Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.OA-2019-0109

Vol. **13**, No. 2, pp. 353-371
May 2020

A Dynamical Method for Solving the Obstacle Problem

Qinghua Ran[1,2,*], Xiaoliang Cheng[1] and Stéphane Abide[3]

[1] *School of Mathematical Science, Zhejiang University, Hangzhou, China*
[2] *School of Mathematics and Statistics, Guizhou University, Guiyang, China*
[3] *Laboratory of Mathematics and Physics, Université de Perpignan, Perpignan, France*

Received 21 July 2019; Accepted (in revised version) 25 October 2019

Abstract. In this paper, we consider the unilateral obstacle problem, trying to find the numerical solution and coincidence set. We construct an equivalent format of the original problem and propose a method with a second-order in time dissipative system for solving the equivalent format. Several numerical examples are given to illustrate the effectiveness and stability of the proposed algorithm. Convergence speed comparisons with existent numerical algorithm are also provided and our algorithm is fast.

AMS subject classifications: 35J86, 65N22, 65N06, 65M06

Key words: Variational inequality, obstacle problem, dynamical system, dynamical functional partical method.

1. Introduction

The obstacle problem is a typical variational inequality of the first kind [1]. It appeared in many fields, such as elastoplastic torsion problems in lubrication theory, membranes in elastic theory, optimal control and option pricing, which means that research into numerical methods for solving the obstacle problem is of great significance.

Let Ω be a bounded domain with a Lipschitz boundary $\partial\Omega$. Given $f \in L^2(\Omega)$ and $\psi \in H^1(\Omega)$ with $\psi \leq 0$ on $\partial\Omega$. The obstacle problem, e.g., describes the equilibrium position of an elastic membrane occupying the domain Ω. The elastic membrane (1) passes through the boundary of Ω; (2) lies above an obstacle of height ψ; and (3) is subject to the action of a vertical force which is proportional to f [2].

We denote by u the vertical displacement component of the membrane. The set of admissible displacements is $K = \{u \in H_0^1(\Omega) | u \geq \psi \text{ a.e. in } \Omega\}$, which is a closed and

*Corresponding author. *Email addresses:* qhran@zju.edu.cn (Q. H. Ran), xiaoliangcheng@zju.edu.cn (X. L. Cheng), stephane.abide@univ-perp.fr (S. Abide)

convex set. According to the principle of energy minimization, the obstacle problem for the membrane can be stated as follows

$$\text{Find} \quad u \in K: \; E(u) = \inf_{v \in K} \left\{ \frac{1}{2} \int_{\Omega} |\nabla v|^2 dx - \int_{\Omega} f v dx \right\}. \tag{1.1}$$

It has a unique solution u that belongs to $H_0^1(\Omega)$ [3]. In addition, if $\psi \in H^2(\Omega)$, then $u \in H^2(\Omega) \cap H_0^1(\Omega)$. It is well known that the solution of (1.1) is also characterized by the variational inequality

$$u \in K, \quad \int_{\Omega} \nabla u \cdot \nabla (v-u) dx \geq \int_{\Omega} f(v-u) dx, \qquad \forall v \in K. \tag{1.2}$$

If the solution $u \in H^2(\Omega) \cap H_0^1(\Omega)$, then the following relations hold [2]:

$$u - \psi \geq 0, \qquad -\Delta u - f \geq 0, \qquad (u-\psi)(-\Delta u - f) = 0 \quad \text{a.e. in } \Omega. \tag{1.3}$$

Consequently, we have $\Omega = \Omega_1 \bigcup \Omega_2$ and

$$\begin{aligned} u - \psi > 0 \quad &\text{and} \quad -\Delta u - f = 0 \quad \text{in} \quad \Omega_1, \\ u - \psi = 0 \quad &\text{and} \quad -\Delta u - f > 0 \quad \text{in} \quad \Omega_2. \end{aligned}$$

Here Ω_2 denotes the coincidence set where the elastic membrane contacts with the obstacle and Ω_1 denotes the non-coincidence set.

Various algorithms have been proposed for solving the obstacle problem. In the following, we provide a brief review of some of the existing methods. The most well-known solution techniques are projected methods such as the relaxation method [4] and multigrid method [5–7], whose convergence rate depend on the mesh refinement. [8] gave active set strategies which can be efficiently implemented by the multigrid approach. [9] proposed a moving obstacle method which considered iterative approximation of the contact region. [10, 11] investigated iterative solution of piecewise linear systems for the numerical solution of the obstacle problem. [12] proposed a direct algorithm with a penalization parameter. [1] studied virtual element method which can work on very general polygonal elements, etc..

The idea of using dynamic systems to solve mathematical problems is well known. Recently, there has been increasing evidence that second-order in time dissipative systems enjoy remarkable optimization properties [13]. They have been developed for solving a variety of problems, e.g., the inverse source problem [13] and nonlinear Schrödinger equation [14]. Combined with the particle method, [15] proposed the dynamical functional particle method (DFPM) and this method is well used in solving many problems such as large linear equations [16] and eigenvalue problems [14]. It was found that this approach is very efficient. The convergence rate of DFPM is fast in all cases (exponential time) [15]. In this paper, we consider the applicability of DFPM for solving the obstacle problem.

Let us introduce the notations used in this paper firstly. Given a bounded domain $D \in R^d$ (here $d = 1, 2$) and two positive integers m and p, $W^{m,p}(D)$ is the Sobolev

space with the corresponding usual norm which is denoted by $\|\cdot\|_{m,p,D}$. When $p=2$, $W^{m,2}(D)$ is written as $H^m(D)$ and the norm is simplified to $\|\cdot\|_{m,D}$. The closure of $C_0^\infty(D)$ in $H^m(D)$ is denoted by $H_0^m(D)$. R^m is the m-dimensional Euclidean space with the inner product $(u,v)=\sum_{i=1}^m u_i v_i$ and norm $\|u\|_2=\sqrt{(u,u)}$. Throughout the text, we adopt $\|\cdot\|$ to denote the above norm in R^m. In most of the paper, we omit to write the spatial variable x for functions.

The rest of the paper is organized as follows. The idea of the dynamical functional particle method is outlined in Section 2. In Section 3 we introduce dynamical method for solving the obstacle problem, where we give an equivalent description of the obstacle problem. Section 4 contains numerical experiments to illustrate the effectiveness and stability of our algorithm. Finally, a summary is given in the last section.

