

Signorini 问题的无网格边界径向点插值法

任彦霖, 李小林

(重庆师范大学 数学学院, 重庆 401331)

摘要: 对一类带非线性互补边界条件的 Signorini 问题, 提出了一种边界型无网格数值方法。该方法首先利用投影算子来处理非线性边界不等式条件, 然后将 Signorini 问题归化为边界积分方程, 并用无网格边界径向点插值法求解。理论证明了该方法的收敛性。数值算例表明该算法具有较高的收敛率和计算效率。

关键词: 无网格法; Signorini 问题; 投影迭代算法; 径向基点插值法; 边界积分方程

中图分类号: O241.82

文献标志码: A

文章编号: 1672-6693(2015)03-0000-00

Signorini 问题出现在许多建立模型的实际工程应用中^{[1][2]}, 该问题的部分边界条件是以函数及其法向导数在一定的不等式约束下交替出现的, 且交替变化的位置是未知的, 这也正是解决这类问题的困难之处。直到现在, 国内外的大多数研究成果是把该问题转化为变分不等式进行理论分析, 再用有限差分法、有限元法或边界元法^{[3][4][5][6]}等数值求解。虽然这些方法已成功应用于许多工程上, 但其有效性依赖于网格的生成, 这样工作量较大, 计算时间较长。

无网格方法^{[6][7][8]}采用基于点的近似, 不需要网格的初始划分和重构, 可以消除网格, 不仅可以保证计算的精度, 还可以减小计算的难度。边界径向点插值法将边界积分方程和基于径向基函数的径向基点插值法相结合, 同时具有边界元法降维和无网格法不需要划分网格的优势^{[9][10]}。另外, 由于径向基点插值具有 Delta 函数性质, 边界径向点插值法克服了边界点法中边界条件不易施加的困难。

1 边界径向点插值

为避免采用多项式基点插值法所引起的奇异性, 径向基函数被采用以形成径向基点插值法形函数^[11]。添加多项式的径向基点插值可表示为

$$v(s) = \sum_{i=1}^n \varphi_i(s) a_i + \sum_{j=1}^m p_j(s) b_j = \boldsymbol{\Phi}^T(s) \mathbf{a} + \mathbf{P}^T(s) \mathbf{b}, \quad (1)$$

式中 s 是 Γ_i 上的一个曲线坐标, 其中边界 Γ 由分段光滑片段 $\Gamma_i (i=1, 2, \dots, N_\Gamma)$ 组成。 $p_j(s) (j=1, 2, \dots, m)$ 是单项式基函数, $\varphi_i(s) (i=1, 2, \dots, n)$ 为径向基函数, a_i 和 b_j 为待定常数, $\mathbf{a} = [a_1, a_2, \dots, a_n]^T$, $\mathbf{b} = [b_1, b_2, \dots, b_m]^T$, $\mathbf{P}^T(s) = [p_1(s), p_2(s), \dots, p_m(s)]$, $\boldsymbol{\Phi}^T(s) = [\varphi_1(s), \varphi_2(s), \dots, \varphi_n(s)]$ 。

这里我们选用 MQ 函数^[11] $p_j(s) = \sqrt{\beta^2 + |s - s_j|^2} (\beta > 0)$ 作为径向基函数。

设点 s 的影响域内有 n 个节点 $s_k (k=1, 2, \dots, n)$, 则

$$\mathbf{v} = \boldsymbol{\Phi} \mathbf{a} + \mathbf{P} \mathbf{b}, \quad (2)$$

其中 $\mathbf{v} = [v(s_1), v(s_2), \dots, v(s_n)]^T$, $\boldsymbol{\Phi} = [\boldsymbol{\Phi}(s_1), \boldsymbol{\Phi}(s_2), \dots, \boldsymbol{\Phi}(s_n)]^T$, $\mathbf{P} = [\mathbf{P}(s_1), \mathbf{P}(s_2), \dots, \mathbf{P}(s_n)]^T$ 。

