_{\epsilon}(\boldsymbol{x}_p, \boldsymbol{x}_q) = \eta^{\boldsymbol{\alpha}}_{\epsilon}(\boldsymbol{x}_q, \boldsymbol{x}_p)$. In DC-PSE, the kernels $\eta^{\boldsymbol{\alpha}}_{\epsilon}$ are thus not determined from continuous moment conditions, as in PSE, but directly from the *discrete moment conditions* that result from substituting Eq. (4) into the quadrature of Eq. (1) [24] for a given set of $\{\boldsymbol{x}_q\}_{q=1}^{N}$. This adapts the kernels to the specific distribution of discretization points (hence the name "discretization-corrected") and avoids the quadrature error of PSE [8], leading to a scheme that is consistent with order r on (almost[1]) arbitrary collocation point sets. This means that at each collocation point, a potentially different kernel is used for the same differential operator if the neighboring collocation points within the kernel support are distributed differently. Evaluating such a kernel at the locations of the collocation points yields a generalized finite-difference stencil, which reduces to the classic compact finite differences on regular grid arrangements of points [24].

DC-PSE kernels are determined at runtime by solving a small system of linear equations for each collocation point, resulting from the discrete moment conditions in its kernel neighborhood. For this, one can choose the function space such that the kernels are compact and symmetric. A frequent choice are polynomials windowed by truncated exponentials [25]:

$$\eta^{\boldsymbol{\alpha}}_{\epsilon}(\boldsymbol{x}_p, \boldsymbol{x}_q) = \eta^{\boldsymbol{\alpha}}_{\epsilon}\left(\frac{\boldsymbol{x}_p - \boldsymbol{x}_q}{\epsilon(\boldsymbol{x}_p)}\right) := \left(\sum_{|\gamma|=\beta_{\min}}^{|\boldsymbol{\alpha}|+r-1} a_{\gamma}(\boldsymbol{x}_p)\left(\frac{\boldsymbol{x}_p - \boldsymbol{x}_q}{\epsilon(\boldsymbol{x}_p)}\right)^{\gamma}\right) \mathrm{e}^{-\left|\frac{\boldsymbol{x}_p - \boldsymbol{x}_q}{\epsilon(\boldsymbol{x}_p)}\right|^2} \tag{5}$$

of finite radius r_c. The polynomial coefficients a_γ are determined for a given $\boldsymbol{\alpha}$ and a given distribution of collocation points, such that the following discrete moment conditions are satisfied:

$$Z_h^{\boldsymbol{\beta}} = \begin{cases} (-1)^{|\boldsymbol{\alpha}|}\boldsymbol{\alpha}!, & \boldsymbol{\beta} = \boldsymbol{\alpha} \\ 0, & \boldsymbol{\beta} \neq \boldsymbol{\alpha}, \quad \beta_{\min} \leqslant |\boldsymbol{\beta}| \leqslant |\boldsymbol{\alpha}| + r - 1, \; \beta_{\min} = \begin{cases} 0, & |\boldsymbol{\alpha}| \text{ odd} \\ 1, & |\boldsymbol{\alpha}| \text{ even} \end{cases} \\ < \infty, & |\boldsymbol{\beta}| = |\boldsymbol{\alpha}| + r \end{cases} \tag{6}$$

where

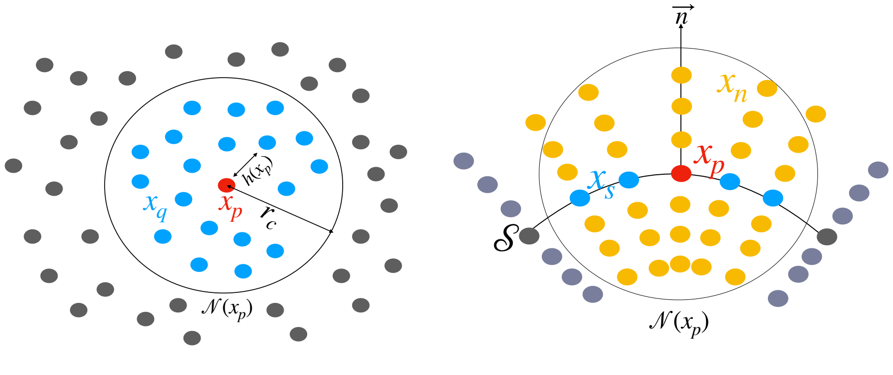
$$Z_h^{\boldsymbol{\beta}}(\boldsymbol{x}) = \frac{1}{\epsilon(\boldsymbol{x}_p)^n} \sum_{\boldsymbol{x}_q \in \mathcal{N}(\boldsymbol{x}_p)} \frac{(\boldsymbol{x}_p - \boldsymbol{x}_q)^{\boldsymbol{\beta}}}{\epsilon(\boldsymbol{x}_p)^{|\boldsymbol{\beta}|}} \, \eta^{\boldsymbol{\alpha}}_{\epsilon}\left(\frac{\boldsymbol{x}_p - \boldsymbol{x}_q}{\epsilon(\boldsymbol{x}_p)}\right) \tag{7}$$

