Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.OA-2019-0126

Vol. **13**, No. 3, pp. 620-643
August 2020

A Simple Semi-Implicit Scheme for Partial Differential Equations with Obstacle Constraints

Hao Liu[1,*] and Shingyu Leung[2]

[1] *School of Mathematics, Georgia Institute of Technology, 686 Cherry Street, Atlanta, GA 30332-0160, USA*
[2] *Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong*

Received 20 August 2019; Accepted (in revised version) 8 December 2019

Abstract. We develop a simple and efficient numerical scheme to solve a class of obstacle problems encountered in various applications. Mathematically, obstacle problems are usually formulated using nonlinear partial differential equations (PDE). To construct a computationally efficient scheme, we introduce a time derivative term and convert the PDE into a time-dependent problem. But due to its nonlinearity, the time step is in general chosen to satisfy a very restrictive stability condition. To relax such a time step constraint when solving a time dependent evolution equation, we decompose the nonlinear obstacle constraint in the PDE into a linear part and a nonlinear part and apply the semi-implicit technique. We take the linear part implicitly while treating the nonlinear part explicitly. Our method can be easily applied to solve the fractional obstacle problem and min curvature flow problem. The article will analyze the convergence of our proposed algorithm. Numerical experiments are given to demonstrate the efficiency of our algorithm.

AMS subject classifications: 65N06, 65N12, 35J60

Key words: Numerical methods, nonlinear elliptic equations, obstacle problem, semi-implicit scheme.

1. Introduction

In this paper, we develop efficient semi-implicit schemes to a class of obstacle problems as stated as follow [2, 5–7, 9, 25]. For a given energy functional $E(u)$, we determine $u \in K$ such that $E(u) = \inf_{v\in K} E(v)$ for some $K = \{v \in H^1 | v \geq \psi \text{ in } \Omega,\ v = g \text{ on } \partial\Omega\}$. The function ψ is a given obstacle function, Ω is the computational domain and g is the boundary condition. This class of problems can be found in various fields including the classical problem of elastic membrane modeling, pricing model in

[*] Corresponding author. *Email addresses:* hao.liu@math.gatech.edu (H. Liu), masyleung@ust.hk (S. Leung)

financial mathematics, porus media computations, computing the torsion of an elastic-plastic cylinder, Stefan problems for crystal growth simulation, min curvature flow in image processing [20], and etc. For example, in the problem of elastic membrane constrained on obstacle, the potential energy $E(u)$ is given by

$$E(u)=\int_\Omega \frac{1}{2}|\nabla u|^2 - fud\mathbf{x}, \tag{1.1}$$

where f is external force on u. For the minimal surface obstacle problem, the potential energy is proportional to its surface area and it leads to the energy functional

$$E(u)=\int_\Omega \sqrt{1+|\nabla u|^2} - fud\mathbf{x}.$$

One possible numerical approach to this class of problems is the projected relaxation method [10, 28] which first reformulates the problem using elliptic variational inequalities [13]. This class of methods is easy to implement and is proven to be convergent. However, its convergence speed depends on the relaxation parameter and the convergent might be slow in practice. To accelerate the algorithms, the multigrid method has been adopted as discussed in [1, 13, 17, 30].

Another way to solve the obstacle problem is via the optimization formulation. In [16], a Langrange multiplier is used to incorporate the constraint in the functional. In [24], a penalty term is introduced in the functional to encourage the solution to satisfy the constraint. The solution obtained by this method is not exact and the penalty parameter needs to be very small, of $\mathcal{O}(h^{-2})$. In [27], an L^1 penalty is added to the functional to relax the constraint of the obstacle. The equivalence of their formulation to (1.1) is proven [8, 21]. A related splitting Bregman algorithm has recently been implemented in [12, 27]. Note that the efficiency of this method depends on the application and also on the choice of the parameters. For linear problems, i.e., when the operator A is linear, this method converges very fast. But for nonlinear problems, i.e., in cases when we do not have any fast algorithm to invert A, the overall algorithm can be less efficient. Anther constraint approach has been developed in [29,31] which iteratively identifies the subdomain where the constraint is active. For the region where the constraint is inactive, the method recomputes the solution to the corresponding Euler-Lagrange equation of the functional. An augmented Lagrangian active set method has been proposed in [19] which takes advantage of the primal-dual formulation of the discretized obstacle problem. In [33], a primal-dual hybrid gradient method is also developed to solve the obstacle problem. Since the starting point of most of these optimization methods is an energy form, they might not be able to easily extend to fractional obstacle problems. Moreover, those methods are designed to solve an optimization problem, they cannot be applied to flow problems where intermediate steps contain the time evolution of the solution such as the min curvature flow problem.

In this work, we determine the minimizer of the variational problem by solving the corresponding Euler-Lagrange equation. In particular, we solve

$$\min(Au-f,u-\psi)=0 \quad \text{on} \quad \Omega \tag{1.2}$$

for some operator A depending on the energy to be minimized, f is some source term and the function ψ defines the obstacle. For example, we have $Au = -\Delta u$ for the elastic membrane problem and

$$Au = -\nabla \cdot \left(\frac{\nabla u}{\sqrt{1 + |\nabla u|^2}} \right) \tag{1.3}$$

for the minimal surface problem. Another interesting example is the fractional obstacle problem when $A = -\Delta^{\alpha/2} u$, $0 < \alpha < 2$. The solution to this fractional obstacle problem is the minimizer of

$$E(u) = \int_{\mathbb{R}} \int_{\mathbb{R}} \frac{|u(x) - u(y)|^2}{|x - y|^{n+\alpha}} dxdy$$

from all u such that $\psi \leq u$. This problem arises from the fractional porous media equation [3, 4]. The approach seems natural and straightforward in the first place. However, even if the operator A is linear as in the elastic membrane problem, the resulting equation is in fact a nonlinear one and it is in general difficult to directly obtain a good numerical solution. One simple numerical strategy is to introduce an artificial time and solve the resulting gradient descent equation until the steady state. The resulting equation is a time dependent partial differential equation (PDE) and, in practice, one might need to impose a strict CFL condition to obtain a stable time evolution solution. It is therefore important to design a numerically efficient algorithm to relax such stability condition.

In this paper, we propose a simple semi-implicit scheme which can handle general obstacle problems in the form of (1.2) while can allow a relatively large time step. Semi-implicit schemes are widely used to relax the strict stability condition on the size of the time-marching step in solving time-dependent PDE's. One simple example is the computations of the curvature motion using the level set method governed by

$$\phi_t = |\nabla \phi| \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) = |\nabla \phi| \kappa, \tag{1.4}$$

where $\kappa = \nabla \cdot (\nabla \phi / |\nabla \phi|)$ is the mean curvature [23]. This nonlinear equation can of course be easily solved by the simple explicit scheme

$$\frac{\phi^{k+1} - \phi^k}{\Delta t} = |\nabla \phi^k| \nabla \cdot \left(\frac{\nabla \phi^k}{|\nabla \phi^k|} \right).$$

However, because the curvature term involves the second derivative of the level set function, the time step constraint for this explicit scheme is of order $\Delta t = \mathcal{O}(\Delta x^2)$. This results in a computationally inefficient numerical method. To relax this time step restriction, [26] has proposed a semi-implicit scheme by first adding-and-subtracting the Laplacian of the level set function to the evolution equation, i.e.,

$$\phi_t = \Delta \phi - (\Delta \phi - |\nabla \phi| \kappa) = \Delta \phi - \mathbf{n} \cdot \nabla (|\nabla \phi|).$$

Then, following the idea of typical semi-implicit methods, the paper proposes to treat the linear part of of this equation implicitly and the nonlinear part explicitly, and finally obtains $\phi^{k+1} = (I - \Delta t \Delta)^{-1}[\phi^k - \Delta t \mathcal{N}(\phi^k)]$ with $\Delta t \gg \mathcal{O}(\Delta x^2)$. The phase-field community has also applied a similar semi-implicit idea to stabilize the evolution of the Cahn-Hilliard equation. For example, [15] has proposed to first add-and-subtract a term $\mathcal{O}(\Delta\phi)$ to the PDE and then treats one of them implicitly while the other one explicitly. Stability and convergence of the approach have also been discussed. As demonstrated in some other examples, the term added-and-substracted from the equation does not necessarily to be a linear one. In [32], we have proposed a different semi-implicit method to approximate the solution of (1.4). Instead of extracting a linear elliptic term, we have proposed to extract a curvature term from the evolution equation, i.e., we consider

$$\phi_t = \beta \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \beta \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + |\nabla \phi| \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right), \tag{1.5}$$

for some constant $\beta > 0$. Numerically, we can treat the first curvature term implicitly and the rest explicitly. At each step, this corresponding update formula can be reformulated as a convex optimization problem and can be solved efficiently using any recently developed fast algorithms such as [11, 12, 14, 22].

This paper proposes a simple strategy to extract a simple linear approximation of the nonlinear obstacle constraint in the PDE. Such simple linear part will be treated implicitly while the nonlinear part will be taken care of explicitly. The proposed semi-implicit schemes will be given in Section 2.1. In Section 2.2 we will discuss the convergence of the numerical algorithms. Various numerical examples will be given in Section 3 to demonstrate the efficiency of the proposed algorithms.