由于式(2)中有 $n+m$ 个未知量, n 个方程, 还需要添加 m 个方程

$$\sum_{i=1}^n p_j(s_i) a_i = \mathbf{P}^T \mathbf{a} = \mathbf{0}, \quad j=1, 2, \dots, m. \quad (3)$$

由式(2)、(3)整理可得 $\mathbf{b} = \mathbf{S}_b \mathbf{v}$, $\mathbf{a} = \mathbf{S}_a \mathbf{v}$ 。其中 $\mathbf{S}_b = [\mathbf{P}^T \boldsymbol{\Phi}^{-1} \mathbf{P}]^{-1} \mathbf{P}^T \boldsymbol{\Phi}^{-1} \mathbf{v}$, $\mathbf{S}_a = \boldsymbol{\Phi}^{-1} - \boldsymbol{\Phi}^{-1} \mathbf{P} \mathbf{S}_b$ 。

因此, 式(1)可表示为

$$v(s) = [\boldsymbol{\Phi}^T(s) \mathbf{S}_a + \mathbf{P}^T(s) \mathbf{S}_b] \mathbf{v} = \boldsymbol{\Phi} \mathbf{v}, \quad (4)$$

其中径向基点插值法形函数

$$\boldsymbol{\Phi}(s) = \boldsymbol{\Phi}^T(s) \mathbf{S}_a + \mathbf{P}^T(s) \mathbf{S}_b = (\phi_1(s), \phi_2(s), \dots, \phi_n(s)) \quad (5)$$

* 收稿日期: 修回日期:

作者简介: 任彦霖, 女, 研究方向为微分方程数值解, E-mail:374940896@qq.com; 通讯作者: 李小林, E-mail:lxlmath@163.com

2 Signorini 问题的无网格方法

2.1 投影迭代格式

考虑如下 Signorini 问题:

$$\begin{cases} \Delta u = 0, \text{ in } \Omega, \\ u = \bar{u}, \text{ on } \Gamma_D, \\ q := \frac{\partial u}{\partial \mathbf{n}} = \bar{q}, \text{ on } \Gamma_N, \end{cases} \quad (6)$$

和 Signorini 边界条件

$$u \leq h, q \leq g, (u-h)(q-g) = 0, \text{ on } \Gamma_S \quad (7)$$

其中 Ω 是 \square^2 中的一有界开区域, 边界 $\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_S$ 且 $\Gamma_S \neq \emptyset$ 。 h 和 g 均为 Signorini 边界 Γ_S 上的已知函数, \bar{u} 和 \bar{q} 分别为 Dirichlet 边界 Γ_D 和 Neumann 边界 Γ_N 上的已知函数。

为了处理 Signorini 边界条件 (7), 定义 $[a]_- = \min(0, a), a \in \square$ 。则条件 (7) 等价于

$$(u-h) - [(u-h) - c(q-g)]_- = 0, \text{ on } \Gamma_S, \quad (8)$$

其中 c 是任意正常数。

利用不动点方程 (8) 引入 Signorini 边界 Γ_S 上的残量函数

$$R(u, q) = (u-h) - [(u-h) - c(q-g)]_-, \text{ on } \Gamma_S. \quad (9)$$

从而构造 Γ_S 上的方程

$$(u-h) + c(q-g) = (u-h) + c(q-g) - \omega R(u, q), \text{ on } \Gamma_S, \quad (10)$$

其中 $\omega \in (0, 2)$ 。显然, 方程 (10) 的解 (u, q) 等价于非光滑连续函数 $R(u, q) = 0$ 的一个零点。因此, 求解 Laplace 算子方程的 Signorini 问题 (6) 和 (7) 的显式投影迭代格式如下

$$\begin{cases} \Delta u^{(k+1)} = 0, \text{ in } \Omega, \\ u^{(k+1)} = \bar{u}, \text{ on } \Gamma_D, \\ q^{(k+1)} = \bar{q}, \text{ on } \Gamma_N, \\ u^{(k+1)} + cq^{(k+1)} = u^{(k)} + cq^{(k)} - \omega R(u^{(k)}, q^{(k)}), \text{ on } \Gamma_S. \end{cases} \quad (11)$$

