

A Continuous Finite Element Method with Homotopy Vanishing Viscosity for Solving the Static Eikonal Equation

Yong Yang^{1,2}, Wenrui Hao^{3,*} and Yong-Tao Zhang⁴

¹ College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China.

² Key Laboratory of Mathematical Modelling and High Performance Computing of Air Vehicles (NUAA), MIIT, Nanjing 211106, China.

³ Department of Mathematics, Pennsylvania State University, University Park, Pennsylvania, USA.

⁴ Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA.

Received 6 August 2021; Accepted (in revised version) 24 November 2021

Abstract. We develop a second-order continuous finite element method for solving the static Eikonal equation. It is based on the vanishing viscosity approach with a homotopy method for solving the discretized nonlinear system. More specifically, the homotopy method is utilized to decrease the viscosity coefficient gradually, while Newton's method is applied to compute the solution for each viscosity coefficient. Newton's method alone converges for just big enough viscosity coefficients on very coarse grids and for simple 1D examples, but the proposed method is much more robust and guarantees the convergence of the nonlinear solver for all viscosity coefficients and for all examples over all grids. Numerical experiments from 1D to 3D are presented to confirm the second-order convergence and the effectiveness of the proposed method on both structured or unstructured meshes.

AMS subject classifications: 65N06, 65N12, 65N15

Key words: Eikonal equation, finite element method, homotopy method.

1 Introduction

We consider the static Eikonal equation

$$\begin{cases} |\nabla u(\mathbf{x})| = f(\mathbf{x}), & \mathbf{x} \in \Omega \setminus \Gamma, \\ u(\mathbf{x}) = g(\mathbf{x}), & \mathbf{x} \in \Gamma, \end{cases} \quad (1.1)$$

*Corresponding author. *Email addresses:* yongyang@nuaa.edu.cn (Y. Yang), wxh64@psu.edu (W. Hao), yzhang10@nd.edu (Y.-T. Zhang)

over the domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, where $f \geq 0$ and g are the given functions, Γ is the “boundary” which is a subset of $\bar{\Omega}$. Moreover, g is assumed to satisfy the compatible condition to guarantee the existence of the physical or viscosity solution of (1.1), see e.g. [29, 31].

Eikonal equation (1.1) has many applications such as computational fluid dynamics, optics, wave propagation, material science, differential geometry (geodesics), image processing and computer graphics [1, 5, 9, 11, 40, 41, 46]. When solving two-phases flow problems [12], the solution u of Eq. (1.1) with $f \equiv 1$ and $g \equiv 0$ is the distance function with Γ as its 0-level set, which makes u be able to track the interface between two phases. In the so-called Shape-from-Shading problem [34], the solution of (1.1) in 2D reconstructs the surface $z = u(x, y)$ based on $f(x, y)$ which is related to the brightness $I(x, y)$ of the surface under a remote vertical light source as $f = \sqrt{1 - I(x, y)^2} / I(x, y)$. In the seismic ray method where the separation of variable is utilized to locate the high-frequency seismic body wave in the media [37], the solution u of (1.1) describes the travel time with $f(x) = 1/v(x)$ as the slowness associated to the velocity $v(x)$ in the media [6].

Mathematically speaking, Eikonal equation (1.1) is a typical example of Hamilton-Jacobi equation $H(x, u, \nabla u) = f$ by taking $H(x, u, \nabla u) = |\nabla u|$. Thus the difficulties of solving Hamilton-Jacobi equation such as the nonlinearity and non-uniqueness apply to the Eikonal equation (1.1). Therefore, the generalized solution or *viscosity solution* has to be sought for solving the Eikonal equation (1.1) [8–11, 29, 31]. The concept of viscosity solution is reasonable and satisfactory since: 1) if u is a smooth solution of (1.1), then it is a viscosity solution; 2) if the viscosity solution u is differentiable at some point, then it satisfies the equality (1.1); 3) the viscosity solution is unique given appropriate boundary condition; 4) the solution obtained by the vanishing viscosity method is the viscosity solution.

There are many numerical methods to solve the Eikonal equation and to compute the viscosity solution. The characteristic method has been developed to solve (1.1) by solving a first-order ODE [25]. However, the method is hard to find the global solution and has to deal with the coupling of the spatial variables and the phase space variables. The level set formulation is utilized to introduce a time variable to solve the static Eikonal equation [28, 32, 33]. A monotone finite difference method and a vanishing viscosity method are used to solve the time-dependent Cauchy problem of the Hamilton-Jacobi equation with the form of $H(\nabla u)$ in [11], where the convergence rate is obtained explicitly. The fast marching method and fast sweeping method coupled with finite difference discretization are developed to solve the Eikonal equation. The fast marching method is based on entropy-satisfying upwind schemes and fast sorting techniques where the solution is updated by sequentially following the causality [41, 44]. The fast sweeping method does not need heap-sort and the updating follows the causality along with the characteristics in a parallel way [26, 27, 46–48]. There are several approaches based on the finite element method for solving Eikonal equation. For example, a continuous finite element method based on minimization of the residual in L^1 norm is proposed to solve stationary Hamilton–Jacobi equations [18–20]; a discontinuous Galerkin method based

on the fast sweeping strategy has been developed to solve (1.1) with a second-order convergence [30]; a continuous piecewise linear finite element method [5] adds a biharmonic regularization in the objective function and Eq. (1.1) is treated as the constraints by the penalty method; a piecewise linear continuous finite element method [7] relies on the Schrödinger transformation and only zero boundary condition over $\Gamma = \partial\Omega$ is considered; based on the local variation principal or the local Hopf-Lax formula a linear finite element method is studied in [4]; a time marching method using the piecewise linear finite element method is proposed to solve the static Eikonal equation in [40, chp. 9] and [3].

In this paper, we use the vanishing viscosity method [29,31] to compute the viscosity solution of (1.1). More specifically, it is to seek the limit of vanishing viscosity solutions u_n of

$$\begin{cases} |\nabla u_n(\mathbf{x})| = f(\mathbf{x}) + v_n \Delta u_n, & \text{in } \Omega \setminus \Gamma, \\ u_n(\mathbf{x}) = g(\mathbf{x}), & \text{on } \Gamma, \end{cases} \quad (1.2)$$

with $v_n \rightarrow 0$ as $n \rightarrow \infty$. Based on this setup, we develop a piecewise linear continuous finite element method, which is easy to implement, can treat problems on irregular domains with various boundary conditions, is flexible to use unstructured meshes in all dimensional spaces, and achieves the second order convergence numerically. The method combines the homotopy method [21,24] and Newton's method to solve the nonlinear discretized system in an effective way, since Newton's method alone does not converge for arbitrary grids. The paper is organized as follows. In Section 2, we explain the motivation of our work by a special 1D problem with $f \equiv 1$, $g = 0$, $\Gamma = \partial\Omega$. In Section 3, the detailed algorithm is presented. In Section 4, several numerical examples in 1D, 2D, and 3D are devoted to illustrate the effectiveness and feasibility of the proposed algorithm. In this paper, we use the standard notations on Sobolev space $H_0^1(\Omega), H^1(\Omega), L^p(\Omega), W^{1,p}(\Omega), C^{0,\gamma}(\Omega)$ and their corresponding norms such as $\|\cdot\|_{H^1}, \|\cdot\|_{L^p}, \|\cdot\|_{W^{1,p}}$ for $p \in [0, \infty]$ (see e.g. [2, 16] for more details).

2 Motivation of the algorithm

In order to design a numerical algorithm with second order convergence, we first examine the convergence rate of the vanishing viscosity method at the continuous level. Assume $\Gamma = \partial\Omega$. The vanishing viscosity method generates a sequence $\{u_n\}_{n=1}^\infty$, where u_n is the viscosity solution or the generalized solution of

$$\begin{cases} |\nabla u_n(\mathbf{x})| = f(\mathbf{x}) + v_n \Delta u_n, & \text{in } \Omega, \\ u_n(\mathbf{x}) = g(\mathbf{x}), & \text{on } \partial\Omega, \end{cases} \quad (2.1)$$

where $\{v_n\}_{n=1}^\infty$ is a positive sequence convergent to 0. For fixed n , the existence and uniqueness of u_n is shown in [29, Theorem 3.2]. Based on some priori estimates and

a compactness argument, the convergence of $\{u_n\}_{n=1}^\infty$, up to a subsequence, is shown in [29, Theorem 3.2]. Its limit u_* is the viscosity solution of (1.1) [10, 29, 31]. The optimal convergence rate is $\|u_n - u_*\|_{L^\infty} \leq c\sqrt{v_n}$ as shown in [31, Proposition 6.3]. It seems that a second-order numerical method requires $v_n = h^4$ where h is the mesh size. The question is whether we have a better estimate for (1.1) if the solution is smooth enough. To answer this question, we examine a 1D Eikonal equation (1.1) as

$$\begin{cases} |u'(x)| = 1, & x \in \Omega := (0, a), \\ u(0) = u(a) = 0. \end{cases} \quad (2.2)$$

The vanishing viscosity approximation

$$\begin{cases} |u'_n(x)| = 1 + v_n u''_n(x), & x \in \Omega := (0, a), \\ u_n(0) = u_n(a) = 0 \end{cases} \quad (2.3)$$

has an explicit solution as follows

$$u_n(x) = \begin{cases} x + v_n(e^{\frac{1}{v_n}(-\frac{a}{2})} - e^{\frac{1}{v_n}(x-\frac{a}{2})}), & x \in [0, \frac{a}{2}], \\ -x + a + v_n(e^{\frac{1}{v_n}(-\frac{a}{2})} - e^{\frac{1}{v_n}(-x+\frac{a}{2})}), & x \in [\frac{a}{2}, a]. \end{cases} \quad (2.4)$$

Thus the rate of convergence can be estimated:

- u_n converges to

$$u_* := \begin{cases} x, & x \in [0, \frac{a}{2}], \\ -x + a, & x \in (\frac{a}{2}, a] \end{cases}$$

pointwisely;

- $\|u'_h - (u_*)'\|_{L^p}^p \leq \min(2\frac{v_n}{p}, a)$ for $p \in [1, \infty)$;
- $\|u_n - u_*\|_{L^p}^p \leq av_n^p$ for $p \in [1, \infty)$.

Furthermore, the Sobolev embedding theorem (see e.g. [16, page 270] or [2, page 108]) implies

$$\|u_n - u_*\|_{C^{0,\gamma}} \leq c\|u_n - u_*\|_{W^{1,p}} \rightarrow 0 \quad (2.5)$$

with $\gamma = 1 - 1/p$ for $p \in (1, \infty)$.

Inspired by this example, it is expected that, if $v_h = ch^2$, the rate of the vanishing viscosity sequence u_h to the limit u_* can reach

- $\mathcal{O}(h)$ in H^1 seminorm;
- $\mathcal{O}(h^2)$ in L^2 norm;

- $\mathcal{O}(h^2)$ in L^1 norm;
- $\mathcal{O}(h^2)$ in L^∞ norm,

as $h \rightarrow 0$. Here, the subscript \cdot_n is replaced by the subscript \cdot_h to highlight the dependence on the mesh size h . After choosing the viscosity coefficient $\nu_h = ch^2$, an appropriate numerical discretization has to be designed such that no more error pollutes the convergence rate.

3 Finite element discretization

Our strategy is to use finite element method to solve the vanishing viscosity problem (1.2) with $\nu_h = ch^2$. Since the piecewise linear polynomial satisfies the approximation estimate $\mathcal{O}(h)$ in H^1 seminorm and $\mathcal{O}(h^2)$ in L^2 norm for $H^2(\Omega)$ functions [14, (11.17)], which is consistent with the convergence rate of $\nu_h = ch^2$, it can be utilized in the algorithm to obtain the goal.

Let $\{\mathcal{T}_h\}$ be a mesh family which are affine, conforming (no hanging nodes), and shape-regular in the sense of Ciarlet [14]. Each mesh \mathcal{T}_h is a set of cells K , which are intervals in one dimensional space, or triangles in two dimensional space, or tetrahedral in three dimensional space. The union of all cells in \mathcal{T}_h is the whole domain $\bar{\Omega}$. The domain Ω is assumed to be a polygon such that the boundary $\partial\Omega$ is partitioned exactly by several mesh edges. Moreover, it is assumed that Γ is partitioned by the mesh explicitly. For example, if Γ is part of the boundary $\partial\Omega$, we assume that Γ is partitioned by several mesh edges; if Γ is only a point, we assume it is a node of the mesh. The general curved Γ is treated later in Section 3.3 with the help of the viscosity solution u_\star of (1.1).

