An algorithm for solving the double obstacle problems

Fei Wang*, Xiao-Liang Cheng

Department of Mathematics, Zhejiang University, Hangzhou 310027, PR China

Abstract

In this paper, we extend the algorithm in Xue and Cheng [L. Xue, X.L. Cheng, An algorithm for solving the obstacle problems, Comput. Math. Appl. 48 (2004) 1651–1657] for solving the double obstacle problem. We try to find the approximated region of the contact in the double obstacle problem with less computation time by iteration. Numerical example is given for the double obstacle problem for the elastic–plastic torsion problem to support the algorithm.

Keywords: The coincidence set; Double obstacle problem; Algorithm

1. Introduction

Let \( \Omega \subset \mathbb{R}^2 \). We consider the double obstacle problem: find \( u \in K \), such that

\[
E(u) = \min_{v \in K} E(v),
\]

where

\[
K = \{ v \in H_0^1(\Omega), \ \varphi \leq v \leq \psi \ \text{a.e in} \ \Omega \},
\]

\[
E(v) = \frac{1}{2} \int_\Omega |\nabla v|^2 \, dx - \int_\Omega f v \, dx,
\]

and the obstacle functions \( \varphi, \psi \in H^1(\Omega) \cap C(\Omega), \varphi \leq 0, \psi \geq 0 \) on \( \partial \Omega \), which is the boundary of the domain \( \Omega \). Problems (1)–(3) denote the displacement \( u \) of elastic membrane lie between \( \varphi \) and \( \psi \) under the distributed force \( f \). The admissible set \( K \) is convex and \( v \in H_0^1(\Omega) \) means \( v = 0 \) on \( \partial \Omega \). Eq. (3) expresses the total potential energy of the deformed membrane. In finding the position of equilibrium, the principle of minimum potential energy reduces the problem to look for, among all functions \( v(x) \) in admissible set \( K \), the one which minimizes the functional (3).

* Corresponding author.

E-mail address: wangfeitwo@163.com (F. Wang).

The projection was supported by National Science Foundation of China (Grant No. 10471129).

© 2007 Elsevier Inc. All rights reserved.

It is well known that (1)–(3) have the equivalent form: find \( u \in K \), such that
\[
\int_{\Omega} \nabla u \cdot \nabla (v - u) \, dx \geq \int_{\Omega} f (v - u) \, dx \quad \forall v \in K,
\]
which is the first kind elliptic variational inequality. The theory and numerical solution of variational inequalities have been widely studied in the monographs, e.g. [2–7]. If the solution \( u \in C^2(\Omega) \cap H^1_0(\Omega) \), the variational inequality (4) also can be transformed to the boundary problem
\[
\begin{align*}
-\Delta u &\geq f, & \Omega_1 & = \{ x \in \Omega | u(x) = \varphi(x) \}, \\
-\Delta u &\leq f, & \Omega_0 & = \{ x \in \Omega | \varphi(x) < u(x) < \psi(x) \}, \\
\phi &\leq u \leq \psi & \text{in } \Omega, \quad \text{and } u = 0 \text{ on } \partial \Omega.
\end{align*}
\]
\( \Omega_1(\Omega_2) \) is the coincidence set about \( u \) with the lower (upper) obstacle. The obstacle problem is not just only introduced for a membrane in elasticity theory, but also for the non-parametric minimal and capillary surfaces as geometrical problems. The elastic–plastic torsion problem and the cavitation problem in the theory of lubrication also can be regarded as obstacle type problems. Since the obstacle problem is highly non-linear, the approximated solution is difficultly computed. The popular method for solving obstacle problems is variable projection method, such as the relaxation method [4], multilevel projection method [8], multigrid method [9–12] and so on.

The authors in [1] propose an algorithm, whose process is similar to simulating of moving obstacle but not same, to find the approximated region of the contact. We give a group of pictures to illustrate the idea of the algorithm for obstacle problem (only one obstacle) in which \( f = 0 \). See Fig. 1, the line marked by little circle “○” denotes the displacement of membrane and the line marked by star “∗” is obstacle, they contact when the star “∗” falls in circle “○” completely. First of all, we compute the equation \(-\Delta u = f\) to get the initial state without obstacle. Then we find \( \Omega_1^{(1)} \) where the obstacle first touches the membrane. By setting \( u(x) = \varphi(x) \) in \( \Omega_1^{(1)} \) we compute \(-\Delta u = f\) in \( \Omega \setminus \Omega_1^{(1)} \) again to get the next state of the membrane. Repeat this process until no new touch region is found.