通过求解 (11) 我们可以得到 Γ_S 上的未知量 $u^{(k+1)}$ 及其法向导数 $q^{(k+1)}$ ($k = 0, 1, 2, \dots$)。

2.2 边界径向点插值法

在问题 (11) 中, 边界上的位势及其法向导数之间的关系可由以下边界积分方程表示

$$\int_{\Gamma} u^*(\mathbf{x}, \mathbf{y}) q^{(k+1)}(\mathbf{y}) d\Gamma - \int_{\Gamma} q^*(\mathbf{x}, \mathbf{y}) [u^{(k+1)}(\mathbf{y}) - u^{(k+1)}(\mathbf{x})] d\Gamma = 0, \mathbf{x} \in \Gamma, \quad (12)$$

其中 $u^*(\mathbf{x}, \mathbf{y}) = -\frac{1}{2\pi} \ln|\mathbf{x} - \mathbf{y}|$, $q^*(\mathbf{x}, \mathbf{y}) = \partial u^*(\mathbf{x}, \mathbf{y}) / \partial \mathbf{n}_y$ 。

设 $\{\mathbf{x}_i\}_{i=1}^N$ 是由 Γ 上的 N 个边界节点组成的集合。由式 (4), 函数值 $u^{(k+1)}$ 及其法向导数 $q^{(k+1)}$ 的无网格插值形式可表示为

$$u^{(k+1)}(\mathbf{x}) = \sum_{j=1}^N \phi_j(\mathbf{x}) u_j^{(k+1)}, \quad q^{(k+1)}(\mathbf{x}) = \sum_{j=1}^N \phi_j(\mathbf{x}) q_j^{(k+1)}, \quad (13)$$

其中 $u_j^{(k+1)} = u^{(k+1)}(\mathbf{x}_j)$, $q_j^{(k+1)} = q^{(k+1)}(\mathbf{x}_j)$, 这是由边界径向点插值形函数具有 δ 函数性质决定的。

对所有的边界点 $\{\mathbf{x}_i\}_{i=1}^N$, 将式 (13) 代入式 (12), 可得到矩阵形式:

$$\mathbf{H}\mathbf{U} = \mathbf{G}\mathbf{Q}, \quad (14)$$

其中 $\mathbf{U} = (u_1^{(k+1)}, u_2^{(k+1)}, \dots, u_N^{(k+1)})^T$, $\mathbf{Q} = (q_1^{(k+1)}, q_2^{(k+1)}, \dots, q_N^{(k+1)})^T$, $G_{ij} = \int_{\Gamma} u^*(\mathbf{x}_i, \mathbf{y}) \phi_j(\mathbf{y}) d\Gamma$,

$$H_{ij} = \int_{\Gamma} q^*(\mathbf{x}_i, \mathbf{y}) [\phi_j(\mathbf{y}) - \phi_j(\mathbf{x}_i)] d\Gamma.$$

利用问题(11)中给出的 Dirichlet 和 Neumann 边界条件, 将式(14)转化成如下的分块形式

$$\begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \mathbf{H}_{13} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \mathbf{H}_{23} \\ \mathbf{H}_{31} & \mathbf{H}_{32} & \mathbf{H}_{33} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{U}}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & \mathbf{G}_{13} \\ \mathbf{G}_{21} & \mathbf{G}_{22} & \mathbf{G}_{23} \\ \mathbf{G}_{31} & \mathbf{G}_{32} & \mathbf{G}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \bar{\mathbf{Q}}_2 \\ \mathbf{Q}_3 \end{bmatrix}, \quad (15)$$