is the discrete moment of order $\boldsymbol{\beta}$ of the kernel $\eta^{\boldsymbol{\alpha}}_{\epsilon}$, and $\beta_{\min}$ is the parity of $|\boldsymbol{\alpha}|$, because the zeroth moment $Z_h^{\boldsymbol{0}}$ vanishes for even operators. Under these conditions, DC-PSE is consistent with order r as long as

$$\frac{h(\boldsymbol{x}_p)}{\epsilon(\boldsymbol{x}_p)} \in O(1), \tag{8}$$

i.e., the kernel width ϵ scales proportionally with the average inter-point distance h around $\boldsymbol{x}_p$.

[1] The collocation point distribution must not be degenerate in the sense that the Vandermonde matrix of the kernel system must have full rank [4]. A trivial example: placing all points along a line and then asking for an approximation of the derivative in the perpendicular direction cannot work.



(a) (b)

Figure 1: **(a) Representation of the DC-PSE method**. The collocation points $\boldsymbol{x}_q$ (blue) are within the symmetric operator support $\mathcal{N}(\boldsymbol{x}_p)$ of radius r_c around the center point $\boldsymbol{x}_p$ (red) at which the discrete operator is to be evaluated. **(b) Representation of the Surface DC-PSE method.** The intrinsic differential operator at a point $\boldsymbol{x}_p$ (red) on the surface $\mathcal{S}$ is discretized with neighbor surface points $\boldsymbol{x}_s$ (blue) and normal extension points $\boldsymbol{x}_n$ (yellow) that are implicitly assumed (but never explicitly created) when determining the kernel. The vector $\boldsymbol{n}$ in the figure shows the normal onto $\mathcal{S}$ at $\boldsymbol{x}_p$.

3 Surface DC-PSE

We generalize DC-PSE to surface differential operators based on the following classic result [22, 16]: Let $\mathcal{S} \subset \mathbb{R}^d$ be a differentiable manifold that has a tubular neighborhood and is orientable[2] and $f : \mathcal{S} \to \mathbb{R}$. Define $F : \mathbb{R}^d \to \mathbb{R}$, such that the restriction $F|_{\mathcal{S}} = f$, and F is constant along the normal direction $\boldsymbol{n}$ of $\mathcal{S}$, i.e., $\nabla F \cdot \boldsymbol{n} = 0$. Then, on the surface $\mathcal{S}$,

$$\nabla_{\mathcal{S}} f = (\nabla F)|_{\mathcal{S}}, \tag{9}$$

where $\nabla_{\mathcal{S}} f$ is the intrinsic surface gradient. A similar result is true for the intrinsic divergence operator ($\nabla_{\mathcal{S}} \cdot$) and for any vector-valued function that is extended by constant tangential extension to all surfaces displaced along the normal of $\mathcal{S}$ [22, 16].

Given this result, it is straightforward to see the advantages of a mesh-free discretization: it allows for conforming discretization of the surface and for exact constant extension by copying points along the normal. One can then derive intrinsic DC-PSE operators at the surface points by constructing a narrow band by orthogonal normal extension, followed by using Eq. (9) to compute the required surface differential operator.

Upon careful inspection of the DC-PSE method, we realize that the constant normal extension can be made internal to the operator evaluation by accumulating the kernel coefficients along the normals. To see this, consider the DC-PSE operator in Eq. (4) in

[2] Every boundary-less smooth surface embedded in $\mathbb{R}^d$ has a tubular neighborhood, and the orientability condition is not restrictive when considered locally [16].

the embedding space. The neighborhood $\mathcal{N}$ for the summation contains both surface points $\boldsymbol{x}_s$ and normally extended points $\boldsymbol{x}_n$, as shown in Fig. 1b. Because the $f(\boldsymbol{x}_n)$ are identical copies of the values of the respective surface points, we note that the prefactors $\big(f(\boldsymbol{x}_s) \pm f(\boldsymbol{x}_p)\big)$ in the summation of Eq. (4) are the same for all extended normal points and the corresponding surface point $\boldsymbol{x}_s$. Hence, they can be factored out and the summation performed over the normally extended kernels:

$$\frac{f(\boldsymbol{x}_s) \pm f(\boldsymbol{x}_p)}{\epsilon(\boldsymbol{x}_p)^{|\alpha|}} \sum_{\boldsymbol{x}_q = \{\boldsymbol{x}_s, \boldsymbol{x}_n : (\boldsymbol{x}_n - \boldsymbol{x}_s) || \boldsymbol{n}(\boldsymbol{x}_s)\}} \eta_\epsilon^{\boldsymbol{\alpha}} \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}_q}{\epsilon\left(\boldsymbol{x}_p\right)} \right) = \frac{f(\boldsymbol{x}_s) \pm f(\boldsymbol{x}_p)}{\epsilon(\boldsymbol{x}_p)^{|\alpha|}} \eta_{\mathcal{S}}(\boldsymbol{x}_p, \boldsymbol{x}_s), \quad (10)$$

