

# 求解单侧障碍问题的自适应投影方法\*

钟艳丽, 严月月, 张守贵

(重庆师范大学 数学科学学院, 重庆 401331)

**摘要:**【目的】单侧障碍问题在变分不等式中具有重要的应用,但不存在或很难求其精确解,所以很有必要进行数值解法的研究。【方法】利用有限差分格式将障碍问题离散为一个线性互补问题,得到该问题的一个投影不动点算法。然后用投影方法得到了变参数的算法,并在迭代过程中自动调整参数,每一步迭代只需求解一个线性方程组。【结果】将障碍问题离散为一个有限维的线性互补问题,而该问题等价于投影问题,于是得到了求解障碍问题的自适应投影算法。【结论】最后用数值算例验证了算法的有效性,与固定参数的投影算法相比较。数值结果表明参数对自适应投影算法影响较小,而且该方法收敛速度更快。

**关键词:**单侧障碍;有限差分;线性互补;投影法;自适应法则

**中图分类号:**O241.82

**文献标志码:**A

**文章编号:**1672-6693(2018)01-0070-07

单侧障碍问题是一类经典的非线性问题,在数学和工程领域具有重要的理论意义和实用价值<sup>[1-3]</sup>。这类问题通常很难求得精确解,因此数值方法显得尤为重要。目前已有的方法主要是采用迭代算法<sup>[4-8]</sup>,文献[8]用有限差分法将求解的问题离散为线性互补问题,提出了一种简便的投影算法,然而该算法含有两个参数 $\beta > 0$ 和 $\eta \in (0, 2)$ ,为了算法收敛性,参数 $\eta$ 的最优范围容易确定为 $(1, 2)$ 。而参数 $\beta$ 太大或太小都将大大减缓收敛速度,本文针对这一算法中如何合理选取参数 $\beta$ 这一问题,提出了求解单侧障碍问题的自适应投影算法<sup>[9-13]</sup>。该方法根据上一步迭代数据自动调整参数 $\beta$ ,达到提高算法效率的目的。相关基本思路是参照文献[8]对单侧障碍问题离散,在基于线性互补问题的简单投影算法基础上,得到一个含可变参数 $\beta_n$ 的新方法及相应的参数自动调整办法,然后通过若干数值算例和已有方法进行了比较。

## 1 单侧障碍问题及其有限差分格式

有以下单侧障碍问题:

$$\begin{cases} -Lv(x) - \varphi(x) \geq 0, x \in \Omega, \\ v(x) \geq 0, x \in \Omega, \\ v(x)(Lv(x) + \varphi(x)) = 0, x \in \Omega, \\ v(x) = \psi(x), x \in \Gamma, \end{cases} \quad (1)$$

其中 $\Omega$ 是 $\mathbf{R}^n$ ( $n=1, 2$ )中的有界区域。当 $n=1$ 时,令 $\Omega=[a, b]$ ,则边界 $\Gamma$ 为 $a, b$ 两端点,算子 $L = \frac{\partial^2}{\partial x^2}$ 。当

$n=2$ 时, $\Omega$ 为平面有界闭区域,它的边界 $\Gamma = \partial\Omega, x = (x_1, x_2)$ ,算子 $L = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}$ 。如果已知函数 $\varphi \in L^2(\Omega), \psi$

$\geq 0$ 且 $\psi \in L^2(\Gamma)$ ,问题(1)等价于变分不等式问题:寻找 $v \in K, K := \{u \in H^1(\Omega) \mid u \geq 0, \text{a.e.在}\Omega\text{中}\}$ ,使它满足

$$\int_{\Omega} \nabla v \cdot \nabla(u - v) dx \geq \int_{\Omega} \varphi(u - v) dx, \forall u \in K, \text{由文献[2-3]知该变分不等式有唯一解,即问题(1)有唯一解。}$$

首先采用有限差分对问题(1)中的微分算子离散化得:

\* 收稿日期:2016-12-10 修回日期:2017-09-22 网络出版时间:2018-01-18 15:21

资助项目:国家自然科学基金(No.11471063)

第一作者简介:钟艳丽,女,研究方向为微分方程数值解法,E-mail: 815517318@qq.com;通信作者:张守贵,副教授,E-mail: shgzhang@cqnu.edu.cn