其中向量 $\bar{\mathbf{U}}_1 = (u_1^{(k+1)}, u_2^{(k+1)}, \dots, u_{N_d}^{(k+1)})^T$, $\bar{\mathbf{Q}}_2 = (q_{N_d+1}^{(k+1)}, q_{N_d+2}^{(k+1)}, \dots, q_{N_d+N_n}^{(k+1)})^T$ 是已知的, 而向量 $\mathbf{U}_2 = (u_{N_d+1}^{(k+1)}, u_{N_d+2}^{(k+1)}, \dots, u_{N_d+N_n}^{(k+1)})^T$, $\mathbf{Q}_1 = (q_1^{(k+1)}, q_2^{(k+1)}, \dots, q_{N_d}^{(k+1)})^T$, $\mathbf{U}_3 = (u_{N_d+N_n+1}^{(k+1)}, u_{N_d+N_n+2}^{(k+1)}, \dots, u_N^{(k+1)})^T$ 及 $\mathbf{Q}_3 = (q_{N_d+N_n+1}^{(k+1)}, q_{N_d+N_n+2}^{(k+1)}, \dots, q_N^{(k+1)})^T$ 是未知的. N_d 和 N_n 分别是边界 Γ_D 和 Γ_N 的节点数目.

另一方面, 将 $\{\mathbf{x}_i\}_{i=N_d+N_n+1}^N \subset \Gamma_S$ 代入问题(11)中的最后一个方程. 整理化简后, 可以表示成矩阵形式

$$\mathbf{U}_3 = \bar{\mathbf{F}} - c\mathbf{Q}_3, \quad (16)$$

其中 $\bar{\mathbf{F}} = (f_1, f_2, \dots, f_{N_n})^T$ 且 $f_i = u_i^{(k)} + cq_i^{(k)} - \omega R(u_i^{(k)}, q_i^{(k)})$.

将(16)代入(15), 并将所有的未知量移到等式的左边, 所有的已知量移到右边, 我们可以得到一个 $N \times N$ 线性方程组

$$\begin{bmatrix} -\mathbf{G}_{11} & \mathbf{H}_{12} & -c\mathbf{H}_{13} - \mathbf{G}_{13} \\ -\mathbf{G}_{21} & \mathbf{H}_{22} & -c\mathbf{H}_{23} - \mathbf{G}_{23} \\ -\mathbf{G}_{31} & \mathbf{H}_{32} & -c\mathbf{H}_{33} - \mathbf{G}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{U}_2 \\ \mathbf{Q}_3 \end{bmatrix} = \begin{bmatrix} -\mathbf{H}_{11} & \mathbf{G}_{12} & -\mathbf{H}_{13} \\ -\mathbf{H}_{21} & \mathbf{G}_{22} & -\mathbf{H}_{23} \\ -\mathbf{H}_{31} & \mathbf{G}_{32} & -\mathbf{H}_{33} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{U}}_1 \\ \bar{\mathbf{Q}}_2 \\ \bar{\mathbf{F}} \end{bmatrix}, \quad (17)$$

由式(17)可以通过求解 N 个方程, 从而得到 N 个未知量 \mathbf{Q}_1 , \mathbf{U}_2 和 \mathbf{Q}_3 , 再将 \mathbf{Q}_3 代入式(16), 可以很容易得到未知量 \mathbf{U}_3 .

2.3 算法步骤

在每一步迭代过程中, 都需要计算(17). 为了提高计算效率和精度, 将求解(17)转化为计算 $\mathbf{A}\mathbf{b}$, 其中 $\mathbf{b} = [\bar{\mathbf{U}}_1^T, \bar{\mathbf{Q}}_2^T, \bar{\mathbf{F}}^T]^T$

$$\mathbf{A} = \begin{bmatrix} -\mathbf{G}_{11} & \mathbf{H}_{12} & -c\mathbf{H}_{13} - \mathbf{G}_{13} \\ -\mathbf{G}_{21} & \mathbf{H}_{22} & -c\mathbf{H}_{23} - \mathbf{G}_{23} \\ -\mathbf{G}_{31} & \mathbf{H}_{32} & -c\mathbf{H}_{33} - \mathbf{G}_{33} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{H}_{11} & \mathbf{G}_{12} & -\mathbf{H}_{13} \\ -\mathbf{H}_{21} & \mathbf{G}_{22} & -\mathbf{H}_{23} \\ -\mathbf{H}_{31} & \mathbf{G}_{32} & -\mathbf{H}_{33} \end{bmatrix}$$

则在迭代过程中, 系数矩阵 \mathbf{A} 只需要计算一次.