Define

$$V_h := \{v \in C(\Omega) : v|_K \in \mathbb{P}_1(K), \forall K \in \mathcal{T}_h, v|_\Gamma = g\}, \quad (3.1)$$

where $\mathbb{P}_k(K)$ means the piecewise polynomial over K with the maximum order less or equal to k , and the parameter h , used as the subscript, denotes the minimal size of edges in the mesh \mathcal{T}_h . Let the Lagrange basis be $\{\phi_1, \dots, \phi_N\}$ associated with the Lagrange nodes of the mesh $\{\mathbf{a}_1, \dots, \mathbf{a}_N\}$. The approximation space V_h is a finite dimensional space and is spanned by $\{\phi_j, \mathbf{a}_j \notin \Gamma\}$. If $w_h \in V_h$, it has the explicit expression $w_h(\mathbf{x}) = \sum_{j=1, \dots, N, \mathbf{a}_j \notin \Gamma} W_j \phi_j(\mathbf{x}) + \sum_{j=1, \dots, N, \mathbf{a}_j \in \Gamma} g(\mathbf{a}_j) \phi_j(\mathbf{x})$, where W_j is the degrees of freedom. See e.g. [13, 14] for the approximation property of V_h and other related details.

The piecewise linear continuous finite element method to solve (1.2) is to find $u_h \in V_h$ such that

$$\int_{\Omega} [(|\nabla u_h(\mathbf{x})| - f)v_h + \nu_h \nabla u_h \cdot \nabla v_h] dx - \int_{\partial\Omega \setminus \Gamma} \nu_h \partial_n u_h v_h ds = 0 \quad (3.2)$$

for all $v_h \in V_h^0 := \{v \in C(\Omega) : v|_K \in \mathbb{P}_1, \forall K \in \mathcal{T}_h, v|_\Gamma = 0\}$. Here integration by parts for the viscous term yields the third term of (3.2) on $\partial\Omega \setminus \Gamma$ since the Dirichlet boundary condition is enforced explicitly in V_h .

3.1 Newton's method

Since (3.2) is a nonlinear system, Newton's method is devoted to solve it. Given the k -th iteration solution $u_h^k \in V_h$, which is an approximation to the solution of (3.2) or u_* , Taylor expansion implies

$$|\nabla u_h| - |\nabla u_h^k| \approx \frac{\nabla u_h^k}{|\nabla u_h^k|} \cdot \nabla (u_h - u_h^k)$$

by neglecting the nonlinear terms. Newton's method is to solve

$$\int_{\Omega} \left[\frac{\nabla u_h^k}{|\nabla u_h^k|} \cdot \nabla u_h^{k+1} v_h - f v_h + v_h \nabla u_h^{k+1} \cdot \nabla v_h \right] dx - \int_{\partial\Omega \setminus \Gamma} v_h \partial_n u_h^k v_h ds = 0 \quad (3.3)$$

for any $v_h \in V_h^0$. The stopping rule of the iteration over k in (3.3) is to check the L^∞ difference between two successive approximations. Given a tolerance TOL , if $\|u_h^{k+1} - u_h^k\|_{L^\infty(\Omega)} < TOL$, then the iteration stops, and u_h^{k+1} is taken as the accepted approximation.

The boundary integral on $\partial\Omega \setminus \Gamma$ in (3.3) is treated explicitly by using u_h^k . In contrast, the fast sweeping method [46] uses the linear extrapolation to obtain the value there by extending the interior value, see Remark 8 of [46]. The linear combination may be used to treat this term as $\int_{\Omega \setminus \Gamma} v_h (\alpha \partial_n u_h^{k+1} + (1-\alpha) \partial_n u_h^k) v_h ds$ for some $\alpha \in [0, 1]$.

The algorithm (3.3) works for the unstructured mesh without modification, which is a remarkable property for problems on the irregular polygon domain. In contrast, it is not trivial to extend the fast sweeping method from the rectangular mesh to unstructured triangulated meshes because there is no obvious way to specify an order for nodes of the unstructured mesh, e.g., [36].

3.2 Homotopy strategy

Solving (3.3) by Newton's method is challenging especially when the viscosity coefficient $\nu_h = ch^2$ becomes small. Moreover, Eq. (3.3) becomes convection-dominant and has to be stabilized [15, chp. 61]. Therefore we need a good initial guess u_h^0 close to the solution for Newton's method. Due to the existence and uniqueness of the viscosity solution corresponding to ν_h [29, Theorem 3.2], Newton's method should converge to the viscosity solution. In order to choose a good initial guess u_h^0 for $\nu_h = ch^2$, we employ the homotopy method by varying ν_h from ch to ch^2 . The homotopy method, as a global method, has been widely used to compute solutions of nonlinear systems such as nonlinear PDEs [21, 22, 42, 43] and applications in biology and physics [23]. More specifically, we solve (3.3) for $\nu_h^1 := ch$ first. After Newton's method converges, its solution is used as the initial guess for $\nu_h^2 = \nu_h^1/2$. In an iterative way, we solve (3.3) until $\nu_h^k \leq ch^2$. The improvement of this strategy can be observed in Example-4 of the next section, where the algorithm converges to the tolerance over all grids, while Newton's method does not converge for finer meshes.

Algorithm 1: The detailed algorithm

Input: Problem dependent parameter c , Stopping tolerance Tol , Maximum iteration of Newton's method N

Output: The numerical solution u_h^0

Initialize $v_h = ch$ and $u_h^0 = 0$

while $v_h > ch^2$ **do**

for $k=0:N$ **do**

 Assemble the matrix A of (3.3) using v_h and u_h^k

 Modify A to enforce the given values for all degree of freedoms in Γ

 Assemble the right hand side term b of (3.3) using v_h and u_h^k

 Solve $AU^{k+1} = b$ to get $u_h^{k+1} = \sum_{j=1 \dots N, a_j \notin \Gamma} U_j^{k+1} \phi_j(\mathbf{x}) + \sum_{j=1 \dots N, a_j \in \Gamma} g(\mathbf{a}_j) \phi_j(\mathbf{x})$

if $\|u_h^{k+1} - u_h^k\|_\infty < TOL$ **then**

└ break

$u_h^k \leftarrow u_h^{k+1}$

$u_h^0 \leftarrow u_h^k$

if $v_h/2 < ch^2$ **then**

$v_h \leftarrow ch^2$

else

$v_h \leftarrow v_h/2$

The whole algorithm for solving the static Eikonal equation is obtained, as summarized in Algorithm 1.

3.3 Extension for curved Γ

When Γ is a curve, it cannot be approximated by some edges of the mesh. The above algorithm still works with the help of u_* . There are two methods to enforce the "boundary" condition $u = g$ on Γ :

- 1) The first method is to assign the value of u_* at all degrees of freedom with a distance less than αh to Γ , say, $\alpha = 1$. Then almost two layers of degrees of freedom are specified with the true value u_* .
- 2) The second method is to assign the value of u_* at all degrees of freedom with distance to Γ less than a constant, say, 0.1. As the mesh become refined, more and more layers of degrees of freedom are specified with the true value u_* .

These two strategies are illustrated in Fig. 1. The choice of which method to use comes down to the types of the singularity of the problem (1.1) on Γ . If the singularity is weak, for example, when two characteristic lines are generated at points of Γ , either choice works fine. Otherwise, if the singularity is strong, then maybe only the second choice

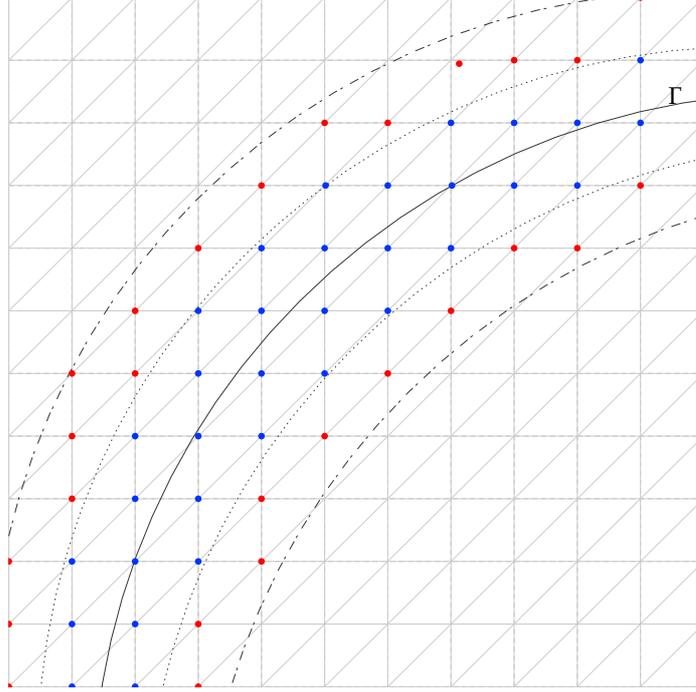


Figure 1: Two strategies to enforce the “boundary condition” on Γ : the first strategy is to specify all degrees of freedom in blue color since they are in the between of two dot lines which have distance αh to Γ ; the second strategy is to specify all degrees of freedom in blue color and red color since they are in the between of two dash-dot lines which have distance 0.1 to Γ .

gives the desired convergence rate. For example, the second strategy obtains the full convergence in Example-8 in Section 4.8. When it is difficult to obtain the exact values u_* to enforce these degrees of freedom close to Γ , a approximation of u_* by some local methods such as the characteristic method can be applied first.

4 Numerical experiments

In this section, several numerical examples are presented to study the second order convergence of the algorithm, and its effectiveness. The mesh used in the computation is consisted of intervals, triangles, or tetrahedrals. For convergence test, the meshes are equidistributed intervals, or structured triangle meshes, and the number of cells in each spatial dimension is $N = 32, 64, 128, 256, 512$. The structured mesh is established by the in-house code, while the unstructured triangular or tetrahedral mesh is generated by Gmesh [17] or Netgen/NGSolve [39]. The numerical errors considered in this section are defined as

$$e_1(\tilde{\Omega}) := \left(\frac{1}{|\tilde{\Omega}|} \int_{\tilde{\Omega}} |u_h - u_*|^2 dx \right)^{1/2}, \quad e_2(\tilde{\Omega}) := \left(\frac{1}{|\tilde{\Omega}|} \int_{\tilde{\Omega}} |\nabla u_h - \nabla u_*|^2 dx \right)^{1/2},$$

$$e_3(\tilde{\Omega}) := \frac{1}{|\tilde{\Omega}|} \int_{\tilde{\Omega}} |u_h - u_*| dx, \quad e_4(\tilde{\Omega}) := \sup_{x \in \tilde{\Omega}} |u_h - u_*|,$$

where $\tilde{\Omega} \subset \Omega$ is chosen to avoid the singular part of u_* in order to study the convergence rate of the method. We choose $\tilde{\Omega} = \Omega$ if u_* has no singularity. Here $e_3(\tilde{\Omega})$ and $e_4(\tilde{\Omega})$ are the standard errors in $L^1(\tilde{\Omega})$ and $L^\infty(\tilde{\Omega})$, while $e_1(\tilde{\Omega}) = \|u_h - u_*\|_{L^2(\tilde{\Omega})} / |\tilde{\Omega}|^{1/2}$, and $e_2(\tilde{\Omega}) = |u_h - u_*|_{H^1(\tilde{\Omega})} / |\tilde{\Omega}|^{1/2}$.

4.1 Example-1

We choose $g=0$, $f = \pi|\cos(\pi x)|$, $\Omega = (0,1)$ and $\Gamma = \partial\Omega$ in the static Eikonal equation (1.1). Its solution $u_*(x) = \sin(\pi x)$ is a smooth function. The numerical errors are shown in Table 1 which indicates the second order convergence in $L^2(\Omega), L^1(\Omega), L^\infty(\Omega)$ and the first order in $H^1(\Omega)$ seminorm. It is consistent with the analysis in Section 1. Although the solution at $x=0.5$ is singular since $f(0.5) = 0$, the convergence order is not effected since u_* is smooth and two characteristic lines merge together from left and right sides of the point $x=0.5$. A typical numerical solution is shown in Fig. 2.