2. The dynamical functional particle method

Let H be a functional, $z: X \to R^k$, $k \in N$. The dynamical functional particle method's main idea is to solve the original equation $H(z(x))=0$, which could be, e.g., a differential, integral or integro-differential equation and possibly nonlinear, by instead defining a dynamical system $z=z(x,t): X\times[0,+\infty)\to R^k$ and solving the following second-order dynamic system

$$\ddot{z}+\eta\dot{z}=H(z) \tag{2.1}$$

in such a way that $\dot{z},\ddot{z}\to 0$, as t increases, i.e., $z(x,t)$ approaches to $z(x)$, the solution of $H(z(x))=0$ [15]. Here the dots are standard notation for time derivatives. The symbol η is damping parameter, which can be constant or function related to time and space depending on the specific problem. For simplicity, we assume that η is a constant here.

Discretizing Eq. (2.1) such that $z_i(t)=z(x_i,t)$ for $i=1,\cdots,n$, the corresponding discrete equation is

$$\ddot{z}_i(t)+\eta\dot{z}_i(t)=H_i(z_1(t),\cdots,z_n(t)),\qquad i=1,\cdots,n.$$

The total notes $I=\{1,\cdots,n\}$ can be viewed as a set of n "particles" where the position of particle i is z_i. The "force" on this particle is given by H_i and may depend on the position of other particles. When the particles become stationary at $t=t_H$, the discrete function $\boldsymbol{z}(t)=(z_1(t),z_2(t),\cdots,z_n(t))^T$ represents the discrete approximation to $z(x)$.

The numerical algorithm based on (2.1) has used to solve some special problems, such as large linear equations, eigenvalue problems and nonlinear Schrödinger problem [14, 16]. The algorithm's stability and cost efficiency is promising [17].

3. The dynamical method for the obstacle problem

3.1. Equivalent description of the obstacle problem

It is well known that solving the obstacle problem according to (1.3) is more difficult than (1.1) and (1.2). But this form is also very important. Inspired by [18], we derived the equivalent form of (1.3), which makes the problem easier to solve.

Let $[\cdot]_+$ denotes the positive part of a scalar quantity $x \in R$:

$$[x]_+ = \begin{cases} x, & \text{if } x > 0, \\ 0, & \text{otherwise.} \end{cases}$$

In the following, we will make use of the property: $|[a]_+ - [b]_+| \leq |a-b|$, $\forall a, b \in R$, where $|a|$ denotes the absolute value of a.

Lemma 3.1. *Let $\varepsilon > 0$, the following complementary problem in Ω*

$$\begin{cases} (i) & u - \psi \geq 0, \\ (ii) & -\Delta u - f \geq 0, \\ (iii) & (u-\psi)(-\Delta u - f) = 0, \end{cases} \tag{3.1}$$

is equivalent to

$$-\Delta u - f - \frac{1}{\varepsilon}[\psi - u - \varepsilon(\Delta u + f)]_+ = 0. \tag{3.2}$$

Proof. Let u be a solution of (3.1). The condition (3.1)(ii) yields either

$$-\Delta u - f > 0 \quad \text{or} \quad -\Delta u - f = 0.$$

Firstly, if $-\Delta u - f > 0$. Then (3.1)(iii) implies that $u - \psi = 0$. In this case we get

$$-\Delta u - f - \frac{1}{\varepsilon}[\psi - u - \varepsilon(\Delta u + f)]_+ = -\Delta u - f - \frac{1}{\varepsilon}[-\varepsilon(\Delta u + f)]_+ = 0.$$

Secondly, if $-\Delta u - f = 0$. According to (3.1)(i) we get $[\psi - u - \varepsilon(\Delta u + f)]_+ = 0$. In this case

$$-\Delta u - f - \frac{1}{\varepsilon}[\psi - u - \varepsilon(\Delta u + f)]_+ = -\Delta u - f = 0.$$

Conversely, let u such that (3.2) holds. This implies

$$-\Delta u - f = \frac{1}{\varepsilon}[\psi - u - \varepsilon(\Delta u + f)]_+ \geq 0,$$

so that (3.1)(ii) holds. Consider first the case $-\Delta u - f > 0$, i.e., $[\psi - u - \varepsilon(\Delta u + f)]_+ > 0$. Then (3.2) can be rewritten as

$$0 = -\Delta u - f - \frac{1}{\varepsilon}(\psi - u - \varepsilon(\Delta u + f)) = -\frac{1}{\varepsilon}(\psi - u),$$

which implies that $\psi - u = 0$. Therefore, (i)-(iii) of (3.1) are satisfied.

We now consider the case $-\Delta u - f = 0$, which implies $\psi - u - \varepsilon(\Delta u + f) \leq 0$, so that $\psi - u \leq 0$. Therefore (3.1)(i)(ii)(iii) are satisfied. □

Suppose $f \in L^2(\Omega)$, $\psi \in H^2(\Omega)$ with the compatibility condition $\psi|_{\partial\Omega} \leq 0$, it is known that the solution of (1.1) satisfies the regularity property $u \in H^2(\Omega)$. In this case, by Lemma 3.1 the obstacle problem can be formulated as the following equation:

$$F(u) = -\Delta u - f - \frac{1}{\varepsilon}[\psi - u - \varepsilon(\Delta u + f)]_+ = 0 \qquad \text{a.e. in } \Omega, \tag{3.3}$$

where $\varepsilon > 0$ is any constant. This format transforms the original complementary problem into an equation which is easier to solve. Obversely the choice of parameter ε is more flexible than the penalty method [12]. We don't need to select the parameter sufficiently small.

In order to obtain a numerical solution of $u(x)$, we discretize the problem (3.3) by using a finite difference five-point scheme (it is possible to use finite elements or any other method). Let $I = \{1, \cdots, m\}$ ($m \in N$) denotes the set of the total nodes. The discretization scheme yields the following finite dimensional problem:

$$\boldsymbol{F}(\boldsymbol{u}) = A\boldsymbol{u} - \boldsymbol{f} - \frac{1}{\varepsilon}[\boldsymbol{\psi} - \boldsymbol{u} + \varepsilon(A\boldsymbol{u} - \boldsymbol{f})]_+ = \boldsymbol{0}. \tag{3.4}$$

Here, $\boldsymbol{u} = (u_1, \cdots, u_m)^T$, $\boldsymbol{\psi} = (\psi_1, \cdots, \psi_m)^T$ and $\boldsymbol{f} = (f_1, \cdots, f_m)^T$ are vectors in R^m. The i^{th} $(i \in I)$ subentries of $\boldsymbol{u}$, $\boldsymbol{\psi}$ and $\boldsymbol{f}$ are the value of $u(x)$, $\psi(x)$ and $f(x)$ at node i, respectively. $A = (a_{ij}) \in R^{m\times m}$ is the stiffness matrix which is symmetric and positive definite [19]. For simplicity of statements, in the following we also use notations F, u, ψ and f to denote their corresponding vectors in R^m.

