

## RESEARCH

## Open Access



# Meshless numerical analysis of partial differential equations with nonlinear inequality constraints

Mei Chen and Xiaolin Li\*

\*Correspondence:  
lxmath@163.com  
College of Mathematics Science,  
Chongqing Normal University,  
Chongqing, 401331, P.R. China

## Abstract

A meshless method for the numerical solution of partial differential equations with nonlinear inequality constraints is discussed in this paper. The original nonlinear inequality problem is linearized as a sequence of linear equality problems, and then discrete linear system of algebraic equations is formed. This meshless method only requires nodes on the boundary of the domain, and it does not require any numerical integrations. Numerical experiments indicate that this method is very effective for nonlinear inequality problems and has good convergence rate and high computational efficiency.

**Keywords:** meshless method; elliptic boundary value problem; nonlinear; inequality

## 1 Introduction

This paper concerns numerical solutions of the following elliptic boundary value problem with nonlinear inequality constraints:

$$\Delta u = 0, \quad \text{in } \Omega, \quad (1)$$

$$u = \bar{u}, \quad \text{on } \Gamma_D, \quad (2)$$

$$q = \bar{q}, \quad \text{on } \Gamma_N, \quad (3)$$

$$u \leq f, \quad q \leq g, \quad (u - f)(q - g) = 0, \quad \text{on } \Gamma_R, \quad (4)$$

where  $\Omega$  is a plane domain bounded by the boundary  $\Gamma$ ,  $u$  is an unknown function,  $q := \partial u / \partial \mathbf{n}$  is the normal derivative of  $u$ ,  $\mathbf{n}$  is the outward normal to  $\Gamma$ ,  $\bar{u}$  is the given potential on  $\Gamma_D$ ,  $\bar{q}$  is the given normal flux on  $\Gamma_N$ , and  $f$  and  $g$  are prescribed functions on  $\Gamma_R = \Gamma - \Gamma_D - \Gamma_N$ .

The inequality constraint (4) is nonlinear and complementary. The nonlinear inequality problem (1)-(4) arises from many kinds of physical and industrial applications such as the groundwater flow problem [1, 2], the electropaint process [3, 4], the contact problem [5], the free surface problem [6] and the variational inequalities theory [7, 8]. The numerical solution of this kind of problems can be obtained using the finite difference method and the finite element method (FEM) [1, 3, 4]. However, these domain-type numerical techniques require domain meshing which is usually arduous and computationally expensive.

In problem (1)-(4),  $u$  and  $q$  alternate on  $\Gamma_R$  in conjunction with nonlinear inequality constraint. To obtain the solution  $u$  in  $\Omega$ , we first need to determine on which parts of  $\Gamma_R$  the boundary conditions  $u = f$  and  $q < g$  apply, and thus on the remaining parts the boundary conditions  $u < f$  and  $q = g$  apply. Therefore, the primary focus in solving this problem is on  $\Gamma_R$  and thus, boundary-type numerical methods such as the boundary element method (BEM) [5, 6, 9–12] are particularly suitable for the solution of such problems. The BEM involves the generation of elements on the boundary surface and the computation of some complex singular integrals on boundary elements. In some cases, these processes can also be very difficult and computationally expensive, especially for free boundary and nonlinear problems.

Boundary-type numerical methods reduce the computational dimensions of the original problem by one and thus simplify the efforts involved in data preparation and CPU time. In the past two decades, meshless (or meshfree) methods for numerical solutions of partial differential equations have been developed for overcoming the meshing-related drawbacks involved in the FEM and the BEM. Some boundary-type meshless methods using meshless shape functions and boundary integral equations have been developed. Among them are the boundary node method (BNM) [12, 13], the boundary point interpolation method (BPIM) [14, 15], the hybrid BNM [16, 17], the Galerkin BNM [18, 19], the dual BNM [20, 21] and the boundary element-free method [22–24]. These boundary-type meshless methods perform very well for the numerical solution of lots of linear problems. However, maybe due to the issues associated with the handling of the nonlinear inequality constraints, not many boundary-type meshless methods have been used to the nonlinear inequality problems. Besides, these boundary-type meshless methods still involve the computation of complex singular boundary integrals.