Table 1: Numerical errors and convergence orders of Example-1 with $\nu_h = h^2$.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
3.13e-02	1.43e-03	-	6.82e-02	-	1.05e-03	-	3.42e-03	-
1.56e-02	3.56e-04	2.01	3.40e-02	1.01	2.63e-04	2.01	8.66e-04	1.99
7.81e-03	8.89e-05	2.00	1.69e-02	1.00	6.55e-05	2.00	2.17e-04	1.99
3.91e-03	2.22e-05	2.00	8.48e-03	1.00	1.64e-05	2.00	5.45e-05	2.00
1.95e-03	5.55e-06	2.00	4.24e-03	1.00	4.09e-06	2.00	1.36e-05	2.00

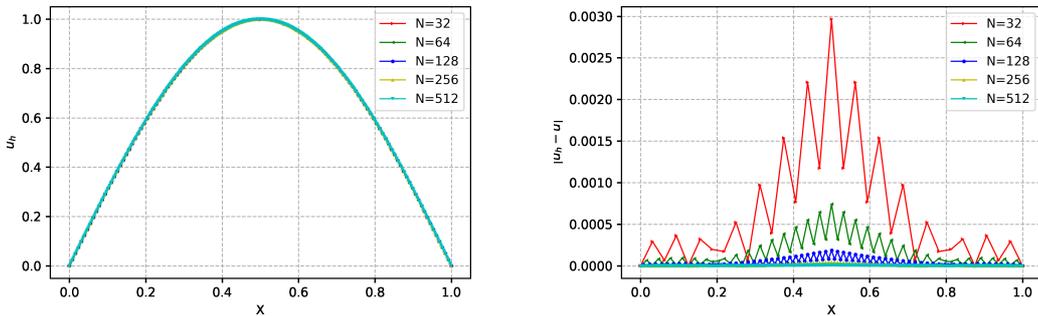


Figure 2: Numerical solutions u_h (left) and $|u_h - u_*|$ (right) of Example-1.

4.2 Example-2

Choosing $f = 1$, $g = 0$, $\Omega = (0,1)$, and $\Gamma = \partial\Omega$, the viscosity solution of (1.1) becomes

$$u_*(x) = \min\{x, 1-x\}. \quad (4.1)$$

It is the distance function to the boundary Γ since $g \equiv 0$. The numerical solutions on different grid points are shown in Fig. 3 while the convergence rate is shown in Table 2 to demonstrate $\mathcal{O}(h^2)$ convergence in $L^2(\Omega), L^1(\Omega), L^\infty(\Omega)$ norm and $\mathcal{O}(h^1)$ in $H^1(\Omega)$ seminorm.

Table 2: Numerical errors and convergence orders of Example-2 with $v = h^2$.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
3.13e-02	3.10e-04	-	3.09e-02	-	2.39e-04	-	8.80e-04	-
1.56e-02	7.73e-05	2.00	1.55e-02	1.00	5.97e-05	2.00	2.31e-04	1.93
7.81e-03	1.93e-05	2.00	7.74e-03	1.00	1.49e-05	2.00	5.93e-05	1.96
3.91e-03	4.83e-06	2.00	3.87e-03	1.00	3.72e-06	2.00	1.50e-05	1.98
1.95e-03	1.21e-06	2.00	1.94e-03	1.00	9.31e-07	2.00	3.78e-06	1.99

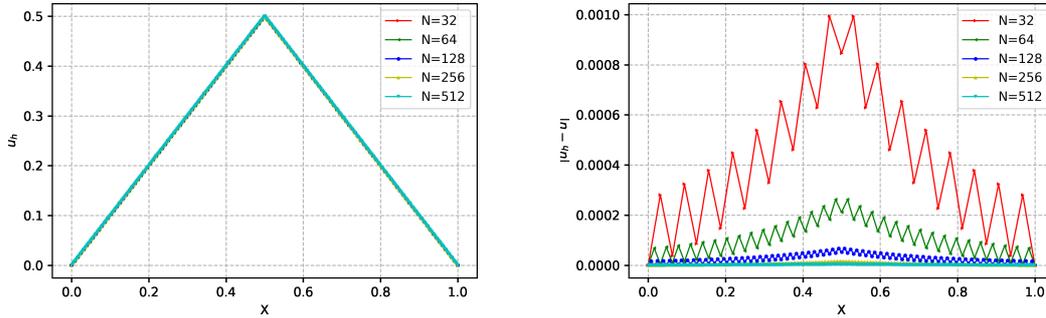


Figure 3: Numerical solutions u_h (left) and $|u_h - u_*|$ (right) of Example-2.

4.3 Example-3

We choose $f = 1$, $g = 0$, $\Omega = (0,1)$, and $\Gamma = \{0.5\}$. The point $x = 0.5$ is a singular point where two characteristic lines are generated and spread out. The outflow boundary condition on $\partial\Omega$ is used since the characteristic lines exit and leave the domain Ω there.

When the number of cells in the mesh is $N = 32, 64, 128, 256, 512$, the singular point $x = 0.5$ is a vertex of the mesh, its value is specified as 0 since it is in Γ . As shown in Table 3, the numerical solution reproduces the true solution $u_* = |x - 0.5|$ exactly since the

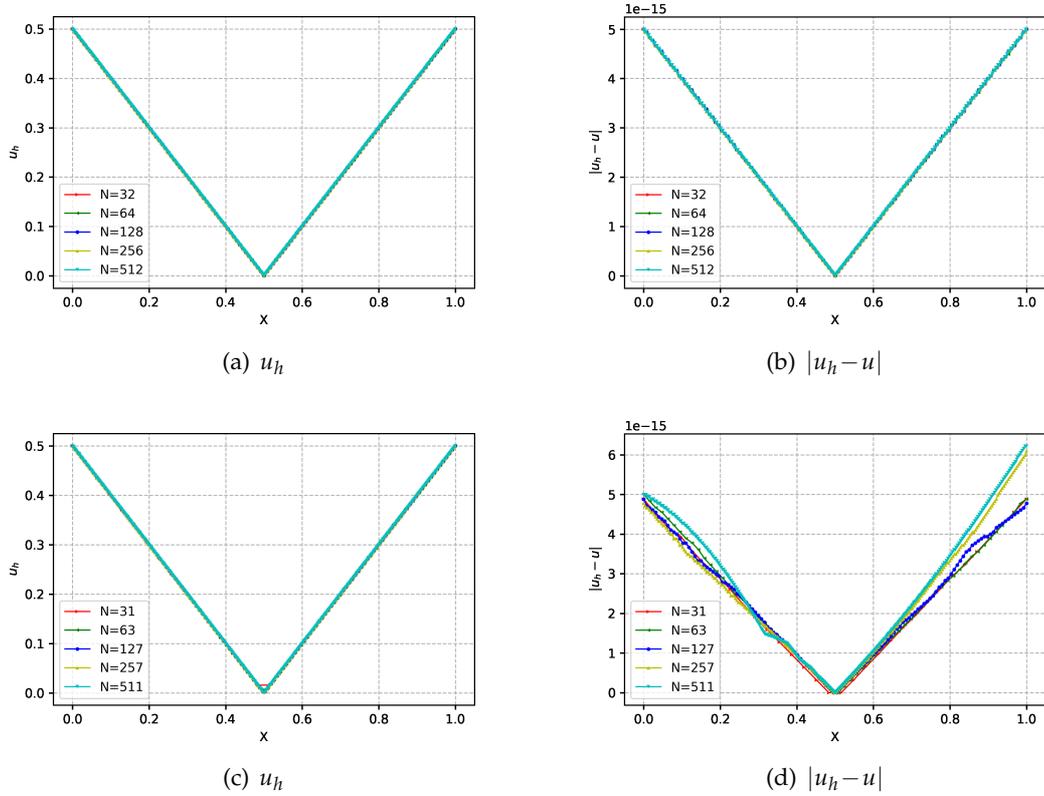


Figure 4: Numerical solutions u_h (a,c) and numerical errors $|u_h - u|$ (b,d) with the number of grid points $N=32,64,128,256,512$ (a,b) and $N=31,63,127,257,511$ (c,d) for Example-3 in 1D.

numerical error is around 10^{-15} no matter how coarse the mesh is. The numerical error is so small and near the machine error that the convergence rate cannot be observed. Numerical solution is shown in Fig. 4.

When $N = 31,63,127,257,511$, the singular point $x = 0.5$ lies inside the center cell of the mesh and is not a vertex of the mesh. In order to enforce the “boundary” condition, the minimal requirement is to fix two degrees of freedom of the center cell using u_* . As shown in Table 4, the convergence rates of the algorithm becomes $\mathcal{O}(h^{1.5}), \mathcal{O}(h^2), \mathcal{O}(h^1)$ and $\mathcal{O}(h^{0.5})$ in $L^2(\Omega), L^1(\Omega), L^\infty(\Omega)$ norm and $H^1(\Omega)$ seminorm, respectively. The order of convergence is reduced by 0.5. It is not surprising since the errors in the table are measured over the whole domain Ω , and there exists a mismatch at the center cell where the value of two degrees of freedom are specified as the given value of u_* which results in a kink since the linear approximation is used in the center cell, as shown in (c) and (d) of Fig. 4. The error implanted by the kink is $\mathcal{O}(h^{1.5}), \mathcal{O}(h^2), \mathcal{O}(h^1), \mathcal{O}(h^{0.5})$ for L^2, L^1, L^∞ norm and H^1 seminorm. If the errors are measured over $\tilde{\Omega} := \Omega \setminus [0.4, 0.6]$, the errors become near machine error as the previous case, as shown in Table 5.

Table 3: Numerical errors and convergence orders of Example-3 with $\nu=h^2$ and $N=32,64,128,256,512$.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
3.13e-02	2.88e-15	-	1.00e-14	-	2.49e-15	-	4.99e-15	-
1.56e-02	2.87e-15	-	1.00e-14	-	2.49e-15	-	4.99e-15	-
7.81e-03	2.88e-15	-	1.03e-14	-	2.49e-15	-	4.99e-15	-
3.91e-03	2.88e-15	-	1.11e-14	-	2.49e-15	-	4.99e-15	-
1.95e-03	2.88e-15	-	1.43e-14	-	2.49e-15	-	4.99e-15	-

Table 4: Numerical errors and convergence orders of Example-3 with $\nu=h^2$ and $N=31,63,127,255,511$.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
3.23e-02	1.56e-03	-	1.79e-01	-	2.4908e-04	-	1.06e-02	-
1.59e-02	5.39e-04	1.50	1.26e-01	0.50	6.03e-05	2.00	5.24e-03	1.00
7.87e-03	1.88e-04	1.50	8.87e-02	0.50	1.48e-05	2.00	2.59e-03	1.00
3.92e-03	6.62e-05	1.50	6.26e-02	0.50	3.68e-06	2.00	1.29e-03	1.00
1.96e-03	2.33e-05	1.50	4.42e-02	0.50	9.17e-07	2.00	6.46e-04	1.00

Table 5: Numerical error and convergence orders of Example-3 with $\nu=h^2$, $N=31,63,127,255,511$ and $\tilde{\Omega}=\Omega \setminus [0.4,0.6]$.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
3.23e-02	2.73e-15	-	9.09e-15	-	2.25e-15	-	4.82e-15	-
1.59e-02	2.81e-15	-	8.96e-15	-	2.34e-15	-	4.88e-15	-
7.87e-03	2.82e-15	-	9.28e-15	-	2.35e-15	-	4.99e-15	-
3.92e-03	2.85e-15	-	1.02e-14	-	2.37e-15	-	4.99e-15	-
1.96e-03	2.86e-15	-	1.35e-14	-	2.39e-15	-	4.99e-15	-

4.4 Example-4

In this example, we choose $f = 12\mathbb{1}_{x \leq 0.25} + 4\mathbb{1}_{x > 0.25}$, $\Omega = (0,1)$ and $\Gamma = \partial\Omega$ for (1.1). Its viscosity solution is $u_*(x) = 12x\mathbb{1}_{x \leq 0.25} + (4-4x)\mathbb{1}_{x > 0.25}$. The example is used to show the advantage of the Algorithm 1 over Newton's method. The viscosity is chosen as $\nu_h = 50h^2$. As shown in Fig. 5, Newton's method converges over two coarse grids, $N=32$ and $N=64$, but does not converge for other finer grids. The reason may be that the viscosity becomes too small to stabilize Newton's method. In contrast, the Algorithm 1 presented in the paper converges to the tolerance over all grids. The tolerance 10^{-14} is utilized here. The improvement comes from the better initial guess by the homotopy method. As shown in Fig. 5, the decrease of the difference of two successive steps is slow only at the beginning stage where $\nu_h^1 \sim h$ since the initial guess is always 0 in the computation. Nevertheless, the algorithm converges in one or two steps to the tolerance for other halved ν_h^k . In fact, the

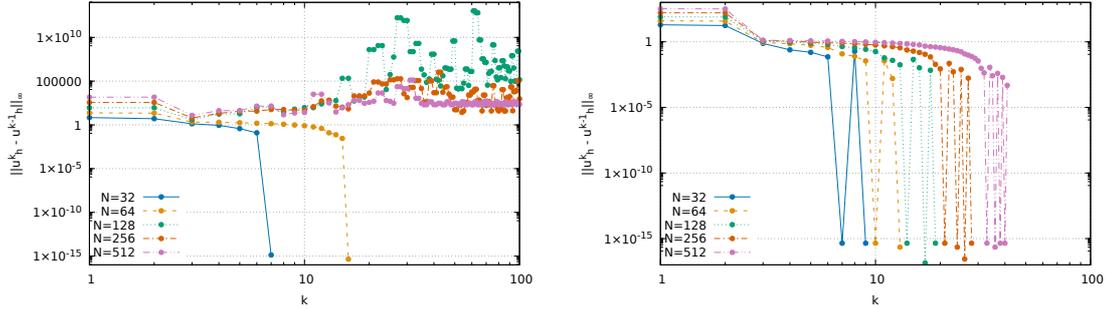


Figure 5: Comparison between Newton's method (left) and Algorithm 1 (right) on Example-4: The horizontal axis represents the number of iterations while the vertical axis is the L^∞ difference between two successive steps.

