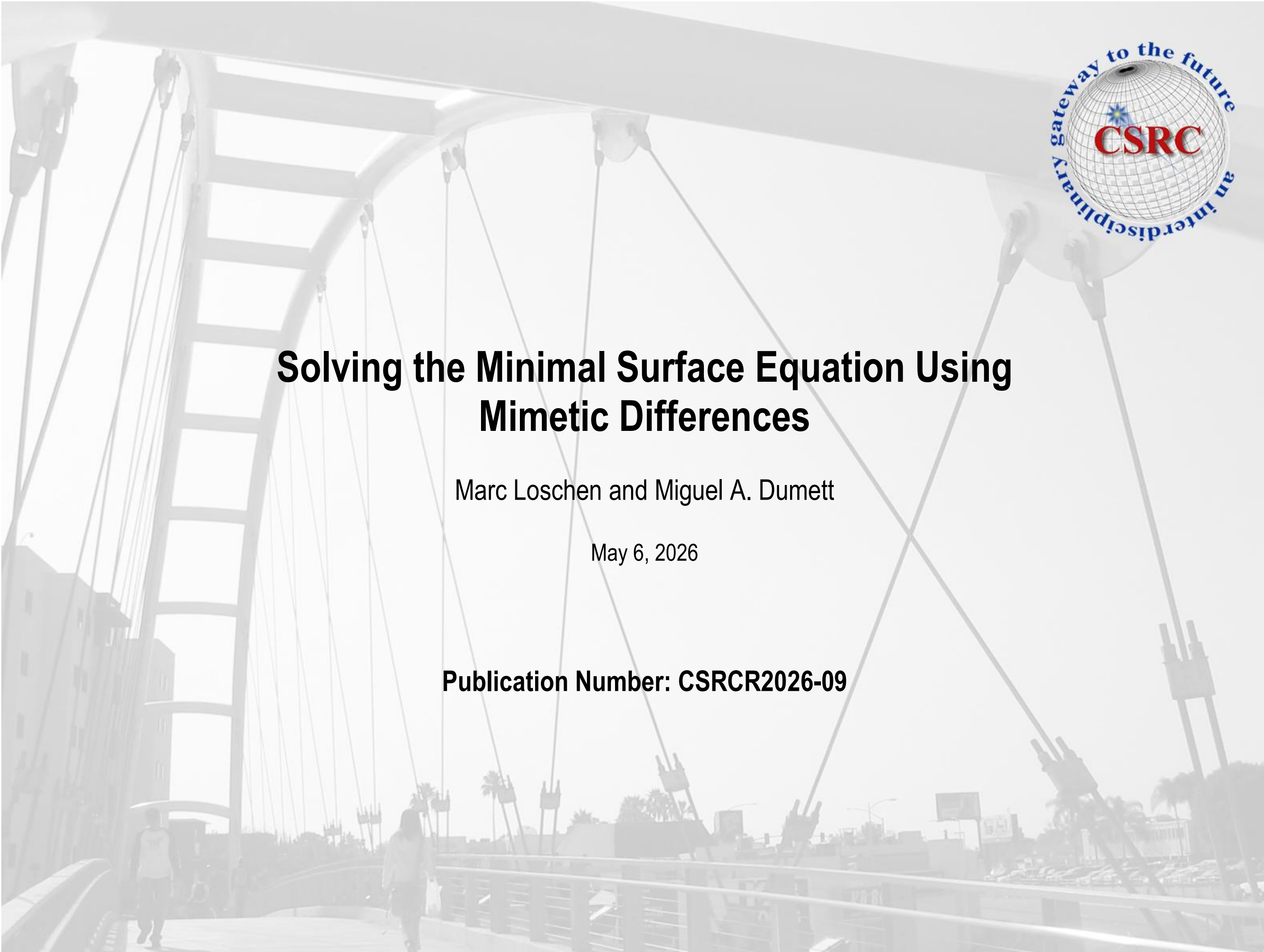


Solving the Minimal Surface Equation Using Mimetic Differences

Marc Loschen and Miguel A. Dumett

May 6, 2026

Publication Number: CSRCR2026-09

Computational Science & Engineering Faculty and Students Research Articles

Database Powered by the Computational Science Research Center Computing Group & Visualization Lab