3.2. DFPM for the $F(u) = 0$ problem

In order to solve the problem (3.4) effectively, an artificial scalar time t is introduced to study the following second-order (in time) dynamical system

$$\ddot{u}(t) + \eta\dot{u}(t) = F(u(t)) \quad \text{in } [0, +\infty), \tag{3.5}$$

where $\eta > 0$ is a damping parameter keeping constant here and the dots are standard notation for time derivatives. Under the proper parameter $\eta > 0$, the solution $u(t)$ of (3.5) converges to u as $t \to \infty$, where u is the solution of (3.4). In contrast to the gradient descent method, the system (3.5) introduces the inertial term $\ddot{u}$, which allows to overcome some of the shortcomings of the gradient descent method, such as the zig-zag phenomenon.

It is noted that the above equation hold in componentwise sense. That is

$$\ddot{u}_i(t) + \eta\dot{u}_i(t) = F_i(u(t)), \qquad i = 1, \cdots, m. \tag{3.6}$$

Then total nodes' set $I = \{1, \cdots, m\}$ can be viewed as a set of m "particles", where the position of particle i is u_i. The "force" on this particle is given by $F_i(u)$. Under the force of F_i, the particle position u_i evolve in order to approach the steady state solution.

Denote $\dot{u}(t) = v(t)$ and rewrite the second-order Eq. (3.5) into an equivalent system which is first-order in time:

$$\begin{cases} \dot{v}(t) = F(u(t)) - \eta v(t), \\ \dot{u}(t) = v(t), \end{cases} \quad \text{in } [0, +\infty). \tag{3.7}$$

This allows to consider Eq. (3.5) when F is non-smooth or subject to constraints [17].

Theorem 3.1. *Given u_0, $v_0 \in R^m$, the first order system* (3.7) *has a unique solution.*

Proof. Denote $E = R^m \times R^m$. We define the space E with linear operations

$$[u_1, v_1]^T + [u_2, v_2]^T = [u_1 + u_2, v_1 + v_2]^T, \qquad \alpha[u_1, v_1]^T = [\alpha u_1, \alpha v_1]^T,$$

scalar product

$$([u_1, v_1]^T, [u_2, v_2]^T)_E = (u_1, u_2) + (v_1, v_2)$$

and the corresponding norm

$$\|[u_1, v_1]^T\|_E = \sqrt{\|u_1\|^2 + \|v_1\|^2},$$

where $[u_i, v_i]^T \in E$, α is any constant, $u_i, v_i \in R^m$, $i = 1, 2$. Obviously, under this definition, E is a Banach space.

As $|[a]_+ - [b]_+| \le |a - b|$, $\forall a, b \in R$, we can get that

$$\begin{aligned} \|[u]_+ - [v]_+\| &= \left(\sum_{i=1}^{m} ([u_i]_+ - [v_i]_+)^2 \right)^{\frac{1}{2}} \\ &\le \left(\sum_{i=1}^{m} (u_i - v_i)^2 \right)^{\frac{1}{2}} = \|u - v\|, \qquad \forall u, v \in R^m. \end{aligned}$$

Using the above conclusion, we obtain

$$\begin{aligned} &\|F(u_1) - F(u_2)\| \\ =&\|Au_1 - f - \frac{1}{\varepsilon}[\psi - u_1 + \varepsilon(Au_1 - f)]_+ - (Au_2 - f - \frac{1}{\varepsilon}[\psi - u_2 + \varepsilon(Au_2 - f)]_+)\| \\ \le&\|Au_1 - Au_2 - \frac{1}{\varepsilon}[\psi - u_1 + \varepsilon(Au_1 - f)]_+ + \frac{1}{\varepsilon}[\psi - u_2 + \varepsilon(Au_2 - f)]_+\| \\ \le&\|Au_1 - Au_2\| + \frac{1}{\varepsilon}\|[\psi - u_2 + \varepsilon(Au_2 - f)]_+ - [\psi - u_1 + \varepsilon(Au_1 - f)]_+\| \\ \le&2\|Au_1 - Au_2\| + \frac{1}{\varepsilon}\|u_1 - u_2\| \\ \le&2\|A\|\|u_1 - u_2\| + \frac{1}{\varepsilon}\|u_1 - u_2\| \\ \le&c_1\|u_1 - u_2\|, \qquad \forall u_1, u_2 \in R^m. \end{aligned}$$

Where c_1 is a constant and $\|A\|$ is induced by the vector norm, i.e.,

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

Let $z = [u(t), v(t)]^T$ and rewrite (3.7) to be a first-order dynamical system in the phase space E:

$$\begin{cases} \dot{z}(t) = G(z(t)) \quad \text{in } [0, +\infty), \\ z(0) = [u_0, v_0]^T, \end{cases} \tag{3.8}$$

where $G(z(t)) = [v(t), F(u(t)) - \eta v(t)]^T$.

Now, we show that G is Lipschitz continuous. $\forall z_1(t), z_2(t) \in E$, we have

$$\begin{aligned} &\|G(z_1(t)) - G(z_2(t))\|_E^2 \\ =&\|v_1(t) - v_2(t)\|^2 + \|F(u_1(t)) - F(u_2(t)) + \eta(v_2(t) - v_1(t))\|^2 \\ \leq&2\|F(u_1(t)) - F(u_2(t))\|^2 + (2\eta^2 + 1)\|v_1(t) - v_2(t)\|^2 \\ \leq&2c_1^2\|u_1(t) - u_2(t)\|^2 + (2\eta^2 + 1)\|v_1(t) - v_2(t)\|^2 \\ \leq&(2c_1^2 + 2\eta^2 + 1)\ (\|u_1(t) - u_2(t)\|^2 + \|v_1(t) - v_2(t)\|^2). \end{aligned}$$

Then, there exist a constant $L = \sqrt{2c_1^2 + (2\eta^2 + 1)}$ satisfying

$$\|G(z_1(t)) - G(z_2(t))\|_E \leq L\|[u_1(t), v_1(t)]^T - [u_2(t), v_2(t)]^T\|_E.$$

That is

$$\|G(z_1(t)) - G(z_2(t))\|_E \leq L\|z_1(t) - z_2(t)\|_E.$$

Therefore, by the Cauchy-Lipschitz-Picard theorem [20], the first-order dynamical systems (3.8) (then (3.7)) has a unique solution. □

As the implicit Euler method is all-damping stability [21], we use the scheme to solve ODE (3.7)

$$\begin{cases} v^{n+1} = v^n + \Delta t(F(u^{n+1}) - \eta v^{n+1}), \\ u^{n+1} = u^n + \Delta t v^{n+1}, \end{cases} \tag{3.9}$$

where $v^n = v(t_n)$, $u^n = u(t_n)$ and Δt is a fixed time step size. It is possible to use the damped symplectic Euler schemes [21] to solve (3.7), such as

$$\begin{cases} v^{n+1} = v^n + \Delta t(F(u^{n+1}) - \eta v^n), \\ u^{n+1} = u^n + \Delta t v^{n+1}. \end{cases} \tag{3.10}$$

Both formats yield good results, but we only discuss the format (3.9) in this article. Obviously, a convergent result u in Eq. (3.9) is indeed a solution to the original problem $F(u) = 0$. Since the high numerical precision of the multi-particle system is not necessary during its evolution towards the stationary state and the only desired characteristic is to approach equilibrium as fast as possible, a large time step Δt can be allowed [16].