![Fig. 1](image-url)
In this paper, we extend this idea for solving double obstacle problem (1)–(3). We present the algorithm in the following section and give one numerical experiment in Section 3.

2. Algorithm

The double obstacle problem describes the equilibrium position $u$ of an elastic membrane constrained to lie between two given obstacles $\phi, \psi$ under an external force $f$. To find the approximated region of the contact $\Omega_1, \Omega_2$, we first consider the no obstacle case as the initial position of the elastic membrane $u^0(x) \in C^2(\Omega) \cap H^1_0(\Omega)$, i.e.

$$
-\Delta u^0(x) = f \quad \text{in } \Omega.
$$

We can imagine that the lower obstacle moves from $-\infty$ and the upper obstacle moves from $+\infty$ to their final position $\phi, \psi$, respectively. Let lower obstacle move first, and find $\Omega_1^{(1)}$, the part of lower coincidence set $\Omega_1$, where the $u^0(x)$ first touches the “moving lower obstacle”. Then, we obtain the position of the elastic membrane, $u^1(x) \in C^2(\Omega) \cap H^1_0(\Omega)$, by solving

$$
\begin{aligned}
-\Delta u^1(x) &= f \quad \text{in } \Omega \setminus \Omega_1^{(1)}, \\
u^1(x) &= \phi(x) \quad \text{in } \Omega_1^{(1)}.
\end{aligned}
$$

We again get the contact part $\Omega_1^{(2)}$, where $u^1(x)$ meets the “lower moving obstacle” in $\Omega \setminus \Omega_1^{(1)}$. $u^2(x) \in C^2(\Omega) \cap H^1_0(\Omega)$, solve

$$
\begin{aligned}
-\Delta u^2(x) &= f \quad \text{in } \Omega \setminus \left(\Omega_1^{(1)} \cup \Omega_1^{(2)}\right), \\
u^2(x) &= \phi(x) \quad \text{in } \Omega_1^{(1)} \cup \Omega_1^{(2)}.
\end{aligned}
$$

Repeat the above procedure until no new touch region is need to be added.

Assume it take $k$ steps of iteration to obtain $\Omega_k = \Omega_1^{(1)} \cup \cdots \cup \Omega_1^{(k)}$ and $u^k$. Then let upper obstacle move, we get $\Omega_1^{(k)}$, the part of region of the contact $\Omega_2$, where the upper obstacle touches the membrane $u(x)$ firstly. By solving

$$
\begin{aligned}
-\Delta u^{k+1}(x) &= f \quad \text{in } \Omega \setminus \left(\Omega_1^{(1)} \cup \Omega_2^{(1)}\right), \\
u^{k+1}(x) &= \phi(x) \quad \text{in } \Omega_1^{(1)}, \\
u^{k+1}(x) &= \psi(x) \quad \text{in } \Omega_2^{(1)},
\end{aligned}
$$

we have the position of the membrane $u^{k+1}(x) \in C^2(\Omega) \cap H^1_0(\Omega)$. Attentively, the moving of upper obstacle makes deformation of membrane, which brings new touch region $\Omega_1^{(k+1)}$ between the lower obstacle and the membrane. $u^{k+2}(x) \in C^2(\Omega) \cap H^1_0(\Omega)$, we solve

$$
\begin{aligned}
-\Delta u^{k+2}(x) &= f \quad \text{in } \Omega \setminus \left(\Omega_1^{(1)} \cup \Omega_2^{(1)} \cup \Omega_1^{(k+1)}\right), \\
u^{k+2}(x) &= \phi(x) \quad \text{in } \Omega_1^{(1)} \cup \Omega_1^{(k+1)}, \\
u^{k+1}(x) &= \psi(x) \quad \text{in } \Omega_2^{(1)},
\end{aligned}
$$

then obtain the displacement of membrane $u^{k+2}$, and $\Omega_2^{(2)}$ the region of contact with upper obstacle. Again repeat the above procedure until no new coincidence set comes to being. Ultimately we obtain the coincidence set $\Omega_1 = \Omega_1^{(1)} \cup \Omega_1^{(k+1)} \cup \cdots$ and $\Omega_2 = \Omega_2^{(1)} \cup \Omega_2^{(2)} \cup \cdots$.