The aim of this paper is to develop a boundary-type meshless method for the numerical solution of the nonlinear inequality problem (1)-(4). This method only requires boundary nodes, and does not need any mesh for either interpolation or integration. Besides, no integrations are involved in the whole solution process. Namely, the present method is truly meshless, integration-free and boundary-only. As a result, this method is expected to have higher computational speed and efficiency. Numerical experiments indicate that this method is very effective for problems with nonlinear inequality constraints and has good convergence rate and high computational efficiency. The following discussions begin with a linearization of the nonlinear inequality problem in Section 2. Then, a detailed meshless numerical implementation is presented in Section 3. Numerical experiments are provided in Section 4. Finally, Section 5 contains some conclusions.

## 2 Linearization

To linearize efficiently the nonlinear inequality constraint (4), let [7]

$$(q - g) - \mathcal{L}((q - g) - c(u - f)) = 0, \quad \text{on } \Gamma_R, \quad (5)$$

where  $c$  is an arbitrary but fixed positive constant, and  $\mathcal{L}$  is an operator defined as

$$\mathcal{L}(v) := \min(0, v), \quad v \in \mathbb{R}. \quad (6)$$

From the numerical point of view, the alternative formulation given by Eq. (5) is very important for the numerical implementation.

Eq. (4) indicates that  $q \leq g$ . If  $q = g$ , then Eq. (5) indicates that  $(q - g) - c(u - f) \geq 0$ , and thus  $u \leq f$ . Otherwise, if  $q < g$ , then  $(q - g) - c(u - f) < 0$ , and thus recalling again Eq. (5) leads to  $u = f$ . As a result, from Eq. (5) we can deduce the nonlinear inequality constraint (4). On the other hand, from constraint (4) we can deduce Eq. (5) immediately. Summarizing, we have shown that the nonlinear inequality constraint (4) is equivalent to Eq. (5).

To tackle the equivalent equation (5) numerically, let

$$R(u, q) = (q - g) - \mathcal{L}((q - g) - c(u - f)), \quad \text{on } \Gamma_R. \tag{7}$$

Then, it can be verified that  $(u, q)$  is a zero of  $R(u, q) = 0$  if and only if  $(u, q)$  is the solution of the following equation:

$$(q - g) + c(u - f) = (q - g) + c(u - f) - \omega R(u, q), \quad \text{on } \Gamma_R, \tag{8}$$

where  $\omega$  is a positive constant.

According to Eq. (8), define

$$(q^{(k)} - g) + c(u^{(k)} - f) = (q^{(k-1)} - g) + c(u^{(k-1)} - f) - \omega R(u^{(k-1)}, q^{(k-1)}), \quad \text{on } \Gamma_R,$$

or

$$q^{(k)} + cu^{(k)} = q^{(k-1)} + cu^{(k-1)} - \omega R(u^{(k-1)}, q^{(k-1)}), \quad \text{on } \Gamma_R, \quad k = 1, 2, \dots \tag{9}$$

Finally, the original nonlinear inequality problem (1)-(4) is reduced to the following linear equality problem:

$$\Delta u^{(k)} = 0, \quad \text{in } \Omega, \tag{10}$$

$$u^{(k)} = \bar{u}, \quad \text{on } \Gamma_D, \tag{11}$$

$$q^{(k)} = \bar{q}, \quad \text{on } \Gamma_N, \tag{12}$$

$$cu^{(k)} + q^{(k)} = r^{(k-1)}, \quad \text{on } \Gamma_R, \tag{13}$$

where  $k = 1, 2, \dots$ ,  $u^{(k)}$  and  $q^{(k)}$  are the unknown functions at the  $k$ th iteration, and

$$r^{(k-1)} := q^{(k-1)} + cu^{(k-1)} - \omega R(u^{(k-1)}, q^{(k-1)}) \tag{14}$$

is known at the  $k$ th iteration.

