Domain Decomposition for the Closest Point Method

Ian May, Ronald D. Haynes, and Steven J. Ruuth

1 Introduction

The discretization of elliptic PDEs leads to large coupled systems of equations. Domain decomposition methods (DDMs) are one approach to the solution of these systems, and can split the problem in a way that allows for parallel computing. Herein, we extend two DDMs to elliptic PDEs posed intrinsic to surfaces as discretized by the Closest Point Method (CPM) [19, 16]. We consider the positive Helmholtz equation

$$(c - \Delta_S)\, u = f, \tag{1}$$

where $c \in \mathbb{R}^+$ is a constant and Δ_S is the Laplace-Beltrami operator associated with the surface $S \subset \mathbb{R}^d$. The evolution of diffusion equations by implicit time-stepping schemes and Laplace-Beltrami eigenvalue problems [14] both give rise to equations of this form. The creation of efficient, parallel, solvers for this equation would ease the investigation of reaction-diffusion equations on surfaces [15], and speed up shape classification [18], to name a couple applications.

Several methods exist for the discretization of surface intrinsic PDEs. The surface may be parametrized to allow the use of standard methods in the parameter space [10]. Unfortunately, many surfaces of interest do not have simple, or even known, parametrizations. Given a triangulation of the surface, a finite element discretization can be formed [9]. This approach leads to a sparse and symmetric system but is sensitive to the quality of the triangulation. Level set methods for surface PDEs [4] solve the problem in a higher dimensional embedding space over a narrow band

Ian May

Simon Fraser University, 8888 University Dr., Burnaby, BC V5A 1S6, e-mail: `mayianm@sfu.ca`

Ronald Haynes

Memorial University of Newfoundland, 230 Elizabeth Ave, St. Johns, NL A1C 5S7 e-mail: `rhaynes@mun.ca`

Steven Ruuth

Simon Fraser University, 8888 University Dr., Burnaby, BC V5A 1S6, e-mail: `sruuth@sfu.ca`

containing the surface. The solution of model equation (1) by this method requires using gradient descent, as the approach was formulated only for parabolic problems. The CPM is also discretized over a narrow band in the embedding space, but has the advantage of using a direct discretization of equation (1).

The solution of the linear system arising from the CPM discretization of the model equation (1) has relied primarily on direct methods, although a multigrid method was discussed in [5]. Herein we formulate restricted additive Schwarz (RAS) and optimized restricted additive Schwarz (ORAS) solvers compatible with the CPM to step towards efficient iterative solvers and to allow for parallelism. The optimized variant of the classical RAS solver uses Robin transmission conditions (TCs) to pass additional information between the subdomains [11, 20], and can accelerate convergence dramatically. This formulation is described in Sections 4 and 5 after reviewing the CPM in Section 2 and (O)RAS solvers in Section 3. Then, we discuss a PETSc [1, 2] implementation and show some numerical examples in Section 6. A more thorough exploration of these solvers, and an initial look at their use as preconditioners, can be found in May's thesis [17].

2 The closest point method

The CPM was introduced in [19] as an embedding method for surface intrinsic PDEs. It allows the reuse of standard flat space discretizations of differential operators and provides a surface agnostic implementation. At the core of this method is the closest point mapping, $CP_{\mathcal{S}}(x) = \arg\min_{y \in \mathcal{S}} |x - y|$ for $x \in \mathbb{R}^d$, which identifies the closest point on the surface for (almost) any point in the embedding space. This mapping exists and is continuous in the subset of $\mathbb{R}^d$ consisting of all points within a distance κ_∞^{-1} of the surface, where κ_∞ is an upper bound on the principal curvatures of the surface [6].

From this mapping, an extension operator E can be defined that sends functions defined on the surface, $f : \mathcal{S} \to \mathbb{R}$, to functions defined on the embedding space via composition with the closet point mapping, $Ef = f \circ CP_{\mathcal{S}}$. The extended functions are constant in the surface normal direction and retain their original values on the surface. This extension operator can be used to define surface intrinsic differential operators from their flat space analogs [19].