Signorini 问题的无网格算法步骤可归纳如下:

第一步: 假定在 Signorini 边界 Γ_S 上的初始条件为 $u^{(0)} = h$, 由边值问题

$$\begin{cases} \Delta u^{(0)} = 0, & \text{in } \Omega, \\ u^{(0)} = \bar{u}, & \text{on } \Gamma_D, \\ q^{(0)} = \bar{q}, & \text{on } \Gamma_N, \\ u^{(0)} = h, & \text{on } \Gamma_S. \end{cases}$$

可以求解出 Γ_S 上的法向导数 $q^{(0)}$ 及系数矩阵 \mathbf{H} 和 \mathbf{G} .

第二步: 选取误差限 $\tau > 0$ 以及参数 $\omega \in (0, 2)$, 置 $k = 0$. 计算 \mathbf{A} , $f_i = u_i^{(0)} + cq_i^{(0)} - \omega R(u_i^{(0)}, q_i^{(0)})$, $\bar{\mathbf{F}} = (f_1, f_2, \dots, f_{N_n})$ 和 $\mathbf{b} = [\bar{\mathbf{U}}_1^T, \bar{\mathbf{Q}}_2^T, \bar{\mathbf{F}}^T]^T$.

第三步: 通过计算 $\mathbf{A}\mathbf{b}$, 得到 $\mathbf{Q}_1^{(k+1)}$, $\mathbf{U}_2^{(k+1)}$ 和 $\mathbf{U}_3^{(k+1)}$. 再由 $\mathbf{U}_3^{(k+1)} = \bar{\mathbf{F}} - c\mathbf{Q}_3^{(k+1)}$ 得到 $\mathbf{Q}_3^{(k+1)}$.

第四步: 如果 $\|\mathbf{U}_3^{(k+1)} - \mathbf{U}_3^{(k)}\|_{\Gamma_S} \leq \tau \|\mathbf{U}_3^{(k+1)}\|_{\Gamma_S}$, 则迭代停止. 得出 $\mathbf{Q}_1^{(k+1)}$, $\mathbf{U}_2^{(k+1)}$, $\mathbf{U}_3^{(k+1)}$ 和 $\mathbf{Q}_3^{(k+1)}$.

第五步: 否则, 计算 $f_i = u_i^{(k+1)} + cq_i^{(k+1)} - \omega R(u_i^{(k+1)}, q_i^{(k+1)})$, $\bar{\mathbf{F}} = (f_1, f_2, \dots, f_{N_n})$ 和 $\mathbf{b} = [\bar{\mathbf{U}}_1^T, \bar{\mathbf{Q}}_2^T, \bar{\mathbf{F}}^T]^T$. 置 $k := k + 1$ 并返回第三步.

类似于文献[12]中定理 5 的证明[12], 可建立上述算法的收敛性定理.

定理 1 设 $\{(u^{(k)}, q^{(k)})\}$ 是迭代格式 (11) 产生的序列, 则该序列收敛于 Signorini 问题 (6) 和 (7) 的唯一解 (u^*, q^*) , 即 $\lim_{k \rightarrow \infty} (u^{(k)}, q^{(k)}) = (u^*, q^*)$ 。

3 数值算例

为了验证我们所提出的算法的可行性和有效性, 在这部分我们给出了两个具有代表性的数值算例, 并将其数值结果与其他方法进行分析比较。

算例 3.1 圆环问题

考虑一个在圆环区域 $\Omega = \{x \in \mathbb{R}^2 : a < |x| < b\}$ 上的 Laplace 方程的 Signorini 问题。这个问题的解析解可以通过复值函数 $u(x_1, x_2) = \text{Im} w(x_1 + ix_2)^3$ 得到^[3], 其中 $w(x_1 + ix_2) = \sqrt{A + B \text{sgn} x_2} + i \sqrt{A - B \text{sgn} x_2}$,

$$A = \frac{1}{2} \sqrt{\left(\frac{x_1^2 - x_2^2}{r^2}\right)^2 + \frac{1}{4} \left(\frac{r^2 - a^2}{a^2 - r^2}\right)^2}, \quad B = \frac{1}{4} \frac{x_1^2 - x_2^2}{r^2} \left(\frac{r^2 + a^2}{a^2 - r^2}\right), \quad r^2 = x_1^2 + x_2^2.$$