In view of Eq. (14), the iterative solution of the linear problem (10)-(13) requires initial values  $u^{(0)}$  and  $q^{(0)}$ . For doing this, assume initially that the boundary condition on  $\Gamma_R$  is  $u^{(0)} = f$ , then obtain  $q^{(0)}$  on  $\Gamma_R$  by solving the following boundary value problem:

$$\Delta u^{(0)} = 0, \quad \text{in } \Omega, \tag{15}$$

$$u^{(0)} = \bar{u}, \quad \text{on } \Gamma_D, \tag{16}$$

$$q^{(0)} = \bar{q}, \quad \text{on } \Gamma_N, \tag{17}$$

$$u^{(0)} = f, \quad \text{on } \Gamma_R. \tag{18}$$

### 3 Meshless numerical implementation

An integration-free and truly meshless method is developed in this section for the non-linear inequality problem (1)-(4).

According to the classical results of partial differential equation, the solution of the governing equations (10) and (15) can be represented as the following integral formulation:

$$u^{(k)}(\mathbf{x}) = \int_{\Gamma} \sigma^{(k)}(\mathbf{y}) \ln |\mathbf{x} - \mathbf{y}| \, d\Gamma_{\mathbf{y}}, \quad \mathbf{x} \in \bar{\Omega} = \Omega \cup \Gamma, \quad k = 0, 1, 2, \dots, \tag{19}$$

where  $\sigma^{(k)}$  is the unknown function defined on  $\Gamma$ .

To approximate  $\sigma^{(k)}$ , the boundary  $\Gamma$  is separated into  $N$  sub-domains  $\Gamma_j$  with  $\bigcup_{j=1}^N \Gamma_j = \Gamma$ . On each sub-domain  $\Gamma_j$ , a boundary node  $\mathbf{x}_j$  is located at its middle. Then, we have  $N$  nodes  $\{\mathbf{x}_j\}_{j=1}^N \subset \Gamma$ . Besides,  $\sigma^{(k)}$  is approximated to be constant on each  $\Gamma_j$ , and the value of  $\sigma^{(k)}$  at  $\mathbf{x}_j$  represents the approximate value of  $\sigma^{(k)}$  on  $\Gamma_j$ . Namely,  $\sigma^{(k)}(\mathbf{y}) \approx \sigma^{(k)}(\mathbf{x}_j) := \sigma_j^{(k)}$  for all  $\mathbf{y} \in \Gamma_j$ . As a result, Eq. (19) can be discretized as

$$u^{(k)}(\mathbf{x}) \approx \sum_{j=1}^N \int_{\Gamma_j} \sigma^{(k)}(\mathbf{y}) \ln |\mathbf{x} - \mathbf{y}| \, d\Gamma_{\mathbf{y}} \approx \sum_{j=1}^N \sigma_j^{(k)} \int_{\Gamma_j} \ln |\mathbf{x} - \mathbf{y}| \, d\Gamma_{\mathbf{y}}, \quad \mathbf{x} \in \bar{\Omega}. \tag{20}$$

Equation (20) has integrations over  $\Gamma_j$ . A simple numerical quadrature is applied as

$$u^{(k)}(\mathbf{x}) \approx \sum_{j=1}^N \sigma_j^{(k)} l_j \ln |\mathbf{x} - \mathbf{y}_j|, \quad \mathbf{x} \in \bar{\Omega}, \tag{21}$$

where  $l_j := \text{mes}(\Gamma_j)$  is the arc length of  $\Gamma_j$ .

When  $\mathbf{x}$  and  $\mathbf{y}_j$  are close to each other or coincide, the logarithmical kernel  $\ln |\mathbf{x} - \mathbf{y}_j|$  in Eq. (21) becomes nearly singular or singular. In this case, the values of this kernel cannot be obtained accurately and directly. To tackle this issue and to establish an integration-free and truly meshless method,  $\mathbf{y}_j$  is located in this paper a distance from  $\Gamma$  and corresponding to the boundary node  $\mathbf{x}_j$ . As in [16, 17], the position of  $\mathbf{y}_j$  can be determined by a scale factor  $d$  using the following equation:

$$\mathbf{y}_j = \mathbf{x}_j + d h \mathbf{n}_{\mathbf{x}_j}, \tag{22}$$

where  $h$  is the nodal spacing. The distribution of the points  $\mathbf{y}_j$  is determined uniquely by  $d$ .