网络出版地址: <http://kns.cnki.net/kcms/detail/50.1165.N.20180118.1521.020.html>



那么在下次迭代时  $\beta_n$  要减少。即令  $\omega_n = \frac{\|\beta_n \mathbf{A}(\mathbf{u}^{(n+1)} - \mathbf{u}^{(n)})\|}{\|\mathbf{u}^{(n+1)} - \mathbf{u}^{(n)}\|}$ , 然后让  $\beta_{n+1} = \begin{cases} (1 + \tau_n)\beta_n, \omega_n < \frac{1}{1 + \mu} \\ \left(\frac{1}{1 + \tau_n}\right)\beta_n, \omega_n > 1 + \mu \\ \beta_n, \text{其他} \end{cases}$

为了分析算法收敛,构造一个非负序列  $\{\tau_n\}$ ,使其满足  $\sum_{n=0}^{\infty} \tau_n < +\infty$ 。下面给出了构造序列  $\{\tau_n\}$  的具体过程。

用  $c_n$  计算  $\beta_n$  改变的次数,令  $c_0 = 0$ , 则  $c_{n+1} = \begin{cases} c_n, \frac{1}{1 + \mu} \leq \omega_n \leq 1 + \mu \\ c_n + 1, \text{其他} \end{cases}$ 。采用如下方法得到非负序列  $\{\tau_n\}$ ,

$$\tau_n = \begin{cases} \mu, c_n < c_{n+1} \text{ 和 } c_{n+1} < c_{\max} \\ \frac{1}{(c_{n+1} - c_{\max})^2}, c_n < c_{n+1} \text{ 和 } c_{n+1} > c_{\max} \\ 0, \text{其他} \end{cases}$$

对于给定的整数  $c_{\max} > 0$ , 很容易知道由以上方法得到的非负序列

$\{\tau_n\}$  满足  $\sum_{n=0}^{\infty} \tau_n < +\infty$ 。

下面给出自适应投影算法的详细过程。

**算法 1** 第一步,任取初值  $\mathbf{u}^{(0)} \in \mathbf{R}^N, \eta \in (1, 2), \beta > 0, \epsilon > 0$ , 并取  $\beta_0 = \beta$ , 令  $n = 0$ ;

第二步,如果  $\|\mathbf{e}(\mathbf{u}^{(n)}, \beta_n)\|_{\infty} \leq \epsilon$ , 则迭代停止; 否则进行下一步;

第三步,求解方程组  $\mathbf{u}^{(n+1)} + \beta_n \mathbf{A} \mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \beta_n \mathbf{A} \mathbf{u}^{(n)} - \eta \mathbf{e}(\mathbf{u}^{(n)}, \beta_n)$ , 得  $\mathbf{u}^{(n+1)}$ ;

第四步,选择下一个参数  $\beta_{n+1}$ , 使得  $\frac{1}{1 + \tau_n} \beta_n \leq \beta_{n+1} \leq (1 + \tau_n) \beta_n$ , 令  $n := n + 1$ 。返回第二步。

在此算法中,由于  $\tau_n \geq 0$  并且  $\sum_{n=0}^{\infty} \tau_n < +\infty$ , 则有  $\prod_{n=0}^{\infty} (1 + \tau_n) < +\infty$ 。令  $C_{\tau} = \prod_{n=0}^{\infty} (1 + \tau_n)$ , 则有  $\beta_n \in$

$\left[\frac{1}{C_{\tau}} \beta_0, C_{\tau} \beta_0\right]$  是有界的。

**定理 1** 设  $\mathbf{u}^*$  为(3)式的解, 自适应投影算法产生的序列  $\{\mathbf{u}^{(n)}\}$  满足

$$\|\mathbf{u}^{(n+1)} - \mathbf{u}^* + \beta_{n+1} \mathbf{A}(\mathbf{u}^{(n+1)} - \mathbf{u}^*)\|^2 \leq (1 + \xi_n) \|\mathbf{u}^{(n)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n)} - \mathbf{u}^*)\|^2 - c \|\mathbf{e}(\mathbf{u}^{(n)}, \beta_n)\|^2, \quad (4)$$