Discretization typically requires a Cartesian grid on the embedding space within a narrow tube surrounding the surface. The extension operator can be defined by any suitable interpolation scheme, with tensor product barycentric Lagrangian interpolation [3] being used here. As such, the computational tube must be wide enough to contain the interpolation stencil for any point on the surface. Using degree p interpolation and a grid spacing of Δx requires that the tube contain all points within a distance of $\gamma = \Delta x (p + 2) \sqrt{d}/2$ from the surface, thus limiting the acceptable grid spacings in relation to κ_∞.

The grid points within the computational tube form the set of active nodes, Σ_A. For (1), we need only discretize the regular Laplacian on $\mathbb{R}^d$. Here we consider the second order accurate centered difference approximation requiring $2d + 1$ points. Around Σ_A and lying outside the tube, a set of ghost nodes, Σ_G, is formed from any incomplete differencing stencils. With a total of N_A active nodes and N_G ghost nodes, we define the discrete Laplacian and extension operators, $\Delta^h : \mathbb{R}^{N_A+N_G} \to \mathbb{R}^{N_A}$ and $\mathbf{E} : \mathbb{R}^{N_A} \to \mathbb{R}^{N_A+N_G}$, where Δ^h applies the centered difference Laplacian over all active nodes, and $\mathbf{E}$ is the discretization of E. $\mathbf{E}$ extends data on the active nodes to both the active and ghost nodes, and has entries consisting of the interpolation weights for each node's closest point.

The Laplace-Beltrami operator can be directly discretized as $\Delta^h_{S,dir} = \Delta^h \mathbf{E}$, which was used successfully for parabolic equations with explicit time-stepping in [19]. However, for implicit time-stepping [16] and eigenvalue problems [14] a modified form is needed. In [16] it was recognized that there was a redundant interpolation being performed, and that its removal could stabilize the discretization. The stabilized form $\Delta^h_S = -\frac{2d}{\Delta x^2}\mathbf{I} + \left(\frac{2d}{\Delta x^2}\mathbf{I} + \Delta^h\right)\mathbf{E}$, will be used in the remainder of this work.

3 (Optimized) Restricted additive Schwarz

Both RAS and ORAS are overlapping DDMs, and can work on the same set of subdomains (given an additional overlap condition for ORAS [20]). We define these solvers from the continuous point of view and subsequently discretize, rather than defining them purely algebraically. This will ease the discussion of TCs within the context of the CPM later in Section 5.

First, the whole surface S is decomposed into N_S disjoint subdomains, $\widetilde{S}_j$, for $j = 1, \ldots, N_S$. These disjoint subdomains are then grown to form overlapping subdomains S_j, whose boundaries are labelled depending on where they lie in the disjoint partitioning. Taking $\Gamma_{jk} = \partial S_j \cap \widetilde{S}_k$ gives $\partial S_j = \bigcup_k \Gamma_{jk}$ and allows the definition of the local problems

$$
\begin{cases}
(c - \Delta_S)\, u_j^{(n+1)} = f, & \text{in } S_j, \\
\mathcal{T}_{jk} u_j^{(n+1)} = \mathcal{T}_{jk} u^{(n)}, & \text{on } \Gamma_{jk},\ k = 1, \ldots, N_S,\ k \neq j,
\end{cases}
\tag{2}
$$

where $\mathcal{T}_{jk}$ are generally linear boundary operators defining the TCs. RAS is achieved by choosing $\mathcal{T}_{jk}$ as identity operators, corresponding to Dirichlet TCs, while ORAS uses Robin TCs, $\mathcal{T}_{jk} = \left(\frac{\partial}{\partial \hat{\mathbf{n}}_{jk}} + \alpha\right)$, where $\hat{\mathbf{n}}_{jk}$ is the outward pointing boundary normal on Γ_{jk} and $\alpha \in \mathbb{R}^+$ is a constant weight on the Dirichlet contribution.