Table 6: Numerical errors and convergence orders of Example-5 with $\nu = h^2$.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
1.25e-01	3.82e-02	-	1.03e-01	-	3.56e-02	-	5.49e-02	-
6.25e-02	1.07e-02	1.83	4.64e-02	1.15	1.01e-02	1.81	1.52e-02	1.85
3.13e-02	2.88e-03	1.90	2.25e-02	1.05	2.74e-03	1.89	3.98e-03	1.94
1.56e-02	7.50e-04	1.94	1.11e-02	1.01	7.15e-04	1.94	1.02e-03	1.96
7.81e-03	1.93e-04	1.96	5.56e-03	1.00	1.84e-04	1.95	2.61e-04	1.97

algorithm works for $\nu_h = h^2$ over all grids, but Newton's method alone does not. And it is worth to mention that this phenomenon also happens for many other problems, which suggests that the Algorithm 1 is more robust.

4.5 Example-5

We choose $\Omega = [-1, 1] \times [-1, 1]$, $\Gamma = \{(0, 0)\}$, $u_\star = \cos(\pi + \frac{\pi}{2}x) + \cos(\pi + \frac{\pi}{2}y)$ as the true solution of (1.1), and $g = u_\star$, which implies $f = \frac{\pi}{2} \sqrt{\sin^2(\pi + \frac{\pi}{2}x) + \sin^2(\pi + \frac{\pi}{2}y)}$. The boundary $\partial\Omega$ is the outflow boundary and no data is specified there. The "boundary" condition on Γ can be enforced by specifying the values of all degree of freedoms in two ways as Section 3.3. The convergence test is shown in Table 6. Although $(0, 0)$ is a singular point where $f(0, 0) = 0$, all numerical errors are computed over the whole domain, i.e. $\tilde{\Omega} = \Omega$, since the singular point does not affect the convergence rate on the whole domain. The numerical solution on the finest mesh ($N = 256$) is shown in Fig. 6.

4.6 Example-6

We choose $\Omega = [-1, 1] \times [-1, 1]$, $f = 1$, $g = 0$, and $\Gamma = \{(0.5\cos\theta, 0.5\sin\theta), \theta \in [0, 2\pi)\}$. The exact solution u_\star is the distance function to the circle Γ . It is clear that u_\star is not smooth at the center $(0, 0)$ and the circle Γ , as indicate by a numerical solution shown in Fig. 7. We

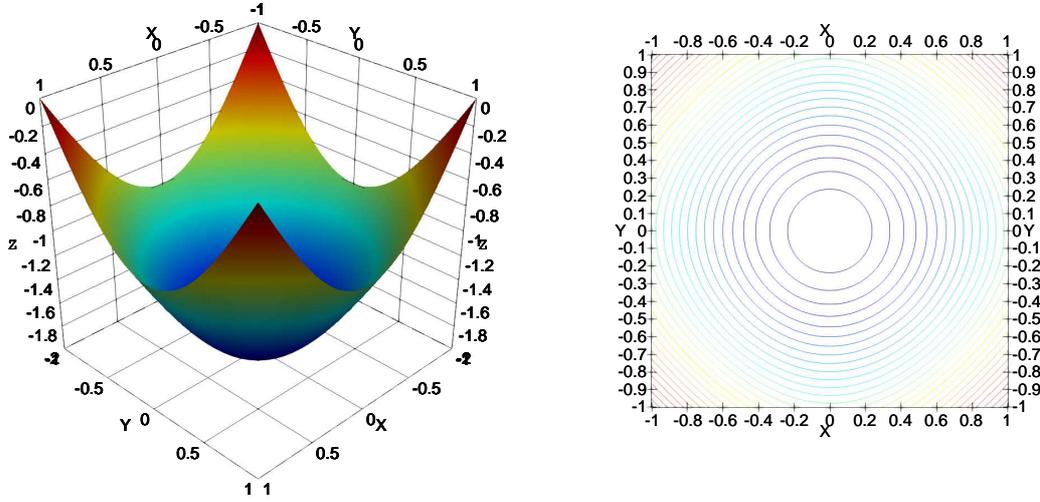


Figure 6: The 3D plot (left) and the contour plot (right) of numerical solution u_h of Example-5 on a uniform mesh with $N = 256$.

Table 7: Numerical errors and convergence orders of Example-6 with $\nu = h^2$ by using the 1st method to enforce the “boundary” condition.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
1.25e-01	6.43e-03	-	8.03e-02	-	5.19e-03	-	3.12e-02	-
6.25e-02	2.20e-03	1.55	3.83e-02	1.07	1.83e-03	1.50	1.00e-02	1.63
3.13e-02	6.18e-04	1.83	1.86e-02	1.04	5.26e-04	1.80	2.53e-03	1.99
1.56e-02	1.64e-04	1.91	9.32e-03	1.00	1.40e-04	1.90	7.94e-04	1.68
7.81e-03	4.22e-05	1.96	4.70e-03	0.99	3.63e-05	1.95	2.47e-04	1.68

choose

$$\tilde{\Omega} := \Omega \setminus (d(\{(0,0)\}, 0.1) \cup d(\Gamma, 0.1)), \tag{4.2}$$

where $d(S, \alpha)$ denotes the subset of Ω containing points having distance to $S \subset \Omega$ less than α . This is because \mathbb{P}_1 finite element method is used in the algorithm and $\tilde{\Omega}$ should not include $d(\Gamma, 0.1)$ as Example-3 in Section 4.3.

We also compare two methods discussed in Section 3.3 to enforce the “boundary” condition on Γ : the first method is to specify all degrees of freedom with distance to Γ less than $2h_{\min}$, while the second method is to specify all degrees of freedom with distance to Γ less than 0.1. The convergence tests are shown in Tables 7 and 8. Both methods give the expected accuracy. The second method produces a smaller numerical error since more degrees of freedom near Γ are specified.

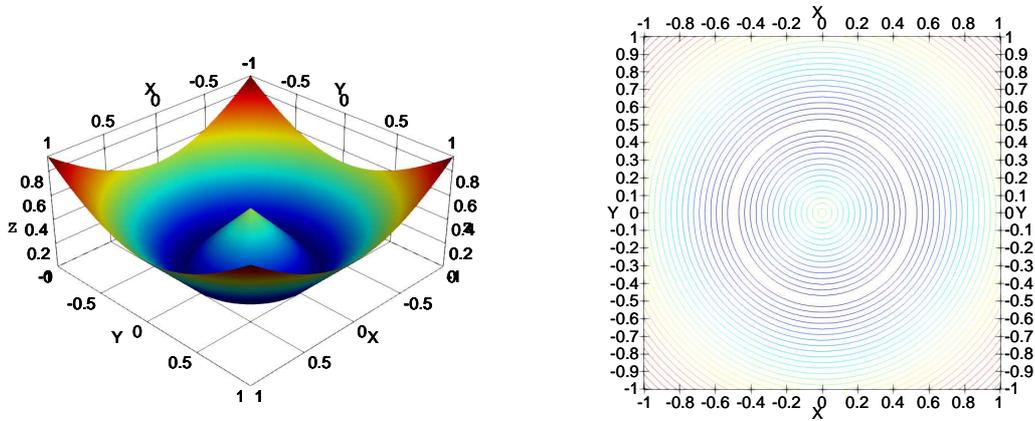


Figure 7: The 3D plot (left) and the 2D contour plot (right) of numerical solution u_h of Example-6 in 2D on a uniform mesh $N = 256$.

Table 8: Numerical errors and convergence orders of Example-6 with $\nu = h^2$ by using the 2nd method to enforce the “boundary” condition.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
1.25e-01	1.25e-02	-	9.02e-02	-	1.02e-02	-	2.82e-02	-
6.25e-02	2.30e-03	2.44	3.83e-02	1.23	1.93e-03	2.40	1.03e-02	1.45
3.13e-02	5.54e-04	2.05	1.86e-02	1.04	4.64e-04	2.06	2.45e-03	2.08
1.56e-02	1.35e-04	2.03	9.32e-03	1.00	1.12e-04	2.04	7.57e-04	1.70
7.81e-03	3.36e-05	2.01	4.70e-03	0.99	2.78e-05	2.02	2.39e-04	1.66

4.7 Example-7

We use $\Omega = [-2, 2]^2$, $f = 1$, $g = 0$, and $\Gamma = \{(\pm 1 + 0.5\cos\theta, 0.5\sin\theta), \theta \in [0, 2\pi]\}$. The true solution u_* of (1.1) is the distance function to Γ , which consists of two circles. Unlike Section 4.6, the true solution u_* is singular not only at the centers and two circles, but also along Y-axis. It suggests to choose $\tilde{\Omega}$ as

$$\tilde{\Omega} = \Omega \setminus (d(\{(-1, 0)\}, 0.1) \cup d(\{(1, 0)\}, 0.1) \cup d(\text{Y-axis}, 0.1) \cup d(\Gamma, 0.1)). \quad (4.3)$$

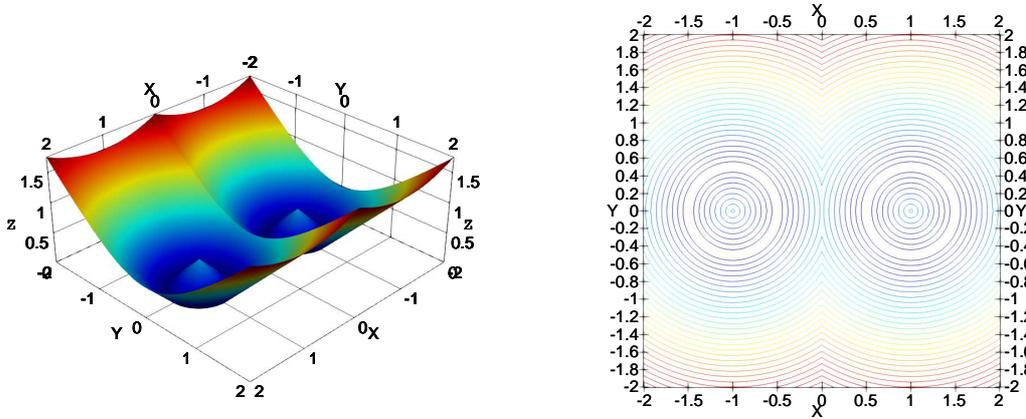
Numerical convergence tests confirms the desired behavior, as shown in Tables 9 and 10. The profile of a numerical solution is plotted in Fig. 8.