在 $\Gamma_D = \{x \in \mathbb{R}^2 : |x| = b\}$ 上为 Dirichlet 边界条件, 在 $\Gamma_S = \{x \in \mathbb{R}^2 : |x| = a\}$ 上为 Signorini 边界条件 $u \geq h, \frac{\partial u}{\partial n} \geq \varphi, (u - h) \left(\frac{\partial u}{\partial n} - \varphi\right) = 0$,

其中
$$h(x_1, x_2) = \min(0, u(x_1, x_2)),$$

$$\varphi(x_1, x_2) = \begin{cases} -\frac{6}{a}, & x_2 > 0, \\ -\frac{6}{a^5} (x_1^2 - x_2^2)^2, & x_2 \leq 0, x_2 \geq -|x_1|, \\ 0, & x_2 < -|x_1|. \end{cases}$$

在 Signorini 边界 Γ_S 上的解析解为

$$u(x_1, x_2) = -\sqrt{\max\left(0, \frac{x_2^2 - x_1^2}{a^2}\right)^3} \text{sgn} x_2, \quad \frac{\partial u}{\partial n}(x_1, x_2) = -\frac{6}{a^3} \sqrt{\max\left(0, \frac{x_2^2 - x_1^2}{a^2}\right)} |x_1| x_2.$$

取 $a = 0.1$ 和 $b = 0.25$, 并引入参数形式 $t \rightarrow (a \cos 2\pi t, -a \sin 2\pi t)$ 来表示 Signorini 边界。当选取 128 个边界节点时, 函数 u 及其法向导数 q 的解析解和数值解分别如图 1 所示。结果表明, 数值解和解析解是吻合的。

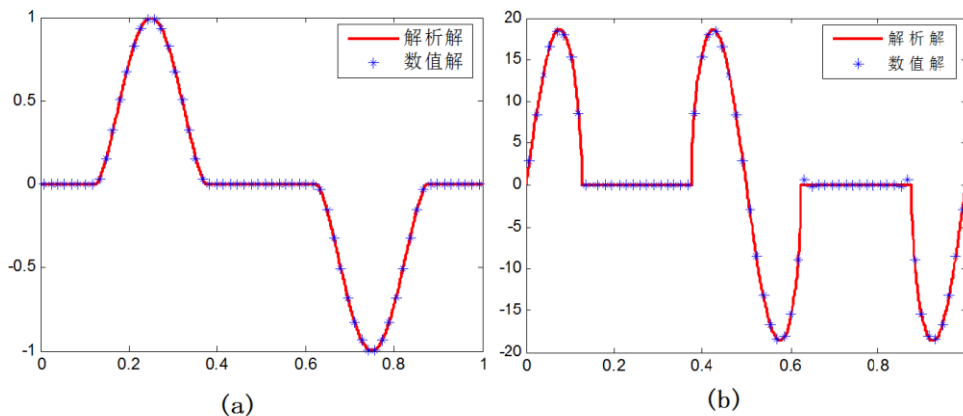


图 1 Γ_S 上函数 u 及其法向导数 q 的解析解和数值解结果

为了研究算法的收敛性, 取不同的节点数 $N = 32, 64, 128, 256$, 利用 $e(u) = \frac{1}{N_s} \sqrt{\sum_{i=1}^{N_s} (u^{(i)} - u_h^{(i)})^2}$, 计算出函数 u 的误差, 并将结果与 GBEM^[3], BE-PIM^[4] 和 BE-LCM^[5] 等边界元方法相比较, 画出误差图 2。结果表明, 本方法 (BRPIM) 比其他三种方法要收敛得更快一些。