For simplify of Eq. (3.9), we can get

$$\begin{cases} v^{n+1} = \dfrac{1}{1+\Delta t\eta} v^n + \dfrac{\Delta t}{1+\Delta t\eta} F(u^{n+1}), \\ u^{n+1} = u^n + \Delta t v^{n+1}. \end{cases} \tag{3.11}$$

Since $F(u^{n+1})$ is nonlinear, let's do some processing on it. Based on Eq. (3.4), we propose a kind of Picard iterative algorithm as follows:

$$F(u^{n+1}) = Au^{n+1} - f - \frac{1}{\varepsilon} R_n[\psi - u^{n+1} + \varepsilon(Au^{n+1} - f)], \tag{3.12}$$

where $R_n = diag(r_1, r_2, \cdots, r_m)$ and

$$r_i = \begin{cases} 1, & \text{if} \quad \psi_i - (u^n)_i + \varepsilon(Au^n - f)_i > 0, \\ 0, & \text{otherwise}, \end{cases} \qquad i = 1, 2, \cdots, m.$$

If the sequence $\{u^n\}$ is the solution of $F(u) = 0$, where $F(u)$ is shown in (3.12). Then, we can get the following conclusion.

Theorem 3.2. *For some $k \geq 0$, if $R_k = R_{k+1}$, then u^{k+1} is an exact solution of Eq.* (3.4).

Proof. Since $R_k = R_{k+1}$, then

$$\begin{aligned} &Au^{k+1} - f - \frac{1}{\varepsilon} R_{k+1}[\psi - u^{k+1} + \varepsilon(Au^{k+1} - f)] \\ =&Au^{k+1} - f - \frac{1}{\varepsilon} R_k[\psi - u^{k+1} + \varepsilon(Au^{k+1} - f)] = 0, \end{aligned}$$

that is u^{k+1} satisfying

$$Au - f - \frac{1}{\varepsilon}[\psi - u + \varepsilon(Au - f)]_+ = 0.$$

Therefore, u^{k+1} is an exact solution of Eq. (3.4). □

Substitute (3.12) into (3.11), the problem (3.4) reduces to the following algebraic system

$$\begin{cases} (i) \quad (E - \Delta t b B)v^{n+1} = av^n + bBu^n - bR_n(\psi - \varepsilon f) - bf, \\ (ii) \quad u^{n+1} = u^n + \Delta t v^{n+1}, \end{cases} \tag{3.13}$$

where $a = \frac{1}{1+\Delta t\eta}$, $b = \frac{\Delta t}{1+\Delta t\eta}$, $B = A + R_n - \varepsilon R_n A$, E is the identity matrix of $R^{m\times m}$. Obvirously, $E - \Delta t b B$ is a strictly diagonally-dominant matrix when Δt is not too large. So, in this case, the algebraic system (3.13) always has a solution.

3.3. Algorithm

Based on the previous theoretical analysis, we will give the steps of the dynamical algorithm for solving the problem (1.1) below.

Algorithm 3.1.

Input : Damping parameter η. Time step size Δt. Parameter ε. The permissible region Ω. The mesh size h. Precision number ϵ_0. Initial values: u^0, v^0. Iterative index : $n \leftarrow 0$.

Output : u^n.

1. $error \leftarrow 2\epsilon_0$
2. while $error > \epsilon_0$ do
3. solve (3.13)(i) with known source u^n, v^n to get v^{n+1}
4. $u^{n+1} \leftarrow u^n + \Delta t v^{n+1}$
5. $error \leftarrow \|u^{n+1} - u^n\|$
6. $n \leftarrow n + 1$
7. **end while**

Remark 3.1. In practice, one may start the algorithm with a random guess for the vector u^0 and simply let $v^0 = 0$.

Remark 3.2. According to Theorem 3.2, it is reasonable to set $\|u^{n+1} - u^n\| < \epsilon_0$ as stop criterion.

Remark 3.3. Without loss of generality, we divide the region Ω into a uniform mesh with a mesh size h, where the grid lines are perpendicular to the coordinate axes.

4. Numerical simulations

In this section, several numerical examples are implemented to illustrate the feasibility and effectiveness of our proposed algorithm. All the computations were performed on a dual core personal computer with 8 GB RAM with Matlab version R2018b. To evaluate the accuracy of the approximate solutions, we define the relative error for the approximate solution u^h :

$$Err2 := \log_2(\|u^h - u^*\|/\|u^*\|).$$

Note in cases where the exact solution u^* is not available, the last iterative result u^n is applied. As we all know, $\|v\|_{L^2(\Omega)} \approx h\|v^h\|$, where v^h is the discrete form of v discussed in this paper. To make the calculation more accurate, we adopt the absolute error $e_{n+1} := h\|u^{n+1} - u^n\| \leq \epsilon_0 = 10^{-7}$ as the stop criterion in the following tests. In order to enhance persuasiveness, we also attach the residual error $e_F := h\|F(u^n)\|$ as a reference.

4.1. Test 1

The purpose of this test is to explore the dependence of the solution accuracy and the convergence speed on damping parameter η, time step size Δt and parameter ε and thus to give a guide on the choices of them in the following tests. In this test, we consider the torsion of an elastic-plastic cylinder of cross-section [22] shown in Example 4.1.