As  $\mathbf{y}_j$  is located outside the domain, the kernel  $\ln |\mathbf{x} - \mathbf{y}_j|$  is always regular. And thus, the solution of the linear problems (10)-(13) and (15)-(18) can be approximated as

$$u^{(k)}(\mathbf{x}) \approx \sum_{j=1}^N \eta_j^{(k)} \ln |\mathbf{x} - \mathbf{y}_j|, \quad \mathbf{x} \in \bar{\Omega}, \quad k = 0, 1, 2, \dots, \tag{23}$$

where  $\eta_j^{(k)} := \sigma_j^{(k)} l_j$  is the  $j$ th unknown coefficient at the  $k$ th iteration. We can observe from Eq. (23) that the approximate solution involves the nodes  $\mathbf{x}_j$  and  $\mathbf{y}_j$  only. Since  $\mathbf{x}_j$  can be located on  $\Gamma$  without the use of  $\Gamma_j$ , the sub-domain  $\Gamma_j$  is not required.

From Eq. (23), the normal derivative  $q^{(k)}(\mathbf{x}) := \partial u^{(k)}(\mathbf{x}) / \partial \mathbf{n}_{\mathbf{x}}$  can be approximated as

$$q^{(k)}(\mathbf{x}) \approx \sum_{j=1}^N \eta_j^{(k)} \frac{\partial \ln |\mathbf{x} - \mathbf{y}_j|}{\partial \mathbf{n}_{\mathbf{x}}}, \quad \mathbf{x} \in \Gamma, \quad k = 0, 1, 2, \dots. \tag{24}$$

To obtain the unknown coefficient  $\eta_j^{(k)}$  and to simplify the representation, we assume that the first  $N_d$  boundary nodes  $\{\mathbf{x}_i\}_{i=1}^{N_d}$  belong to  $\Gamma_D$ , the next  $N_n$  boundary nodes  $\{\mathbf{x}_i\}_{i=N_d+1}^{N_d+N_n}$  belong to  $\Gamma_N$  and the remaining  $N_r = N - N_d - N_n$  boundary nodes  $\{\mathbf{x}_i\}_{i=N_d+N_n+1}^N$  belong to  $\Gamma_R$ .

At the initial stage  $k = 0$ , using Eqs. (23) and (24) and collocating boundary conditions (16)-(18) yield

$$\sum_{j=1}^N \eta_j^{(0)} \ln |\mathbf{x}_i - \mathbf{y}_j| = \bar{u}(\mathbf{x}_i), \quad i = 1, 2, \dots, N_d, \tag{25}$$

$$\sum_{j=1}^N \eta_j^{(0)} \frac{\partial \ln |\mathbf{x}_i - \mathbf{y}_j|}{\partial \mathbf{n}_{\mathbf{x}_i}} = \bar{q}(\mathbf{x}_i), \quad i = N_d + 1, \dots, N_d + N_n, \tag{26}$$

$$\sum_{j=1}^N \eta_j^{(0)} \ln |\mathbf{x}_i - \mathbf{y}_j| = f(\mathbf{x}_i), \quad i = N_d + N_n + 1, \dots, N. \tag{27}$$

Equations (25)-(27) contain a set of coupled  $N$  linear algebraic equations, which are solved together for  $N$  unknowns  $\eta_j^{(0)}$ . Then, the initial values  $u^{(0)}(\mathbf{x}_i)$  and  $q^{(0)}(\mathbf{x}_i)$  at  $\mathbf{x}_i \in \Gamma_R$  can be approximated as

$$u^{(0)}(\mathbf{x}_i) = f(\mathbf{x}_i), \quad q^{(0)}(\mathbf{x}_i) \approx \sum_{j=1}^N \eta_j^{(0)} \frac{\partial \ln |\mathbf{x}_i - \mathbf{y}_j|}{\partial \mathbf{n}_{\mathbf{x}_i}}, \quad i = N_d + N_n + 1, \dots, N, \tag{28}$$

and thus  $r^{(0)}(\mathbf{x}_i)$  can be computed using Eq. (14).

Once  $u^{(0)}(\mathbf{x}_i)$  and  $q^{(0)}(\mathbf{x}_i)$  are found, the linear problem (10)-(13) can be solved iteratively. The details follow.