其中  $\xi_n = 2\tau_n + \tau_n^2, c = \eta(2 - \eta)$ 。

**证明** 由自适应投影算法  $\mathbf{u}^{(n+1)} + \beta_n \mathbf{A} \mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \beta_n \mathbf{A} \mathbf{u}^{(n)} - \eta \mathbf{e}(\mathbf{u}^{(n)}, \beta_n)$  及文献[8]中引理 2 的结论

$$\langle \mathbf{u} - \mathbf{u}^* + \beta \mathbf{A}(\mathbf{u} - \mathbf{u}^*), \mathbf{e}(\mathbf{u}, \beta) \rangle \geq \|\mathbf{e}(\mathbf{u}, \beta)\|^2 + \beta(\mathbf{u} - \mathbf{u}^*)^T \mathbf{A}(\mathbf{u} - \mathbf{u}^*)$$

则有  $\langle \mathbf{u} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u} - \mathbf{u}^*), \mathbf{e}(\mathbf{u}, \beta_n) \rangle \geq \|\mathbf{e}(\mathbf{u}, \beta_n)\|^2 + \beta_n(\mathbf{u} - \mathbf{u}^*)^T \mathbf{A}(\mathbf{u} - \mathbf{u}^*)$ , 从而

$$\begin{aligned} \|\mathbf{u}^{(n+1)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n+1)} - \mathbf{u}^*)\|^2 &= \|\mathbf{u}^{(n)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n)} - \mathbf{u}^*) - \eta \mathbf{e}(\mathbf{u}^{(n)}, \beta_n)\|^2 = \\ &\|\mathbf{u}^{(n)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n)} - \mathbf{u}^*)\|^2 - 2\eta \langle \mathbf{u}^{(n)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n)} - \mathbf{u}^*), \mathbf{e}(\mathbf{u}^{(n)}, \beta_n) \rangle + \eta^2 \|\mathbf{e}(\mathbf{u}^{(n)}, \beta_n)\|^2 \leq \\ &\|\mathbf{u}^{(n)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n)} - \mathbf{u}^*)\|^2 - 2\eta \beta_n (\mathbf{u}^{(n)} - \mathbf{u}^*)^T \mathbf{A}(\mathbf{u}^{(n)} - \mathbf{u}^*) - 2\eta \|\mathbf{e}(\mathbf{u}^{(n)}, \beta_n)\|^2 + \eta^2 \|\mathbf{e}(\mathbf{u}^{(n)}, \beta_n)\|^2 = \\ &\|\mathbf{u}^{(n)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n)} - \mathbf{u}^*)\|^2 - 2\eta \beta_n (\mathbf{u}^{(n)} - \mathbf{u}^*)^T \mathbf{A}(\mathbf{u}^{(n)} - \mathbf{u}^*) - \eta(2 - \eta) \|\mathbf{e}(\mathbf{u}^{(n)}, \beta_n)\|^2, \end{aligned}$$

因为  $\mathbf{A}$  是正定的, 所以

$$\|\mathbf{u}^{(n+1)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n+1)} - \mathbf{u}^*)\|^2 \leq \|\mathbf{u}^{(n)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n)} - \mathbf{u}^*)\|^2 - \eta(2 - \eta) \|\mathbf{e}(\mathbf{u}^{(n)}, \beta_n)\|^2, \quad (5)$$

又因为  $0 < \beta_{n+1} \leq (1 + \tau_n) \beta_n$ , 所以有

$$\|\mathbf{u}^{(n+1)} - \mathbf{u}^* + \beta_{n+1} \mathbf{A}(\mathbf{u}^{(n+1)} - \mathbf{u}^*)\|^2 \leq (1 + \tau_n)^2 \|\mathbf{u}^{(n+1)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n+1)} - \mathbf{u}^*)\|^2, \quad (6)$$

由(5), (6)式有:

$$\|\mathbf{u}^{(n+1)} - \mathbf{u}^* + \beta_{n+1} \mathbf{A}(\mathbf{u}^{(n+1)} - \mathbf{u}^*)\|^2 \leq (1 + \tau_n)^2 \|\mathbf{u}^{(n)} - \mathbf{u}^* + \beta_n \mathbf{A}(\mathbf{u}^{(n)} - \mathbf{u}^*)\|^2 - \eta(2 - \eta) \|\mathbf{e}(\mathbf{u}^{(n)}, \beta_n)\|^2. \quad \text{证毕}$$