Example 4.1. The cross-section of the prism is $\Omega = [0,1] \times [0,1]$,

$$u \in K_1, \quad \int_\Omega \nabla u \cdot \nabla (v-u) dx \geq C \int_\Omega (v-u) dx, \qquad \forall v \in K_1, \tag{4.1}$$

where

$$K_1 = \left\{ v \in H_0^1(\Omega) || \nabla v(x)| \leq 1 \quad \text{a.e. in } \Omega \right\}. \tag{4.2}$$

By the result of [23], the problem (4.1) and (4.2) can be equivalently expressed as

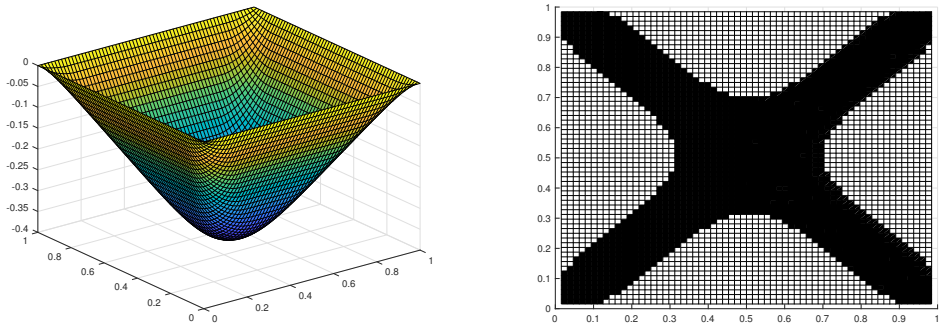
$$u = Arg \inf_{v \in K} \left\{ \frac{1}{2} \int_\Omega |\nabla v|^2 dx - \int_\Omega C v dx \right\}, \tag{4.3}$$

where

$$K = \left\{ v \in H_0^1(\Omega) |\ v \geq \psi \quad \text{a.e. in } \Omega \right\}. \tag{4.4}$$

If we choose the constant twist angle $C < 0$, then (4.3) and (4.4) form a lower obstacle problem with $\psi(x,y) = -dist((x,y), \partial\Omega)$.

Let $f(x,y) = C = -8$, we use the new algorithm to obtain an approximated solution with size $h = \frac{1}{64}$ and $u^0 = v^0 = 0$. The results are shown in Fig. 1. In Fig. 1(b), we show the plastic region ($|\nabla u| = 1$) in white and its complement represents the elastic region.



(a) The numerical result (b) The coincidence sets (white)

Figure 1: The solution of Example 4.1.

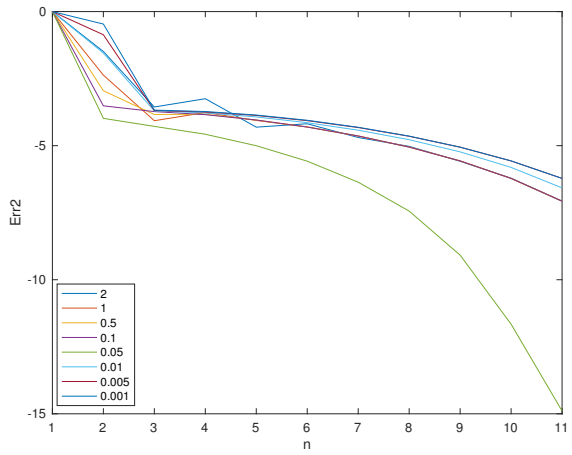


Figure 2: Evolution of $Err2$ versus n for different values of ε.

We first investigate the influence of ε on the convergence rate. For this purpose, we fix $\Delta t = 10$, $\eta = 1$. The evolutions of $Err2$ with respect to the iterative number for the different values of ε are shown in Fig. 2. The iterative number n, absolute error e_n and residual error e_F for different values of ε are shown in Table 1. The results show that for all $\varepsilon > 0$, the iteration converges. As ε gets smaller, the iterative number, error e_n and e_F remain the same. Comparably, we set $\varepsilon = 0.05$ in the following tests.

We next consider the influence of damping parameter η on the convergence rate. Similarly, we fix $\Delta t = 10$ and $\varepsilon = 0.05$. The evolutions of $Err2$ with respect to the iterative number for the different values of η are shown in Fig. 3. The iterative number and error for different values of η are shown in Table 2. According to Table 2 and Fig. 3, we can get that the convergence rate depends less on the values of η. Thus, we set $\eta = 1$ in all numerical experiments below.

We finally consider the influence of time step size Δt on the convergence rate. Here, we fix $\eta = 1$ and $\varepsilon = 0.05$. The evolutions of $Err2$ with respect to the iterative number for the different values of Δt are shown in Fig. 4. The iterative number and error for

Table 1: Error and iteration number for different values of ε.

ε	2	1	0.5	0.1	0.05	0.01	0.005	0.001
n	16	16	16	16	12	16	17	17
e_n	7.5823e-08	4.7137e-09	4.7155e-09	4.7155e-09	1.8907e-09	4.7365e-09	4.7155e-09	4.7155e-09
e_F	5.8872e-08	6.4374e-08	6.4375e-08	6.4375e-08	6.3844e-08	6.4461e-08	6.4375e-08	6.4375e-08

Table 2: Error and iteration number for different values of η.

η	10	5	1	0.5	0.1	0.05	0.01	0.005
n	12	12	12	12	12	12	12	12
e_n	4.2964e-08	2.0629e-08	1.8907e-09	8.4496e-10	2.6185e-09	2.8560e-09	3.0467e-09	3.0706e-09
e_F	1.0001e-07	7.1571e-08	6.3844e-08	6.3780e-08	6.3874e-08	6.3895e-08	6.3913e-08	6.3915e-08

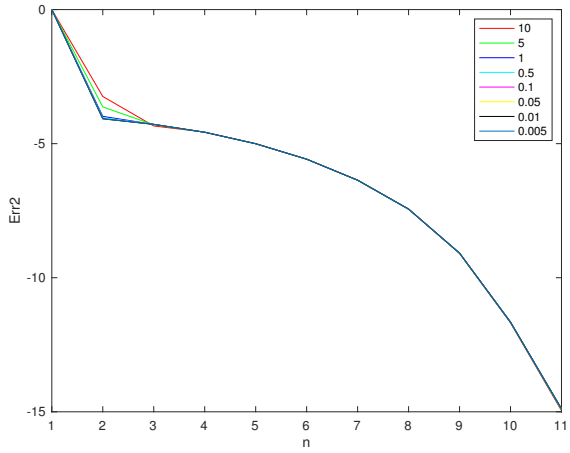


Figure 3: Evolution of $Err2$ versus n for different values of η.

different values of Δt are presented in Table 3. The results show that with the decrease of Δt, the convergence speed slows down when $\Delta t < 1$. It may even lead to non-convergence due to the continuous accumulation of errors along with the increasing iterative times. So Δt can not be too small. Inspite of this, our algorithm is not affected. Since the high numerical precision of the multi-particle system is not necessary during its evolution towards the stationary state and the only desired characteristic is to approach equilibrium as fast as possible, a large time step Δt can be allowed [16]. We also can get that the bigger the time step size Δt is, the faster the iteration is. However, our experiments suggest that too big value of Δt is not necessary. For example, $\Delta t = 10, 15$ and 20 almost have the same convergence rate. By comparing with each other, we set $\Delta t = 10$ in the following tests.

Table 3: Error and iteration number for different values of Δt.