2. The semi-implicit method

2.1. The proposed schemes

In this work, we consider the Euler-Lagrange formulation of the obstacle problem by solving the nonlinear equation (1.2). We consider the steady state of the following time dependent PDE:

$$u_t + \min(u - \psi, Au - f) = 0. \tag{2.1}$$

The simplest way to update the evolution equation is to use the explicit Euler method given by

$$u^{k+1} = u^k - \Delta t \min(u^k - \psi, Au^k - f)$$

for some time step $\Delta t > 0$. Such constant should depends on the operator A. For the elastic membrane and minimal surface problems, the corresponding stability constraints due to this CFL condition are both given by $\mathcal{O}(h^2)$ with h representing the mesh size. To improve the efficiency by a semi-implicit scheme, we rewrite the equation using the identity

$$\min(a,b)=\frac{a+b}{2}-\frac{|a-b|}{2}$$

and obtain

$$u_t+\frac{(u-\psi)+(Au-f)}{2}-\frac{|(u-\psi)-(Au-f)|}{2}=0. \tag{2.2}$$

When A is a linear operator, we can easily apply the semi-implicit idea and obtain the following update formula

$$u^{k+1}=\left[\left(1+\frac{2}{\Delta t}\right)I+A\right]^{-1}\left[\frac{2}{\Delta t}u^k+\left|(I-A)u^k-\psi+f\right|+\psi+f\right]. \tag{2.3}$$

In the case when A is nonlinear, we linearlize it using $A(u^{k+1})=B_{u^k}u^{k+1}$ and it leads to

$$u^{k+1}=\left[\left(1+\frac{2}{\Delta t}\right)I+B_{u^k}\right]^{-1}\left[\frac{2}{\Delta t}u^k+\left|(I-B_{u^k})\,u^k-\psi+f\right|+\psi+f\right]. \tag{2.4}$$

For most time dependent PDE's, the efficiency and convergence rate depend on the choice of time step. One always prefer a stable scheme under a relatively large time step. In the extreme case when $\Delta t\to+\infty$, these Δt-depending updating formulas reduce to

$$u^{k+1}=(I+A)^{-1}\left[\left|(I-A)u^k-\psi+f\right|+\psi+f\right] \tag{2.5}$$

and

$$u^{k+1}=(I+B_{u^k})^{-1}\left[\left|(I-B_{u^k})\,u^k-\psi+f\right|+\psi+f\right] \tag{2.6}$$

for the linear and nonlinear operator A, respectively.

2.2. Convergence analysis

In this section, we give a simple convergence analysis of our proposed schemes. For simplicity, we consider the discretized versions of the above numerical methods applied to the one dimensional case. Instead of introducing a new notation, we keep A as the matrix representing the discretization of the differential operator A.

Theorem 2.1. *Let $\{\lambda_i : i=1,\cdots,n\}$ be the set of eigenvalues of the matrix A. We have*

$$\left\|u^{k+1}-u^k\right\|\le C\left\|u^k-u^{k-1}\right\|$$

with the positive constant C given by

$$\frac{1+\frac{\Delta t}{2}\max_i|1-\lambda_i|}{\min_i\{\left|1+\frac{\Delta t}{2}(1+\lambda_i)\right|\}}\quad \textit{and}\quad \frac{\max_i|1-\lambda_i|}{\min_i|1+\lambda_i|}$$

for schemes (2.3) *and* (2.5)*, respectively. Scheme* (2.3) *converges if*

$$1+\frac{\Delta t}{2}\max_i|1-\lambda_i|<\min_i\left\{\left|1+\frac{\Delta t}{2}(1+\lambda_i)\right|\right\}.$$

Scheme (2.5) *converges if* $\max_i|1-\lambda_i|<\min_i|1+\lambda_i|$.

Proof. Denote

$$M_{\Delta t}=\left(1+\frac{2}{\Delta t}\right)I+A \quad \text{and} \quad N=I-A.$$

Then the updating formula of scheme (2.3) can be written as

$$u^{k+1}=M_{\Delta t}^{-1}\left(\frac{2}{\Delta t}u^k+\left|Nu^k+f-\psi\right|+f+\psi\right),$$

which implies

$$u^{k+1}-u^k=M_{\Delta t}^{-1}\left[\frac{2}{\Delta t}\left(u^k-u^{k-1}\right)+\left|Nu^k+f-\psi\right|-\left|Nu^{k-1}+f-\psi\right|\right],$$

and therefore

$$\begin{aligned}\left\|u^{k+1}-u^k\right\|\leq&\|M_{\Delta t}^{-1}\|\left\|\frac{2}{\Delta t}\left(u^k-u^{k-1}\right)+\left|Nu^k+f-\psi\right|-\left|Nu^{k-1}+f-\psi\right|\right\|\\ \leq&\frac{2}{\Delta t}\|M_{\Delta t}^{-1}\|\left\|u^k-u^{k-1}\right\|+\|M_{\Delta t}^{-1}\|\left\|\left|Nu^k+f-\psi\right|-\left|Nu^{k-1}+f-\psi\right|\right\|.\end{aligned}$$

Since

$$\left|\left|Nu^k+f-\psi\right|-\left|Nu^{k-1}+f-\psi\right|\right|\leq\left|N\left(u^k-u^{k-1}\right)\right|,$$

we have

$$\begin{aligned}&\left\|\left|Nu^k+f-\psi\right|-\left|Nu^{k-1}+f-\psi\right|\right\|\\ \leq&\left\|\left|N\left(u^k-u^{k-1}\right)\right|\right\|=\left\|N\left(u^k-u^{k-1}\right)\right\|\leq\|N\|\left\|u^k-u^{k-1}\right\|.\end{aligned}$$

This leads to

$$\left\|u^{k+1}-u^k\right\|\leq\|M_{\Delta t}^{-1}\|\left(\frac{2}{\Delta t}+\|N\|\right)\left\|u^k-u^{k-1}\right\|. \tag{2.7}$$

Since the eigenvalues of $M_{\Delta t}$ are given by $\left(1+\frac{2}{\Delta t}+\lambda_i\right)$ for $i=1,\cdots,n$, we have

$$\|M_{\Delta t}^{-1}\|\leq\frac{1}{\min_i\left\{\left|\frac{2}{\Delta t}+1+\lambda_i\right|\right\}}.$$

Similarly, since $N = I - A$, we have $\|N\| \le \max_i \{|1-\lambda_i|\}$. Substituting the relations of $\left\|M_{\Delta t}^{-1}\right\|$ and $\|N\|$ into Eq. (2.7), we get

$$\begin{aligned}\left\|u^{k+1}-u^k\right\| \le& \left(\frac{1}{\min_i\left\{\left|1+\frac{\Delta t}{2}(1+\lambda_i)\right|\right\}}\right)\left(1+\frac{\Delta t}{2}\max_i\{|1-\lambda_i|\}\right)\left\|u^k-u^{k-1}\right\| \\ =&\frac{1+\frac{\Delta t}{2}\max_i\{|1-\lambda_i|\}}{\min_i\left\{\left|1+\frac{\Delta t}{2}(1+\lambda_i)\right|\right\}}\left\|u^k-u^{k-1}\right\| = C_1\left\|u^k-u^{k-1}\right\|\end{aligned}$$

with

$$C_1 = \frac{1+\frac{\Delta t}{2}\max_i\{|1-\lambda_i|\}}{\min_i\left\{\left|1+\frac{\Delta t}{2}(1+\lambda_i)\right|\right\}}.$$

If

$$1+\frac{\Delta t}{2}\max_i|1-\lambda_i| < \min_i\left\{\left|1+\frac{\Delta t}{2}(1+\lambda_i)\right|\right\},$$

it holds

$$C_1 = \frac{1+\frac{\Delta t}{2}\max_i\{|1-\lambda_i|\}}{\min_i\left\{\left|1+\frac{\Delta t}{2}(1+\lambda_i)\right|\right\}} < 1.$$

The scheme converges.

The proof for scheme (2.5) is similar. Denote $I+A$, $I-A$ by M and N respectively. The updating formula (2.5) can be written as

$$u^{k+1} = M^{-1}\left(\left|Nu^k-\psi+f\right|+\psi+f\right).$$

We have

$$u^{k+1}-u^k = M^{-1}\left(\left|Nu^k-\psi+f\right|-\left|Nu^{k-1}-\psi+f\right|\right)$$

and

$$\left\|u^{k+1}-u^k\right\| \le \|M^{-1}\|\left\|\left|Nu^k-\psi+f\right|-\left|Nu^{k-1}-\psi+f\right|\right\|. \tag{2.8}$$

Focusing on the second factor in Eq. (2.8), we have

$$\left|\left|Nu^k-\psi+f\right|-\left|Nu^{k-1}-\psi+f\right|\right| \le \left|N\left(u^k-u^{k-1}\right)\right|,$$

and therefore

$$\left\|\left|Nu^k-\psi+f\right|-\left|Nu^{k-1}-\psi+f\right|\right\| \le \left\|\left|N\left(u^k-u^{k-1}\right)\right|\right\| = \left\|N\left(u^k-u^{k-1}\right)\right\|.$$

Substituting this to (2.8), we obtain

$$\left\|u^{k+1}-u^k\right\| \le \|M^{-1}\|\left\|N\left(u^k-u^{k-1}\right)\right\| \le \|M^{-1}\|\,\|N\|\left\|u^k-u^{k-1}\right\|.$$