**定理 2** 自适应投影算法产生的序列  $\{\mathbf{u}^{(n)}\}$  收敛于(3)的解  $\mathbf{u}^*$ , 即  $\lim_{n \rightarrow \infty} \mathbf{u}^{(n)} = \mathbf{u}^*$ 。

**证明** 因为  $\sum_{n=0}^{\infty} \xi_n < +\infty$ , 显然有  $\prod_{n=0}^{\infty} (1 + \xi_n) < +\infty$ , 令  $C_s = \sum_{i=0}^{\infty} \xi_i, C_p = \prod_{i=n_0}^{\infty} (1 + \xi_i)$ . 设  $\tilde{u}$  是问题(3)的一个解, 利用(4)式可以得到:

$$\|u^{(n+1)} - \tilde{u} + \beta_{n+1}A(u^{(n+1)} - \tilde{u})\|^2 \leq \prod_{i=n_0}^n (1 + \xi_i) \|u^{(n_0)} - \tilde{u} + \beta_{n_0}A(u^{(n_0)} - \tilde{u})\|^2 \leq C_p \|u^{(n_0)} - \tilde{u} + \beta_{n_0}A(u^{(n_0)} - \tilde{u})\|^2,$$

因此, 存在常数  $C > 0$ , 有:

$$\|u^{(n)} - \tilde{u} + \beta_n A(u^{(n)} - \tilde{u})\|^2 \leq C, \forall n \geq 0, \tag{7}$$

显然  $\{u^{(n)}\}$  是有界的, 利用(4)和(6)式, 则:

$$c \sum_{n=n_0}^{\infty} \|e(u^{(n)}, \beta_n)\|^2 \leq \|u^{(n_0)} - \tilde{u} + \beta_{n_0}A(u^{(n_0)} - \tilde{u})\|^2 + \sum_{n=n_0}^{\infty} \xi_n \|u^{(n)} - \tilde{u} + \beta_n A(u^{(n)} - \tilde{u})\|^2 \leq C + C \sum_{n=n_0}^{\infty} \xi_n \leq (1 + C_s)C,$$

其中  $c > 0$  是一个常数, 由上式可知  $\lim_{n \rightarrow \infty} e(u^{(n)}, \beta_n) = 0$ .

设  $u^*$  是  $\{u^{(n)}\}$  的一个聚点且子序列  $\{u^{(n_j)}\}$  收敛于  $u^*$ , 因为  $e(u, \beta_n)$  是连续的, 则有  $e(u^*, \beta_n) = \lim_{j \rightarrow \infty} e(u^{(n_j)}, \beta_n) = 0$ , 所以  $u^*$  是问题(3)的一个解.

假设  $u^{**} \neq u^*$  是  $\{u^{(n)}\}$  的另一个聚点, 因为  $u^*$  是  $\{u^{(n)}\}$  的聚点, 所以存在  $n_1 > n_0$ , 使得

$$\|u^{(n_1)} - u^* + \beta_{n_1}A(u^{(n_1)} - u^*)\| \leq \frac{1}{2\sqrt{C_p}} \|u^{**} - u^*\|.$$

由(4)式可知, 对任意  $n > n_1$ , 有:

$$\|u^{(n)} - u^*\| \leq \|u^{(n)} - u^* + \beta_n A(u^{(n)} - u^*)\| \leq \left(\prod_{i=n_1}^{n-1} (1 + \xi_i)\right)^{\frac{1}{2}} \|u^{(n_1)} - u^* + \beta_{n_1}A(u^{(n_1)} - u^*)\| \leq \sqrt{C_p} \|u^{(n_1)} - u^* + \beta_{n_1}A(u^{(n_1)} - u^*)\| \leq \frac{1}{2} \|u^{**} - u^*\|.$$

因此, 有  $\|u^{(n)} - u^{**}\| \geq \|u^{**} - u^*\| - \|u^{(n)} - u^*\| \geq \frac{1}{2} \|u^{**} - u^*\| > 0, \forall n \geq n_1$ , 显然与假设矛盾, 因此  $u^{**}$  不是  $\{u^{(n)}\}$  的聚点, 即  $\{u^{(n)}\}$  收敛于  $u^*$ .

### 3 算例分析

为了检验自适应投影算法的性能, 对它进行数值验证, 算法中的参数分别取  $\eta = 1.8, \mu = 1, \epsilon = 10^{-4}$ .