For  $k = 1, 2, \dots$ , using Eq. (23) and collocating the Dirichlet boundary condition (11) for boundary nodes  $\{\mathbf{x}_i\}_{i=1}^{N_d} \subset \Gamma_D$  yield

$$\sum_{j=1}^N \eta_j^{(k)} \ln |\mathbf{x}_i - \mathbf{y}_j| = \bar{u}(\mathbf{x}_i), \quad i = 1, 2, \dots, N_d. \tag{29}$$

Next, using Eq. (24) and collocating the Neumann boundary condition (12) for boundary nodes  $\{\mathbf{x}_i\}_{i=N_d+1}^{N_d+N_n} \subset \Gamma_N$  yield

$$\sum_{j=1}^N \eta_j^{(k)} \frac{\partial \ln |\mathbf{x}_i - \mathbf{y}_j|}{\partial \mathbf{n}_{\mathbf{x}_i}} = \bar{q}(\mathbf{x}_i), \quad i = N_d + 1, \dots, N_d + N_n. \tag{30}$$

Finally, using Eqs. (23) and (24) and collocating the Robin boundary condition (13) for boundary nodes  $\{\mathbf{x}_i\}_{i=N_d+N_n+1}^N \subset \Gamma_R$  yield

$$\sum_{j=1}^N \eta_j^{(k)} \left( c \ln |\mathbf{x}_i - \mathbf{y}_j| + \frac{\partial \ln |\mathbf{x}_i - \mathbf{y}_j|}{\partial \mathbf{n}_{\mathbf{x}_i}} \right) = r^{(k-1)}(\mathbf{x}_i), \quad i = N_d + N_n + 1, \dots, N. \tag{31}$$

Equations (29)-(31) contain a set of coupled  $N$  linear algebraic equations and can be represented in matrix form as

$$\mathbf{A}\boldsymbol{\eta}^{(k)} = \mathbf{b}^{(k-1)}, \tag{32}$$

where  $\boldsymbol{\eta}^{(k)} = (\eta_1^{(k)}, \eta_2^{(k)}, \dots, \eta_N^{(k)})^T$ ,  $\mathbf{A}$  is the coefficient matrix, and

$$\mathbf{b}^{(k-1)} = [\bar{\mathbf{u}}, \bar{\mathbf{q}}, \mathbf{r}^{(k-1)}]^T, \tag{33}$$

with

$$\bar{\mathbf{u}} = (\bar{u}(\mathbf{x}_1), \bar{u}(\mathbf{x}_2), \dots, \bar{u}(\mathbf{x}_{N_d})), \tag{34}$$

$$\bar{\mathbf{q}} = (\bar{q}(\mathbf{x}_{N_d+1}), \bar{q}(\mathbf{x}_{N_d+2}), \dots, \bar{q}(\mathbf{x}_{N_d+N_n})), \tag{35}$$

and

$$\mathbf{r}^{(k-1)} = (r^{(k-1)}(\mathbf{x}_{N_d+N_n+1}), r^{(k-1)}(\mathbf{x}_{N_d+N_n+2}), \dots, r^{(k-1)}(\mathbf{x}_N)). \tag{36}$$

After obtaining  $\eta_j^{(k)}$  from Eq. (32),  $u^{(k)}(\mathbf{x})$  at an internal point  $\mathbf{x} \in \Omega$  can be computed approximately using Eq. (23), and  $u^{(k)}(\mathbf{x})$  and  $q^{(k)}(\mathbf{x})$  at a boundary point  $\mathbf{x} \in \Gamma$  can be approximated respectively using Eqs. (23) and (24). Especially,  $u^{(k)}(\mathbf{x})$  and  $q^{(k)}(\mathbf{x})$  at boundary nodes  $\mathbf{x}_i \in \Gamma_R$  can be approximated as

$$u^{(k)}(\mathbf{x}_i) \approx \sum_{j=1}^N \eta_j^{(k)} \ln |\mathbf{x}_i - \mathbf{y}_j|, \quad q^{(k)}(\mathbf{x}_i) \approx \sum_{j=1}^N \eta_j^{(k)} \frac{\partial \ln |\mathbf{x}_i - \mathbf{y}_j|}{\partial \mathbf{n}_x}, \tag{37}$$

where  $i = N_d + N_n + 1, \dots, N$ .