defining the surface kernels $\eta_{\mathcal{S}}(\boldsymbol{x}_p, \boldsymbol{x}_s)$. These can be evaluated over only the surface points $\boldsymbol{x}_s = \mathcal{N}_{\mathcal{S}}(\boldsymbol{x}_p)$ in the in-surface neighborhood $\mathcal{N}_{\mathcal{S}}(\boldsymbol{x}_p)$ around the surface point $\boldsymbol{x}_p$, see Fig. 1b, yielding the Surface DC-PSE operator:

$$\boldsymbol{Q}_{\mathcal{S}}^{\boldsymbol{\alpha}} f(\boldsymbol{x}_p) = \frac{1}{\epsilon(\boldsymbol{x}_p)^{|\alpha|}} \sum_{\boldsymbol{x}_s \in \mathcal{N}_{\mathcal{S}}(\boldsymbol{x}_p)} \big(f(\boldsymbol{x}_s) \pm f(\boldsymbol{x}_p)\big) \eta_{\mathcal{S}}(\boldsymbol{x}_p, \boldsymbol{x}_s). \quad (11)$$

Importantly, the surface kernels $\eta_{\mathcal{S}}(\boldsymbol{x}_p, \boldsymbol{x}_s)$, summed over all orthogonally extended points, can directly be computed when determining the kernel weights and without explicitly creating or storing the normally extended points $\boldsymbol{x}_n$. Evaluation of the Surface DC-PSE operators involves only the neighboring points on the surface and requires no embedding, even though the derivation of the method uses an embedding. This is detailed in Algorithm 1.

Algorithm 1: Surface Discretization-Corrected Particle Strength Exchange

Input:

1. Point distribution on a surface $\mathcal{S}$.
2. Cutoff radius for the operator support r_c.
3. Number of normal copies of each surface point during operator construction.
4. Index $\boldsymbol{\alpha}$ of the surface differential operator and order of convergence r.
5. Optional: spacing δn between the normal points. Default: average lateral spacing h between surface points.

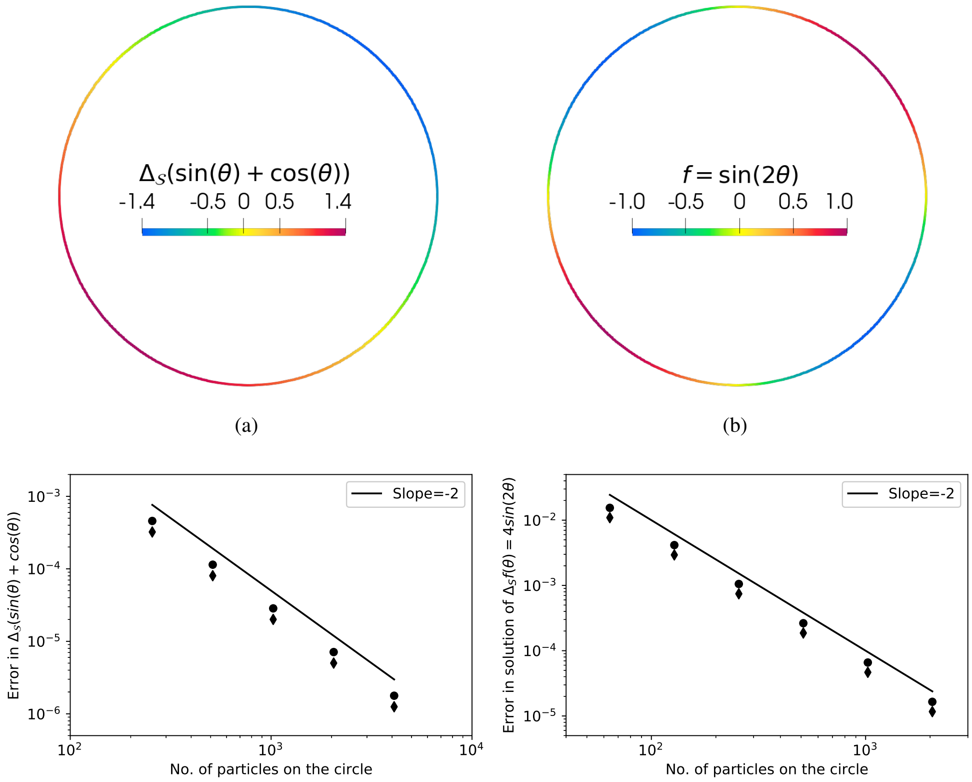
Output: Surface DC-PSE discrete operator with convergence order r.
Algorithm: For each point p on the surface $\mathcal{S}$,

1. Use the provided r_c, r, and $\boldsymbol{\alpha}$ to compute the DC-PSE kernel coefficients for $\eta_\epsilon()$ (Eq. (5)) along with δn-spaced virtual normal points.
2. Sum the kernel coefficients to compute $\eta_{\mathcal{S}}()$ as per Eq. (10).