Since $M = I + A$ and $N = I - A$, we have

$$\left\|M^{-1}\right\| \leq \left(\min_j \{|1+\lambda_j|\}\right)^{-1} \quad \text{and} \quad \|N\| \leq \max_i \{|1-\lambda_i|\}.$$

Denote $\frac{\max_i\{|1-\lambda_i|\}}{\min_j\{|1+\lambda_j|\}}$ by C_2, we have

$$\left\|u^{k+1} - u^k\right\| \leq C_2 \left\|u^k - u^{k-1}\right\|.$$

If $\max_i \{|1-\lambda_i|\} < \min_j \{|1+\lambda_j|\}$, we have $C_2 = \max_i \{|1-\lambda_i|\}/\min_j \{|1+\lambda_j|\} < 1$ and therefore the scheme (2.5) converges. □

For scheme (2.5), the convergence condition fully depends on the eigenvalues of matrix A. Denote the smallest and largest eigenvalues as $\lambda_{\min}$, $\lambda_{\max}$, respectively. For a special case that $\lambda_{\min} > 0$ and $\lambda_{\max} - \lambda_{\min} < 2$, the condition is satisfied. For scheme (2.3), the condition and convergence rate depend on both the eigenvalues and time step. If $\lambda_{\min} > 0$ and $\lambda_{\max} - \lambda_{\min} < 2$,

$$\min_i \left\{\left|1 + \frac{\Delta t}{2}(1+\lambda_i)\right|\right\} = 1 + \frac{\Delta t}{2} \min_i \{(1+\lambda_i)\}.$$

Then

$$\frac{1 + \frac{\Delta t}{2} \max_i \{(1-\lambda_i)\}}{\min_i \left\{\left|1 + \frac{\Delta t}{2}(1+\lambda_i)\right|\right\}}$$

is a strictly decreasing function of Δt and is smaller than 1 for any $\Delta t > 0$. One can choose Δt as large as possible to get faster convergence rate. As $\Delta t \to +\infty$, the convergence rate convergence to $\max_i\{1-\lambda_i\}/\min_i\{1+\lambda_i\}$, the convergence rate of scheme (2.5).

For the case when A is nonlinear and can be linearized as $A(u) = B_u u$, the updating formula is quite similar to the linear A case. We just replace A in both schemes (2.3) and (2.5) by B_{u^k} to get:

$$u^{k+1} = \left[I + \frac{\Delta t}{2}(I + B_{u^k})\right]^{-1} \left[\frac{\Delta t}{2}\left|\left(I - B^n_{u^k}\right)u^k - \psi + f\right| + \frac{\Delta t}{2}(\psi + f)\right], \tag{2.9}$$

and

$$u^{k+1} = (I + B_{u^k})^{-1} \left[\left|(I - B_{u^k})\,u^k - \psi + f\right| + \psi + f\right]. \tag{2.10}$$

Their convergence are given by the following theorem.

Theorem 2.2. *Let n denote the number of eigenvalues of B_{u^k} and denote all eigenvalues by λ_i^k, $i = 1, \cdots, n$. We have*

$$\left\|u^{k+1} - u^k\right\| \leq C_1^k \left\|u^k - u^{k-1}\right\| \quad \textit{and} \quad \left\|u^{k+1} - u^k\right\| \leq C_2^k \left\|u^k - u^{k-1}\right\|$$

for (2.3) *and* (2.5)*, where*

$$C_1^k = \frac{1+\frac{\Delta t}{2}\max_i\left\{\left|1-\lambda_i^k\right|\right\}}{\min_i\left\{\left|1+\frac{\Delta t}{2}(1+\lambda_i^k)\right|\right\}} \quad \textit{and} \quad C_2^k = \frac{\max_i\left\{\left|1-\lambda_i^k\right|\right\}}{\min_j\left\{\left|1+\lambda_j^k\right|\right\}},$$

respectively. Scheme (2.9) *converges if*

$$1+\frac{\Delta t}{2}\max_i\left\{\left|1-\lambda_i^k\right|\right\} < \min_i\left\{\left|1+\frac{\Delta t}{2}(1+\lambda_i)\right|\right\}$$

for all u^k. Scheme (2.10) *converges if*

$$\max_i\left\{\left|1-\lambda_i^k\right|\right\} < \min_j\left\{\left|1+\lambda_j^k\right|\right\}$$

for all u^k.

The proof of this theorem is similar to that of the previous one and is omitted here. Note that since the condition depends on B_{u^k} and therefore u^k, we are not able to provide a more relaxed condition on the convergence of the iteration. Having said that, the above schemes still work well for many applications as will be demonstrated later in the example section. For cases that the conditions in the theorem are indeed satisfied, the numerical convergence speed is in fact better than the estimates as stated in the above theorems.

If we compare the expressions between C_1 and C_2 in the above two theorems, we can see that as the time step goes to infinity, the constant C_1 approaches to C_2. Similar property is observed also for the constants C_1^k and C_2^k. In other words, the schemes (2.3) and (2.9) behave similar to (2.5) and (2.10) respectively as Δt goes to infinity. This behaviour in convergence speed related to time step is also observed in our numerical experiments. In some experiments we monitor the residual error instead of the error, similar convergence behavior is also observed.

3. Numerical experiements

In this section, we test our schemes for different applications. In all experiments, we set the external force $f = 0$. In these discretized obstacle problems, the residual matrix **r** is defined as $r_{i,j} = \min(Au_{i,j} - f, u_{i,j} - \psi_{i,j})$ and the residual error is defined as

$$e_{resid} = \sqrt{h\sum_i r_i^2} \quad \text{and} \quad e_{resid} = \sqrt{h^2\sum_{i,j} r_{i,j}^2}$$

for the 1D and 2D cases, respectively. For those examples whose exact solution u_{exact} is known, we compute the error as $e = \|u - u_{exact}\|$. L_∞ norm is used if not stated otherwise. For all experiments, we take our initial condition as the shape of the obstacle.

3.1. A membrane constrained obstacle problem

For the membrane constrained obstacle problem, A is the negative Laplacian operator which is linear. The corresponding Euler-Lagrange equation can be written as:

$$\min(-\Delta u - f, u - \psi) = 0.$$

We denote the mesh size by h and use the standard central difference to approximate Au.

3.1.1. One dimensional cases

Here, we test two examples with the zero Dirichlet boundary condition given in [27, 31]. The first obstacle is given by

$$\psi_1(x) = \begin{cases} 100x^2 & \text{for} \quad 0 \le x \le 0.25, \\ 100x(1-x) - 12.5 & \text{for} \quad 0.25 \le x \le 0.5, \\ \psi_1(1-x) & \text{for} \quad 0.5 \le x \le 1.0, \end{cases}$$

with exact solution

$$u_{1,exact}(x) = \begin{cases} (100 - 50\sqrt{2})x & \text{for} \quad 0 \le x \le \dfrac{1}{2\sqrt{2}}, \\ 100x(1-x) - 12.5 & \text{for} \quad \dfrac{1}{2\sqrt{2}} \le x \le 0.5, \\ u_{1,exact}(1-x) & \text{for} \quad 0.5 \le x \le 1.0. \end{cases}$$

Fig. 1 shows the result by schemes (2.5) and (2.3) with different size of the time steps. For comparison, we also provide the plots of the error and the residual in Figs. 1(b) and (c), respectively. We observe that the explicit scheme has difficulty in converging to the exact solution for a relatively large time step given by $\Delta t = h^2$ (the orange solid line with plus signs). The residual in the solution seems to get stuck with no improvement in the accuracy. When we increase the step size to $2h^2$, the iteration is unstable and the solution diverges. The convergence is improved when we reduce the time step to $\Delta t = 0.5h^2$ (the blue solid line with circles). Concerning the scheme (2.3) with a small time step, we find that the method behaves similar like an explicit scheme (the yellow solid line with cross signs overlaps with the blue solid line with circles). But since the semi-implicit scheme allows one to use a significantly large time step, we are able to converge to the steady state solution in a faster speed. As we can see from Fig. 1(b), the solution converges much faster in the first few hundred iterations when we increase the size from $\Delta t = h^2$ to $5h$. The accuracy of scheme (2.5) does not seem to be as good as some other results. But the scheme (2.5) still converges to the exact solution as we refine the mesh as shown in Fig. 2.