Table 9: Numerical errors and convergence orders of Example-7 with $\nu = h^2$ by using the 1st method to enforce the boundary condition.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
2.50e-01	3.33e-02	-	1.74e-01	-	2.63e-02	-	1.02e-01	-
1.25e-01	1.13e-02	1.56	6.75e-02	1.37	9.40e-03	1.49	3.12e-02	1.72
6.25e-02	3.31e-03	1.77	3.29e-02	1.03	2.83e-03	1.73	1.00e-02	1.63
3.13e-02	9.01e-04	1.88	1.59e-02	1.05	7.84e-04	1.86	2.53e-03	1.99
1.56e-02	2.34e-04	1.94	7.91e-03	1.01	2.05e-04	1.93	7.94e-04	1.68

Table 10: Numerical errors and convergence orders of Example-7 with $\nu = h^2$ by using the 2nd method to enforce the boundary condition.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
2.50e-01	5.70e-02	-	1.92e-01	-	4.95e-02	-	1.11e-01	-
1.25e-01	1.65e-02	1.78	7.41e-02	1.38	1.43e-02	1.79	3.35e-02	1.73
6.25e-02	3.39e-03	2.29	3.30e-02	1.17	2.93e-03	2.29	1.03e-02	1.69
3.13e-02	8.36e-04	2.02	1.59e-02	1.05	7.20e-04	2.02	2.45e-03	2.08
1.56e-02	2.05e-04	2.02	7.92e-03	1.01	1.76e-04	2.03	7.57e-04	1.70

Figure 8: The 3D plot (left) and the 2D contour (right) of numerical solution u_h of Example-7 in 2D on a uniform mesh $N = 256$.

4.8 Example-8

We use $\Omega = [-1, 1]^2$, $f = 1$, $g = 0$, and $\Gamma = \{(0, 0)\}$. The exact solution u_* is the distance function to the origin point $(0, 0)$. Since u_* is singular at $(0, 0)$, we choose

$$\tilde{\Omega} = \Omega \setminus d(\{(0, 0)\}, 0.1). \quad (4.4)$$

Table 11: Numerical errors and convergence orders of Example-8 with $\nu = h^2$ by using the 1st method to enforce the boundary condition.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
1.25e-01	2.95e-02	-	8.63e-02	-	2.86e-02	-	4.16e-02	-
6.25e-02	1.11e-02	1.41	4.15e-02	1.05	1.09e-02	1.39	1.50e-02	1.47
3.13e-02	3.99e-03	1.48	2.03e-02	1.03	3.93e-03	1.48	5.38e-03	1.48
1.56e-02	1.42e-03	1.49	1.05e-02	0.95	1.37e-03	1.51	2.01e-03	1.42
7.81e-03	5.29e-04	1.43	5.81e-03	0.86	4.78e-04	1.53	7.96e-04	1.34

Table 12: Numerical errors and convergence orders of Example-8 with $\nu = h^2$ using the 2nd method to enforce the boundary condition.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
1.25e-01	4.46e-02	-	9.05e-02	-	4.36e-02	-	6.25e-02	-
6.25e-02	1.11e-02	2.00	4.15e-02	1.12	1.09e-02	1.99	1.50e-02	2.06
3.13e-02	2.61e-03	2.09	1.99e-02	1.06	2.56e-03	2.10	3.74e-03	2.00
1.56e-02	6.50e-04	2.01	9.88e-03	1.01	6.36e-04	2.01	9.39e-04	2.00
7.81e-03	1.58e-04	2.04	4.92e-03	1.00	1.55e-04	2.04	2.24e-04	2.07

Because all characteristic lines are generated from $(0,0)$, its singularity is stronger than the previous two examples in Section 4.6 and Section 4.7. Thus the first method of enforcing Γ condition is not sufficient to ensure the desired convergence rate as shown in Table 11. However, the second method produces the full convergence rate as shown in Table 12. A numerical solution is plotted in Fig. 9.

4.9 Example-9

We use $\Omega = [-1,1]^2, g = 0, f = 1$ and Γ as the curve shown in the Fig. 10. The true solution u_* is the distance function to Γ . It is not smooth along Γ and the shock wave \mathcal{S} that is indicated in Fig. 11 in bold line. Therefore we choose

$$\tilde{\Omega} = \Omega \setminus (d(\Gamma, 0.1) \cup d(\mathcal{S}, 0.1)) \tag{4.5}$$

to measure the convergence of the algorithm. Because of three strong singular points $\{(1,0), (0,1), (0,0)\}$ producing infinite many characteristic lines, the first strategy to enforce the boundary condition on Γ does not produce the full convergence rate in Table 13, but the second strategy obtains the desired convergence as shown in Table 14. Here, we choose $\nu_h = 5h^2$ which is bigger than previous examples.

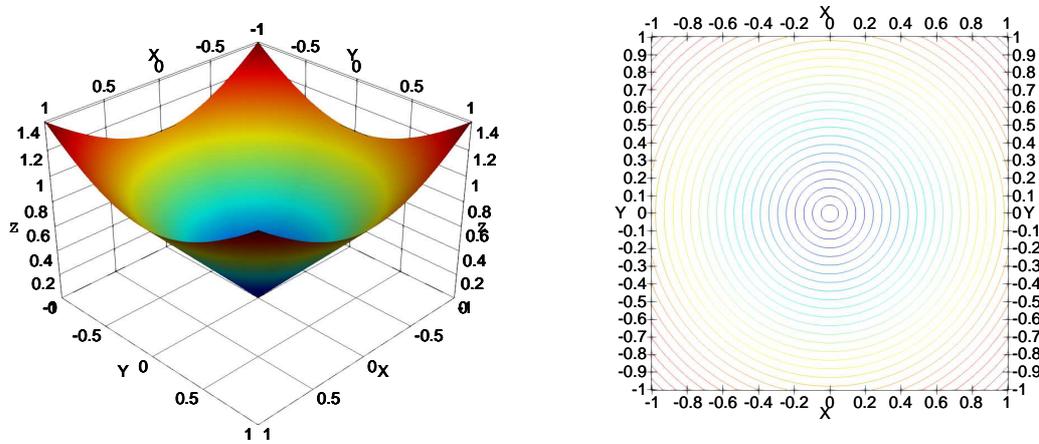


Figure 9: The 3D plot (left) and the 2D contour plot (right) of numerical solution u_h of Example-8 in 2D on a uniform mesh $N=256$.

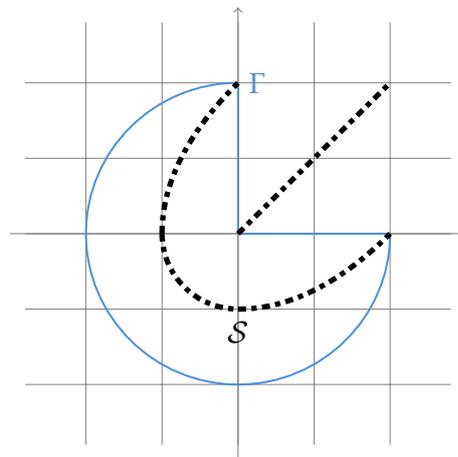


Figure 10: The curve Γ and the shock S of Example-9.

4.10 Example-10

This example is a typical problem of shape-from-shading over the domain $\Omega = [0,1]^2$, see e.g. [38, 46]. The true solution u_* is the shape function reconstructed from (1.1) with $f(x,y) = \sqrt{[\cos(2\pi x)\sin(2\pi y)]^2 + [\sin(2\pi x)\cos(2\pi y)]^2}$ which is related to the brightness of the shape. The set Γ is chosen as

$$\Gamma = \partial\Omega \cup \{(0.25,0.25), (0.25,0.75), (0.75,0.25), (0.75,0.75), (0.5,0.5)\}.$$

Table 13: Numerical errors and convergence orders of Example-9 in 2D with $\nu=5h^2$ using the 1st method to enforce the boundary condition.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
2.50e-01	9.72e-02	-	2.30e-01	-	7.06e-02	-	2.69e-01	-
1.25e-01	4.88e-02	0.99	1.12e-01	1.04	3.53e-02	1.00	1.43e-01	0.91
6.25e-02	1.93e-02	1.33	3.67e-02	1.61	1.30e-02	1.44	5.70e-02	1.33
3.13e-02	6.56e-03	1.56	1.48e-02	1.30	4.12e-03	1.66	1.93e-02	1.56
1.56e-02	2.11e-03	1.63	7.10e-03	1.07	1.25e-03	1.72	6.11e-03	1.66

Table 14: Numerical errors and convergence orders of Example-9 with $\nu=5h^2$ using the 2nd method to enforce the boundary condition.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
2.50e-01	2.06e-01	-	3.18e-01	-	1.74e-01	-	4.12e-01	-
1.25e-01	6.70e-02	1.63	1.31e-01	1.28	4.99e-02	1.81	1.84e-01	1.16
6.25e-02	1.95e-02	1.78	3.67e-02	1.84	1.32e-02	1.91	5.70e-02	1.69
3.13e-02	5.12e-03	1.93	1.25e-02	1.55	3.36e-03	1.98	1.55e-02	1.87
1.56e-02	1.30e-03	1.97	5.74e-03	1.13	8.45e-04	1.99	3.95e-03	1.98

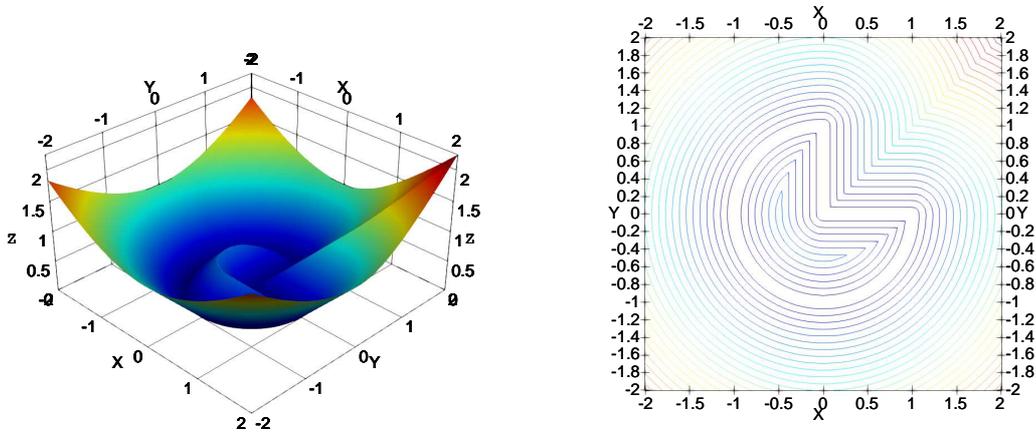


Figure 11: The 3D plot (left) and the 2D contour plot (right) of numerical solution u_h of Example-9 in 2D on a uniform mesh $N=256$.

The function g is the restriction on Γ of the exact shape function $u_{\star}^i, i=1,2$. Two cases are considered

a) $u_{\star}^1 = \sin(2\pi x)\sin(2\pi y);$

Table 15: Numerical errors and convergence orders of Example-10 a) with $\nu=20h^2$.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
6.25e-02	1.19e-01	-	1.75e+00	-	9.32e-02	-	2.38e-01	-
3.13e-02	6.12e-02	0.97	1.10e+00	0.67	4.72e-02	0.98	1.17e-01	1.02
1.56e-02	1.94e-02	1.65	5.39e-01	1.03	1.52e-02	1.63	4.04e-02	1.54
7.81e-03	4.92e-03	1.98	2.58e-01	1.06	3.88e-03	1.97	1.20e-02	1.75
3.91e-03	1.21e-03	2.02	1.30e-01	0.99	9.61e-04	2.02	4.10e-03	1.55

Table 16: Numerical errors and convergence orders of Example-10 b) with $\nu=20h^2$.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
6.25e-02	9.84e-02	-	1.69e+00	-	7.63e-02	-	2.51e-01	-
3.13e-02	6.26e-02	0.65	1.07e+00	0.66	4.28e-02	0.83	1.67e-01	0.59
1.56e-02	2.43e-02	1.36	5.40e-01	0.99	1.64e-02	1.38	5.68e-02	1.55
7.81e-03	7.28e-03	1.74	2.66e-01	1.02	4.96e-03	1.73	1.51e-02	1.91
3.91e-03	1.96e-03	1.89	1.37e-01	0.96	1.34e-03	1.89	4.42e-03	1.78

$$b) u_{\star}^2 = \begin{cases} \max(|\sin(2\pi x)\sin(2\pi y)|, 1 + \cos(2\pi x)\cos(2\pi y)), & \text{if } |x+y-1| < \frac{1}{2} \text{ and } |x-y| < \frac{1}{2}, \\ |\sin(2\pi x)\sin(2\pi y)|, & \text{otherwise.} \end{cases}$$

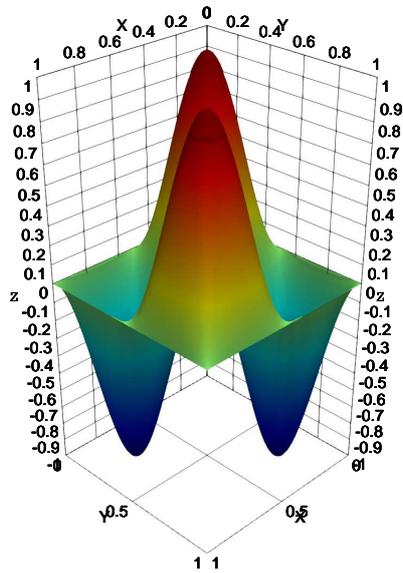
The zero condition of g on $\partial\Omega$ is implemented as usual finite element methods, while the condition over other five points of Γ is enforced by the second method of Section 3.3 since the strong singularity happens at Γ . As shown in Tables 15 and 16, the convergence rates are $\mathcal{O}(h^2), \mathcal{O}(h^2), \mathcal{O}(h^2)$ and $\mathcal{O}(h^1)$ in L^2, L^1, L^∞ norm and H^1 seminorm, respectively. However, the convergence rate of case b) is slightly smaller than case a) since u_{\star}^2 is not as smooth as u_{\star}^1 . A numerical solution is shown in Fig. 12.