At each iteration, Eq. (32) needs to be solved. To improve the computational efficiency, an effectual approach for solving Eq. (32) would consist of computing the matrix

$$\mathbf{K} = \mathbf{A}^{-1}$$

at the initial stage and then computing the product  $\boldsymbol{\eta}^{(k)} = \mathbf{K}\mathbf{b}^{(k-1)}$ . Therefore, the system matrix and its inverse in the present method only have to be computed once.

From the above analysis, we obtain the following detailed meshless algorithm for the nonlinear inequality problem (1)-(4).

1. Choose a tolerance  $\varepsilon > 0$  and  $N$  boundary nodes  $\{\mathbf{x}_i\}_{i=1}^N \subset \Gamma$ , and obtain  $\{\mathbf{y}_j\}_{j=1}^N$  using Eq. (22).
2. Form the linear algebra system given by Eqs. (25)-(27), solve the system to obtain  $\eta_j^{(0)}$ , and compute  $u^{(0)}(\mathbf{x}_i)$  and  $q^{(0)}(\mathbf{x}_i)$  using Eq. (28) for all nodes  $\mathbf{x}_i \in \Gamma_R$ .
3. Compute  $\mathbf{A}$ ,  $\bar{\mathbf{u}}$ ,  $\bar{\mathbf{q}}$  and  $\mathbf{K}$ , and set  $k = 1$ .
4. Compute  $\mathbf{r}^{(k-1)}$  and form  $\mathbf{b}^{(k-1)}$ , obtain  $\boldsymbol{\eta}^{(k)}$  by computing  $\mathbf{K}\mathbf{b}^{(k-1)}$ , and then obtain  $u^{(k)}(\mathbf{x}_i)$  and  $q^{(k)}(\mathbf{x}_i)$  using Eq. (37) for all nodes  $\mathbf{x}_i \in \Gamma_R$ .
5. If  $e(u) := \sqrt{\sum_{\mathbf{x}_i \in \Gamma_R} (u^{(k)}(\mathbf{x}_i) - u^{(k-1)}(\mathbf{x}_i))^2 / \sum_{\mathbf{x}_i \in \Gamma_R} (u^{(k)}(\mathbf{x}_i))^2} \leq \varepsilon$ , stop the iterative process and return  $\boldsymbol{\eta}^{(k)}$  for computing  $u^{(k)}(\mathbf{x})$ .
6. Otherwise, update  $k$  to  $k + 1$ , and go to step 4.

The iterative process continues until the stopping criterion  $e(u) \leq \varepsilon$  is satisfied. In the subsequent numerical examples, the tolerance is taken as  $\varepsilon = 10^{-6}$ . Obviously, the number of iteration is determined uniquely by the stopping criterion.

From the above discussion, we can conclude that no elements are used and no integrations are computed in the whole iterative process. Thus, the present numerical method is integration-free and truly meshless.

#### 4 Numerical experiments

##### 4.1 Analysis of accuracy and convergence of our method

To demonstrate the accuracy and convergence of our meshless method, we consider the following nonlinear inequality problem [9]:

$$\begin{cases} \Delta u = 0, & \text{in } \Omega = \{a < \rho := \sqrt{x_1^2 + x_2^2} < b\}, \\ u = \bar{u}, & \text{on } \Gamma_D = \{\rho = b\}, \\ u \geq f, \quad q \geq g, \quad (u - f)(q - g) = 0, & \text{on } \Gamma_R = \{\rho = a\}, \end{cases}$$

where

$$g(\mathbf{x}) = \begin{cases} -6/a, & x_2 > 0, \\ -6(x_1^2 - x_2^2)^2/a^5, & x_2 \leq 0, x_2 \geq -|x_1|, \\ 0, & x_2 < -|x_1|, \end{cases}$$

and  $\bar{u}$  and  $f = \min(0, u)$  are given by the following analytical solution:

$$u(x_1, x_2) = \frac{1}{2\rho^3}(v + w)\sqrt{2v - w}\text{sign}(x_2), \quad (x_1, x_2) \in \Omega = \{a < \rho < b\},$$

where  $v = \sqrt{(x_1^2 - x_2^2)^2 + \frac{1}{4a^4}(\rho^4 - a^4)^2}$  and  $w = (x_1^2 - x_2^2)(\frac{\rho^2}{a^2} + \frac{a^2}{\rho^2})$ .