Now, we consider the number of iterations required for different schemes to achieve a residual error of 10^{-3} on different grids, as shown in Table 1. Our semi-implicit

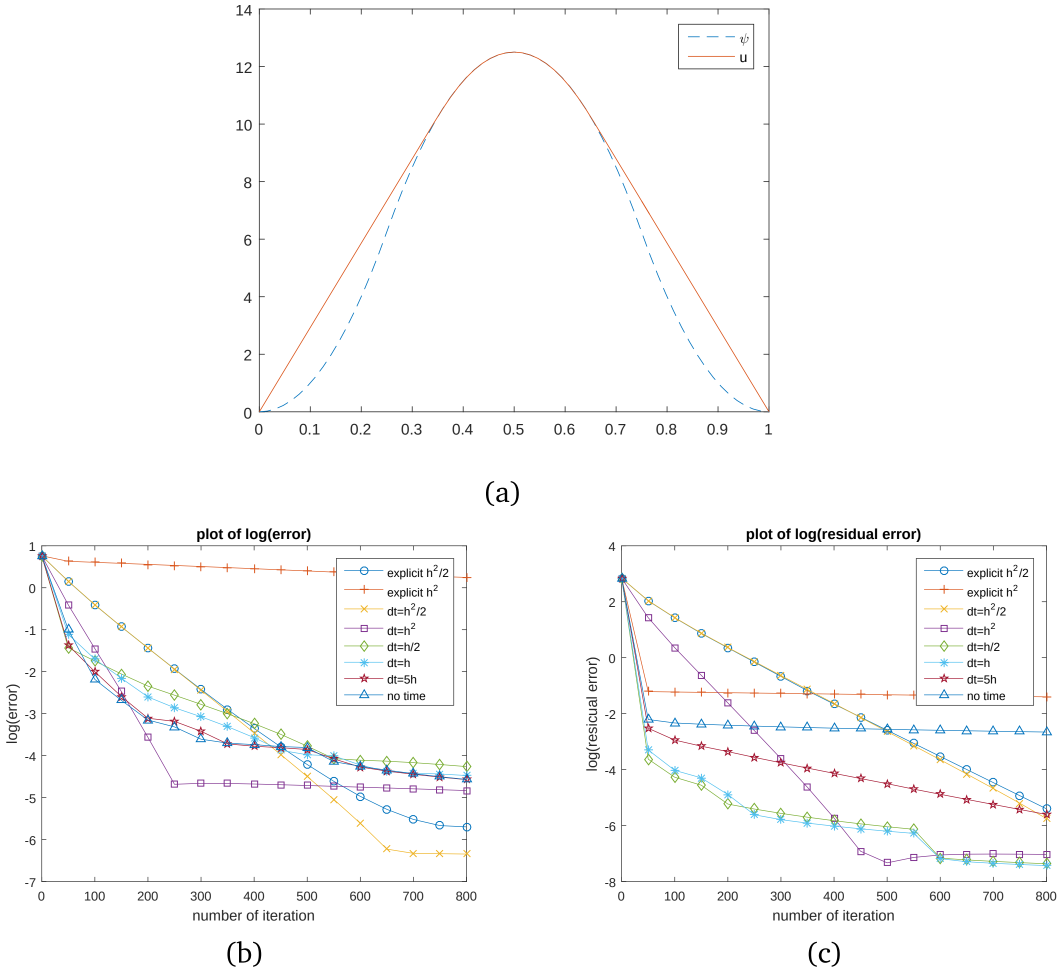


Figure 1: (Example 3.1.1 with the first obstacle ψ_1) The number of grids is 64. (a) The result by scheme (2.5) after 1000 iterations. (b) The log(error) of scheme (2.5) and scheme (2.3) with different time steps with respect to the number of iterations. (c) The residual error.

Table 1: (Example 3.1.1 with the first obstacle ψ_1) The number of iterations required by different schemes to achieve a residual error of 10^{-3}.

	h	1/64	1/128	1/256	1/512
$\Delta t = 0.5h^2$	Semi	898	3630	13959	53788
	Explicit	961	3626	13977	53760
$\Delta t = h^2$	Semi	450	1816	6987	26891
$\Delta t = h$	Semi	562	546	619	751
$\Delta t = 5h$	Semi	1191	2036	3148	4777
$\Delta t = 10h$	Semi	2083	4062	7463	11741
Without time	Semi	30958	109896	418497	1154881

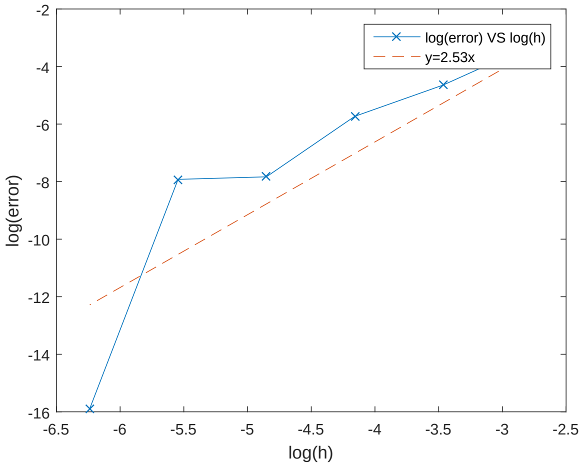


Figure 2: (Example 3.1.1 with the first obstacle ψ_1) The plot of error with respect to h with the numerical solution obtained by scheme (2.5).

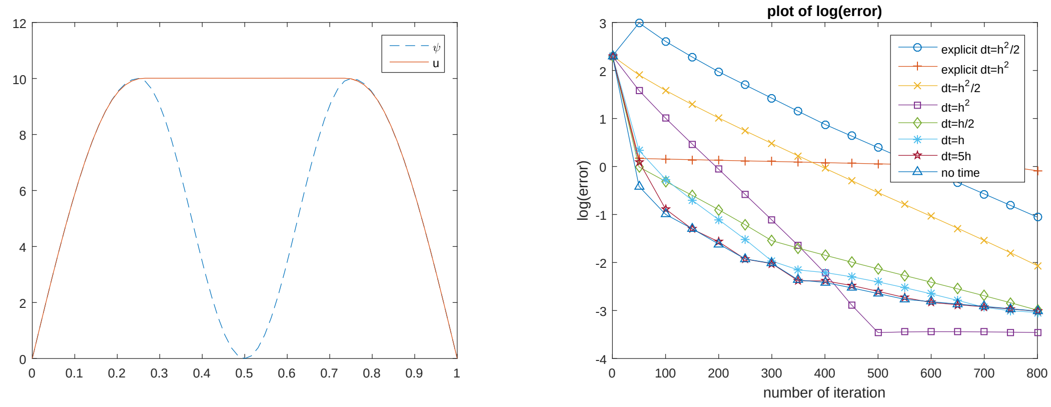
Table 2: (Example 3.1.1 with the first obstacle ψ_1) The corresponding L_1 error when schemes achieve a residual error of 10^{-3}.

	h	1/64	1/128	1/256	1/512
$\Delta t = 0.5h^2$	Semi	3.67E-2	4.00E-3	1.04E-4	1.92E-4
	Explicit	1.25E-3	2.49E-4	2.84E-4	2.17E-4
$\Delta t = h^2$	Semi	3.43E-3	9.05E-4	1.24E-4	1.77E-4
$\Delta t = h$	Semi	5.88E-3	1.08E-2	1.21E-2	1.07E-2
$\Delta t = 5h$	Semi	3.09E-3	1.63E-3	8.60E-4	6.40E-4
$\Delta t = 10h$	Semi	5.48E-4	4.62E-4	2.78E-4	2.58E-4
Without time	Semi	1.12E-3	1.41E-4	1.30E-4	3.21E-7

Table 3: (Example 3.1.1 with with the first obstacle ψ_1) Number of iterations required to achieve L_1 error less than h.

	h	1/128	1/256	1/512
$\Delta t = 0.5h^2$	Explicit	1874	8663	39171
	Semi	1852	8593	39095
$\Delta t = h^2$	Semi	918	4272	19506
$\Delta t = 5h^2$	Semi	173	826	3827
$\Delta t = 20h^2$	Semi	963	198	919
$\Delta t = 5h$	Semi	502	1006	1941
$\Delta t = 10h$	Semi	477	963	1820
Without time	Semi	447	907	1656

method with $\Delta t = h$ can efficiently reduce the number of iterations. Table 2 gives the corresponding L_1 error to the exact solution when achieve this residual error accuracy. Although our semi-implicit scheme with $\Delta t = h$ has a larger error, a slightly more number of iterations can already effectively reduce the error to the same order as that



(a) (b)

Figure 3: (Example 3.1.1 with the second obstacle ψ_2) The computational mesh has 64 points. (a) The result by scheme (2.5) after 1000 iterations. (b) The log-error of scheme (2.5) with respect to the number of iterations.

Table 4: (Example 3.1.1 with the first obstacle ψ_1) The L_1 error by different schemes after 1000 iterations.

	h	1/128	1/256	1/512
$\Delta t = 0.5h^2$	Explicit	6.47E-2	4.25E-1	7.13E-1
	Semi	6.43E-2	4.25E-1	7.13E-1
$\Delta t = h^2$	Semi	5.00E-3	2.23E-1	5.97E-1
$\Delta t = 5h^2$	Semi	4.20E-3	1.07E-3	1.63E-1
$\Delta t = 20h^2$	Semi	7.59E-3	4.33E-3	1.07E-3
$\Delta t = 5h$	Semi	3.13E-3	3.94E-3	4.61E-3
$\Delta t = 10h$	Semi	2.95E-3	3.66E-3	4.16E-3
Without time	Semi	2.75E-3	3.34E-3	3.61E-3

of other schemes. In Table 3, we look at different schemes and obtain the number of iterations required so that the L_1 error is less than h. To achieve the same accuracy, for small Δt such as $\Delta t = 0.5h^2$ or $\Delta t = h^2$, there is no big difference among these schemes. As the time step goes larger, we can see that the number of iterations is significantly reduced. In Table 3, as we refine the mesh, the number of iterations seems to increase. But note that the stopping criterion (chosen as h) is more demanding as the mesh is refined. Next, we compare the convergence behavior of different schemes and various Δt's and h's with a fixed number of iterations. The L_1 error of each setting after 1000 iterations are shown in Table 4. For most choices of Δt's and h's, the resulting L_1 error is in the order of 10^{-3}. If Δt is small, the convergence rate is affected and is slower (the right upper part of Table 4). As long as Δt is large enough and for a fixed number of iterations, the convergence behavior of scheme (2.3) is not sensitive to the choice of Δt and h, and achieves error with similar order to that of scheme (2.5).