**例 1** 考虑  $\Omega = [0, 4]$  中的一维问题<sup>[8,13]</sup>

$$\begin{cases} -Lv(x) - \varphi(x) \geq 0, x \in \Omega \\ v(x) \geq 0, x \in \Omega \\ v(x)(Lv(x) + \varphi(x)) = 0, x \in \Omega \\ v(x) = 0, x \in \Gamma \end{cases}$$

$$\text{其中, } v(x) = \begin{cases} -\frac{x^2}{2} + (2 - \sqrt{2})x, & x \in [0, 1] \\ \frac{x^2}{2} - \sqrt{2}x + 1, & x \in (1, \sqrt{2}] \\ 0, & x \in (\sqrt{2}, 4] \end{cases}, \varphi(x) = \begin{cases} -1, & x \in [0, 1] \\ 1, & x \in (1, 4] \end{cases}$$

为解析解, 图 1 给出了当步长  $h = 0.02$  的数值解与精确解结果, 它们是吻合的.

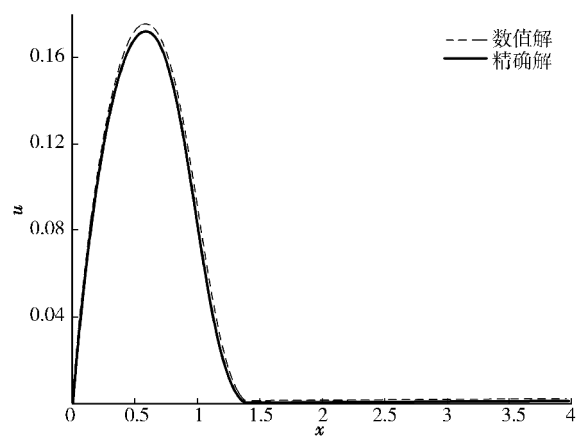


图 1 例 1 数值解和精确解结果

Fig. 1 Results of numerical solution and exact solution for example 1

为了考察不同参数  $\beta$  和不同网格剖分对算法收敛性的影响,表 1 和表 2 分别对本文方法和文献[8]中的算法所需的迭代次数和 CPU 时间(单位:s)进行了比较。结果表明参数  $\beta$  对自适应投影算法影响较小,该方法的迭代次数和 CPU 时间都比文献[8]中的算法少,而且算法比较稳定。

例 2 考虑在正方形区域  $\Omega = (-1.5, 1.5) \times (-1.5, 1.5)$  中的二维问题<sup>[8]</sup>

$$\begin{cases} -Lv(x) + 2 \geq 0, x \in \Omega, \\ v(x) \geq 0, x \in \Omega, \\ v(x)(Lv(x) - 2) = 0, x \in \Omega, \\ v(x) = \psi(x), x \in \Gamma. \end{cases}$$

其中 Dirichlet 边界条件由解析解  $v = \begin{cases} \frac{r^2}{2} - \ln(r) - \frac{1}{2}, r > 1 \\ 0, r \leq 1 \end{cases}$  确定,  $r = \sqrt{x_1^2 + x_2^2}$ 。图 2 给出了当步长  $h = 0.05$

自由边界( $v = 0$  和  $v > 0$  的分界线)的数值解与精确解结果,它们也是吻合的。

表 1 两种算法随步长变化所需迭代次数情况

Tab. 1 Number of iteration with respect to step  $h$  for each method

$\beta$	改进投影算法			自适应投影算法		
	$h=0.04$	$h=0.02$	$h=0.01$	$h=0.04$	$h=0.02$	$h=0.01$
$10^{-3}$	>200	>200	>200	34	46	44
$10^{-2}$	76	101	100	32	34	34
$10^{-1}$	>200	>200	>200	29	31	31
$10^0$	>200	>200	>200	31	33	33
$10^1$	>200	>200	>200	33	36	34
$10^2$	>200	>200	>200	33	52	43
$10^3$	>200	>200	>200	39	42	41