Δt	20	15	10	5	1	0.8	0.6
n	12	12	12	12	14	20	44
e_n	1.6697e-09	1.8958e-09	1.8907e-09	3.3947e-09	8.8247e-08	4.7474e-08	9.3329e-08
e_F	1.6015e-08	2.8438e-08	6.3844e-08	2.5385e-07	4.0812e-07	2.9671e-07	7.7774e-07

4.2. Test 2

In this test, we still consider the problem of the torsion of an elastic-plastic cylinder shown in Example 4.1. We first investigate the stability of approximate solutions with respect to the selection of initial values u^0 and v^0. As indicated by Test 1, we set $\varepsilon = 0.05$, $\eta = 1$, $\Delta t = 10$ and the mesh size $h = \frac{1}{64}$. Then we set different values of u^0 and v^0 and implement algorithm repeatedly. As is shown in Table 4 where $rand_i$, $(i = 1, 2, \cdots, 5)$ means random value, the iteration numbers, e_n and e_F changed barely,

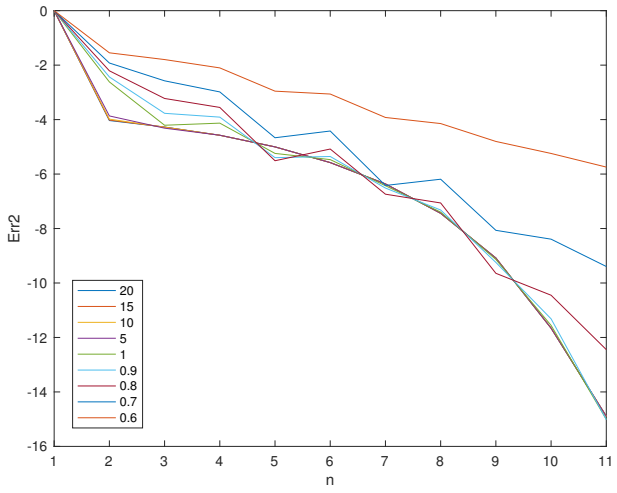


Figure 4: Evolution of $Err2$ versus n for different values of Δt.

which indicates that our algorithm is quite stable. The evolutions of $Err2$ with respect to the iterative number for the different values of u^0 and v^0 are shown in Fig. 5, which shows that no matter how we choose the initial values, the method proposed in this paper will quickly (about 2 to 4 steps) determine the approximate coincidence and non-coincidence sets and then refine the interface between the coincidence and non-coincidence parts, so there isn't much difference in the number of iterations under different initial conditions.

We finally set $\varepsilon = 0.05$, $\eta = 1$, $\Delta t = 10$ and $u^0 = v^0 = 0$. Then we implement algorithm repeatedly with different values of h. The iterative result showed in Table 5. We can see that iterative number n is independent of mesh size and the algorithm is robust.

Table 4: Error and iteration number for different values of u^0 and v^0.

u^0 and v^0	0	$rand_1$	$rand_2$	$rand_3$	$rand_4$	$rand_5$
n	12	15	16	15	16	16
e_n	1.8907e-09	2.2129e-09	2.2130e-09	4.7232e-09	4.7236e-09	4.7261e-09
e_F	6.3844e-08	3.9813e-08	3.9813e-08	6.4408e-08	6.4408e-08	6.4419e-08

Table 5: Error and iteration number for different values of h.

h	$1/8$	$1/16$	$1/32$	$1/50$	$1/64$	$1/128$
n	6	6	7	10	12	20
e_n	1.5209e-09	2.3770e-09	4.5019e-08	4.6321e-09	1.8907e-09	1.9356e-10
e_F	6.1958e-09	1.0704e-09	1.1736e-08	3.0316e-08	6.3844e-08	2.2907e-08

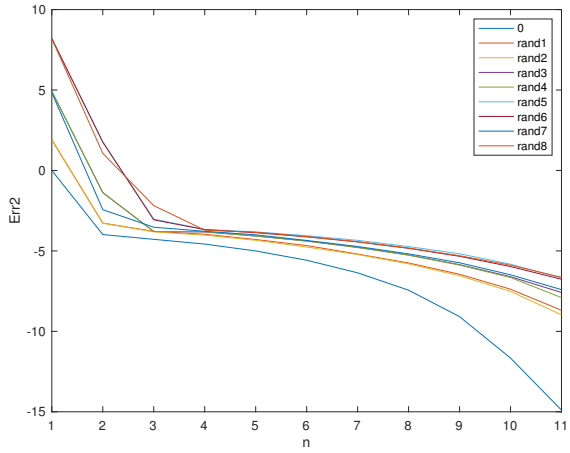


Figure 5: Evolution of $Err2$ versus n for different values of u^0 and v^0.

4.3. Test 3

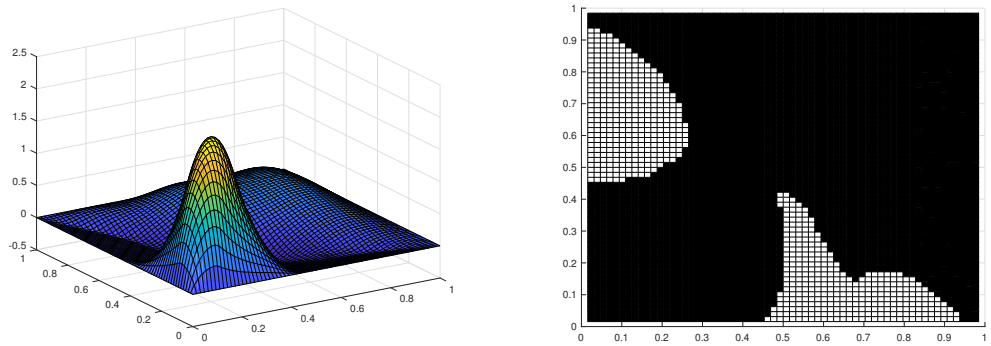
The above experiments show the good behavior of our proposed method. In this test, we compare the convergence rates between the method proposed in this paper and the penalization method [12] by solving other four problems. Indicated by Test 1 and Test 2, for our method, we set $\Delta t = 10$, $\eta = 1$, $\varepsilon = 0.05$, $u^0 = v^0 = 0$ and $h = \frac{1}{64}$. We firstly give the other four examples and their numerical results are shown in Figs. 6-9 respectively, which show that our algorithm is accurate and convergent.