COMPUTATIONAL SCIENCE & ENGINEERING

SAN DIEGO STATE UNIVERSITY

Computational Science Research Center
College of Sciences
5500 Campanile Drive
San Diego, CA 92182-1245
(619) 594-3430

Solving the Minimal Surface Equation Using Mimetic Differences

Marc Loschen*   Miguel A. Dumett‡

May 6, 2026

Abstract

In this report, we demonstrate how mimetic difference methods can be used to solve the minimal surface problem. After discretizing the general minimal surface equation using mimetic operators, this method is applied to two example problems (one of which is separated into two subparts). A discussion of how to handle the nonlinear term in the equation is provided, along with how to incorporate the boundary conditions into the method. Numerical results are presented, and the quality of solutions and behavior of the convergence is discussed.

1 Introduction

The minimal surface problem consists of finding the surface of minimal area that is contained within a given space curve. This can be reformulated as solving a boundary value problem, where the boundary conditions come from the aforementioned space curve. This is a quasilinear partial differential equation (PDE), which in practical settings requires the use of numerical methods to find approximate solutions. One such method for solving this type of problem involves finite differences, as demonstrated by Concus[1].

In this present study, we examine the same example problems considered in Concus' paper and construct mimetic difference methods using MOLE[2, 3] to find approximate solutions. The first example, which may be considered (a) in its entirety, or (b) in half using symmetry, shows how the boundary conditions can further increase the nonlinearity of the problem. The second example admits an exact solution, which can be used for comparing the accuracy of our method.

*Computational Science PhD Program at San Diego State University (mloschen2001@sdsu.edu).
†Editor: Jose E. Castillo.
‡Computational Science Research Center at San Diego State University (mdumett@sdsu.edu).

2 Mathematical Formulation

2.1 Motivation

We first consider Plateau's problem in nonparametric form, as stated in [1]. Let $f(x, y)$ be a single-valued function, defined on the boundary C of a simply connected region R in the xy-plane, that represents the height of a given space curve Γ above the point (x, y) on C. Let $u(x, y)$ represent the height above the point $(x, y) \in \mathbb{R}$ of the surface of minimal area through Γ. Then, we want to find a function $u(x, y)$ twice continuously differentiable in R that satisfies

$$u(x, y) = f(x, y) \text{ on } C \tag{1}$$

and minimizes the surface area

$$A = \iint_R (1 + u_x^2 + u_y^2)^{1/2}\, dxdy \tag{2}$$

Hence, this defines a problem of finding a "minimal surface" through Γ.

2.2 Differential Equation

Using the calculus of variations, we can derive from (2) the following Euler equation:

$$\nabla \cdot \left[(1 + |\nabla u|^2)^{-1/2}\, \nabla u \right] = 0. \tag{3}$$

which is the PDE we aim to solve. We note that this is a quasilinear, elliptic equaiton in two variables.

2.3 Boundary Conditions

The full PDE problem involving (3) will be fully specified by the boundary conditions, i.e. the curve Γ through which we want to find the minimal surface. In this study, we consider two sets of examples similarly experimented with in [1]. The first example, which may be broken further into parts (a) and (b), is as follows: Let (3) be defined in the rectangle $(0, 2) \times (0, 1)$ with boundaries

$$\begin{aligned} &u(0, y) = 0, \quad y \in [0, 1] \\ &u(2, y) = 0, \quad y \in [0, 1] \\ &u(x, 0) = K \sin\left(\frac{\pi x}{2}\right), \quad x \in [0, 2] \\ &u(x, 1) = 0, x \in [0, 2] \end{aligned} \tag{4}$$

where K is some real number. Then Example 1(a) is the Dirichlet problem given by (3) and (4).