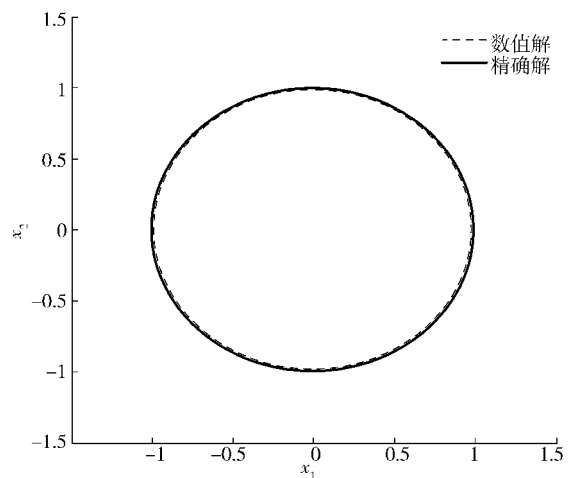


图 2 例 2 自由边界数值解和精确解结果  
Fig. 2 Free boundary of numerical solution and exact solution for example 2

表 2 两种算法随步长变化所需 CPU 时间情况

Tab. 2 CPU time with respect to step  $h$  for each method

$\beta$	改进投影算法			自适应投影算法		
	$h=0.04$	$h=0.02$	$h=0.01$	$h=0.04$	$h=0.02$	$h=0.01$
$10^{-3}$	>0.012 64	>0.015 35	>0.017 08	0.004 68	0.002 72	0.004 57
$10^{-2}$	0.004 73	0.011 25	0.007 96	0.001 66	0.002 81	0.002 47
$10^{-1}$	>0.015 24	>0.011 17	>0.015 09	0.001 41	0.003 46	0.002 21
$10^0$	>0.011 99	>0.024 57	>0.018 06	0.003 67	0.001 92	0.002 35
$10^1$	>0.010 74	>0.014 69	>0.021 67	0.002 17	0.002 09	0.002 48
$10^2$	>0.014 12	>0.014 08	>0.020 06	0.001 66	0.002 95	0.003 41
$10^3$	>0.014 21	>0.011 44	>0.016 68	0.001 90	0.002 41	0.003 02

对不同参数  $\beta$  和不同网格剖分,表 3 和表 4 分别给出了本文方法和文献[8]中的算法所需迭代次数和 CPU

时间。结果同样表明参数  $\beta$  对自适应投影算法影响较小,自适应投影算法的迭代次数和 CPU 时间都比文献[8]中的算法少得多,而且算法也很稳定。

表 3 两种算法随步长变化所需迭代次数情况

Tab. 3 Number of iteration with respect to step  $h$  for each method

$\beta$	改进投影算法			自适应投影算法		
	$h=0.2$	$h=0.1$	$h=0.05$	$h=0.2$	$h=0.1$	$h=0.05$
$10^{-3}$	185	184	183	19	25	31
$10^{-2}$	27	28	51	17	20	34
$10^{-1}$	48	160	>200	18	28	34
$10^0$	>200	>200	>200	16	30	36
$10^1$	>200	>200	>200	22	34	43
$10^2$	>200	>200	>200	21	34	51
$10^3$	>200	>200	>200	26	40	45

表 4 两种算法随步长变化所需 CPU 时间情况

Tab. 4 CPU time with respect to step  $h$  for each method

$\beta$	改进投影算法			自适应投影算法		
	$h=0.2$	$h=0.1$	$h=0.05$	$h=0.2$	$h=0.1$	$h=0.05$
$10^{-3}$	0.154 65	5.425 13	153.625 84	0.020 40	0.768 76	26.774 53
$10^{-2}$	0.023 05	0.796 73	43.015 95	0.015 60	0.688 17	29.458 08
$10^{-1}$	0.040 92	4.647 09	>175.163 45	0.016 57	0.869 72	29.236 07
$10^0$	>0.163 38	>6.010 14	>173.660 75	0.013 03	0.887 92	31.250 32
$10^1$	>0.158 70	>3.165 36	>177.087 63	0.018 45	1.011 88	37.249 13
$10^2$	>0.158 46	>5.673 44	>178.026 27	0.016 99	0.996 52	44.883 13
$10^3$	>0.157 51	>5.732 01	>170.177 90	0.023 16	1.187 29	40.086 81

## 4 结论

本文提出了求解单侧障碍问题的自适应投影算法,该方法先采用中心差分格式将问题离散为一个线性互补问题,然后在改进投影迭代算法基础上进一步改进,给出自动调整参数的法则。数值算例结果表明参数对自适应投影算法影响较小,该方法明显加快了收敛速度。