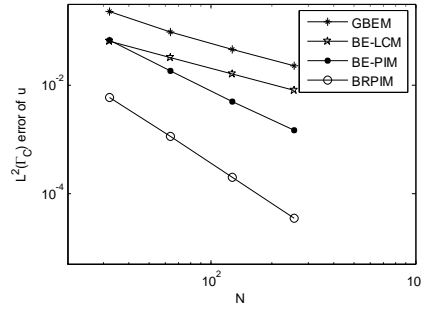


图 2 函数 u 的误差对比图

在表 1 中, 对比了 BE-LCM 及 BE-PIM 与本文方法(BRPIM)的计算时间, 我们可以发现, 本文方法比 BE-PIM 和 BE-LCM 耗费计算时间更短。

N	32	64	128	256	秒
BE-PIM	0.842000	0.297000	0.936000	4.368000	
BE-LCM	0.156000	0.234000	0.749000	4.758000	
BRPIM	0.174000	0.290000	0.700000	2.562000	

算例 3.2 经典的自由水坝问题

文献[2]中给出了自由水坝问题的详细介绍, 该问题的数学模型为

$$\begin{cases} \Delta u = 0, \text{ in } \Omega = \{(x, y) : 0 \leq x \leq 1, 0 \leq y \leq 1\}, \\ u = 0, \text{ on } \Gamma_D = \{(x, y) : x = 0, 0 \leq y \leq 1\}, \\ \frac{\partial u}{\partial y} = 0, \text{ on } \Gamma_N = \{(x, y) : 0 \leq x \leq 1, y = 0\}, \\ u = G(1), \text{ on } \Gamma_D = \{(x, y) : x = 1, 0 \leq y \leq 1\}, \\ u \leq G(x), \frac{\partial u}{\partial y} \leq 0, (u - G(x)) \frac{\partial u}{\partial y} = 0, \text{ on } \Gamma_S = \{(x, y) : 0 \leq x \leq 1, y = 1\}, \end{cases}$$

其中 $G(x)$ 为描述表面轮廓的已知函数。

在数值求解过程中, 在正方形的每条边上采用 21 个边界节点, 并将其数值结果与其数值解进行比较。图 3 分别比较了两种不同的表面轮廓函数与文献^[1]中采用开关算法得到的数值解。其中 $G_1(x_1) = \left(\frac{1}{2} - x_1\right)(1 - x_1) - x_1$, $G_3(x_1) = \sin(12x_1) - 2$ 。从图 3 可以看出, 本文所得到的图形和用开关算法得到的数值结果的图形是一致的。

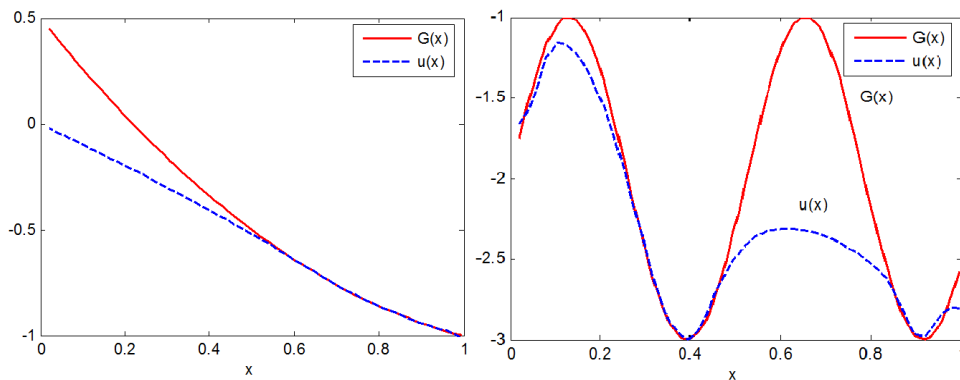


图 3 表面轮廓函数分别为 $G_1(x)$ 和 $G_3(x)$ 时自由水坝问题的数值解

为了进一步验证本文算法的收敛性, 将本文的数值结果与文献[2]中采用基本解方法(MFS)所得结果进行比较。表 2 说明, 固定误差限 $\tau=10^{-6}$ 不变, 在采用相同边界节点数目时, 本文方法(BRPIM)收敛速度比 MFS 更快, 并且 BRPIM 中边界节点数目的变化对迭代次数影响较小。表 3 说明, 固定边界节点数 $N = 64$ 不变, 在采用相同误差限时, BRPIM 也比 MFS 收敛得更快。

