

Solving PDEs on Deconstructed Domains

S. Sellán¹, H. Y. Cheng², Y. Ma³, M. Dembowski⁴, A. Jacobson³

¹University of Oviedo, ²Massachusetts Institute of Technology, ³University of Toronto, ⁴Ryerson University

Abstract

When finding analytical solutions to Partial Differential Equations (PDEs) becomes impossible, it is useful to approximate them via a discrete mesh of the domain. Sometimes a robust triangular (2D) or tetrahedral (3D) mesh of the whole domain is a hard thing to accomplish, and in those cases we advocate for breaking up the domain in various different subdomains with nontrivial intersection and to find solutions for the equation in each of them individually. Although this approach solves one issue, it creates another, i.e. what constraints to impose on the separate solutions in a way that they converge to true solution on their union. We present a method that solves this problem for the most common second and fourth order equations in graphics.

1. Background

Partial Differential Equations (PDEs) arise in many areas of science, from physics to biology to computer graphics. In the latter, they can be used for data smoothing [SGWJ17], shape deformation [Jac13] and even computing geodesic distances [CWW13], among many others. By far the most common PDE present in computer graphics and geometry processing applications is the Poisson equation ($\Delta u = h$ for some known function h) or its particular case, the Laplace equation ($\Delta u = 0$). Our objective is usually to find the solution for these equations in a two-dimensional or three-dimensional domain Ω subjected to some Dirichlet boundary condition $u|_{\partial\Omega} = g$ for a known function g .

The preferable method to numerically approximate solutions to these equations consists in discretizing the domain in a mesh of small elements (triangles in 2D, tetrahedra in 3D) in a way that the analytical equation can be converted into a quadratic energy minimization problem where the variables are the values of the function

u at the elements' vertices, which we denote by the vector u . This problem can be solved with the standard techniques of quadratic programming to obtain the minimizer u^* , from which we can build a piece-wise linear function which can be shown to be a discrete approximation of the PDE's analytical solution u .

The accurateness of this approximation is directly related with the *goodness* of the discretization, measured mainly by its coarseness (size of the elements) and its regularity (aspect ratio of edges of each element). In practice, manually constructing quality meshes for complex three-dimensional domains is an arduous and generally undesirable process, while the automatic software available fail too often. Most of the times, the final complex domain Ω we intend to solve the PDE on (see Fig. 1) can be described via the union of very simple primitive domains Ω_i , the discretizations of which are either known or easy to find. Inspired by this, we present a method for solving PDEs on the final domain which makes use only of the meshes of the primitive shapes, thus eliminating the need for discretizing the complex domain itself.

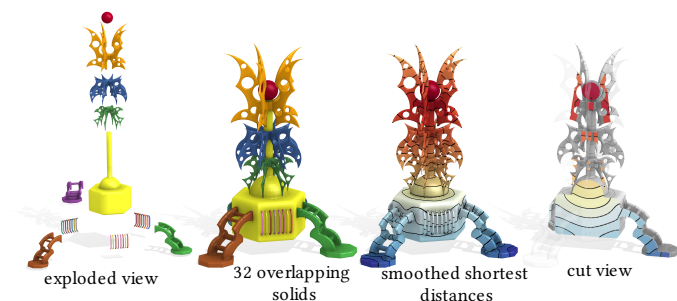


Figure 1: Our method allows for computation of geodesic distances on a shape without requiring a tetrahedral mesh of the complex domain.

2. The consistency requirement

Our first step is constructing a combined quadratic energy which functions as an analogue of the single-domain energy described above. To achieve this, we add the individual energies we would obtain from solving the PDE on each primitive and weigh them in a way that we avoid accounting for the same area twice in the zones where the primitives intersect.

Simply minimizing this combined energy subject to the original PDE's boundary condition is not enough. The problem lies precisely in those regions where the primitives intersect, where we would obtain various different solution values u^1, \dots, u^n (corresponding to the n intersecting primitives) which need not be consistent with each other. We will explore different methods of impos-

Log-log plot of error convergence on 2D annulus

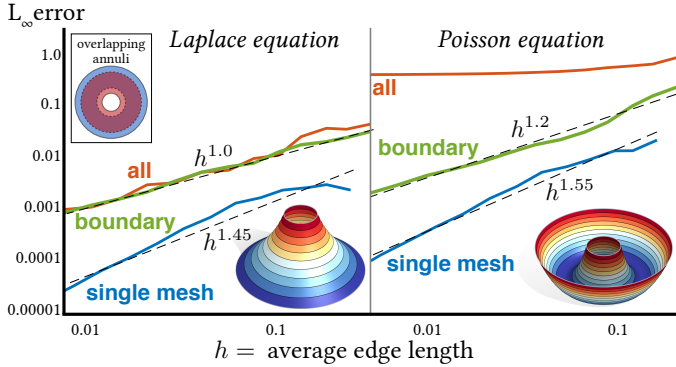


Figure 2: Convergence plots of our linear equality constraint methods for both the Poisson and Laplace equations.

ing this consistency requirement within the energy minimization in a way that the individual minimizers on each primitive do approximate the analytical solution on the final domain.