In computation, we use  $a = 0.1$  and  $b = 0.25$ . When 128 boundary nodes are used, the numerical results for  $u$  and  $q$  on  $\Gamma_R$  are shown in Figure 1. Here, the results are shown as plots over the arc angle  $2\pi t, t \in (0, 1)$ . It is evident that the numerical solution is in excellent agreement with the analytical solution.

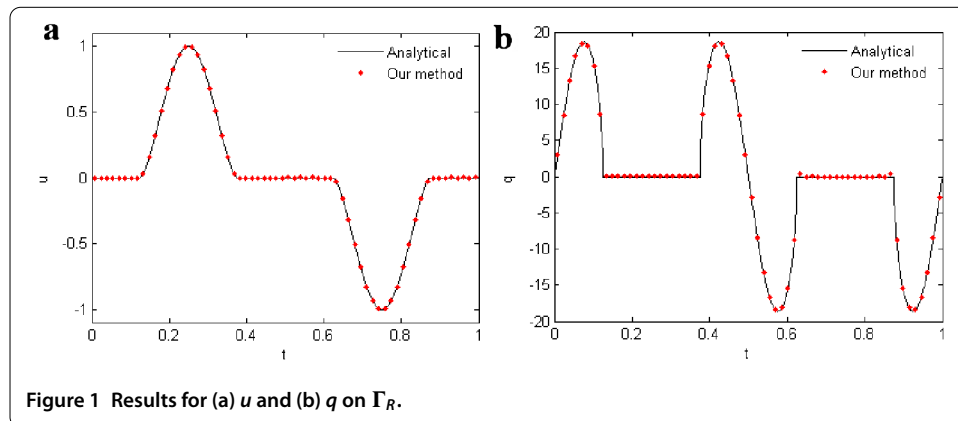


Figure 1 Results for (a)  $u$  and (b)  $q$  on  $\Gamma_R$ .

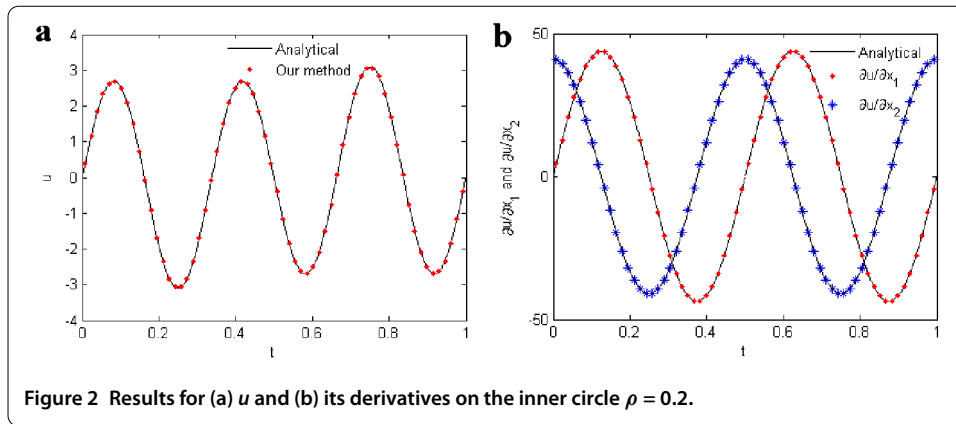


Figure 2 Results for (a)  $u$  and (b) its derivatives on the inner circle  $\rho = 0.2$ .