During this procedure we first let one obstacle move completely and then let the other obstacle move, so we call this procedure “One the other”. In what follows we introduce another “Up down” procedure to get the solution $u$ and the coincidence sets $\Omega_1, \Omega_2$.

At the beginning, $u^0(x) \in C^2(\Omega) \cap H^1_0(\Omega)$, solve

$$
-\Delta u^0(x) = f \quad \text{in } \Omega.
$$
we get \( u^0 \) the displacement of elastic membrane under the external force \( f \) without obstacle as initial state. Let lower obstacle move, \( \Omega_1^{(1)} \) is the part of the region of contact between \( u \) and \( \varphi \), where lower obstacle first touches membrane. By solving

\[
\begin{aligned}
-\Delta u^1(x) &= f & \text{in } \Omega \setminus \Omega_1^{(1)}, \\
u^1(x) &= \varphi(x) & \text{in } \Omega_1^{(1)},
\end{aligned}
\]  

(12)

we obtain displacement of membrane \( u^1(x) \in C^2(\Omega) \cap H_0^1(\Omega) \). Now lower obstacle stops, let the upper move, after finding \( \Omega_2^{(1)} \) the part of touch region of \( u \) with \( \psi \), we can get the next state of \( u^2(x) \in C^2(\Omega) \cap H_0^1(\Omega) \) by solving

\[
\begin{aligned}
-\Delta u^2(x) &= f & \text{in } \Omega \setminus (\Omega_1^{(1)} \cup \Omega_2^{(1)}), \\
u^2(x) &= \varphi(x) & \text{in } \Omega_1^{(1)}, \\
u^2(x) &= \psi(x) & \text{in } \Omega_2^{(1)},
\end{aligned}
\]  

(13)

it is upper obstacle's turn to stop and lower obstacle moves.

Repeat this procedure until no new touch region is added. We obtain the approximated coincidence set \( \Omega_1 = \Omega_1^{(1)} \cup \Omega_2^{(1)} \cup \cdots \cup \Omega_1^{(k)} \), \( \Omega_2 = \Omega_2^{(1)} \cup \Omega_2^{(2)} \cup \cdots \cup \Omega_2^{(k)} \), and solution \( u^k \) we want to have.

The above procedure will stop in finite steps for the discreted double obstacle problem. Following, we present the algorithm for solving the discreted obstacle problem.

Let \( V_h \subseteq H_0^1(\Omega) \) be a finite element space. The discrete admissible set is

\[
K_h = v_k \in V_h, \quad \varphi(x) \leq v_k(x) \leq \psi(x), \quad \text{for any node } x.
\]  

(14)

Then, the approximation of problem (4) is to find \( u_k \in K_h \), such that

\[
\int_{\Omega} \nabla u_k \cdot \nabla (v_k - u_k) \, dx \geq \int_{\Omega} f(v_k - u_k) \, dx \quad \forall v_k \in K_h.
\]  

(15)

Denote \( \phi_1, \phi_2, \ldots, \phi_N \), the basis functions at nodes \( x_1, x_2, \ldots, x_N \) of \( V_h \) and \( A \) be the \( N \times N \) matrix with entries \( a_{ij} = \int_{\Omega} \nabla \phi_i \cdot \nabla \phi_j \, dx \). The load vector \( f \) is \( f_i = \int_{\Omega} f \nabla \phi_i \, dx \). Now, we can rewrite (15) in matrix form: find \( u = (u_1, \ldots, u_N) \in \mathbb{R}^N \), \( \varphi(x_i) \leq u_i \leq \psi(x_i) \), such that

\[
\frac{1}{2} (Au, v - u) \geq (f, v - u),
\]  

(16)

for all \( v = (v_1, \ldots, v_N) \in \mathbb{R}^N \), \( \varphi(x_i) \leq v_i \leq \psi(x_i) \), \( i = 1, \ldots, N \).