Example 4.2. This two-dimensional unilateral obstacle problem has been reported in [24, 25]. Let $\Omega = [0,1] \times [0,1]$, $\psi(x,y) = -dist((x,y), \partial\Omega)$, and

$$f(x,y) = \begin{cases} 300, & \text{if} \quad (x,y) \in S = \{(x,y) \in \Omega : |x-y| \le 0.1 \quad \text{and} \quad x \le 0.3\}, \\ -70e^y g(x), & \text{if} \quad x \le 1-y \quad \text{and} \quad (x,y) \notin S, \\ 15e^y g(x), & \text{if} \quad x > 1-y \quad \text{and} \quad (x,y) \notin S, \end{cases}$$

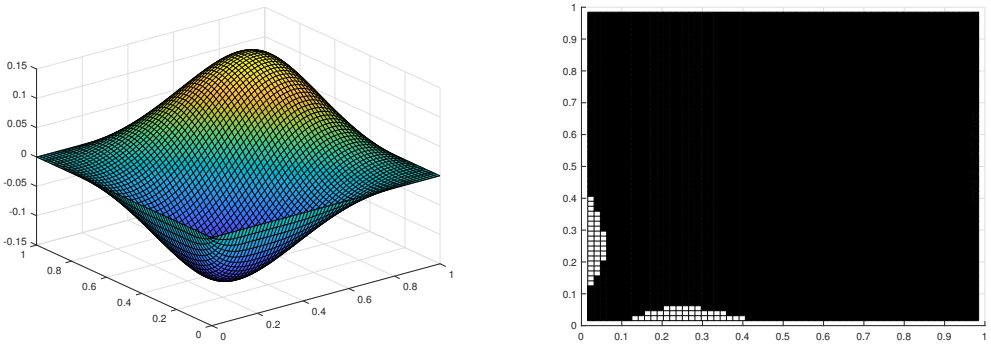
where

$$g(x) = \begin{cases} 6x, & \text{if} \quad 0 < x \le \dfrac{1}{6}, \\ 2(1-3x), & \text{if} \quad \dfrac{1}{6} < x \le \dfrac{1}{3}, \\ 6\Big(x - \dfrac{1}{3}\Big), & \text{if} \quad \dfrac{1}{3} < x \le \dfrac{1}{2}, \\ 2\Big(1 - 3\Big(x - \dfrac{1}{3}\Big)\Big), & \text{if} \quad \dfrac{1}{2} < x \le \dfrac{2}{3}, \\ 6\Big(x - \dfrac{2}{3}\Big), & \text{if} \quad \dfrac{2}{3} < x \le \dfrac{5}{6}, \\ 2\Big(1 - 3\Big(x - \dfrac{2}{3}\Big)\Big), & \text{if} \quad \dfrac{5}{6} < x \le 1. \end{cases}$$



(a) The numerical result (b) The coincidence sets (white)

Figure 6: The solution of Example 4.2.



(a) The numerical result (b) The coincidence sets (white)

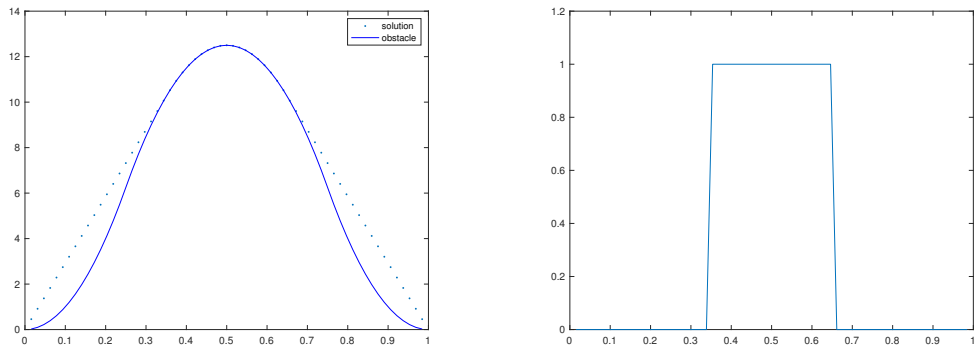
Figure 7: The solution of Example 4.3.

Example 4.3. Let $\Omega = [0,1] \times [0,1]$, $\psi(x,y) = -dist((x,y), \partial\Omega)$, and set $f(x,y) = 11(x+y-1)$ [26].

Example 4.4. This one-dimensional unilateral obstacle problem has been reported in [9]. Set $\Omega = [0,1]$, $f(x) = 0$ and

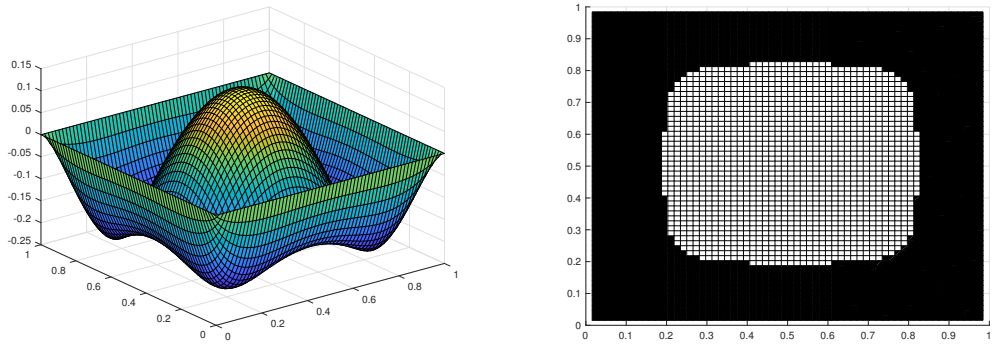
$$\psi(x) = \begin{cases} 100x^2, & \text{if} \quad 0 \leq x \leq 0.25, \\ 100x(1-x) - 12.5, & \text{if} \quad 0.25 < x \leq 0.5, \\ \psi(1-x), & \text{if} \quad 0.5 < x \leq 1. \end{cases}$$

In Fig. 8(a), the dotted line represents the numerical solution u and the solid line represents the obstacle ψ. In Fig. 8(b), the abscissa of the part whose function value is 1 represents the coincidence part.



(a) The numerical result (b) The coincidence sets

Figure 8: The solution of Example 4.4.



(a) The numerical result (b) The coincidence sets (white)

Figure 9: The solution of Example 4.5.

Example 4.5. Let $\Omega = [0,1] \times [0,1]$. The load and obstacle function are $f(x,y) = -20$ and $\psi(x,y) = 10x(1-x)y(1-y) - \frac{1}{2}$, respectively [27].

For the sake of understanding, we briefly introduce the penalization method. More details can be found in [12,26]. The penalization method's idea is writing the obstacle problem as a multi-valued equation

$$-\Delta u + \beta(u-\psi) \ni f \quad \text{in } \Omega, \quad \text{where } \beta(x) = \begin{cases} [-\infty, 0], & \text{if } x = 0, \\ 0, & \text{if } x > 0, \\ \emptyset, & \text{if } x < 0, \end{cases}$$

Table 6: Iteration number of different examples for different methods.