The comparison between the execution times of the proposed method and a DG fast sweeping method is reported in Table 17. The second column of the table is from Table 3.13 of [45], where the same problem over the same mesh is considered. It shows that the second order DG fast sweeping method is more efficient. However, the execution time of the proposed method is also linear with respect to the mesh size, and more importantly, it works for the unstructured mesh without additional special manipulations.

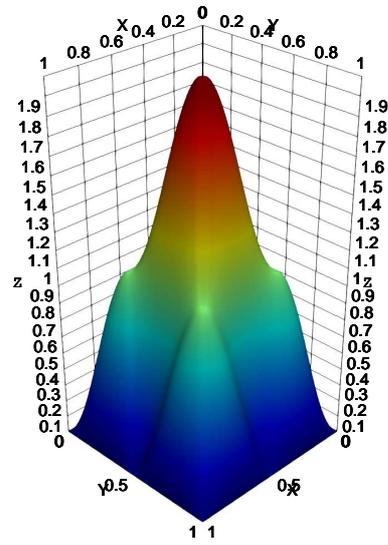
4.11 Example-11

In this example, we use $\Omega = [0,1]^2$, $\Gamma = \partial\Omega$, $g=0$, and u_{\star}, f are defined as

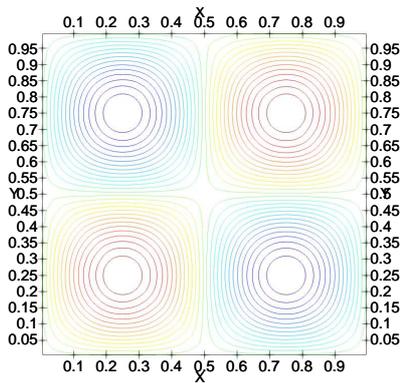
$$a) \begin{cases} f = 2\sqrt{y^2(1-x^2)^2 + x^2(1-y^2)^2}, \\ u_{\star} = (1-x^2)(1-y^2); \end{cases}$$



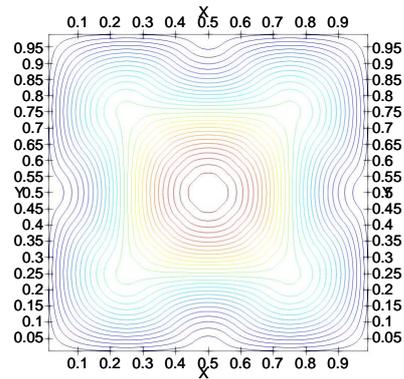
(a) 2D Example-10a



(b) 2D Example-10b



(c) 30 contours of 2D Example-10a



(d) 30 contours of 2D Example-10b

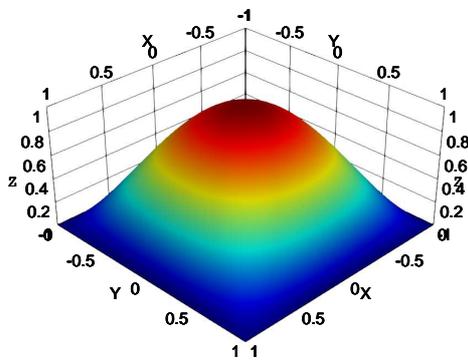
Figure 12: The 3D plot (upper) and the 2D contour plot (lower) of numerical solution u_h of Example-10 a) & b) in 2D on a uniform mesh $N = 256$.

$$b) \begin{cases} f = \sqrt{(1-|x|)^2 + (1-|y|)^2}, \\ u_* = (1-|x|)(1-|y|). \end{cases}$$

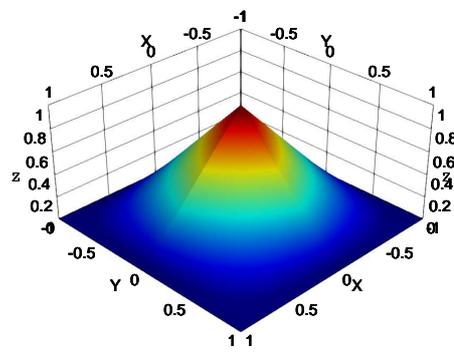
The convergence tests are shown in Tables 18 and 19. A numerical solution on the structured mesh is shown in Fig. 13.

Table 17: Numerical performance of Example-10a.

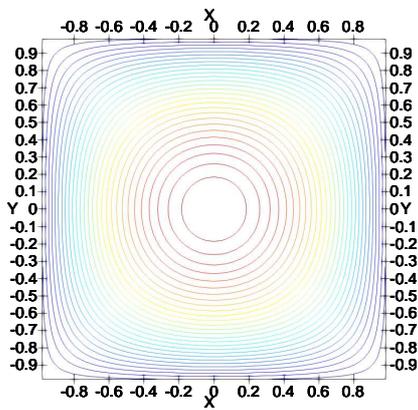
Mesh size	Execution time of the	Execution time of
	DG Fast sweeping method [45]	the proposed algorithm
	(unit: sec)	(unit: sec)
40×40	0.22	0.314
80×80	0.87	2.268
160×160	3.40	13.127
320×320	13.62	73.245



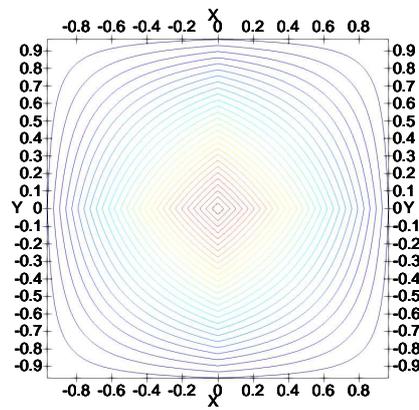
(a) 2D Example-11a



(b) 2D Example-11b



(c) 30 contours of 2D Example-11a



(d) 30 contours of 2D Example-11b

Figure 13: The 3D plot (upper) and the 2D contour plot (lower) of numerical solution u_h of Example-11 a) & b) in 2D on a uniform mesh $N=256$.

Table 18: Numerical errors and convergence orders of Example-11a with $\nu=h^2$.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
1.25e-01	2.13e-02	-	1.28e-01	-	1.70e-02	-	5.63e-02	-
6.25e-02	5.33e-03	2.00	6.20e-02	1.06	4.25e-03	2.01	1.51e-02	1.90
3.13e-02	1.33e-03	2.00	3.06e-02	1.02	1.06e-03	2.00	3.93e-03	1.94
1.56e-02	3.33e-04	2.00	1.52e-02	1.01	2.65e-04	2.00	1.02e-03	1.94
7.81e-03	8.33e-05	2.00	7.61e-03	1.00	6.63e-05	2.00	2.67e-04	1.94

Table 19: Numerical errors and convergence orders of Example-11b with $\nu=h^2$.

h	e_1	Order	e_2	Order	e_3	Order	e_4	Order
1.25e-01	5.32e-03	-	1.14e-01	-	2.94e-03	-	3.89e-02	-
6.25e-02	1.11e-03	2.25	5.88e-02	0.96	6.69e-04	2.14	9.52e-03	2.03
3.13e-02	2.56e-04	2.13	2.95e-02	1.00	1.62e-04	2.04	2.37e-03	2.00
1.56e-02	6.21e-05	2.04	1.47e-02	1.00	4.02e-05	2.01	5.97e-04	1.99
7.81e-03	1.53e-05	2.01	7.38e-03	1.00	1.00e-05	2.00	1.50e-04	1.99

4.12 Example-12

In this example, we choose $f = 1$ to compute the distance function to the boundary of several irregular domains. Four different cases [5] are considered

1. L-shape domain: $\Omega_1 := [0,2]^2 \setminus [1,2]^2$;
2. Unit disk: $\Omega_2 := \{(x,y) : x^2 + y^2 \leq 1\}$;
3. Ellipse: $\Omega_3 := \{(x,y) : x^2 + y^2/4 \leq 1\}$;
4. Half ellipse: $\Omega_4 := \{(x,y) : x^2 + y^2/4 \leq 1, x \geq 0\}$.

The unstructured meshes are shown in Fig. 14. The numerical results are shown in Figs. 15, 16, 17, and 18. The number of iterations and the execution time for the first case is shown in Table 20, which shows that the method is almost close to the linear computational complexity. It demonstrates the efficiency of the proposed method.

4.13 Example-13

In this example, the computational domain is the ellipse $\Omega = \{(x,y) : x^2 + y^2/4 \leq 1\}$. The true solution is the distance function to the two foci $(0, \pm\sqrt{3})$. The "boundary condition" over $\Gamma = \{(0, \pm\sqrt{3})\}$ is enforced by specifying the values of all degree of freedoms with distance less than 0.1 to the two foci. The numerical results are shown in Fig. 19.

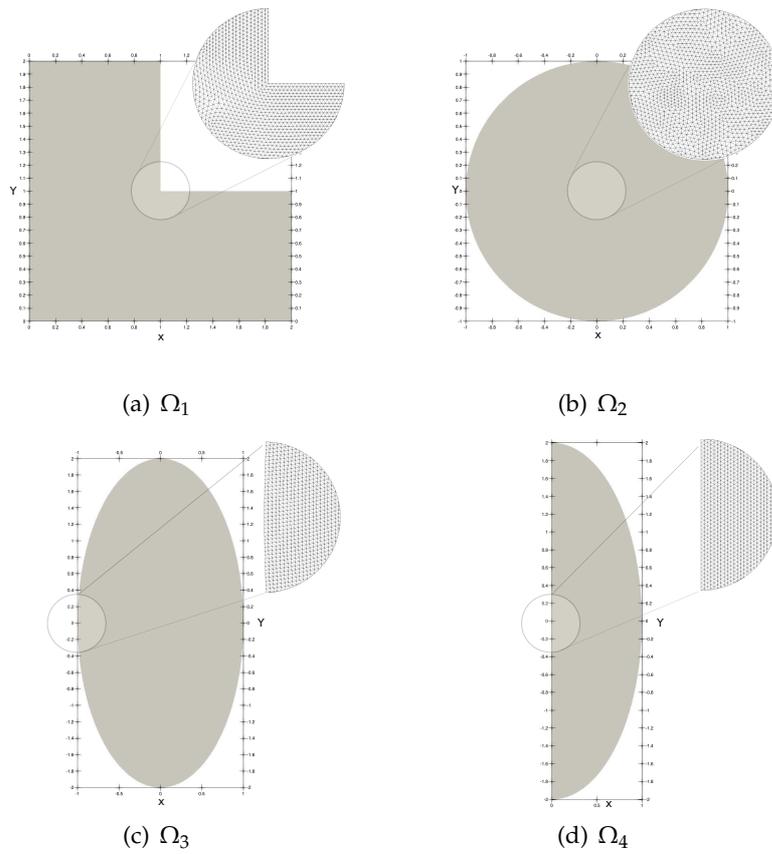


Figure 14: Unstructured meshes used in four different domains of Example-12. The zoom-in meshes highlight in specific sub-domains.