Observing the symmetry about $x = 1$, we can instead consider half of the region enclosed by (4) with a modified condition along $x = 1$. Specifically, we have

$$\begin{aligned} u(0,y) &= 0, \quad y \in [0,1] \\ \frac{\partial u}{\partial x}(1,y) &= 0, \quad y \in [0,1] \\ u(x,0) &= K \sin\left(\frac{\pi x}{2}\right), \quad x \in [0,1] \\ u(x,1) &= 0, x \in [0,1] \end{aligned} \tag{5}$$

Then Example 1(b) is the mixed problem (Dirichlet and Neumann conditions) given by (3) and (5). In both cases of Example 1, we will consider the values $K = \frac{1}{2}$, 1, and 5.

In Example 2, we will examine equation (3) paired with the following boundary conditions:

$$\begin{aligned} u(0,y) &= \cosh(y), \quad y \in [0,1] \\ u(1,y) &= \left[\cosh^2(y) - 1\right]^{1/2}, \quad y \in [0,1] \\ u(x,0) &= \left[1 - x^2\right]^{1/2}, \quad x \in [0,1] \\ u(x,1) &= \left[\cosh^2(1) - x^2\right]^{1/2}, x \in [0,2] \end{aligned} \tag{6}$$

The solution of this example will give the minimal surface bounded by the curve $u(x,y) = \left[\cosh^2(y) - x^2\right]^{1/2}$. It should be noted that the equation for the curve is also the exact solution for this problem.

3 Numerical Approach

The discretization of all our examples will be facilitated by the MOLE[3] software package. This package provides, among other features, implementations of the discrete mimetic operators D (divergence), G (gradient), and L (laplacian), as well as utility functions for incorporating boundary conditions into the discretization.

3.1 Discretization Using Mimetic Operators

Starting from equation (3), we replace the divergence and gradient operators with their mimetic equivalent, and set our unknown u to the discrete variable U, which yields

$$D\left[(1 + |GU|^2)^{-1/2}\, GU\right] = 0. \tag{7}$$

To handle the squared norm of the gradient, we utilize the Hadamard product[4], denoted by the symbol $\circ$, which is defined such that for same-size matrices $A = [a_{ij}]$ and $B = [b_{ij}]$, their Hadamard product is $A \circ B := [a_{ij} b_{ij}]$ (i.e. the matrix of elementwise products of A and B). This gives

$$|GU|^2 = (GU) \circ (GU). \tag{8}$$

Denoting by $\mathbf{1}$ a vector of ones (of appropriate length), we can then rewrite our discretization as

$$D\left[\text{diag}\{\mathbf{1}+(GU)\circ(GU)\}^{-1/2}\,GU\right]=0 \tag{9}$$

where $\{\mathbf{1}+(GU)*(GU)\}^{-1/2}$ is understood to be the elementwise reciprocal square root and diag$(\cdot)$ represents a diagonal matrix whose entries are those of the input vector. Finally, regrouping terms, we arrive at

$$\left[D\left(\text{diag}\{\mathbf{1}+(GU)\circ(GU)\}^{-1/2}\right)G\right]U=0 \tag{10}$$

which is our mimetic difference equation.

3.2 Treatment of Nonlinear Problem and Boundary Conditions

Note that the mimetic discretization (10) is nonlinear in the unknown U. Thus, an iterative method is required in order to obtain our desired solution. If we set

$$A=A(U)=\text{diag}\{\mathbf{1}+(GU)\circ(GU)\}^{-1/2} \tag{11}$$

then we can rewrite (10) as

$$[DAG]U=0. \tag{12}$$

From here, we can construct a fixed-point iteration. At each step, we will use MOLE's `addScalarBC2D` function to incorporate the boundary conditions into this system of equations. The matrix $DA(U^n)G$ and right-hand side vector 0 are used as input; the resulting output is L^n and b^n, representing the updated matrix and right-hand side, respectively. Observe that L^n will depend on U^n at each step of the iteration. Then, we compute

$$U^{n+1}=(L^n)^{-1}b^n=F(U^n) \tag{13}$$

as our fixed point iteration. (Note that in practice, we do not invert the matrix L^n, but instead solve the corresponding linear problem using, e.g., Gaussian elimination.)