4 Results

We validate and benchmark the Surface DC-PSE method. First, we verify its convergence in simple test cases with known analytical solution. Then, we show applications to cases with more general surfaces where no analytical solution is available.

4.1 Laplace-Beltrami operator on a circle and a sphere



(a)

(b)

(c)

(d)

Figure 2: **Visualization and convergence of the Laplace-Beltrami operator on the unit circle.** **(a)** Visualization of $\Delta_{\mathcal{S}}(\sin(\theta) + \cos(\theta))$ computed using second-order accurate Surface DC-PSE operators. **(b)** Visualization of the solution of the Poisson equation $\Delta_{\mathcal{S}} f = 4\sin(2\theta)$ solved with second-order accurate Surface DC-PSE operators. **(c)** Convergence plot of the Laplace-Beltrami operator in (a). L_∞(●) and L_2(⧫) norms of the absolute errors are computed against the analytical solution in Eq. (13) for increasing numbers of points on the circle. **(d)** Convergence plot of the Poisson equation solution in (b). L_∞(●) and L_2(⧫) norms of the absolute errors are computed against the analytical solution in Eq. (17) for increasing numbers of points on the circle.

We start by verifying convergence for the Laplace-Beltrami operator on a the unit circle S^1. The collocation points are distributed regularly using equi-angular spacing. We use a normal spacing of $\delta n = 3/(N_p - 1)$ to compute the surface operators in Eq. (11) in a narrow band with 4 layers on each side of the surface. N_p is the number of surface points $\boldsymbol{x}_s$. We choose $r_c = 4.1\delta n$ as the operator support and $r = 2$ as the desired order of convergence. The Laplace-Beltrami operator is characterized by the multi-index $\boldsymbol{\alpha} = (2, 0) + (0, 2)$. Note that this multi-index is 2-dimensional, despite the circle being one-dimensional, since the operators are constructed in the embedding space, but evaluated intrinsically.

We test the numerical approximation of the surface operator on the function

$$f(\theta) = \sin(\theta) + \cos(\theta) \tag{12}$$

in polar coordinates. The error is computed against the analytical solution

$$\Delta_{\mathcal{S}} f(\theta) = \nabla_{\mathcal{S}} \cdot (\nabla_{\mathcal{S}} f(\theta)) = \nabla_\theta^2 f(\theta) = -(\sin(\theta) + \cos(\theta)). \tag{13}$$

The visualization of the numerical solution and the convergence plot of the absolute errors are shown in Fig. 2a,c. We observe second order convergence to the analytical solution, as expected for $r = 2$.

We further test the method on the unit sphere S^2. The collocation points are distributed using the Fibonacci sphere technique [9]. We use a normal spacing of $\delta n = 0.8/(\sqrt[3]{N_p} - 1)$ to determine the surface operators in Eq. (11) in a narrow band with 2 layers on each side of the surface. N_p is the number of points on the sphere. We choose $r_c = 2.9\delta n$ as the operator support and $r = 2$ and $r = 4$ as the desired orders of convergence. The Laplace-Beltrami operator is characterized by the three-dimensional multi-index $\boldsymbol{\alpha} = (2, 0, 0) + (0, 2, 0) + (0, 0, 2)$. We test the numerical approximation of the surface operator on the scalar spherical harmonic function

$$f(\theta, \phi) = Y_{lm} \tag{14}$$

in spherical coordinates for the mode $l = 4$, $m = 0$ (Fig. 3a). The error is computed against the analytical solution

$$\Delta_{\mathcal{S}} f = \nabla_{\mathcal{S}} \cdot (\nabla_{\mathcal{S}} f(\theta, \phi)) = -l(l + 1)Y_{lm} \tag{15}$$

and is plotted in Fig. 3c.

We also use this test case to benchmark against the Closest Point method [22] with L_2 and L_∞ errors plotted in Fig. 3c. Surface DC-PSE is one to two orders of magnitude more accurate than the Closest Point method for the same operator order.

Finally, we perform a strong scaling benchmark of the computation time with increasing numbers of CPU cores with both codes implemented in the parallel computing library OpenFPM [11, 26] in C++ and run on the same hardware. The results in Fig. 3b show that one evaluation of the Surface DC-PSE operators over the whole domain is about one order of magnitude faster than using the Closest-Point method [22] and scales better with increasing numbers of parallel CPU cores. When including also the time required to determine the Surface DC-PSE operator kernels, the Surface DC-PSE method is about 1.5 orders of magnitude slower than constructing the Closest-Point representation in a narrow band, but shows similar parallel scalability. However, for Eulerian simulations, the kernels have to be determined only once at the beginning of a simulation, or they can be loaded from files for standard point distributions.