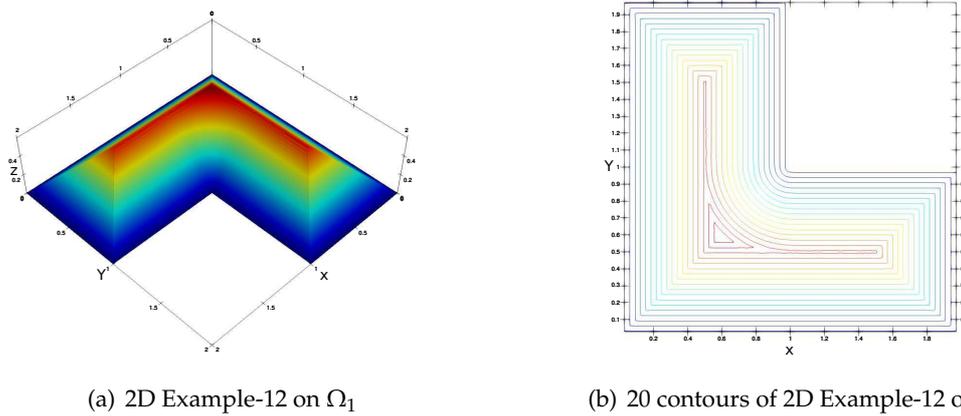
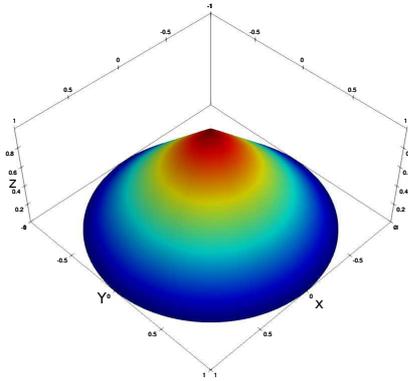
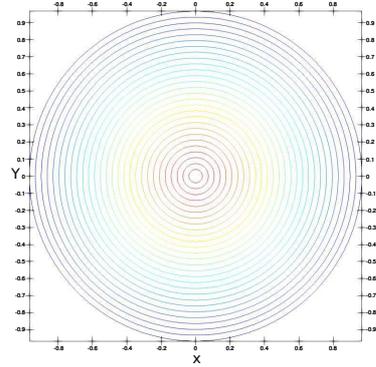


Figure 15: The 3D plot and the 2D contour plot of numerical solution u_h of Example-12 on Ω_1 with a unstructured mesh (69932 cells and 209796 nodes).

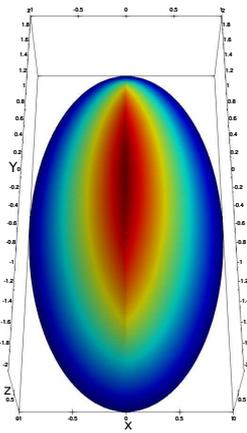


(a) 2D Example-12 on Ω_2

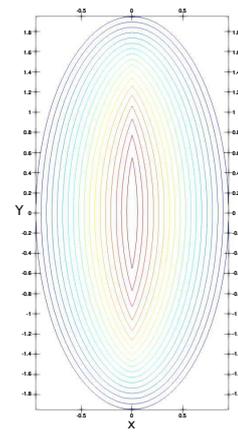


(b) 20 contours of 2D Example-12 on Ω_2

Figure 16: The 3D plot and the 2D contour plot of numerical solution u_h of Example-12 on Ω_2 with a unstructured mesh (71672 cells and 215016 nodes).



(a) 2D Example-12 on Ω_3



(b) 20 contours of 2D Example-12 on Ω_3

Figure 17: The 3D plot and the 2D contour plot of numerical solution u_h of Example-12 on Ω_3 with a unstructured mesh (141094 cells and 423282 nodes).

Table 20: Numerical performance of Example-12 on L-shape domain.

Mesh size h	Degrees of freedom	Total number of Newton iterations	Number of homotopy reduction	Execution time (unit: sec)
0.08	610	12	3	0.13
0.04	2271	17	4	0.62
0.02	8893	21	5	4.07
0.01	35367	30	6	21.86

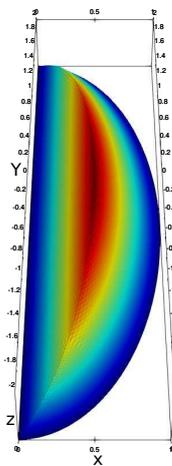
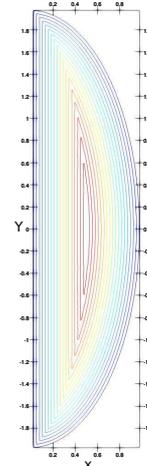
(a) 2D Example-12 on Ω_4 (b) 20 contours of 2D Example-12 on Ω_4

Figure 18: The 3D plot and the 2D contour plot of numerical solution u_h of Example-12 on Ω_4 with a unstructured mesh (70780 cells and 212340 nodes).

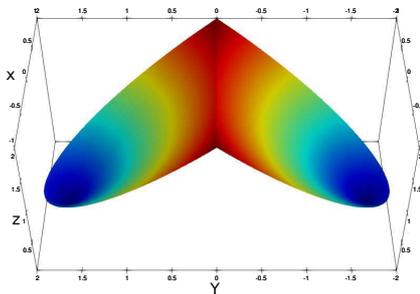
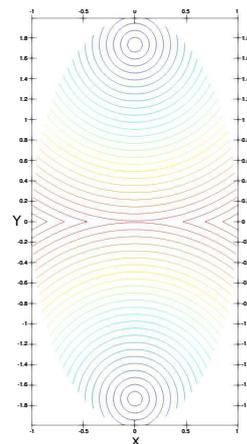
(a) 2D Example-13 on Ω_3 (b) 20 contours of 2D Example-13 on Ω_3

Figure 19: The 3D plot and the 2D contour plot of numerical solution u_h of Example-13 on Ω_3 with a unstructured mesh (19726 cells and 19777 nodes).

4.14 Example-14

In this example, an anisotropic Eikonal equation is considered

$$\begin{cases} \sqrt{au_x^2 + bu_y^2} - 2cu_xu_y = 1, & (x,y) \in \Omega = (-2,2)^2, \\ u(0,0) = 0, \end{cases} \quad (4.6)$$

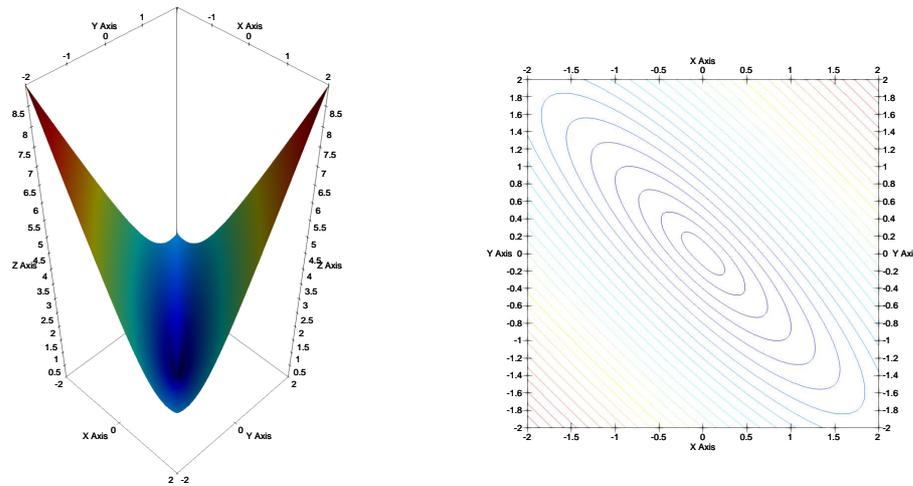


Figure 20: Numerical solution u_h of Example-14 over a structured mesh $N=101$ and its 2D contour plot.

where $a=b=1, c=0.9$. The example is from [35] where a fast sweeping method is designed for structured and unstructured meshes. The exact solution u_* has the following form

$$u_*(x,y) = x \frac{\sqrt{ap^2 + bq^2 - 2cpq}}{ap - cq} \quad \text{or} \quad u_*(x,y) = y \frac{\sqrt{ap^2 + bq^2 - 2cpq}}{bq - cp}, \quad (4.7)$$

where $p = cy + bx$ and $q = cx + ay$. It is used to specify the value of degrees of freedom with distance less than 0.1 to the center $(0,0)$ since it is a singular point of the problem. We solve the problem on a structured mesh with $100 \times 100 \times 2$ triangle cells and plot the solutions in Fig. 20.

4.15 Example-15

In this example, we compute the 3D distance functions to the two spheres S_{\pm} with center $(\pm 1, 0, 0)$ and radius 0.5 over the domain $\Omega := [-2, 2]^3 \setminus (S_+ \cup S_-)$. The unstructured mesh and several isosurfaces are shown in Fig. 21. The efficiency of the method is shown in Table 21, which indicates that the method is close to the linear computational complexity.

Table 21: Numerical performance of Example-15.

Mesh size h	Degrees of freedom	Total number of Newton iterations	Number of homotopy reduction	Execution time (unit: sec)
0.4	24464	10	2	1.22
0.2	220832	10	2	7.81
0.1	1773828	11	2	78.24
0.05	14144920	16	3	1098.87

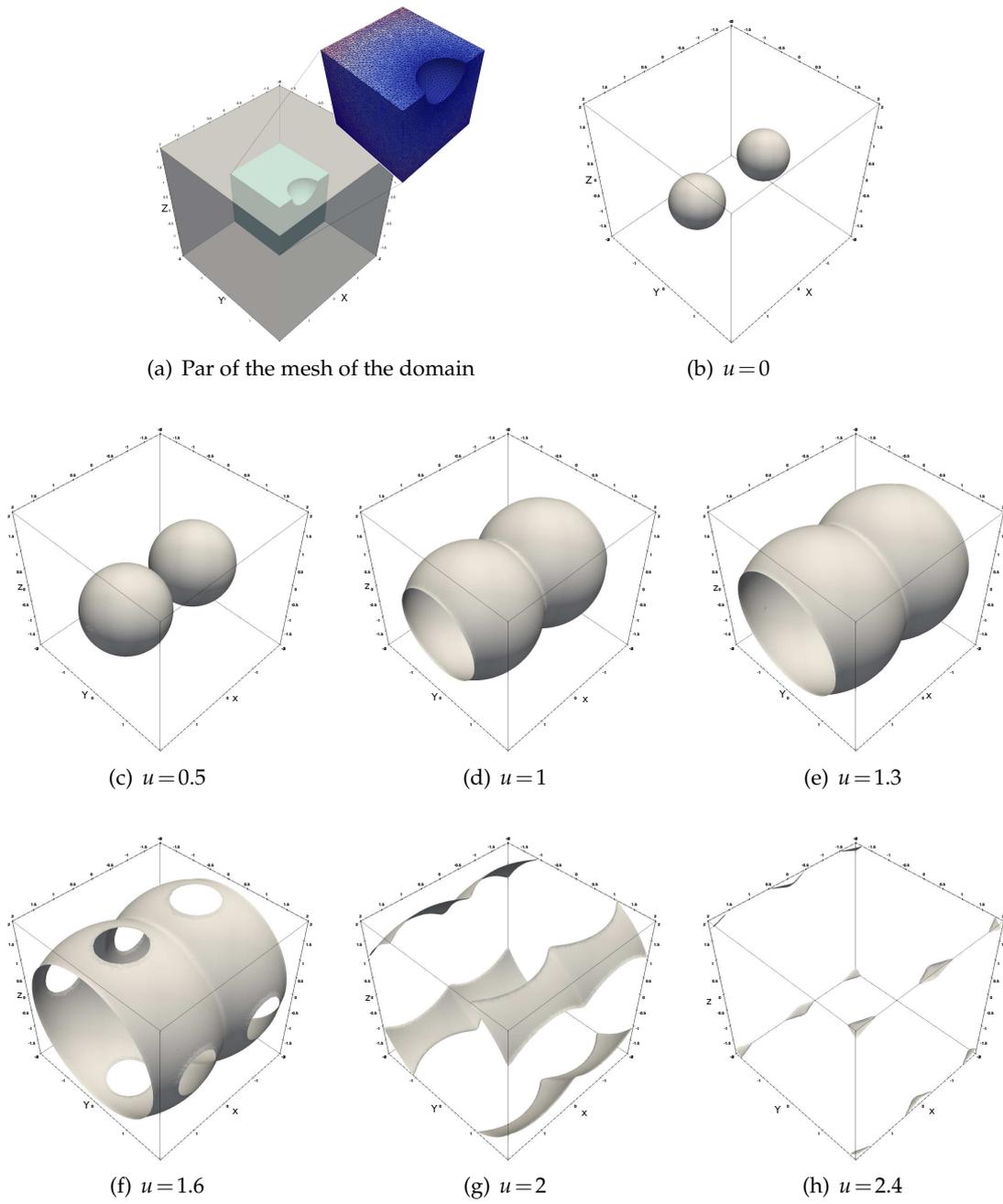


Figure 21: Part of the mesh with $h = 0.05$ and some isosurfaces of the numerical solution of Example-15.