This iterative process is run until the difference in successive approximations reaches a provided tolerance, or until the number of iterations reaches a provided maximum threshold. As an initial guess to the fixed-point iteration, we will use the solution of obtained by solving Laplace's problem with the same boundary conditions. The mimetic difference equation for Laplace's problem is given by

$$LU=0. \tag{14}$$

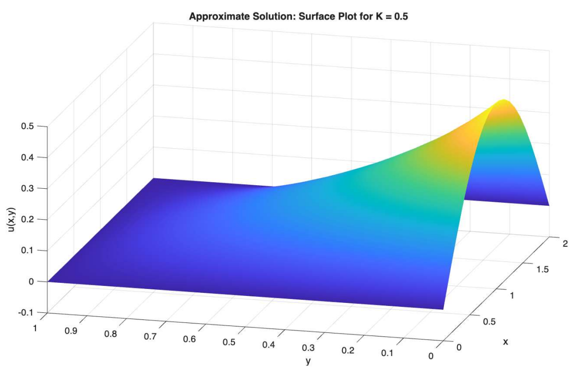
and can be readily solved using methods for linear systems. Before computing the solution, which will be our initial guess U^0, we use `addScalarBC2D` to incorporate the boundary conditions in L (left-hand side) and 0 (right-hand side).

4 Results and Discussion

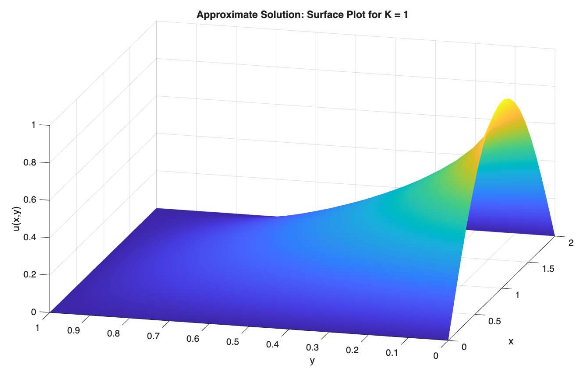
In all of the results presented below, second-order ($k = 2$) mimetic operators were used in the implementations. Additionally, the following error tolerance tol is used in all test cases:

$$tol = 1 \times 10^{-12} \tag{15}$$

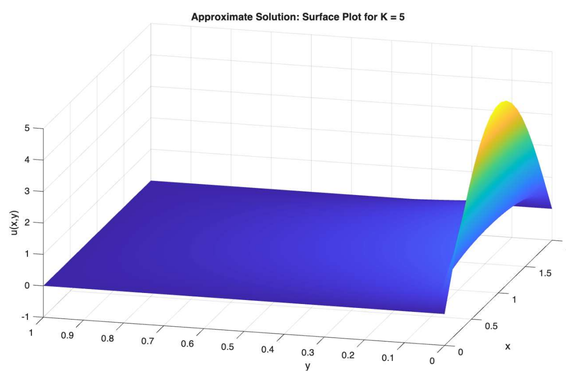
4.1 Example 1a



(a) $K = 1/2$



(b) $K = 1$



(c) $K = 5$

Figure 1: Surface plots of Example 1(a) for 3 values of K

Surface plots for the solution of Example 1(a), i.e. equations (3) and (4), are provided in Figure 1. Each subfigure corresponds to a different value of K. The calculations were performed on a grid of size $m = 20$ cells in the x direction and $n = 20$ cells in the y direction. We observe that, as K increases, the minimal surface's slope changes much faster as we approach the edge $y = 0$. As Concus notes in [1], larger values of K result in this problem becoming more nonlinear. In Table 1, we report the required number of fixed-point iterations required to meet the convergence criterion

$$\|U^n - U^{n-1}\| < tol \tag{16}$$

We indeed find that more iterations are needed to converge as K increases.