### 参考文献:

- [1] 韩继业,修乃华,戚厚铎.非线性互补理论与算法[M].上海:上海科学技术出版社,2006:17-141.  
HAN J Y, XIU N H, QI H D. Nonlinear complementary theory and algorithm[M]. Shanghai: Shanghai Science and Technology Press, 2006, 17-141.
- [2] 韩渭敏,程晓良.变分不等式简介[M].北京:高等教育出版社,2007.  
HAN W M, CHENG X L. Introduction to variational inequation[M]. Beijing: Higher Education Press, 2007.
- [3] GLOWINSKI G. Numerical methods for nonlinear variational problems[M]. Berlin: Springer-Verlag, 2008.
- [4] LIAN X P, CEN Z D, CHENG X L. Some iterative algorithms for obstacle problems[J]. INT J COMPUT MATH, 2010, 87(11): 2493-2502.

- [5] 张守贵.自由边界问题的线性互补投影迭代算法[J].西南师范大学学报(自然科学版),2013,38(7):15-19.  
ZHANG S G.The linear complementary-projection iterative algorithm for seepage with free boundary problem[J].Journal of Southwest China Normal University(Natural Science Edition),2013,38(7):15-19.
- [6] 张守贵.自由边界问题的自适应预测-校正算法[J].西南师范大学学报(自然科学版),2014,39(9):1-5.  
ZHANG S G.On a self-adaptive prediction-correct algorithm for free boundary problem[J].Journal of Southwest China Normal University(Natural Science Edition),2014,39(9):1-5.
- [7] 张守贵.求解自由边界问题的投影收缩算法[J].重庆师范大学学报(自然科学版),2015,32(2):50-52.  
ZHANG S G.A projection and contraction algorithm for the free boundary problem[J].Journal of Chongqing Normal University(Natural Science),2015,32(2):50-52
- [8] 严月月,钟艳丽,张守贵.自由边界问题的改进投影算法[J].重庆师范大学学报(自然科学版),2017,34(1):70-73.  
YAN Y Y,ZHONG Y L,ZHANG SG.An Improved Projection Algorithm for the Free Boundary Problem[J].Journal of Chongqing Normal University (Natural Science),2017,34(1):70-73.
- [9] HE B S, LIAO L Z, WANG S L. Self-adaptive operator splitting methods for monotone variational inequalities[J]. Mumer Math,2003,94(4):715-737.
- [10] HAN D R. Solving linear variational inequality problems by a self adaptive projection method[J].Appl Math Comput,2006,182(2):1765-1771.
- [11] LI M, BNOUHACHEM A. A modified inexact operator splitting method for monotone variational inequalities [J]. Glob Optim,2008,41(3):417-426.
- [12] LIM, BNOUHACHEM A, KHALFAOUI M, et al. A modified inexact implicit method for mixed variational Inequalities[J].Comput Appl Math,2010,234(12):3356-3365.
- [13] MERMRI E B, HAN W. Numerical approximation of a unilateral obstacle problem [J]. J Optim Theory Appl,2012,153(1):177-194.

## A Self-Adaption Projection Method for the Unilateral Obstacle Problem

ZHONG Yanli, YAN Yueyue, ZHANG Shougui

(School of Mathematics Science, Chongqing Normal University, Chongqing 401331, China)

**Abstract:** [Purposes]The unilateral obstacle problem plays an important application in variation inequalities and the exact solution does not exist or is difficult to be obtained, so it is necessary to find a numerical method for the problem. [Methods]For this problem, a projection method is proposed for the numerical solution of the problem by using the finite difference method, and formulated as a linear complementary problem. Then an algorithm with variable parameters is obtained by the projection method, and the parameters are adjusted automatically in the iterative process. The method only needs to solve a system of linear equations for each iteration. [Findings]The problem is discredited as a finite dimensional linear complementary problem which is equivalent to projection problem. Then a self-adaption projection problem for the solution is obtained. The detail process of the algorithm is presented. It is very simple for us to prove the convergence results with projection properties. [Conclusion]Finally, some numerical examples are used to verify the validity of the algorithm, compared with the projection algorithm with fixed parameters. The numerical results show that the parameter has little effect on the self-adaptive projection method. Furthermore, our method converges faster.

**Keywords:** unilateral obstacle; finite difference; linear complementary; projection algorithm; self-adaption rule

(责任编辑 黄颖)