**Remark 1.** As the author said in [1], their algorithm can be proved to be convergent in finite steps of iteration towards the exact solution of the discrete problem (2.7) in the same way as suggested in [13], the algorithm we proposed here also can be proved to be convergent in the same way.

Following, we propose the algorithm based on the idea described above for solving the problem (16).

**Algorithm 2.1 (One the other type).** Solving \( Au^0 = f \), we get the initial position of the elastic membrane \( u^0 \) without obstacle. Then, let \( k = 0, \ l = 1 \).

(Lower obstacle move)

(i) computing \( d_i^{(k)} = u_i^0 - \varphi(x_i), i = 1, 2, \ldots, N \), and \( d_{\min}^{(k)} = \min_{1 \leq i \leq N} d_i^{(k)} \)

(ii) if \( d_{\min}^{(k)} \geq 0, \Omega_1^{(k)} = \emptyset, \) and go to (v), else

\[
\Omega_1^{(k)} = \{ x_i | d_i(k) = d_{\min}^{(k)} \},
\]

(iii) for all \( i \in \Omega_1^{(k)} \), modify matrix \( A \) and load vector \( f \) by

\[
a_{ij} = 1, \quad a_{ij} = 0, \quad j \neq i, \quad f_i = \varphi(x_i),
\]

(iv) \( k = k + 1 \), solve \( Au^k = f \). Go to (i)

(Suppose it takes \( n \) steps to go to (v), so we get \( \Omega_1 = \Omega_1^{(0)} \cup \Omega_1^{(1)} \cup \cdots \cup \Omega_1^{(n)} \) and the \( u^{n+1} \). Then upper obstacle move).
Algorithm 2.2 (Up down type). Solving \( Au^0 = f \), we get the initial position of the elastic membrane \( u^0 \). Then, for \( k = 0, 1, \ldots \):

(i) computing \( d^{(2k)}_i = u^{2k} - \varphi(x_i) \), \( i = 1, 2, \ldots, N \), and \( d^{(2k)}_i = \min_{1 \leq i \leq N} d^{(2k)}_i \)

(ii) if \( d^{(2k)}_i \geq 0, \Omega^{(2k)}_1 = \emptyset \), and go to (v), else

\[
\Omega^{(2k)}_1 = \left\{ x_i \mid d^{(2k)}_i = d^{(2k)}_i \right\},
\]

(iii) for all \( i \in \Omega^{(2k)}_1 \), modify matrix \( A \) and load vector \( f \) by

\[
a_{ij} = 1, \quad a_{jj} = 0, \quad j \neq i, \quad f_i = \varphi(x_i),
\]

(iv) solving \( Au^{2k+1} = f \)

(v) computing \( d^{(2k+1)}_i = \psi(x_i) - u^{2k+1} \), \( i = 1, 2, \ldots, N \), and \( d^{(2k+1)}_i = \min_{1 \leq i \leq N} d^{(2k+1)}_i \)

(vi) if \( d^{(2k+1)}_i \geq 0, \Omega^{(2k+1)}_1 = \emptyset \), and \( \Omega^{(2k+1)}_1 \cup \Omega^{(2k+1)}_2 = \emptyset \), STOP, else go to (i); else

\[
\Omega^{(2k+1)}_2 = \left\{ x_i \mid d^{(2k+1)}_i = d^{(2k+1)}_i \right\},
\]

(vii) for all \( i \in \Omega^{(2k+1)}_2 \), modify matrix \( A \) and load vector \( f \) by

\[
a_{ij} = 1, \quad a_{jj} = 0, \quad j \neq i, \quad f_i = \varphi(x_i),
\]

(viii) solving \( Au^{2k+2} = f \).

Ultimately, we get the solution \( u^{2k} \), \( \Omega_1 = \Omega_1^{(0)} \cup \cdots \cup \Omega_1^{(2k)} \), \( \Omega_2 = \Omega_2^{(1)} \cup \cdots \cup \Omega_2^{(2k+1)} \).