The second case has the following obstacle with two peaks:

$$\psi_2(x) = \begin{cases} 10\sin(2\pi x) & \text{for} \quad 0 \le x \le 0.25, \\ 5\cos(\pi(4x-1)) + 5 & \text{for} \quad 0.25 \le x \le 0.5, \\ \psi_2(1-x) & \text{for} \quad 0.5 \le x \le 1.0, \end{cases}$$

with the exact solution given by

$$u_{2,exact}(x) = \begin{cases} 10\sin(2\pi x) & \text{for} \quad 0 \le x \le 0.25, \\ 10 & \text{for} \quad 0.25 \le x \le 0.5, \\ u_{2,exact}(1-x) & \text{for} \quad 0.5 \le x \le 1.0. \end{cases}$$

The result by the scheme (2.5) and error behavior of different schemes are shown in Fig. 3. The numerical solution matches very well with the exact solution. We observe also that a larger time step provides faster convergence at the beginning meaning that the method can capture the macroscopic structure of the solution quickly in the first few hundred iterations.

3.1.2. A two dimensional case

We now consider the following two dimensional obstacle problem with the zero Dirichlet boundary condition where the obstacle is given by $\psi = \max\{0, 0.6 - 8|(x-0.5)^2 + (y-0.5)^2|\}$. In Fig. 4, we plot the obstacle, the result by the scheme (2.5) and the residual error behavior. To initialize the time-dependent evolution, we use the obstacle as the initial condition so that the initial residual error is quite large. But it damps significantly after even the first iteration. In Fig. 4, we show only the residual error after the second iteration.

3.2. A minimal surface obstacle problem

For the minimal surface obstacle problem, we write the problem in the HJB form $\min(Au - f, u - \psi) = 0$, where A is defined as in (1.3).

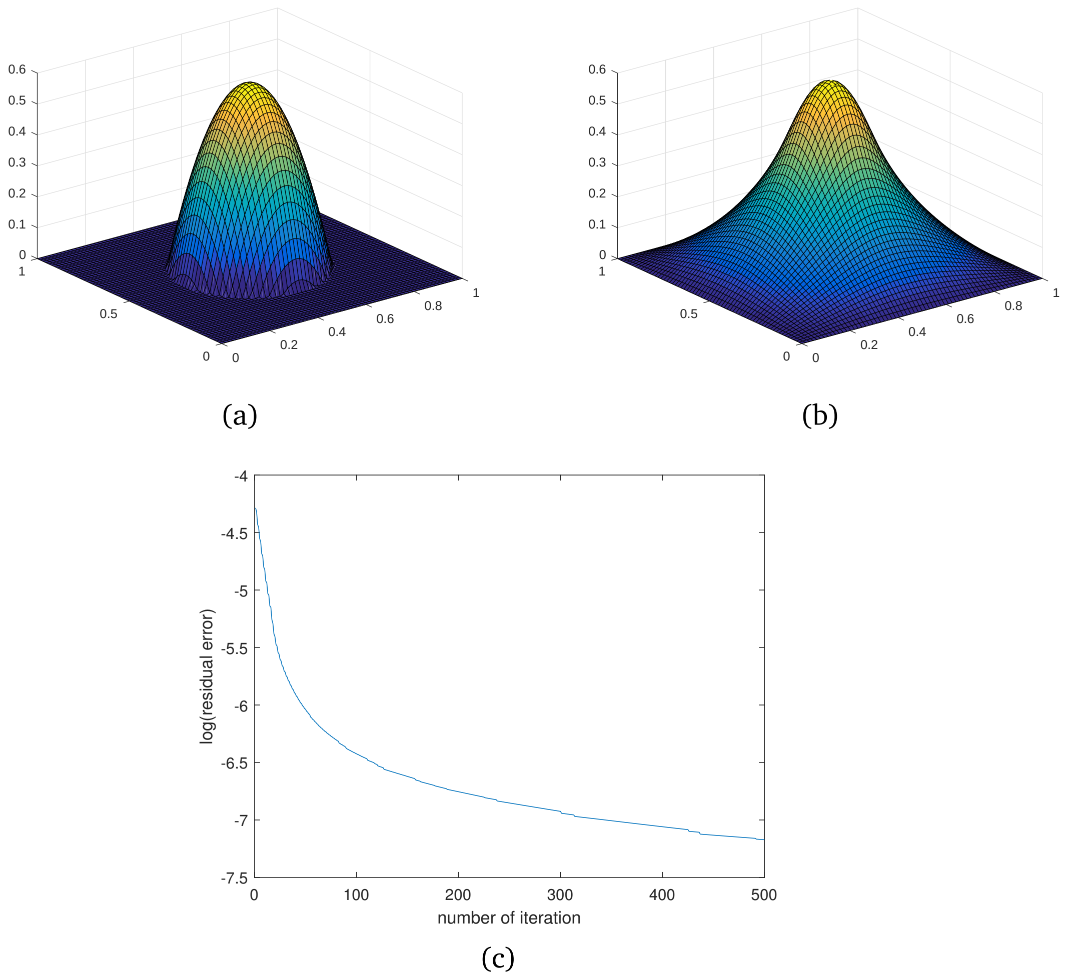
3.2.1. A one dimensional case

For the discretization in 1D case, to compute u^{k+1}, we discretize $(Au)_i$ as:

$$(Au)_i = \left[-\nabla \cdot \left(\frac{\nabla u}{\sqrt{1 + |\nabla u^k|^2}} \right) \right]_i = a_l u_{i-1} + a u_i + a_r u_{i+1},$$

with $a = -(a_l + a_r)$ and

$$a_l = -\frac{1}{h^2} \frac{1}{\sqrt{1 + ((u_i^k - u_{i-1}^k)/h)^2}} \quad \text{and} \quad a_r = -\frac{1}{h^2} \frac{1}{\sqrt{1 + ((u_{i+1}^k - u_i^k)/h)^2}}.$$



(a) (b) (c)

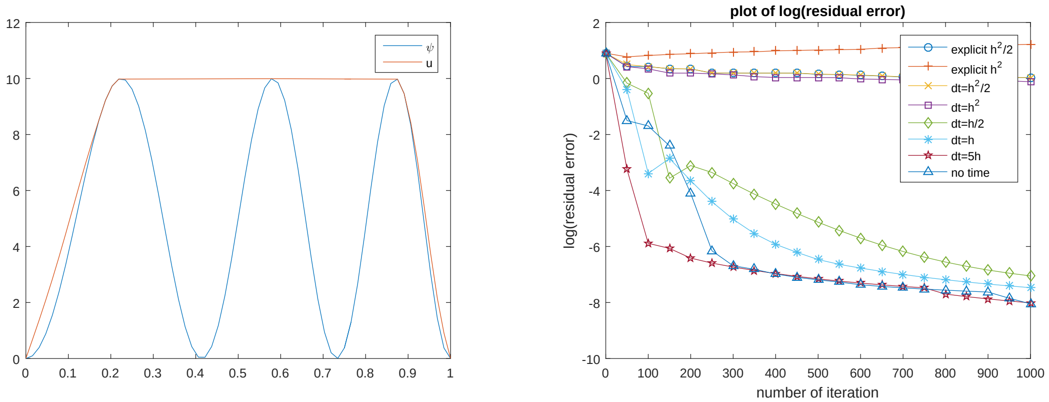
Figure 4: (Example 3.1.2) The domain is $[0,1]^2$ and the mesh is 64×64. (a) The shape of the obstacle. (b) The result by scheme (2.5) after 500 iterations. (c) The log residual error with respect to the number of iterations.

The obstacle is given by $\psi = 10\sin^2(\pi(x+1)^2)$ for $x\in[0,1]$. The result and residual error behavior of scheme (2.6) are shown in Fig. 5. The residual error of the explicit scheme with different time steps is also plotted for comparison. In this example, we can see the advantage of our scheme. The residual error of our scheme with a large time step reduces fast.