4.2 Poisson equation on a circle and a sphere

Given the DC-PSE surface operators, implicit equations can be solved by solving a linear system of equations with a system matrix constructed using the discrete operators. We test this by solving the Poisson equation on the unit circle S^1:

$$\Delta_{\mathcal{S}} f = 4\sin(2\theta) \qquad \theta \in \Omega = S^1 \backslash (1,0) \tag{16}$$

with Dirichlet boundary condition at one point $(1,0)$ conforming to the analytical solution

$$f(\theta) = \sin(2\theta) \qquad \theta \in \Omega \cup (1,0). \tag{17}$$

We use the same Surface DC-PSE operators as in the previous subsection to construct the system of equations, which is then solved using the KSPGMRES solver from PETSc [2]. Fig. 2b,d show the numerically computed solution f, along with the convergence plot of the absolute error toward the analytical solution.

Next, we test the method in three dimensions by solving the Poisson equation on the sphere S^2:

$$\Delta_{\mathcal{S}} f = -20Y_{4,0} \tag{18}$$

with Dirichlet boundary condition along the equatorial circle parallel to the $y-z$ plane conforming to the analytical solution

$$f = Y_{40}. \tag{19}$$

We solve the resulting linear system with KSPGMRES from PETSc [2] without preconditioning. The convergence plots for orders $r = 2$ and $r = 4$ are shown in Fig. 3d.

4.3 Mean curvature computation

We verify Surface DC-PSE for vector fields by computing the mean curvature H of an ellipsoid

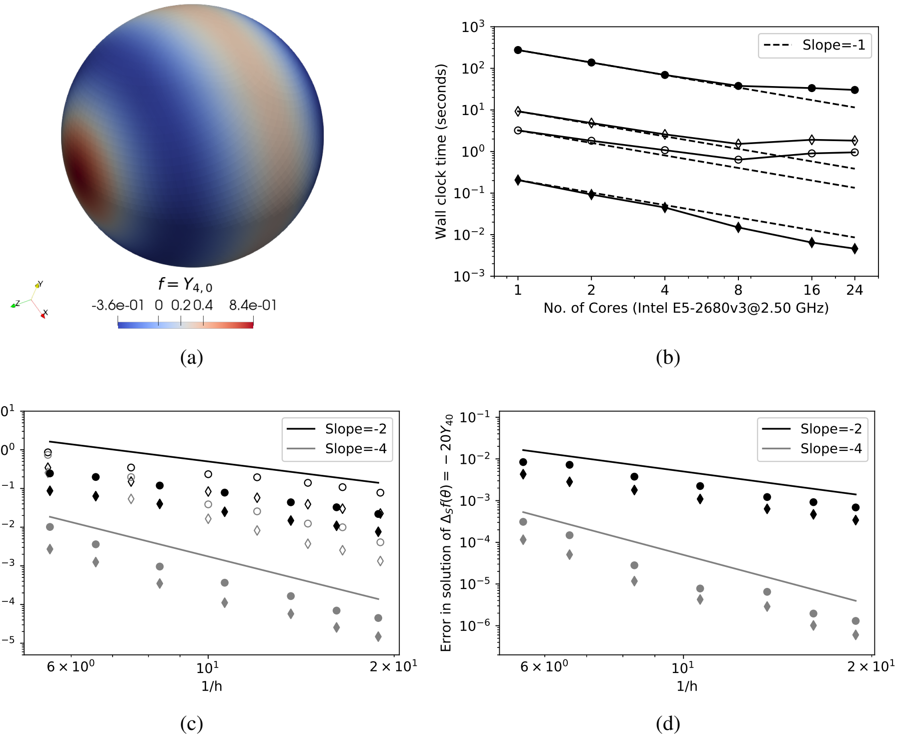
$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{20}$$

with $a = 1$, $b = 0.8$, $c = 0.75$ and parametrization (u, v)

$$x = a\cos u \sin v, \qquad y = b \sin u \sin v, \qquad z = c\cos v. \tag{21}$$

We compute mean curvature as the divergence of the surface normal field and compare it with the analytical solution

$$\begin{aligned} H &= -\frac{1}{2}\nabla_{\mathcal{S}} \cdot \boldsymbol{n} \\ &= \frac{abc[3(a^2+b^2) + 2c^2 + (a^2+b^2-2c^2)\cos(2v) - 2(a^2-b^2)\cos(2u)\sin^2 v]}{8[a^2b^2\cos^2 v + c^2(b^2\cos^2 u + a^2\sin^2 u)\sin^2 v]^{3/2}}. \end{aligned} \tag{22}$$



(a)