The subproblems in equation (2) are initialized with a guess for the global solution $u^{(0)}$ (defined at least over the boundaries $\Gamma_{jk}, \forall j, k$), which is usually just taken as $u^{(0)} = 0$. After all of the subproblems have been solved a new global solution is constructed with respect to the disjoint partitioning, $u^{(n+1)} = \sum_j u_j^{(n+1)}\Big|_{\widetilde{S}_j}$, where

the use of the term *restricted* indicates that the portion of the local solutions in the overlap regions are discarded. From this new approximation for the global solution the local problems may be solved again with new boundary data, and the process repeats until the global solution is satisfactory.

4 Subdomain construction

To solve problems arising from the CPM we first need to decompose the global set of active nodes Σ_A. (O)RAS solvers rely on both a disjoint partitioning of the active nodes and an induced overlapping partitioning. Following the notation in Section 3, disjoint partitions will be denoted by $\widetilde{\Sigma}_j$, overlapping partitions by Σ_j, and the boundaries of the overlapping partitions by Λ_j.

To ensure the solvers work on a variety of surfaces, we seek an automated and surface agnostic partitioning scheme to generate the disjoint partitions. METIS [13] is a graph partitioner that is frequently used within the DD community to partition meshes [8]. The stencils of Δ^h and $\mathbf{E}$ may be used to induce connectivity between the active nodes and define a graph. Here we only consider nearest neighbor coupling through the stencil for Δ^h. Fig. 1 shows a portion of one such disjoint partition, in black circles, for a circular surface.

With $\widetilde{\Sigma}_j$ obtained from METIS, overlapping subdomains Σ_j can be formed. This construction proceeds in the following steps:

1. All nodes in $\widetilde{\Sigma}_j$ are added to Σ_j.
2. N_O layers of overlap nodes are added around Σ_j. Layers are added one at a time from globally active nodes neighboring Σ_j.
3. A subset of the ghost nodes, Σ_G, are placed in Σ_j^G which consists of nodes that neighbor a member of Σ_j.
4. The shapes of the disjoint and overlapping subdomains are not known in advance. The boundary $\partial \mathcal{S}_j$ is approximated discretely by the closest points of the final layer of overlap nodes, and held in the set Λ_j.
5. Nodes needed to complete stencils from the ambient Laplacian or extension operator, including extension from the points $x_i \in \Lambda_j$, are placed in the set Σ_j^{BC}.
6. For ORAS a layer of ghost nodes around Σ_j^{BC} are also placed in Σ_j^{BC}.

The active nodes in the jth subdomain consist of Σ_j and the active portion of Σ_j^{BC}. Σ_j^{BC} is kept separate as that is where the TCs in Section 5 are defined. Each of these sets are shown in Fig. 1, which shows a portion of one subdomain on a circle in the vicinity of the points in Λ_j at one of its boundaries.

The Robin TCs, to be defined in Section 5, need some final information about the subdomain. Every node in Σ_j^{BC} is identified with the point in Λ_j that is closest to it. This identification will be used to override the global closest point function in the following section. For each point in Λ_j we also need to know the direction that is simultaneously orthogonal to the boundary and the surface normal direction. We call this the conormal direction. It is in this direction that the Neumann component of the