3.2.2. A two dimensional case

For the 2D case, we use the finite difference discretization [30]

$$Au_{i,j} = a_c u_{i,j} + a_w u_{i-1,j} +_e u_{i+1,j} + a_s u_{i,j-1} + a_n u_{i,j+1},$$



(a) (b)

Figure 5: (Example 3.2.1) The domain is $[0,1]$ and the grid size is 64. (a) The obstacle and the result by scheme (2.6). (b) The residual error with respect to the number of iterations.

where $a_c = 1 - (a_w + a_e + a_s + a_n)$ and

$$a_w = -\frac{1}{2h^2}\left[\frac{1}{\sqrt{(\frac{u_{i,j}-u_{i-1,j}}{h})^2 + (\frac{u_{i,j}-u_{i,j-1}}{h})^2 + 1}} + \frac{1}{\sqrt{(\frac{u_{i,j}-u_{i-1,j}}{h})^2 + (\frac{u_{i-1,j+1}-u_{i-1,j}}{h})^2 + 1}}\right],$$

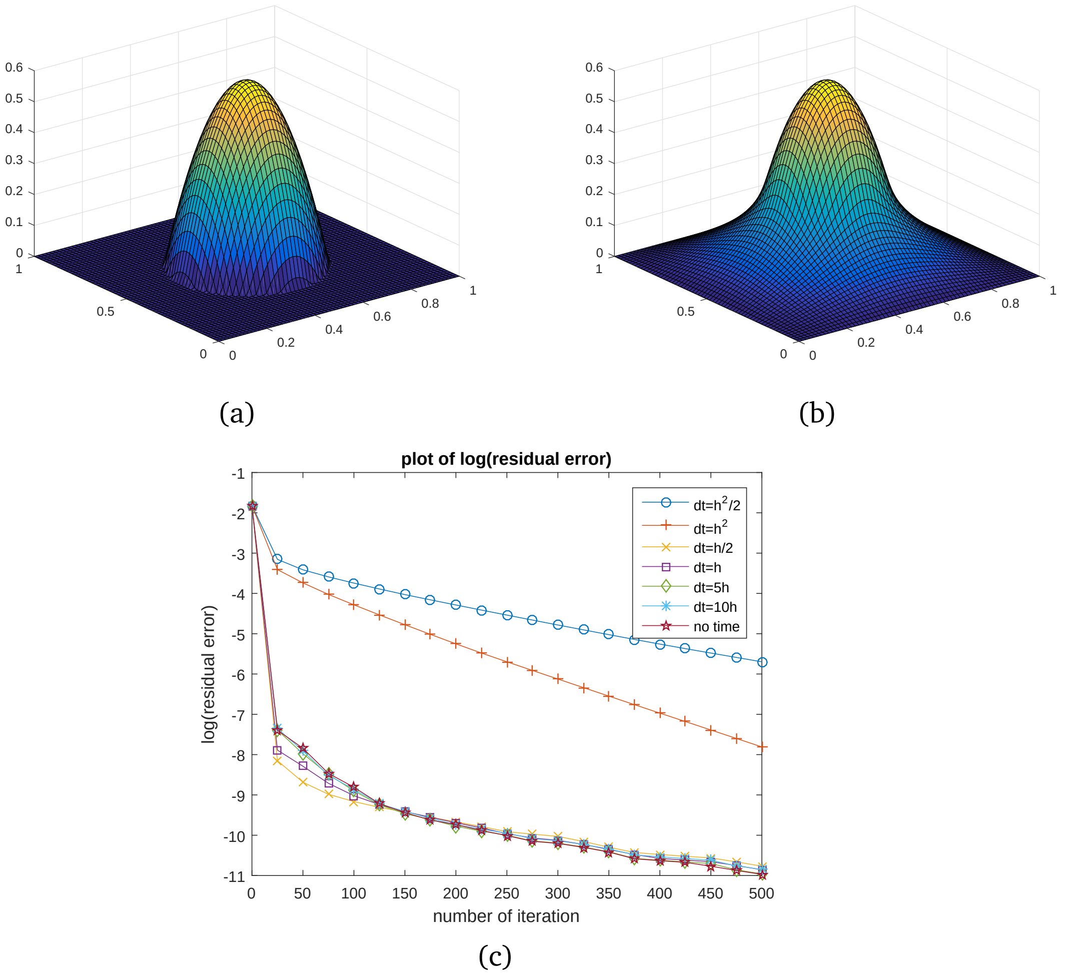
$$a_e = -\frac{1}{2h^2}\left[\frac{1}{\sqrt{(\frac{u_{i+1,j}-u_{i,j}}{h})^2 + (\frac{u_{i+1,j}-u_{i+1,j-1}}{h})^2 + 1}} + \frac{1}{\sqrt{(\frac{u_{i+1,j}-u_{i,j}}{h})^2 + (\frac{u_{i,j+1}-u_{i,j}}{h})^2 + 1}}\right],$$

$$a_s = -\frac{1}{2h^2}\left[\frac{1}{\sqrt{(\frac{u_{i,j}-u_{i-1,j}}{h})^2 + (\frac{u_{i,j}-u_{i,j-1}}{h})^2 + 1}} + \frac{1}{\sqrt{(\frac{u_{i+1,j-1}-u_{i,j-1}}{h})^2 + (\frac{u_{i,j}-u_{i,j-1}}{h})^2 + 1}}\right],$$

$$a_n = -\frac{1}{2h^2}\left[\frac{1}{\sqrt{(\frac{u_{i+1,j}-u_{i,j}}{h})^2 + (\frac{u_{i,j+1}-u_{i,j}}{h})^2 + 1}} + \frac{1}{\sqrt{(\frac{u_{i,j+1}-u_{i-1,j+1}}{h})^2 + (\frac{u_{i,j+1}-u_{i,j}}{h})^2 + 1}}\right].$$

This scheme is monotonic and thus convergent.

In our experiment, we use $f = 0$ and the zero Dirichlet boundary condition. The computational domain is $(x,y) \in [0,1]^2$. Here we use the same obstacle $\psi = \max\{0, 0.6 - 8|(x-0.5)^2 + (y-0.5)^2|\}$ as in Section 3.1.2. The obstacle and our result obtained by the scheme (2.6) are showed in Figs. 6(a) and (b). In this example, we also monitor the residual error. We plot in Fig. 6(c) the convergence of the residual error from different schemes and with different time step. For the residual error, a larger time step in general gives a better convergence speed. For scheme (2.4), as the time step goes to infinity, the convergence curve converges to that of scheme (2.6).



(a) (b)

(c)

Figure 6: (Example 3.2.2) The computational domain is $(0,1)^2$. The mesh is 64×64. (a) The obstacle and (b) the result by scheme (2.6). (c) Comparison of residual error for different scheme with different time step.

3.2.3. Comparison of different semi-implicit schemes

Our schemes (2.4) and (2.6) are just one possible semi-implicit discretization for the minimal surface obstacle problem. In this subsection, we compare our scheme with some other semi-implicit schemes. The first scheme is the one used in [26] which reads as follows:

$$\frac{u^{k+1}-u^k}{\Delta t} - \beta \Delta u^{k+1} + \beta \Delta u^k = \min(Au^k - f, u - \psi), \tag{3.1}$$

where Δu is the Laplacian of u, β is some positive constant. This is the simplest semi-implicit scheme.

If we take a closer look on the operator A for this problem, we have

$$A=-\nabla\cdot\frac{\nabla u}{\sqrt{1+|\nabla u|^2}}=-\nabla\frac{1}{\sqrt{1+|\nabla u|^2}}\cdot\nabla u-\frac{1}{\sqrt{1+|\nabla u|^2}}\Delta u.$$

Based on (2.2), if we replace the first operator A using the above relation, the PDE becomes

$$\begin{aligned}&u_t+\frac{u-\psi}{2}-\frac{1}{2}\nabla\frac{1}{\sqrt{1+|\nabla u|^2}}\cdot\nabla u-\frac{1}{2}\frac{1}{\sqrt{1+|\nabla u|^2}}\Delta u\\=&\frac{|(u-\psi)-(Au-f)|}{2}+\frac{f}{2}.\end{aligned}\tag{3.2}$$

Then we have the following semi-implicit scheme:

$$\begin{aligned}&\frac{u^{k+1}-u^k}{\Delta t}+\frac{u^{k+1}}{2}-\frac{1}{2}\frac{1}{\sqrt{1+|\nabla u^k|^2}}\Delta u^{k+1}\\=&\frac{1}{2}\nabla\frac{1}{\sqrt{1+|\nabla u^k|^2}}\cdot\nabla u^k+\frac{|(u^k-\psi)-(Au^k-f)|}{2}+\frac{\psi+f}{2}.\end{aligned}\tag{3.3}$$

The terms on the left hand side of (3.3) are very similar to that of (3.1). In (3.1), the implicit term is $\beta\Delta u^{k+1}$ whereas in (3.3) the implicit terms are

$$\frac{u^{k+1}}{2}\quad\text{and}\quad-\frac{1}{2}\frac{1}{\sqrt{1+|\nabla u^k|^2}}\Delta u^{k+1}.$$

In both schemes, we have a Laplacian term as the implicit term. The difference is that in (3.1), the coefficient of the Laplacian term is a constant and in (3.3) the coefficient varies in each iteration. In some sense, (3.3) can be taken as an expanded version of (3.1) with adaptive coefficient.