Value of K	# Iterations
$K = 1/2$	34
$K = 1$	260
$K = 5$	326

Table 1: Number of fixed-point iterations required for Example 1(a)

4.2 Example 1b

Contour maps for the solution of Example 1(b), i.e. equations (3) and (5), are provided in Figure 2. As before, each subfigure corresponds to a different value of K. The calculations were performed on a grid of size $m = 10$ cells in the x direction and $n = 10$ cells in the y direction.

In qualitatively comparing the results obtained here with those in Figure 1 of [1], we observe that similar results are obtained from both the finite difference approach and mimetic difference approach. Table 1 shows the required number of fixed-point iterations required to meet the same convergence criterion as (16). Additionally, we plot here in Figure 3 the measurement $\|U^n - U^{n-1}\|$ for each iteration, across all three values of K. As with Example 1(a), we note again that more iterations are required for convergence as K increases.

Value of K	# Iterations
$K = 1/2$	32
$K = 1$	149
$K = 5$	171

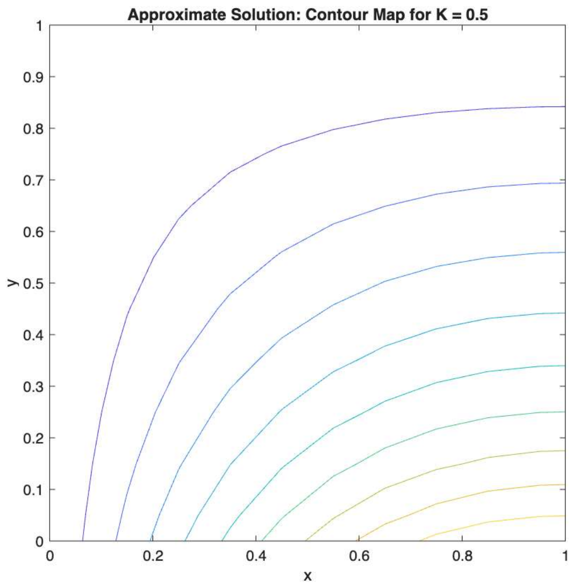
Table 2: Number of fixed-point iterations required for each value of K

4.3 Example 2

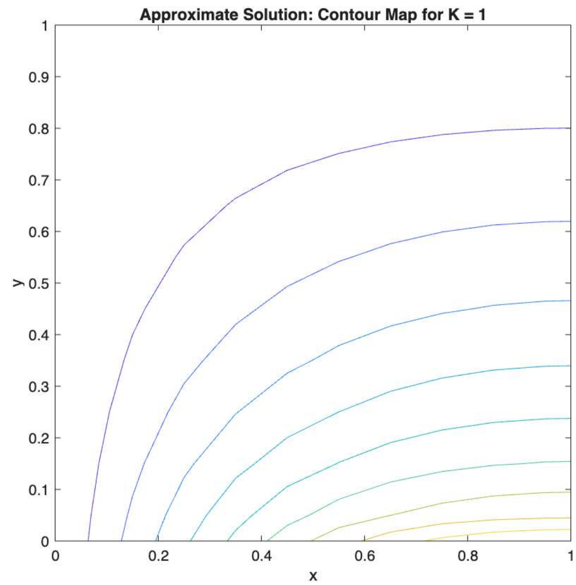
For Example 2, i.e. equations (3) and (6), contour maps of both the computed approximate solution and the analytical solution are provided in Figure 4. The approximate solution was computed on a grid of size $m = 10$ cells in the x direction and $n = 10$ cells in the y direction. We see that visually, the approximate solution is very close to the exact solution, although upon a closer inspection we find that they do not perfectly overlay on each other.