表 2 当 $\tau=10^{-6}$ 时对表面轮廓函数 $G_1(x)$ 的收敛性分析

N	28	48	64	84	128	512	1024
BRPIM	25	25	24	24	25	26	24
MFS	72	119	893	946			

表3 当 $N = 64$ 时对表面轮廓函数 $G_1(x)$ 的收敛性分析

τ	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
BRPIM	5	10	14	19	24
MFS	77	84	270	345	893

4 总结

本文研究了用边界径向点插值法求解Signorini问题的无网格算法。该算法利用投影算子将Signorini问题转化为边界积分方程,然后用径向点插值法来构造边界变量的无网格近似。该方法是一种边界型无网格法,同时具有边界元法降维和无网格法不需要划分网格的优势。数值结果表明该方法比边界元方法具有更高的计算精度和更快的收敛速度,并且计算时间更短。因此,这种无网格法是处理Signorini问题的一种有效的解法。

参考文献:

- [1] Aitchison J M, Poole M W. A numerical algorithm for the solution of Signorini problems[J]. Journal of Computational and Applied Mathematics, 1998, 94: 55-67.
- [2] A. Poullikkas, A. Karageorghis, G. Georgiou. The method of fundamental solutions for Signorini problems[J]. IMA Journal of Numerical Analysis, 1998, 18: 273-285.
- [3] Spann W. On the boundary element method for the Signorini problem of the Laplacian[J]. Numerische Mathematik, 1993, 65: 337-356.
- [4] S.G. Zhang, J.L. Zhu. A projection iterative algorithm boundary element method for the Signorini problem[J]. Engineering Analysis with Boundary Elements, 2013, 37: 176-181.
- [5] S.G. Zhang, J.L. Zhu. The boundary element-linear complementarity method for the Signorini problem[J]. Engineering Analysis with Boundary Elements, 2012, 36: 112-117.
- [6] 张守贵. 用边界元法求解Signorini问题的线性互补法[J]. 重庆师范大学学报:自然科学版, 2012, 29(5): 41-45.
Zhang S G. The linear complementarity method for the Signorini problem using boundary element method[J]. Journal of Chongqing Normal University: Natural Science, 2012, 29(5): 41-45.
- [7] Xiaolin Li. Meshless Galerkin algorithms for boundary integral equations with moving least square approximations[J]. Applied Numerical Mathematics, 2011, 61(12): 1237-1256.
- [8] Xiaolin Li, Jialin Zhu. A Galerkin boundary node method and its convergence analysis[J]. Journal of Computational and Applied Mathematics, 2009, 230(1): 314-328.
- [9] G.R. Liu, Y.T. Gu. Boundary meshfree methods based on the boundary point interpolation methods[J]. Eng Anal Bound Elem, 2004, 28: 475-87.
- [10] Xiaolin Li, Jialin Zhu, Shougui Zhang. A meshless method based on boundary integral equations and radial basis functions for biharmonic-type problems. Applied Mathematical Modelling, 2011, 35(2): 737-751.
- [11] Xiong Zhang, Yan Liu. The meshfree method[M]. Beijing: Tsinghua University Press, 2004.
- [12] Yanlin Ren, Xiaolin Li. A Meshfree Method for Signorini Problems Using Boundary Integral Equations[J]. Mathematical Problems in Engineering. Volume 2014, Article ID 490127, 12 pages, 2014.

A Meshfree Boundary Radial Point Interpolation Method for Signorini Problems

REN Yanlin, LI Xiaolin

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

Abstract: A boundary type meshfree method is developed for the numerical solution of Signorini problems. In this method, a projection operator, which is used to convert the boundary inequality constraints into a fixed point equation, is introduced. Then, the Signorini problem is reformulated as BIEs and solved by the meshfree boundary radial point interpolation method. Thus, the new method is easy to be implied effectively and quickly. The convergence of the method is proven. Numerical examples are given to illustrate the performance and usefulness of the method.

Keywords: Meshfree method; Signorini problem; Projection iterative algorithm; Boundary radial point interpolation method; Boundary integral equation