In the discretization, we approximate Au in both schemes using the formula introduced in Section 3.2.1. For Δu, we use central difference. For the coefficient of the Laplacian term in (3.3), simply using central difference on ∇u makes the scheme not working. Instead, we use the following averaged approximation:

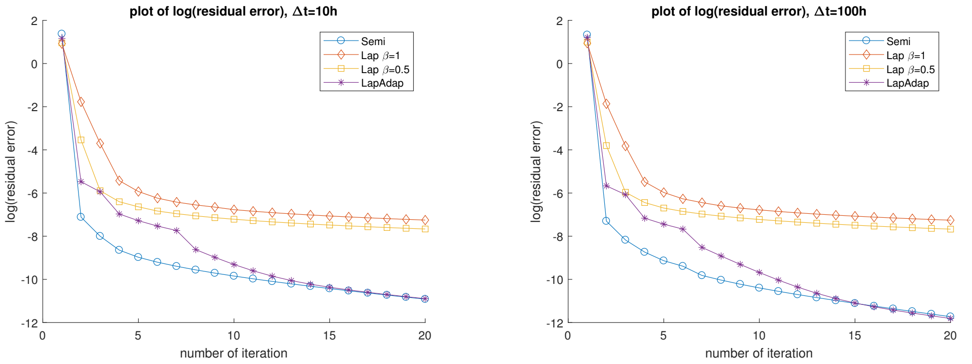
$$\left(\frac{1}{2}\frac{1}{\sqrt{1+|\nabla u^k|^2}}\right)_i=\frac{1}{4}\left[\frac{1}{\sqrt{1+\left((u_{i+1}^k-u_i^k)/h\right)^2}}+\frac{1}{\sqrt{1+\left((u_i^k-u_{i-1}^k)/h\right)^2}}\right].$$

Then we approximate the first term on the right-hand side of (3.3) by

$$\left(\frac{1}{2}\nabla\frac{1}{\sqrt{1+|\nabla u^k|^2}}\cdot\nabla u^k\right)_i=\frac{1}{2h^2}\left(b_{i+\frac{1}{2}}^k u_{i+1}^k-b_{i-\frac{1}{2}}^k u_{i-1}^k\right),$$

where

$$b_{i+\frac{1}{2}}^k=\left(1+\frac{u_{i+1}^k-u_i^k}{h}\right)^{-\frac{1}{2}}\quad\text{and}\quad b_{i-\frac{1}{2}}^k=\left(1+\frac{u_i^k-u_{i-1}^k}{h}\right)^{-\frac{1}{2}}.$$



(a) (b)

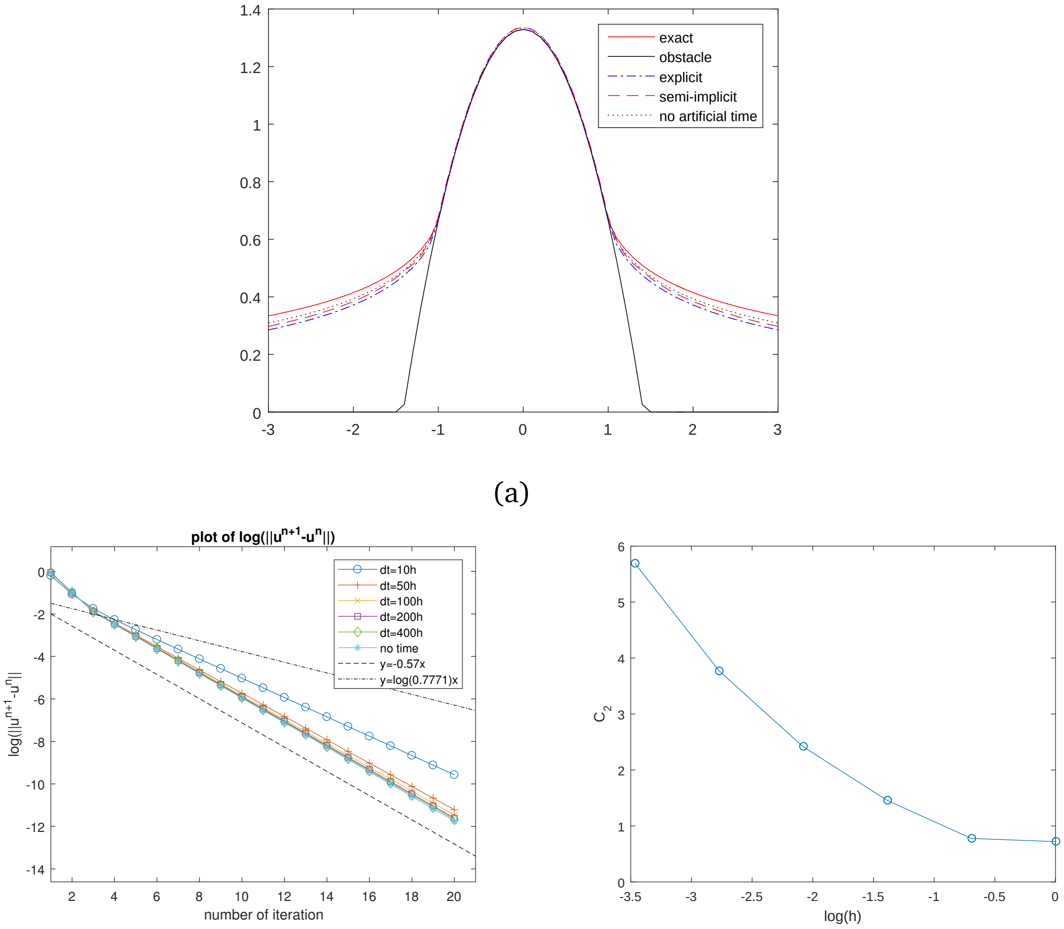
Figure 7: (Example 3.2.3) The domain is $[0,1]$ and the grid size is 64. Comparison of residual error for different semi-implicit schemes for the one dimensional minimal surface problem. (a) $\Delta t = 10h$. (b) $\Delta t = 100h$. In the legend, "Lap" denotes scheme (3.1). "LapAdap" denotes scheme (3.3). "Semi" denotes scheme (2.4).

We test the performance of the above two schemes and compare them with our semi-implicit scheme. In Fig. 7, the residual error behavior for different schemes with the same time step are shown. We test scheme (3.1) with $\beta = 0.5$ and 1, scheme (3.3) and scheme (2.4) with time step $10h$, $100h$ and grid size 64. In Fig. 7, we can see that the two residual errors by scheme (3.1) give the two slowest convergence rates. Surprisingly, smaller β gives better convergence rate. We think it might because the Laplacian term is just an artificial term. This term only stabilizes the scheme and has no contribution to the convergence behavior. Compared to scheme (3.1), scheme (3.3) gives better result which implies that adapting the linearized operator helps the convergence. Among all of the schemes, our proposed scheme (2.4) gives the best convergence rate. The residual error by scheme (2.4) decreases much faster than that of the other schemes. Since the implicit terms in scheme (3.3) and scheme (2.4) are separated from the nonlinear operator, they contain some information of the operator and thus gives better results than (3.1), which is an artificial scheme. Similarly, since (2.4) linearizes more portion from the nonlinear operator than (3.1), it gives better result.

Numerically, scheme (3.3) only needs to store the matrix representation of the Laplacian. In each iteration, one just multiplies the matrix with a diagonal matrix whose nonzero elements representing the updated coefficients. But scheme (2.4) needs to regenerate every element of the matrix representing the linearized operator. So compared with (2.4), scheme (3.3) is indeed more efficient.

3.3. A fractional obstacle problem

In this example, we consider the fractional obstacle problem given by $\min(Au - f, u - \psi) = 0$ where $A = -\Delta^{\alpha/2}$ is the negative fractional Laplacian for $0 < \alpha < 2$.



(a)

(b) (c)

Figure 8: (Example 3.3) (a) The numerical solution from different schemes. We use $h=0.5$ and $\Delta t=400h^{\alpha/2}$ for the semi-implicit scheme, $\Delta t=0.5h^{\alpha/2}$ for the explicit scheme. (b) Comparison of the convergence behavior where the error refers to $\|u_{i+1}-u_i\|$. We also plot the reference curve $y=-0.57x$ using a black dash line. (c) The constant C_2 versus the mesh size h.

A linear quadrature-finite-difference numerical scheme has been introduced in [18] to compute fractional Laplacian which is adopted in our experiment. In the experiment, we use $f=0$ and $A=-\Delta^{\alpha/2}$. The obstacle ψ is given by

$$\psi(x)=2^{-\alpha}\pi^{-1/2}\Gamma\left(\frac{1-\alpha}{2}\right)\Gamma\left(\frac{4-\alpha}{2}\right)(1-(1-\alpha)x^2)_+.$$

We choose $\alpha=0.5$ and use computational domain $x\in[-8,8]$. If we use mesh size $h=0.5$, the condition in Theorem 2.1 on A is actually satisfied with $C_2=0.7771$.

Fig. 8(a) shows the results from different schemes. We can see in Fig. 8(b), the scheme (2.5) converges better than our estimate. As Δt goes to infinity, the scheme

(2.3) behaves similar to that of the scheme (2.5), the same as our prediction. And the scheme (2.5) gives the best convergence speed. In our experiment, if we use $h = 0.1$, the condition on A in Theorem 2.1 is no longer satisfied. But both schemes still work well with almost the same convergence behavior as $h = 0.5$. The behavior of C_2 with respect to h is shown in Fig. 8(c). We can see C_2 increases as the mesh refines. But, scheme (2.5) converges well in all cases.