For clarity, let us restrict ourselves to the case where $n = 2$, i.e. our final domain can be written as $\Omega = \Omega_1 \cup \Omega_2$ and we wish to formulate a constraint that guarantees us that our energy minimizers u_1 and u_2 take similar values in the region $\Omega_1 \cap \Omega_2$. Since the vertices of each domain's discretization need not coincide, enforcing an *equality* between u_1 and u_2 is not as straight-forward as it may seem. In order to be able to make this comparison, we will need to construct the piece-wise linear functions $u^1(x)$ and $u^2(x)$ from the vertex values u_1 and u_2 .

Enforcing $u^1|_{\Omega \cap \Omega_2} \equiv u^2|_{\Omega_1 \cap \Omega_2}$ (equality on all points in the intersection) can easily be shown to be too harsh a constraint since it will effectively restrict our minimization to the space of functions which are linear over the intersection (see Fig. 3). This is an artificial restriction which persists even as the discretizations are refined. We will refer to this type of undesirable consequences as *locking*.

The first formulation of this consistency that we present is that which enforces equality between both functions *only at all of the discretizations' vertices*. This requirement can easily be expressed as a linear equality constraint to add on to our previous quadratic program. We show that this method also results in locking when attempting to solve the Poisson equation, although it performs surprisingly well when solving the Laplace equation (see Fig. 2). The reason for this lies in the similarity between our imposed linear constraint and the energy associated with the Laplace equation (usually known as the *cotangent laplacian* or *discrete laplacian*), which we intuitively show by a study of each's eigenmodes. Softening the requirement by exchanging the linear equality constraint for a quadratic penalty term to add to the PDE's energy we are minimizing. Naturally, the severity of the penalty will have to be governed by a positive parameter λ . If λ is small, u_1 and u_2 become increasingly disconnected; on the other hand, if λ is too large, adding this term is equivalent to the linear equality constraint seen to fail above. The optimal value of λ may depend on the equation itself and even on the characteristics of the discretization.

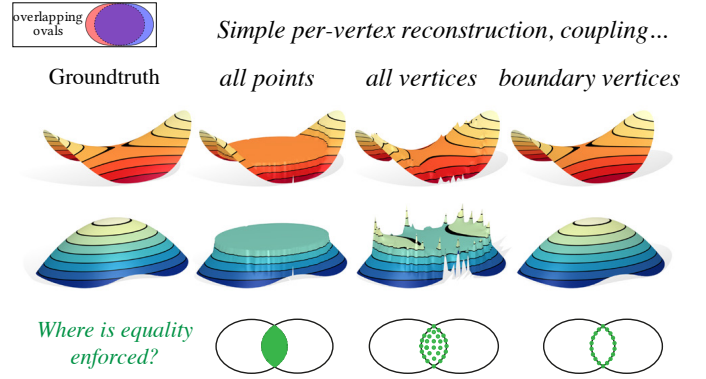


Figure 3: Behaviour of our three suggested solvers for saddle-like and paraboloid-like shapes.

3. Our method

Finally, we present our *boundary* method, which consists in enforcing equality between the solutions *only at the vertices in the intersection's boundary*, which can also be expressed as a linear equality constraint to add to our quadratic program. We show that, contrary to the previously suggested formulations, this requirement results in a convergent solver which is parameterless and consistently achieves low error. We also compare this method's convergence to the one we would obtain if we used a standard solver on a discretization of Ω (blue line in Fig. 2). We find that the expected loss in accuracy is small compared to the benefit of not requiring a mesh of the final domain, thus validating our method.

We end by showing a sample of computer graphics applications of our solver for complex three-dimensional domains constructed as the union of simpler ones, and present generalizations of our method to other PDEs such as the Wave or Heat equations, as well as higher-order ones like the Bi-Laplace $\Delta^2 u = 0$. We also exhibit possible future extensions of our boundary-only constraints to domains generated via Constructive Solid Geometry, which includes other basic set operations apart from unions, such as intersections and subtractions of basic shapes.

References

- [CWW13] CRANE K., WEISCHEDEL C., WARDETZKY M.: Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph.* (2013). 1
- [Jac13] JACOBSON A.: *Algorithms and Interfaces for Real-Time Deformation of 2D and 3D Shapes*. PhD thesis, ETH Zurich, 2013. 1
- [SGWJ17] STEIN O., GRINSPUN E., WARDETZKY M., JACOBSON A.: Natural boundary conditions for smoothing in geometry processing. *ACM Trans. Graph. to appear* (2017). 1