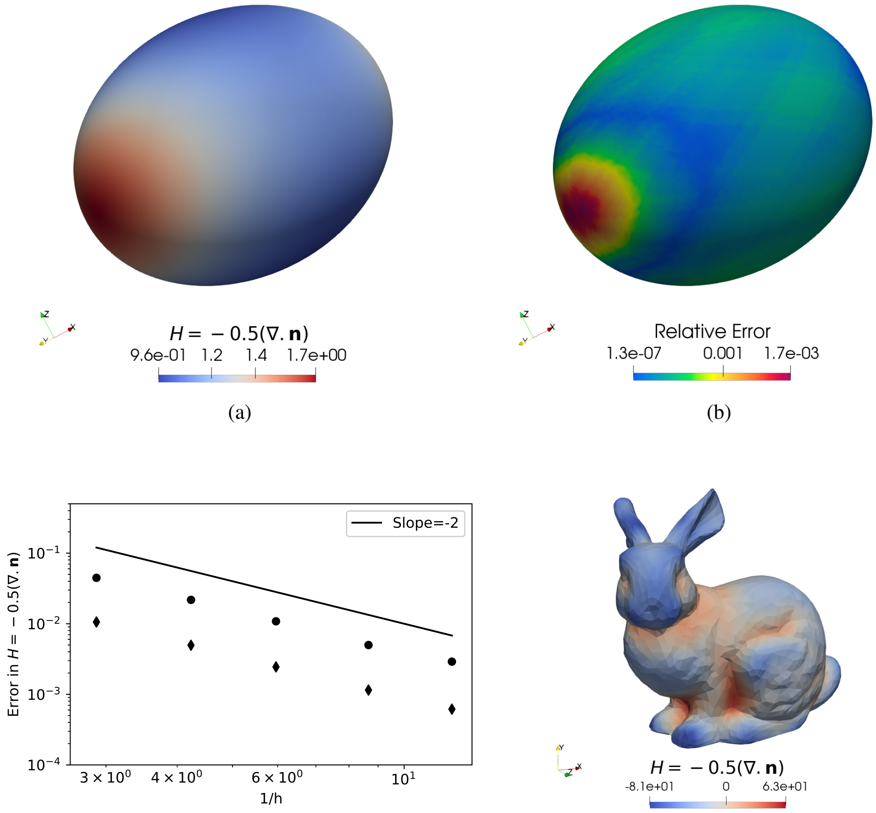
(b)

(c)

(d)

Figure 3: **Visualization and convergence of the Laplace-Beltrami operator on the unit sphere.** **(a)** Visualization of the spherical harmonic function $Y_{4,0}$. **(b)** Results of a strong scalability test for the computation of $\Delta_{\mathcal{S}}(Y_{4,0})$ for average spacing $h = 0.05$ using Surface DC-PSE operators (●, ⧫) and the Closest Point (CP) method (○, ◊) [22] on increasing numbers of CPU cores. Times for the creation of Surface DC-PSE operators (●) and for the creation of the CP representation (○) are required once in Eulerian simulations. Times for the computation of $\Delta_{\mathcal{S}}(Y_{4,0})$ for Surface DC-PSE operators (⧫) include evaluation of the operators, and for the CP method (◊) include extension of initial condition, computation of the surface Laplacian, and extension of the result. Dashed lines show the ideal linear speedup. **(c)** Convergence plot of $\Delta_{\mathcal{S}}(Y_{4,0})$ for Surface DC-PSE (L_∞(●), L_2(⧫)) and the Closest Point method (L_∞(○), L_2(◊)) using second-order (black) and fourth-order (gray) approximations. Norms of the absolute errors are computed against the analytical solution in Eq. (15) for increasing numbers of points (decreasing average spacing h). **(d)** Convergence plot of the solution of the Poisson equation $\Delta_{\mathcal{S}} f = -20Y_{4,0}$ using second-order (black) and fourth-order (gray) approximations. L_∞(●) and L_2(⧫) norms of the absolute errors are computed against the analytical solution in Eq. (19) for increasing numbers of surface points.

We approximate the surface divergence operator using Surface DC-PSE with $\delta n = 4.5/(N_p - 1)$, $r_c = 3.1\delta n$, and $r = 2$. The result and the convergence plot of the



(a)

(b)

(c)

(d)

Figure 4: **Mean curvature computation using the Surface DC-PSE divergence operator. (a)** Visualization of the mean curvature of an ellipsoid, numerically computed as the divergence $\nabla \cdot \boldsymbol{n}$ of the surface normal vector field $\boldsymbol{n}$ using second-order accurate Surface DC-PSE operators. **(b)** Visualization of the relative error in the computed curvature from (a) on 32,258 surface points in comparison with the analytical solution in Eq. (22). **(c)** Convergence plot of the mean curvature computation. $L_\infty(\bullet)$ and $L_2(\blacklozenge)$ norms of the absolute errors are computed against the analytical solution in Eq. (22) for decreasing average surface-point spacing h. **(d)** Visualization of the mean curvature computed on the Stanford bunny with 2960 points using second-order accurate Surface DC-PSE operators.

absolute errors are shown in Fig. 4a,c. As specified by r, we observe second-order convergence to the analytical solution when decreasing the average spacing h between the points. The relative errors are visualized in Fig. 4b. They concentrate around extremal points of the curvature, as expected.