Fig. 1 A portion of a subdomain from a circular problem with $N_S = 8$ subdomains and $N_O = 4$ layers of overlap is shown here. The nodes are marked according to their role in the subdomain as described in Section 4. The points belonging to the set Λ_j are shown by (nearly coincident) black stars.
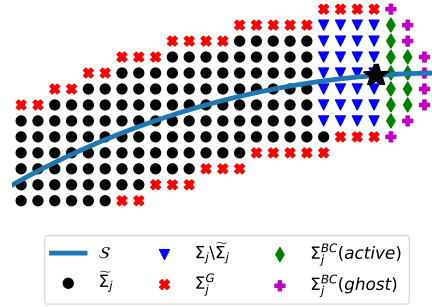
Robin condition will be enforced. However, the discrete nature of Λ_j makes this construction difficult. Instead we define the conormal vectors from the point of view of the boundary nodes. Take $x_i \in \Sigma_j^{BC}$ as a node whose associated conormal direction, $\hat{q}_i$, is sought. Let y_i be its closest point in Λ_j, and $\hat{n}_i$ be the unit surface normal there. Connecting the boundary location to the boundary node via $d_i = x_i - y_i$, we obtain a usable approximation to the conormal by computing the component of d_i that is orthogonal to $\hat{n}_i$ and normalizing, i.e., $\hat{q}_i = (d_i - (d_i \cdot \hat{n}_i)\,\hat{n}_i)/(|d_i - (d_i \cdot \hat{n}_i)\,\hat{n}_i|)$. In the unlikely event that d_i lies perfectly in the surface normal direction, we set $\hat{q}_i = 0$ which recovers the natural boundary condition on the computational tube as discussed at the end of Section 5.

5 Transmission conditions

Boundary conditions in the CPM are imposed by modifying the extension operator over the nodes Σ_j^{BC} beyond the surface boundary [14]. As such, the local operators will take the form

$$\mathbf{A}_j = \left(c + \frac{2d}{h^2}\right) - \left(\frac{2d}{h^2} + \Delta_j^h\right)\begin{bmatrix}\mathbf{E}_j \\ \mathbf{T}_j\end{bmatrix}, \tag{3}$$

where $\mathbf{E}_j$ is the extension operator for the nodes in Σ_j as inherited from the global operator and $\mathbf{T}_j$ is the modified extension operator for the nodes in Σ_j^{BC}. When solving for the local correction to the solution the right hand side of the local problem, $\mathbf{A}_j v_j = r_j$, will be the restriction of the residual to Σ_j. The final rows of the right hand side, those lying over Σ_j^{BC}, become zeros corresponding to the homogenous TCs.

Homogeneous Dirichlet TCs can be enforced to first order accuracy by extending zeros over all of Σ_j^{BC}. With the right hand side already set to zero there, the modified extension reduces to the identity mapping, $\mathbf{T}_j = \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}$, with the zero matrix padding the columns corresponding to the interior nodes.

We discretize the Robin condition

$$\left.\frac{\partial u}{\partial \hat{q}_i}\right|_{CP_{S_j}(x_i)} + \alpha u \left(CP_{S_j}(x_i) \right) = 0, \tag{4}$$

using a forward difference in the $\hat{q}_i$ direction for each node in Σ_j^{BC} and the first order accurate Dirichlet condition from above. Taking the partial derivative $\frac{\partial u}{\partial d_i}$, and applying the change of variables $d_i = \hat{q}_i + \hat{n}_i$, allows one to write the Neumann term in equation (4) in terms of the displacement vector d_i from Section 4. Assuming for the moment that d_i and $\hat{q}_i$ are not perpendicular, the derivative in the conormal direction can be approximated by $\left.\frac{\partial u}{\partial \hat{q}_i}\right|_{CP_{S_j}(x_i)} \approx \frac{u(x_i) - u\left(CP_{S_j}(x_i)\right)}{d_i \cdot \hat{q}_i}$ where CP_{S_j} denotes the modified closest point function identifying points in Σ_j^{BC} with points in Λ_j. Combining this with (4), and applying the identity extension for the Dirichlet component, $u(x_i) = u(CP_{S_j}(x_i))$, we find that $\mathbf{T}_j$ must enforce the extension $u(x_i) = \frac{u\left(CP_{S_j}(x_i)\right)}{1 + \alpha d_i \cdot \hat{q}_i}$, with $u\left(CP_{S_j}(x_i)\right)$ replaced by the same interpolation used in the global scheme discussed in Section 2.