A semi-logarithmic plot the measurement $\|U^n - U^{n-1}\|$ is given in Figure 5. In total, 83 iterations were required to reach the convergence criteria with this setup. Additionally, we compute the absolute error between the approximate solution and the exact solution as

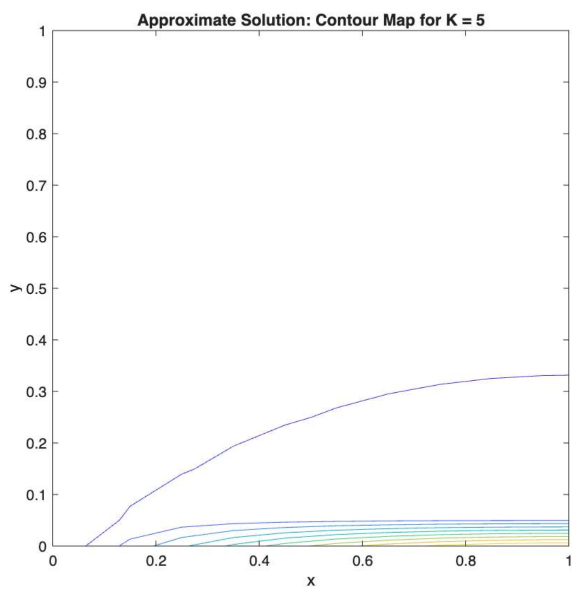
$$\|U^n - U_{exact}\| \approx 0.12161 \tag{17}$$