Finally, we apply the same mean-curvature computation to an arbitrary surface with no analytical solution, the Stanford bunny from the Stanford Computer Graphics Laboratory. We use the down-sampled version of the original data set with 2960 points on the surface, obtained from `https://www.stlfinder.com/model/stanford-bunny-S4kAUsKI/3091553`. The result is visualized in Fig. 4d, showing that the Surface DC-PSE method qualitatively works also for non-algebraic surfaces.

5 Conclusions

We have presented a mesh-free collocation scheme for consistently approximating intrinsic differential operators on curved surfaces. Our scheme is based on the DC-PSE method for discretizing bulk differential operators on irregular point clouds [24]. We have derived the present surface-intrinsic version by realizing that the kernel evaluations can be factored out across points created by constant normal extension, and that the partial sums over the kernels can be pre-computed and stored on the surface points only. This yields a method that is easy to implement and is computationally efficient, as it only requires collocation points *on* the surface. In this sense, Surface DC-PSE combines features from embedding methods with features from embedding-free methods. The operators are determined in an embedding formulation, but result in an embedding-free algorithm for operator evaluation.

We have verified the method in different test cases with known analytical solutions. This included evaluating the Laplace-Beltrami operator on the unit circle and the unit sphere, solving Poisson equations on the unit circle and the unit sphere using an implicit solver, and computing mean curvature of an ellipsoid as the divergence of the normal vector field. In all cases, the scheme converged as expected. We then applied the method to compute the mean curvature of the Stanford bunny, showing an application to a non-analytical surface.

Despite its advantages, Surface DC-PSE also has a number of limitations: First, the normal field is required as an input, which can be limiting or introduce additional errors in cases where the normals are not analytically know or given in the data. Second, for a given point distribution, the numerical error is limited by the curvature of the given surface and depends on the average spacing between the surface points and the normal extension points (see Figs. 1b and 4b). This can be alleviated by using higher resolution in higher-curvature areas. The required minimum resolution can be pre-determined based on an approximation of the curvature. Lastly, determining the DC-PSE kernels is computationally expensive, as it involves solving a small linear system of equations for each point. For Eulerian simulations, where the collocation points do not move, the kernels have to be determined once at the beginning of the simulation. However, if points move, e.g. in a Lagrangian simulation, the kernels need to be recomputed at each time step. While the cost may be amortized by a gain in accuracy and stability [24], it is still significant.

In future work, we will consider extensions of Surface DC-PSE to Lagrangian problems involving moving and deforming surfaces. This also includes simulations of deformable surfaces, where the surface deformation itself results from intrinsic force-balance equations [23, 15]. We will also consider coupling Surface DC-PSE with regular DC-PSE in the surrounding space in order to describe coupled bulk-surface phenomena.

In summary, we have extended the mesh-free collocation method DC-PSE to problems on curved surfaces, requiring only intrinsic surface points. Like DC-PSE, also Surface DC-PSE computes the operator kernels numerically at runtime and is consistent for any desired order of convergence r. This makes the presented algorithm particularly attractive for higher-order intrinsic operators, such as the fourth-order operators in Fig. 3, and for determining the system matrices of implicit equations on surfaces or implicit time integration schemes.

Acknowledgments

This work was funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) under grants FOR-3013 ("Vector- and tensor-valued surface PDEs"), GRK-1907 ("RoSI: role-based software infrastructures"), and SB-350008342 ("OpenPME"), and by the Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) under funding code 031L0160 (project "SPlaT-DM – computer simulation platform for topology-driven morphogenesis").

References

[1] Elena Bachini, Matthew W. Farthing, and Mario Putti. Intrinsic finite element method for advection-diffusion-reaction equations on surfaces. *Journal of Computational Physics*, 424:109827, January 2021.

[2] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries. In Erlend Arge, Are Magnus Bruaset, and Hans Petter Langtangen, editors, *Modern Software Tools for Scientific Computing*, pages 163–202. Birkhäuser, Boston, MA, 1997.

[3] Michael Bergdorf, Ivo F. Sbalzarini, and Petros Koumoutsakos. A Lagrangian particle method for reaction–diffusion systems on deforming surfaces. *Journal of Mathematical Biology*, 61(5):649–663, November 2010.

[4] George C. Bourantas, Bevan L. Cheeseman, Rajesh Ramaswamy, and Ivo F. Sbalzarini. Using DC PSE operator discretization in Eulerian meshless collocation methods improves their robustness in complex geometries. *Computers & Fluids*, 136:285–300, September 2016.