3.4. The min curvature flow

In this example, we consider the min curvature flow problem in the level set formulation [20] where we try to solve the following time dependent obstacle problem given by

$$u_t = |\nabla u| \min(0, \kappa),$$

where $\kappa = \nabla \cdot \frac{\nabla u}{|\nabla u|}$ is the curvature of u. Using our approach, the PDE becomes

$$u_t = \frac{|\nabla u|}{2} \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right) - \frac{|\nabla u|}{2} \left| \nabla \cdot \frac{\nabla u}{|\nabla u|} \right|$$

and is discretized as

$$\frac{u^{k+1} - u^k}{\Delta t} = \frac{|\nabla u^k|}{2} \left(\nabla \cdot \frac{\nabla u^{k+1}}{|\nabla u^k|} \right) - \frac{|\nabla u^k|}{2} \left| \nabla \cdot \frac{\nabla u^k}{|\nabla u^k|} \right|.$$

The term ∇u is approximated by the central difference. To approximate $\nabla \cdot (a \nabla u)$, we use the following scheme:

$$\begin{aligned} [\nabla \cdot (a\nabla u)]_{i,j} =& \frac{1}{h} \left[a_{i+\frac{1}{2},j} \left(\frac{u_{i+1,j} - u_{i,j}}{h} \right) - a_{i-\frac{1}{2},j} \left(\frac{u_{i,j} - u_{i-1,j}}{h} \right) \right. \\ & \left. + a_{i,j+\frac{1}{2}} \left(\frac{u_{i,j+1} - u_{i,j}}{h} \right) - a_{i,j-\frac{1}{2}} \left(\frac{u_{i,j} - u_{i,j-1}}{h} \right) \right] \end{aligned}$$

with $a_{i+\frac{1}{2},j} = \frac{1}{2}(a_{i,j} + a_{i+1,j})$ and $a_{i,j+\frac{1}{2}} = \frac{1}{2}(a_{i,j} + a_{i,j+1})$. In our experiment, we use a 4-folded star as our initial shape. The computational domain is $[0, 1]^2$. The mesh is 128×128. If the explicit scheme is used, due to the CFL condition, the time step should be of $\mathcal{O}(h^2)$. For our semi-implicit scheme, the time step is chosen as $\Delta t = h/16$. The initial shape and and its evolution by our method is given in Fig. 9.

4. Conclusions

We introduced a semi-implicit method to solve a class of obstacle problems. The idea is to extract a good linear approximation of the nonlinear obstacle constraint in the PDE based on a simple formula which expressed the min-operator by the absolute operator. We have demonstrated that our scheme can efficiently solve different kinds of

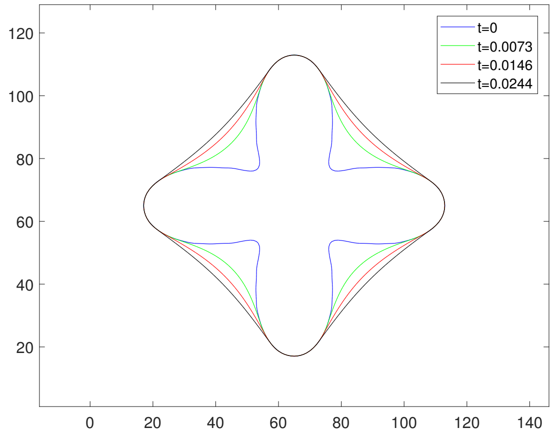


Figure 9: (Example 3.4) The computational domain is $(0,1)^2$ with the grid given by 128×128. The time step is chosen as $\Delta t = h/16$.

obstacle problems, including the membrane constrained obstacle problem and the minimal surface obstacle problem. Our methods can be easily extended to solve the fractional obstacle problem and the min curvature flow problem. Because the semi-implicit discretization allows a relatively large time step, the overall algorithm is computationally very efficient. We have also analyzed the convergence of our scheme. Although the constraint on C_1 or C_2 might seem to be quite restrictive, numerical examples show that our scheme works well even if the discretization might sometimes violate these conditions. From our numerical experiment, our scheme has the best convergence behavior among several semi-implicit schemes.

Acknowledgements The work of Leung was supported in part by the Hong Kong RGC.

References

[1] A. Brandt and C. W. Cryer, *Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems,* SIAM J. Sci. Stat. Comput., 4(4) (1983), pp. 655–684.

[2] L. A. Caffarelli, S. Salsa and L. Silvestre, *Regularity estimates for the solution and the free boundary of the obstacle problem for the fractional laplacian,* Inventiones Math., 171(2) (2008), pp. 425–461.

[3] L. A. Caffarelli and J. L. Vázquez, *Asymptotic behaviour of a porous medium equation with fractional diffusion,* arXiv preprint arXiv:1004.1096, 2010.

[4] L. A. Caffarelli and J. L. Vazquez, *Nonlinear porous medium flow with fractional potential pressure,* arXiv preprint arXiv:1001.0410, (2010).

[5] L. A. Caffarelli, The Obstacle Problem, Lezioni Fermiane 11, Scuola Normale Superiore, Pisa, 1998.

[6] L. A. Caffarelli, *The obstacle problem revisited,* Fourier Anal. Appl., 4 (1998), pp. 383–402.

[7] G. Duvaut and J.-L. Lions, Inequalities in Mechanics and Physics, Springer-Verlag, New York, 1976.

[8] M. P. Friedlander and P. Tseng, *Exact regularization of convex programs,* SIAM J. Optimization, 18(4) (2007), pp. 1326–1350.

[9] R. Glowinski, Numerical Methods for Non-Linear Variational Problems, Springer Verlag, New York, 1984.

[10] R. Glowinski, Lectures on Numerical Methods for Non-Linear Variational Problems, Springer Science & Business Media, 2008.

[11] D. Goldfarb and W. Yin, *Second-order cone programming methods for total variation based image restoration,* SIAM J. Sci. Comput., 27 (2005), pp. 622–645.

[12] T. Goldstein and S. Osher, *The split Bregman method for $L1$ regularized problems,* SIAM J. Imag. Sci., 2 (2009), pp. 323–343.

[13] W. Hackbusch and H. D. Mittelmann, *On multi-grid methods for variational inequalities,* Numer. Math., 42(1) (1983), pp. 65–76.

[14] E. T. Hale, W. Yin, and Y. Zhang, *Fixed-point continuation for l_1-minimization: Methodology and convergence,* SIAM J. Optimization, 19 (2008), pp. 1107–1130.

[15] Y. He, Y. Liu, and T. Tang, *On large time-stepping methods for the cahn-hilliard equation,* Appl. Numer. Math., 57 (2007), pp. 616–628.

[16] V. A. Hintermüller, M. Kovtunenko and K. Kunisch, *Obstacle problems with cohesion: a hemivariational inequality approach and its efficient numerical solution,* SIAM J. Optimization, 21(2) (2011), pp. 491–516.

[17] R. Hoppe, *Multigrid algorithms for variational inequalities,* SIAM J. Numer. Anal., 24(5) (1987), pp. 1046–1065.

[18] Y. Huang and A. Oberman, *Numerical methods for the fractional laplacian: a finite difference-quadrature approach,* SIAM J. Numer. Anal., 52(6) (2014), pp. 3056–3084.

[19] T. Kärkkäinen, K Kunisch and P. Tarvainen, *Augmented lagrangian active set methods for obstacle problems,* J. Optimization Theory Appl., 119(3) (2003), pp. 499–533.

[20] R. Malladi and J. A. Sethian, *Flows under $\min/\max$ curvature flow and mean curvature: applications in image processing,* Lecture Notes Comput. Sci., 1064 (1996).

[21] O. L. Mangasarian, *Sufficiency of exact penalty minimization,* SIAM J. Control Optimization, 23(1) (1985), pp. 30–37.

[22] S. Osher, Y. Mao, B. Dong and W Yin, *Fast linearized Bregman iteration for compressive sensing and sparse denoising,* Commun. Math. Sci., 8 (2010), pp. 93–111.

[23] S. J. Osher and R. P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, New York, 2003.

[24] R. Scholz, *Numerical solution of the obstacle problem by the penalty method,* Numer. Math., 49(2-3) (1986), pp. 255–268.

[25] L. Silvestre, *Regularity of the obstacle problem for a fractional power of the laplace operator,* Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, 60(1) (2007), pp. 67–112.

[26] P. Smereka, *Semi-implicit level set methods for curvature and surface diffusion motion,* J. Sci. Comput., 1-3 (2003), pp. 439–456.

[27] G. Tran, H. Schaeffer, W. M. Feldman and S. J. Osher, *An l^1 penalty method for general obstacle problems,* SIAM J. Appl. Math., 75(4) (2015), pp. 1424–1444.

[28] R. Tremolieres, J.-L. Lions and R. Glowinski, Numerical Analysis of Variational Inequalities, volume 8, Elsevier, 2011.

[29] F. WANG AND X.-L. CHENG, *An algorithm for solving the double obstacle problems,* Appl. Math. Comput., 201(1) (2008), pp. 221–228.

[30] C. WU AND J. WAN, *Multigrid methods with newton-gauss-seidel smoothing and constraint preserving interpolation for obstacle problems,* Numer. Math. Theory Methods Appl., 8(02) (2015), pp. 199–219.

[31] L. XUE AND X.-L. CHENG, *An algorithm for solving the obstacle problems,* Comput. Math. Appl., 48(10) (2004), pp. 1651–1657.

[32] G. YOU AND S. LEUNG, *A fast semi-implicit level set method for curvature dependent flows with an application to limit cycles extraction in dynamical systems,* Commu. Comput. Phys., 18(1) (2015), pp. 203–229.

[33] D. ZOSSO, B. OSTING, M. XIA AND S. OSHER, *An efficient primal-dual method for the obstacle problem,* J. Sci. Comput., 73(1) (2017), pp. 416–437.