(a) $K = 1/2$



(b) $K = 1$



(c) $K = 5$

Figure 2: Contour maps of Example 1(b) for 3 values of K

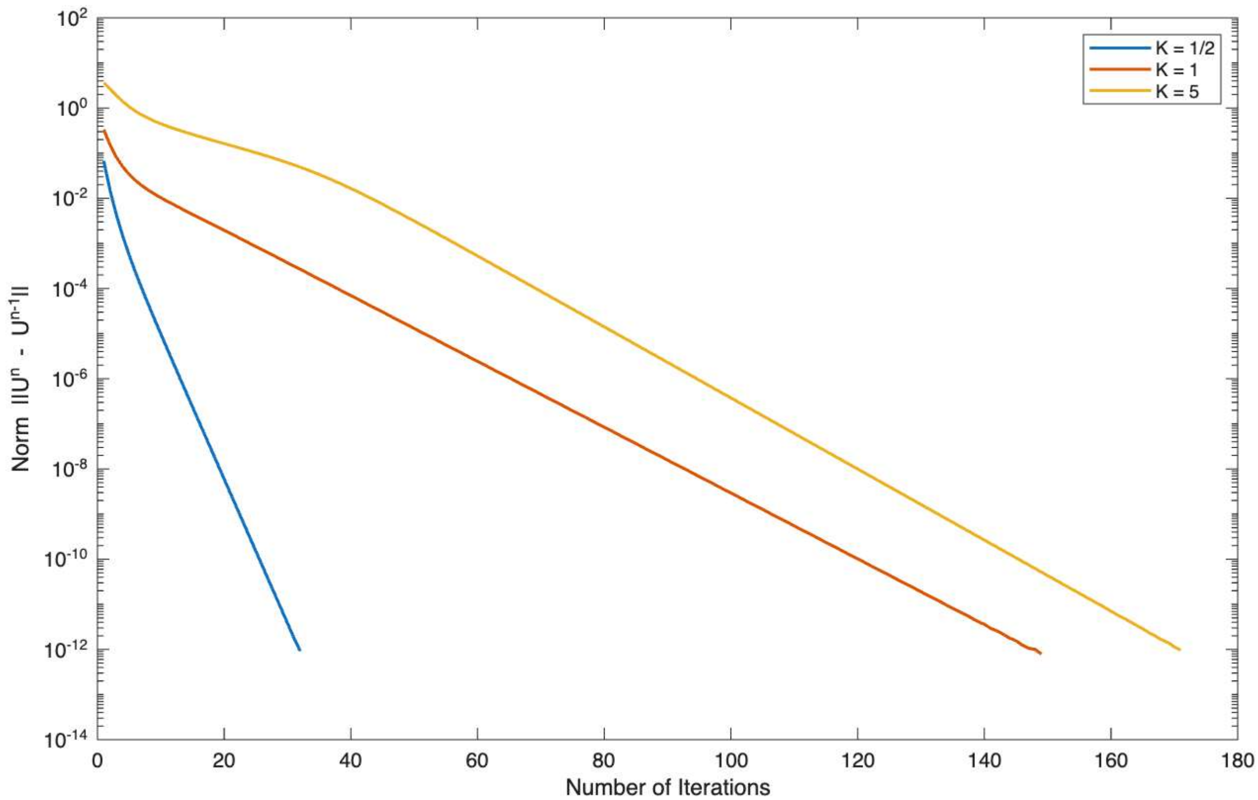


Figure 3: Semi-logarithmic plot of $\|U^n - U^{n-1}\|$ per iteration, for each K

5 Conclusion

As shown above, the mimetic difference method provides, at minimum, a comparable alternative to traditional numerical PDE solvers. Beyond the discretization using mimetic operators, care must be taken when handling the nonlinear term in the equation. While fixed-point iterations were used for the present study, other iterative techniques could certainly be used instead. Additionally, second-order mimetic operators were only considered in the experiments in this report, but MOLE provides implementations for fourth-, sixth-, and eighth-order operators that could also be utilized. The nonlinear quality of the minimal surface problem provides an excellent test case for exploring the efficacy of MOLE and of mimetic difference methods.

References

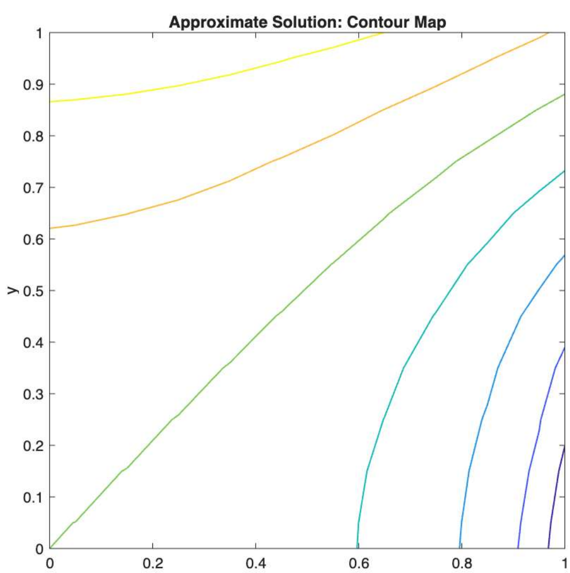
[1] Paul Concus. Numerical Solution of the Minimal Surface Equation. *Math. Comp.*, 21:340–350, 1967.

[2] J. Corbino and J. E. Castillo. High-order mimetic finite-difference operators satisfying the extended gauss divergence theorem. *Journal of Computational and Applied Mathematics*,

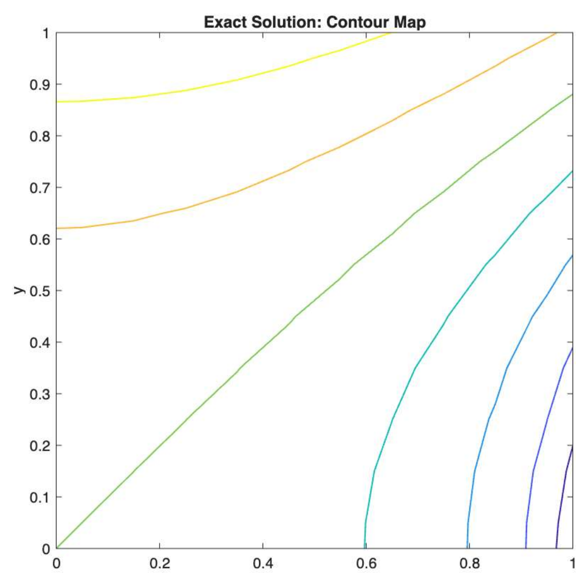
364, 01 2020.