As d_i approaches the surface normal direction, $d_i \cdot \hat{q}_i$ will tend to zero. In this event, the extension reduces to $u(x_i) = u\left(CP_{S_j}(x_i)\right)$, which is just the standard extension corresponding to the interior. Fortuitously, this case arises when the point x_i lies adjacent to the interior points where this condition would be applied anyway, and in our experience this ensures that the method remains robust.

6 Results

The solvers described in the previous sections were implemented in C++, with PETSc [1, 2] providing the linear algebra data structures and MPI parallelization, and Umfpack [7] providing the local solutions. Here we focus on evaluating the solver, though in practice one should accelerate the solver with a Krylov method. The (O)RAS solver was placed into a PETSc PCSHELL preconditioner, allowing it to be embedded in any of their Krylov methods, and we have found coupling with GMRES to be a favorable pair.

Equation (1) was solved over the Stanford Bunny [21], which has been scaled to be two units tall. The original triangulation has not been modified in any way beyond this scaling. This surface has several holes and is complicated enough to stress the solvers, making it a good test case. Our chosen grid spacing was $\Delta x = 1/120$, which paired with tri-quadratic interpolation gives $N_A = 947, 964$ active nodes in the global problem. The origin was placed at the center of the bounding box containing the bunny and the right hand side $f = \phi(\pi - \phi) \sin(3\phi)(\sin \theta + \cos(10\theta))/2$ was used after extending it to be constant along the surface normals.

Table 1 shows the effects of subdomain count N_S, overlap width N_O, and Robin parameter α. For comparison, GMRES preconditioned with the standard block-Jacobi method with 64, 96, and 128 blocks requires more than 10000 iterations. The

	$N_O = 4, \alpha = 16$			$N_S = 64, \alpha = 16$			$N_S = 64, N_O = 4$		
N_S	64	96	128	N_O 4	6	8	α 16	32	64
RAS	992	1237	1533	992	747	610	992	992	992
ORAS	526	672	833	526	418	393	526	707	868

Table 1: Here the iterations to convergence of the ORAS solver are gathered for various parameters. Convergence was declared when the 2−norm of the residual was reduced by a factor of 10^6.

solvers display the expected behavior with the iteration count increasing for larger subdomain counts and decreasing with larger overlap widths. ORAS consistently requires fewer iterations than RAS, though the final sub-table shows the dependence of this performance on the appropriate choice of Robin weight α. The partitioning, the initial error, and the error in the approximate solution after 10 and 550 iterations are visible in Fig. 2 for one run of the solver.

Choosing an optimal value for α is non-trivial as it depends on the value of c, the mesh width, and the geometry. Additionally, the presence of cross points in decomposition, where more than two subdomains meet, complicate the matter. From the planar case, it is known that $\alpha \sim O\left(\Delta x^{-1/2}\right)$, but determining precise values *a priori* is limited to simple splittings [11, 12]. An upcoming work from the same authors explores this in much greater detail.
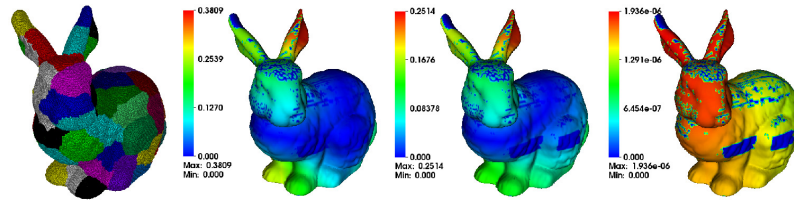

Fig. 2: The Stanford Bunny test problem solved with ORAS using $N_S = 64$, $N_O = 4$, $\alpha = 16$. The first panel shows the disjoint partitioning from METIS. The second, third, and fourth panels show the error in the solution after the 1st, 10th, and 500th iterations compared to the converged solution.