In Algorithm 2.1 (Algorithm 2.2), we find very few new contact points in \( \Omega_1^{(k)} \), \( \Omega_2^{(l)} \), \( \Omega_1^{(e+1)} \left( \Omega_1^{(2k)} \cup \Omega_2^{(2k+1)} \right) \) in each iteration and need much iteration. To accelerate the algorithm, we introduce the small parameter \( \varepsilon > 0 \). We think the nodes are contact points if \( d^{(k)}_i \leq d^{(k)}_i + \varepsilon \), thus, we obtain useful algorithm through taking place of \( \Omega_1^{(k)} = \left\{ x_i \mid d^{(k)}_i = d^{(k)}_i \right\} \) by \( \Omega_1^{(k)} = \left\{ x_i \mid d^{(k)}_i \leq d^{(k)}_i + \varepsilon \right\} \).

In the above algorithm, we need to solve linear system for each iteration. We can use some know methods to solve the linear system, for instance, the fast Poisson solver, incomplete LU decomposition, preconditioned conjugate gradient method, multigrid method, and domain decomposition method.

Remark 2. The parameter \( \varepsilon \) in the algorithms determine the pace of “moving obstacle”, we do not set that it is too small to need much iteration and do not make it large to influence the precision of the algorithm as well.
For obtaining more exact solution as soon as possible, we can set the \(e\) larger in the beginning, and make it smaller in the last steps.

**Remark 3.** In above algorithms, the touch region which is found in one step, is regarded as the part of contact region all the times in the following steps. But the algorithms in \([13,14]\) are different, the touch point in one step may become non-contact point in the next steps.

### 3. Numerical result

In this section, one numerical example is given for the elastic–plastic torsion problem. It is seen that the contact region of the obstacle problem is approximated by implementing the new algorithm on the computer, in which we suppose \(e = 0.01\), e.g. suppose \(\Omega = [0, 1] \times [0, 1]\). Let \(\varphi(x,y) = -\text{dist}(x, \partial \Omega)\), \(\psi(x,y) = 0.2\), and set

\[
    f(x,y) = \begin{cases} 
        300, & \text{if } (x,y) \in S = \{(x,y) \in \Omega : |x-y| \leq 0.1 \text{ and } x \leq 0.3\}, \\
        -70 \exp(y)g(x), & \text{if } x \leq 1 - y \text{ and } (x,y) \not\in S, \\
        15 \exp(y)g(x), & \text{if } x > 1 - y \text{ and } (x,y) \not\in S, 
    \end{cases}
\]

where

\[
g(x) = \begin{cases} 
    6x, & \text{if } 0 < x \leq 1/6, \\
    2(1 - 3x), & \text{if } 1/6 < x \leq 1/3, \\
    6(x - 1/3), & \text{if } 1/3 < x \leq 1/2, \\
    2(1 - 3(x - 1/3)), & \text{if } 1/2 < x \leq 2/3, \\
    6(x - 2/3), & \text{if } 2/3 < x \leq 5/6, \\
    2(1 - 3(x - 2/3)), & \text{if } 5/6 < x \leq 1. 
\end{cases}
\]

Here we use the bilinear rectangle finite element space to approach \(H^1_0(\Omega)\). It is the 3D picture of the approximated solution \(u\) computed by “One the other” type algorithm in Fig. 2. In Figs. 3 and 4 the lower and upper coincidence sets are marked with star “*” and dot “.”, respectively. They all have the similar figure with Fig. 3 in Example 5.4 in \([14]\).
Table 1
Comparison of the two algorithm

<table>
<thead>
<tr>
<th>Algorithm type</th>
<th>Consumed time (s)</th>
<th>Compute times of $Au = f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>One the other</td>
<td>1275.1</td>
<td>65</td>
</tr>
<tr>
<td>Up down</td>
<td>916.8</td>
<td>46</td>
</tr>
</tbody>
</table>
Comparing the two types of algorithm, we find the consumed times of “Up down” type is 71.9% of “One the other” type’s. Detailed information is in Table 1.

References