[3] Johnny Corbino, Miguel A. Dumett, and Jose E. Castillo. MOLE: Mimetic Operators Library Enhanced. *Journal of Open Source Software*, 9(99):6288, 2024.

[4] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, repr. with corrections edition, 1990.



(a) Approximate solution



(b) Exact solution

Figure 4: Comparison of approximate and exact solutions for Example 2

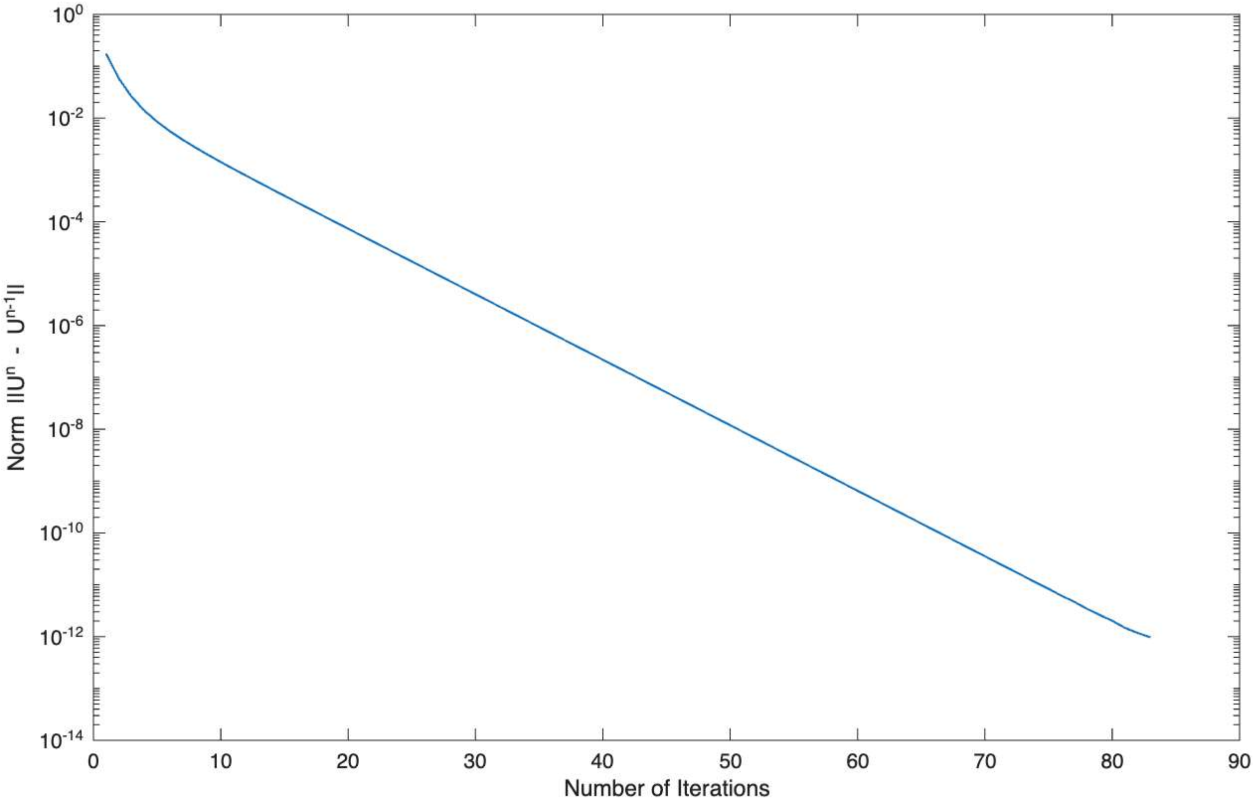


Figure 5: Semi-logarithmic plot of $\|U^n - U^{n-1}\|$ per iteration, for Example 2