	Example 4.1	Example 4.2	Example 4.3	Example 4.4	Example 4.5
our method	12	23	16	10	17
penalization method	19	25	17	21	20

and use

$$\beta_\varepsilon(x) = \begin{cases} \dfrac{1}{\varepsilon}x, & \text{if} \quad x \leq 0, \\ 0, & \text{if} \quad x > 0, \end{cases}$$

to approximate $\beta(x)$. At step n of iteration, the authors use finite element method to solve a linear elliptic equation

$$-\Delta u_n + \frac{1}{\varepsilon_n}\chi_{\Omega\setminus\Omega^n}(u_n - \psi) = f \quad \text{in } \Omega,$$

then they let $u_n = \max\{u_n, \psi\}$, $\Omega^{n+1} = \{x \in \Omega : u_n(x) > \psi(x)\}$, $\varepsilon_{n+1} = \frac{\varepsilon_n}{2}$ and iterated until $\|u_n - u_{n-1}\|_{1,\Omega} < 10^{-3}$. In this method, how the parameter ε is selected is critical. Based on this process, we set an algorithm for the penalization method which keeps its original process and optimal parameter ε. For comparison, we use finite difference five-point scheme to solve the linear elliptic equation at each iteration and set $e_n = h\|u_n - u_{n-1}\| \leq 10^{-7}$ as the stop criterion and $h = \frac{1}{64}$, $u^0 = 0$ for the penalization method. The iterative numbers for different methods are shown in Table 6. Obviously, our algorithm is as fast as the penalization method. But the choice of parameter ε is more flexible than the penalization method. Our method converges quickly in many cases.

5. Conclusions

We construct a equivalent format of the well-known elliptic obstacle problem and propose a robust algorithm based on the dynamical functional particle method (DFPM). Several numerical experiments illustrate the effectiveness of the algorithm. The experimental results show that our algorithm converges as fast as the penalization method under the condition of more flexible selection of parameter. And our algorithm barely depend on the parameters and initial values.

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 11571311 and by the European Unions Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie Grand Agreement No. 823731 CONMECH.

References

[1] F. WANG AND H. WEI, *Virtual element methods for the obstacle problem,* IMA J. Numer. Anal., 2018.

[2] W. HAN AND X. CHENG, An Introduction to Variational Inequalities: Theory, Numerical Analysis and Applications, Beijing: Higher Education Press, 2007.

[3] G. DUVAUT, J. LIONS AND C. JOHN ET AL., Inequalities in Mechanics and Physics, Springer-Verlag, Berlin, 1976.

[4] R. GLOWINSKI, Numerical Methods for Non-linear Variational Problems, Springer-Verlag, 1984.

[5] W. HACKBUSCH AND A. REUSKEN, *Analysis of a damped non-linear multilevel method,* Numer. Math., 55(2) (1989), pp. 225–246.

[6] R. KORNHUBER, *Monotone multigrid methods for elliptic variational inequalities I,* Numer. Math., 69 (1994), pp. 167–184.

[7] R. KORNHUBER, *Monotone multigrid methods for elliptic variational inequalities II,* Numer. Math., 72 (1996), pp. 481–499.

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

[9] L. XUE AND X. CHENG, *An algorithm for solving the obstacle problems,* Comput. Math. Appl., 48(10-11) (2004), pp. 1651–1657.

[10] L. BRUGNANO AND A. SESTINI, *Iterative solution of piecewise linear systems for the numerical solution of obstacle problems,* Math., 31(3-4) (2012), pp. 67–82.

[11] D. YUAN AND X. CHENG, *An iterative algorithm based on the piecewise linear system for solving bilateral obstacle problems,* Int. J. Comput. Math., 89 (2012), pp. 2374–2384.

[12] C. MUREA AND D. TIBA, *A direct algorithm in some free boundary problems,* J. Numer. Math., 24 (2016), pp. 253–271.

[13] Y. ZHANG, R. GONG AND X. CHENG ET AL., *A dynamical regularization algorithm for solving inverse source problems of elliptic partial differential equations,* Inverse Probl., 34(6) (2018), 65001.

[14] P. SANDIN, M. ÖGREN AND M. GULLIKSSON, *Numerical solution of the stationary multicomponent nonlinear Schrödinger equation with a constraint on the angular momentum,* Phys. Rev. E., 93(3) (2016), 033301.

[15] S. EDVARDSSON, M. GULLIKSSON AND J. PERSSON, *The dynamical functional particle method: an approach for boundary value problems,* J. Appl. Mech., 79(2) (2012), 021012.

[16] S. EDVARDSSON, M. NEUMAN AND P. EDSTRÖM ET AL., *Solving equations through particle dynamics,* Comput. Phys. Commun., 197 (2015), pp. 169–181.

[17] F. ALVAREZ, H. ATTOUCH AND J. BOLTE ET AL., *A second-order gradient-like dissipative dynamical system with Hessian-driven damping: application to optimization and mechanics,* J. Math. Pure. Appl., 81(8) (2002), pp. 747–779.

[18] F. CHOULY AND P. HILD, *A Nitsche-based method for unilateral contact problems: numerical analysis,* SIAM J. Numer. Anal., 51(2) (2013), pp. 1295–1307.

[19] G. WINDISCH, M-matrices in Numerical Analysis, Wiesbaden: 1989.

[20] H. BRÉZIS, Functional Analysis, Soblev Spaces and Partial Differential Equations, Universitext, Springer, second edition, 2010.

[21] E. HAIRER, G. WANNER AND C. LUBICH, Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations, Berlin: 2006.

[22] R. GLOWINSKI, J. LIONS AND R. TREÉMOLIÈRE, Numerical Analysis of Variational Inequalities, 2nd Edition, 1981.

[23] H. BRÉZIS AND M. SIBONY, *Equivalence de deux inéquations variationnelles et applications,* Arch. Ration. Mech. An., 41 (1971), pp. 254–265.
[24] T. KÄRKKÄINEN, K. KUNISCH AND P. TARVAINEN, *Augmented Lagrangian active set methods for obstacle problems,* J. Optimiz. Theory Appl., 119 (2003), pp. 499–533.
[25] F. WANG AND X. CHENG, *An algorithm for solving the double obstacle problems,* Appl. Math. Comput., 201(1) (2008), pp. 221–228.
[26] C. MUREA AND D. TIBA, *A penalization method for the elliptic bilateral obstacle problem,* IFIP Adv. Inform. Commun. Tech., 443 (2014), pp. 189–198.
[27] B. IMORO, *Discretized obstacle problems with penalties on nested grids,* Appl. Numer. Math., 32 (2000), pp. 21–34.