7 Conclusion

Restricted additive Schwarz and optimized restricted additive Schwarz solvers were formulated for the closest point method applied to (1). These solvers provide a solution mechanism for larger problem sizes and will allow users of the CPM to leverage large scale parallelism. Table 1 shows the dramatic reduction in iteration count when Robin TCs are used. These solvers were more completely evaluated in [17], which includes an exploration of their utility as preconditioners. The optimized conditions come at the cost of some additional complexity in the implementation, and even the standard RAS solver brings parallel capabilities to the user. Interesting extensions to this work include multiplicative methods, non-overlapping Robin schemes, two-level solvers, and inclusion of advective terms in the model equation.

References

1. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., May, D.A., McInnes, L.C., Mills, R.T., Munson, T., Rupp, K., Sanan, P., Smith, B.F., Zampini, S., Zhang, H., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.9, Argonne National Laboratory (2018)
2. Balay, S., Gropp, W.D., McInnes, L.C., Smith, B.F.: Efficient management of parallelism in object oriented numerical software libraries. In: E. Arge, A.M. Bruaset, H.P. Langtangen (eds.) Modern Software Tools in Scientific Computing, pp. 163–202. Birkhäuser Press (1997)
3. Berrut, J.P., Trefethen, L.N.: Barycentric Lagrange interpolation. SIAM Rev. **46**(3), 501–517 (2004)
4. Bertalmio, M., Cheng, L.T., Osher, S., Sapiro, G.: Variational problems and partial differential equations on implicit surfaces. J. Comput. Phys. **174**(2), 759–780 (2001)
5. Chen, Y., Macdonald, C.: The closest point method and multigrid solvers for elliptic equations on surfaces. SIAM J. Sci. Comput. **37**(1), A134–A155 (2015)
6. Chu, J., Tsai, R.: Volumetric variational principles for a class of partial differential equations defined on surfaces and curves. Res. Math. Sci. **5**(2), 19 (2018)
7. Davis, T.A.: Algorithm 832: UMFPACK v4.3—an unsymmetric-pattern multifrontal method. ACM Transactions on Mathematical Software (TOMS) **30**(2), 196–199 (2004)
8. Dolean, V., Jolivet, P., Nataf, F.: An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2015)
9. Dziuk, G., Elliott, C.: Surface finite elements for parabolic equations. Journal of Computational Mathematics **25**(4), 385–407 (2007)
10. Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. Math. Vis. Advances in Multiresolution for Geometric Modelling pp. 157–186 (2005)
11. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**(2), 699–731 (2006)
12. Gander, M.J., Kwok, F.: Best Robin parameters for optimized Schwarz methods at cross points. SIAM J. Sci. Comput. **34**(4), 1849–1879 (2012)
13. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Compt. **20**(1), 359–392 (1998)
14. Macdonald, C.B., Brandman, J., Ruuth, S.J.: Solving eigenvalue problems on curved surfaces using the closest point method. J. Comput. Phys. **230**(22), 7944–7956 (2011)
15. Macdonald, C.B., Merriman, B., Ruuth, S.J.: Simple computation of reaction diffusion processes on point clouds. Proceedings of the National Academy of Sciences **110**(23) (2013)
16. Macdonald, C.B., Ruuth, S.J.: The implicit closest point method for the numerical solution of partial differential equations on surfaces. SIAM J. Sci. Comput. **31**(6), 4330–4350 (2010)
17. May, I.: Domain decomposition solvers and preconditioners for the implicit closest point method. Master's thesis, Simon Fraser University (2018)
18. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace–Beltrami spectra as 'Shape-DNA' of surfaces and solids. Computer-Aided Design **38**(4), 342–366 (2006)
19. Ruuth, S.J., Merriman, B.: A simple embedding method for solving partial differential equations on surfaces. J. Comput. Phys. **227**(3), 1943–1961 (2008)
20. St-Cyr, A., Gander, M.J., Thomas, S.J.: Optimized multiplicative, additive, and restricted additive Schwarz preconditioning. SIAM J. Sci. Comput. **29**(6), 2402–2425 (2007)
21. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: Proceedings of the 21st annual conference on computer graphics and interactive techniques, SIGGRAPH '94, pp. 311–318. ACM (1994)