[5] Sehun Chun. Method of moving frames to solve conservation laws on curved surfaces. *Journal of Scientific Computing*, 53(2):268–294, 2012.

[6] Sehun Chun and Claes Eskilsson. Method of moving frames to solve the shallow water equations on arbitrary rotating curved surfaces. *Journal of Computational Physics*, 333:1–23, 2017.

[7] Georges-Henri Cottet and Emmanuel Maitre. A semi-implicit level set method for multiphase flows and fluid–structure interaction problems. *Journal of Computational Physics*, 314:80–92, June 2016.

[8] Jeff D. Eldredge, Anthony Leonard, and Tim Colonius. A General Deterministic Treatment of Derivatives in Particle Methods. *Journal of Computational Physics*, 180(2):686–709, August 2002.

[9] Álvaro González. Measurement of areas on a sphere using Fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 42(1):49–64, November 2009.

[10] J. Grande, M.A. Olshanskii, and A. Reusken. A space-time fem for pdes on evolving surfaces. pages 211–222, 2014.

[11] Pietro Incardona, Antonio Leo, Yaroslav Zaluzhnyi, Rajesh Ramaswamy, and Ivo F. Sbalzarini. OpenFPM: A scalable open framework for particle and particle-mesh codes on parallel computers. *Computer Physics Communications*, 241:155–177, August 2019.

[12] Ashley J. James and John Lowengrub. A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant. *J. Comput. Phys.*, 201:685–722, 2004.

[13] Shingyu Leung and Hongkai Zhao. A grid based particle method for moving interface problems. *J. Comput. Phys.*, 228(8):2993 – 3024, 2009.

[14] Colin B. Macdonald, Jeremy Brandman, and Steven J. Ruuth. Solving eigenvalue problems on curved surfaces using the Closest Point Method. *Journal of Computational Physics*, 230(22):7944–7956, September 2011. arXiv: 1106.4351.

[15] Alexander Mietke, Frank Jülicher, and Ivo F. Sbalzarini. Self-organized shape dynamics of active surfaces. *Proceedings of the National Academy of Sciences*, 116(1):29–34, January 2019.

[16] Thomas März and Colin B. Macdonald. Calculus on surfaces with general closest point functions. *SIAM Journal on Numerical Analysis*, 50(6):3303–3328, 2012.

[17] Michael Nestler, Ingo Nitschke, and Axel Voigt. A finite element approach for vector- and tensor-valued surface PDEs. *Journal of Computational Physics*, 389:48–61, July 2019.

[18] Ingo Nitschke, Sebastian Reuther, and Axel Voigt. Discrete Exterior Calculus (DEC) for the Surface Navier-Stokes Equation. In Dieter Bothe and Arnold Reusken, editors, *Transport Processes at Fluidic Interfaces*, pages 177–197. Springer International Publishing, Cham, 2017.

[19] Ingo Nitschke, Sebastian Reuther, and Axel Voigt. Liquid Crystals on Deformable Surfaces. *arXiv:1911.11859 [cond-mat]*, November 2019. arXiv: 1911.11859.

[20] Maxim A. Olshanskii and Arnold Reusken. Trace Finite Element Methods for PDEs on Surfaces. In Stéphane P. A. Bordas, Erik Burman, Mats G. Larson, and Maxim A. Olshanskii, editors, *Geometrically Unfitted Finite Element Methods and Applications*, Lecture Notes in Computational Science and Engineering, pages 211–258, Cham, 2017. Springer International Publishing.

[21] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155:410–438, 1999.

[22] Steven J. Ruuth and Barry Merriman. A simple embedding method for solving partial differential equations on surfaces. *Journal of Computational Physics*, 227(3):1943–1961, January 2008.

[23] Guillaume Salbreux and Frank Jülicher. Mechanics of active surfaces. *Physical Review E*, 96(3):032404, September 2017. Publisher: American Physical Society.

[24] Birte Schrader, Sylvain Reboux, and Ivo F. Sbalzarini. Discretization correction of general integral PSE Operators for particle methods. *Journal of Computational Physics*, 229(11):4159–4182, June 2010.

[25] Birte Schrader, Sylvain Reboux, and Ivo F. Sbalzarini. Choosing the Best Kernel: Performance Models for Diffusion Operators in Particle Methods. *SIAM Journal on Scientific Computing*, 34(3):A1607–A1634, January 2012.

[26] Abhinav Singh, Pietro Incardona, and Ivo F. Sbalzarini. A C++ expression system for partial differential equations enables generic simulations of biological hydrodynamics. *The European Physical Journal E*, 44(9):117, September 2021.

[27] A. Voigt. Fluid deformable surfaces. *Journal of Fluid Mechanics*, 878:1–4, November 2019.

[28] Ruimin Wang, Zhouwang Yang, Ligang Liu, and Qing Chen. Discretizing Laplace–Beltrami Operator from Differential Quantities. *Communications in Mathematics and Statistics*, 1(3):331–350, September 2013.