5 Conclusion

We propose a continuous finite element method based on the vanishing viscosity strategy to solve the static Eikonal equation. The new method utilizes the homotopy method and Newton's method to efficiently solve the discretized nonlinear system. The homotopy method guarantees the convergence of the nonlinear solver for all $v_h = ch^2$ in all examples and over all grids, while Newton's method usually converges for just special big $v_h = ch$ on very coarse grids, or simple 1D examples only. Extensive numerical examples show the second order convergence of the algorithm problems on structured or unstructured meshes. In this paper, we focus on the linear finite element method only. We will extend it to \mathbb{P}_2 finite element along with the high order viscosity as $v_h = \mathcal{O}(h^3)$, which may require more advanced homotopy techniques and will be investigated with further efforts in the future.

Acknowledgments

YY is supported by Natural Science Foundation of Jiangsu Province (Nos. KFR21026, PAF20042), National Natural Science Foundation of China (Nos. GBA20029, GCA20004), Science Challenge Project (No. TZ2018002), and National Science and Technology Major Project (No. J2019-II-0007-0027). WH is supported by NSF DMS-1818769.

References

- [1] H.-K. ZHAO, S. OSHER, B. MERRIMAN, AND M. KANG, *Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method*, Computer Vision and Image Understanding, 80 (2000), pp. 295–314.
- [2] R. ADAMS AND J. FOURNIER, *Sobolev Spaces*, Elsevier Science, 2003.
- [3] T. J. BARTH AND J. A. SETHIAN, *Numerical schemes for the Hamilton–Jacobi and level set equations on triangulated domains*, Journal of Computational Physics, 145 (1998), pp. 1–40, <https://doi.org/10.1006/jcph.1998.6007>.
- [4] F. BORNEMANN AND C. RASCH, *Finite-element discretization of static Hamilton–Jacobi equations based on a local variational principle*, Computing and Visualization in Science, 9 (2006), pp. 57–69, <https://doi.org/10.1007/s00791-006-0016-y>.
- [5] A. CABOUSSAT AND R. GLOWINSKI, *A penalty-regularization-operator splitting method for the numerical solution of a scalar Eikonal equation*, Chinese Annals of Mathematics, Series B, 36 (2015), pp. 659–688, <https://doi.org/10.1007/s11401-015-0930-8>.
- [6] V. CERVENY, I. A. MOLOTKOV, AND I. PSENCIK, *Ray Method in Seismology*, Charles University Press, Prague, 1977.
- [7] A. G. CHURBANOV AND P. N. VABISHCHEVICH, *Numerical solution of boundary value problems for the Eikonal equation in an anisotropic medium*, Journal of Computational and Applied Mathematics, 362 (2019), pp. 55–67, <https://doi.org/10.1016/j.cam.2019.05.016>.
- [8] M. G. CRANDALL, L. C. EVANS, AND P. L. LIONS, *Some properties of viscosity solutions of Hamilton–Jacobi equations*, Transactions of the American Mathematical Society, 282 (1984), pp. 487–502.

- [9] M. G. CRANDALL, H. ISHII, AND P.-L. LIONS, *User's guide to viscosity solutions of second order partial differential equations*, Bulletin of the American Mathematical Society, 27 (1992), pp. 1–68, <https://doi.org/10.1090/s0273-0979-1992-00266-5>.
- [10] M. G. CRANDALL AND P.-L. LIONS, *Viscosity solutions of Hamilton-Jacobi equations*, Transactions of the American Mathematical Society, 277 (1983), pp. 1–42, <https://doi.org/10.1090/s0002-9947-1983-0690039-8>.
- [11] M. G. CRANDALL AND P.-L. LIONS, *Two approximations of solutions of Hamilton-Jacobi equations*, Mathematics of Computation, 43 (1984), pp. 1–19, <https://doi.org/10.1090/s0025-5718-1984-0744921-8>.
- [12] R. CROCE, M. GRIEBEL, AND M. A. SCHWEITZER, *A parallel level-set approach for two-phase flow problems with surface tension in three space dimensions*, Preprint 157, Sonderforschungsbereich 611, University Bonn, 2004.
- [13] A. ERN AND J.-L. GUERMOND, *Theory and Practice of Finite Elements*, Springer New York, 2004, <https://doi.org/10.1007/978-1-4757-4355-5>.
- [14] A. ERN AND J.-L. GUERMOND, *Finite Elements I: Approximation and Interpolation*, vol. 72 of Texts in Applied Mathematics, Springer International Publishing, Cham, 2021, <https://doi.org/10.1007/978-3-030-56341-7>.
- [15] A. ERN AND J.-L. GUERMOND, *Finite Elements III: First-Order and Time-Dependent PDEs*, vol. 74 of Texts in Applied Mathematics, Springer International Publishing, Cham, 2021, <https://doi.org/10.1007/978-3-030-57348-5>.
- [16] L. C. EVANS, *Partial Differential Equations*, American Mathematical Society, 2010.
- [17] C. GEUZAIN AND J.-F. REMACLE, *Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*, International Journal for Numerical Methods in Engineering, 79 (2009), pp. 1309–1331, <https://doi.org/10.1002/nme.2579>.
- [18] J.-L. GUERMOND, F. MARPEAU, AND B. POPOV, *A fast algorithm for solving first-order PDEs by L1-minimization*, Communications in Mathematical Sciences, 6 (2008), pp. 199–216, <https://doi.org/10.4310/CMS.2008.v6.n1.a10>.
- [19] J.-L. GUERMOND AND B. POPOV, *L1-minimization methods for Hamilton-Jacobi equations: The one-dimensional case*, Numerische Mathematik, 109 (2008), pp. 269–284, <https://doi.org/10.1007/s00211-008-0142-1>.
- [20] J.-L. GUERMOND AND B. POPOV, *L1-approximation of stationary Hamilton-Jacobi equations*, SIAM Journal on Numerical Analysis, 47 (2009), pp. 339–362, <https://doi.org/10.1137/070681922>.
- [21] W. HAO, J. D. HAUENSTEIN, C.-W. SHU, A. J. SOMMESE, Z. XU, AND Y.-T. ZHANG, *A homotopy method based on weno schemes for solving steady state problems of hyperbolic conservation laws*, Journal of Computational Physics, 250 (2013), pp. 332–346.
- [22] W. HAO, J. HESTHAVEN, G. LIN, AND B. ZHENG, *A homotopy method with adaptive basis selection for computing multiple solutions of differential equations*, Tech. Report 1, 2020.
- [23] W. HAO AND C. XUE, *Spatial pattern formation in reaction-diffusion models: A computational approach*, Journal of Mathematical Biology, (2020), pp. 1–23.
- [24] W. HAO AND C. ZHENG, *An adaptive homotopy method for computing bifurcations of nonlinear parametric systems*, Journal of Scientific Computing, 82 (2020), pp. 1–19.
- [25] W.-K. JEONG AND R. T. WHITAKER, *A fast iterative method for Eikonal equations*, SIAM Journal on Scientific Computing, 30 (2008), pp. 2512–2534.
- [26] C. Y. KAO, S. OSHER, AND J. QIAN, *Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations*, 196 (2004), pp. 367–391, <https://doi.org/10.1016/j.jcp.2003.11.007>.
- [27] C.-Y. KAO, S. OSHER, AND Y.-H. TSAI, *Fast sweeping methods for static Hamilton-Jacobi equa-*

- tions, 42 (2005), pp. 2612–2632, <https://doi.org/10.1137/s0036142902419600>.
- [28] R. KIMMEL, K. SIDDIQI, B. B. KIMIA, AND A. M. BRUCKSTEIN, *Shape from shading: Level set propagation and viscosity solutions*, International Journal of Computer Vision, 16 (1995), pp. 107–133, <https://doi.org/10.1007/BF01539551>.
- [29] S. N. KRUŽKOV, *Generalized solutions of the Hamilton-Jacobi equations of Eikonal type. I. Formulation of the problems. Existence, uniqueness and stability theorems; some properties of the solutions*, Math. USSR Sb., 27 (1975), pp. 406–446, <https://doi.org/10.1070/sm1975v027n03abeh002522>.
- [30] F. LI, C.-W. SHU, Y.-T. ZHANG, AND H. ZHAO, *A second order discontinuous Galerkin fast sweeping method for Eikonal equations*, Journal of Computational Physics, 227 (2008), pp. 8191–8208, <https://doi.org/10.1016/j.jcp.2008.05.018>.
- [31] P. L. LIONS, *Generalized Solutions of Hamilton-Jacobi Equations (Research Notes in Mathematics Series, 69)*, Pitman, Oct 1982.
- [32] S. OSHER, *A level set formulation for the solution of the Dirichlet problem for Hamilton–Jacobi equations*, SIAM Journal on Mathematical Analysis, 24 (1993), pp. 1145–1152, <https://doi.org/10.1137/0524066>.
- [33] S. OSHER AND J. A. SETHIAN, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, Journal of Computational Physics, 79 (1988), pp. 12–49, [https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2).
- [34] E. PRADOS AND O. FAUGERAS, *Shape from Shading*, in Handbook of Mathematical Models in Computer Vision, Springer, 2006, pp. 375–388.
- [35] J. QIAN, Y.-T. ZHANG, AND H. ZHAO, *A fast sweeping method for static convex Hamilton–Jacobi equations*, Journal of Scientific Computing, 31 (2007), pp. 237–271.
- [36] J. QIAN, Y.-T. ZHANG, AND H.-K. ZHAO, *Fast sweeping methods for Eikonal equations on triangular meshes*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 83–107, <https://doi.org/10.1137/050627083>.
- [37] F. QIN, Y. LUO, K. B. OLSEN, W. CAI, AND G. T. SCHUSTER, *Finite-difference solution of the Eikonal equation along expanding wavefronts*, Geophysics, 57 (1992), pp. 478–487.
- [38] E. ROUY AND A. TOURIN, *A viscosity solutions approach to shape-from-shading*, SIAM Journal on Numerical Analysis, 29 (1992), pp. 867–884, <https://doi.org/10.1137/0729053>.
- [39] J. SCHÖBERL, *NETGEN an advancing front 2D/3D-mesh generator based on abstract rules*, Computing and Visualization in Science, 1 (1997), pp. 41–52, <https://doi.org/10.1007/s007910050004>.
- [40] J. A. SETHIAN, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, 1996.
- [41] J. A. SETHIAN, *Fast marching methods*, SIAM Review, 41 (1999), pp. 199–235, <https://doi.org/10.1137/s0036144598347059>.
- [42] Y. WANG, W. HAO, AND G. LIN, *Two-level spectral methods for nonlinear elliptic equations with multiple solutions*, SIAM Journal on Scientific Computing, 40 (2018), pp. B1180–B1205.
- [43] Y. YANG AND W. HAO, *Convergence of a homotopy finite element method for computing steady states of Burgers’ equation*, ESAIM: Mathematical Modelling and Numerical Analysis, (2018).
- [44] L. YATZIV, A. BARTESAGHI, AND G. SAPIRO, *O(n) implementation of the fast marching algorithm*, Journal of Computational Physics, 212 (2006), pp. 393–399.
- [45] Y.-T. ZHANG, S. CHEN, F. LI, H. ZHAO, AND C.-W. SHU, *Uniformly accurate discontinuous Galerkin fast sweeping methods for Eikonal equations*, SIAM Journal on Scientific Computing, 33 (2011), pp. 1873–1896, <https://doi.org/10.1137/090770291>.
- [46] Y.-T. ZHANG, H.-K. ZHAO, AND J. QIAN, *High order fast sweeping methods for static Hamilton-*

- Jacobi equations*, Journal of Scientific Computing, 29 (2006), pp. 25–56, <https://doi.org/10.1007/s10915-005-9014-3>.
- [47] H. ZHAO, *A fast sweeping method for Eikonal equations*, Math. Comput., 74 (2005), pp. 603–627.
- [48] H. ZHAO, *The Fast Sweeping Method for Stationary Hamilton–Jacobi Equations*, in Handbook of Numerical Analysis, vol. 17, Elsevier, 2016, pp. 585–601, <https://doi.org/10.1016/bs.hna.2016.09.012>.