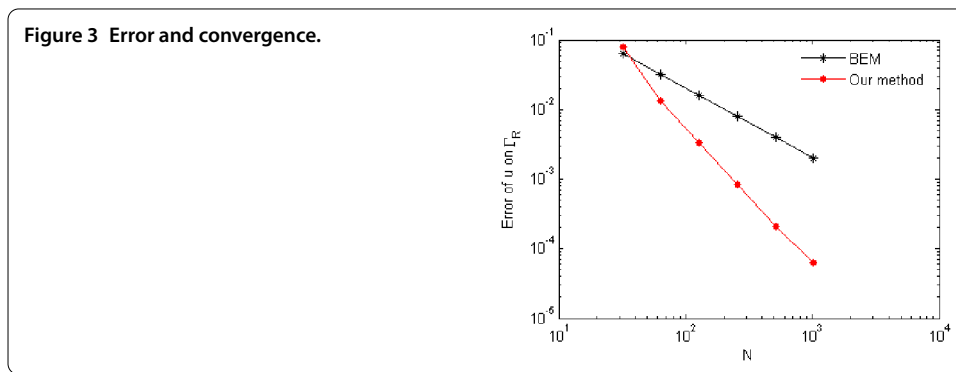


Figure 3 Error and convergence.

Table 1 Number of iteration

$N$	32	64	128	256	512	1,024
BEM	56	91	141	203	263	319
Our method	20	28	37	48	54	47

The numerical results for  $u$  and its derivatives  $\partial u/\partial x_1$  and  $\partial u/\partial x_2$  on the inner circle  $\rho = 0.2$  are shown in Figure 2. It is again evident that the numerical solution agrees well with the analytical solution.

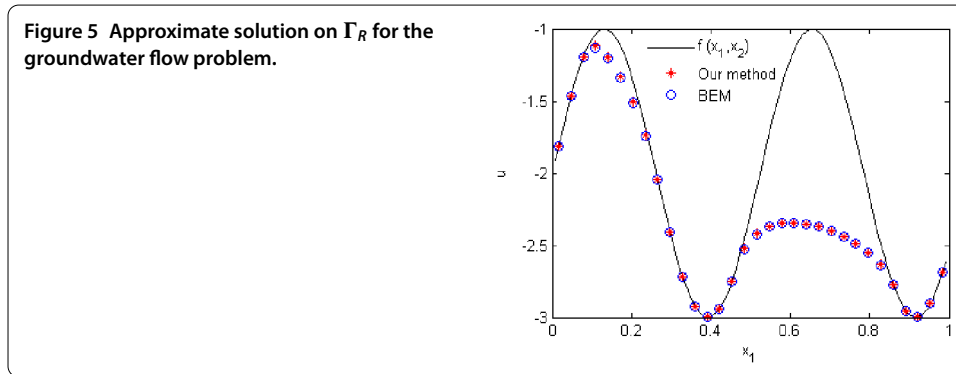
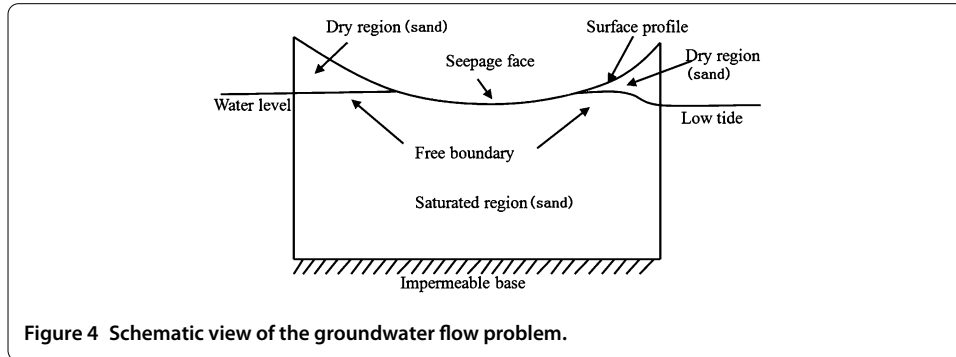
For error estimation and convergence analysis, Figure 3 gives the log-log plot of the  $L^2$  relative error norms of  $u$  on  $\Gamma_R$  with respect to the boundary nodes number  $N$ . The error of the BEM [10] is also plotted for comparison. We can observe that the error of our method is the least, and the experimental convergence rates obtained from the BEM and our method are 1.01 and 2.04, respectively.

The corresponding number of iteration and CPU time are tabulated in Tables 1 and 2, respectively. Table 1 indicates that the number of iteration of our method is little influenced by the number of boundary nodes, and is much less than that of the BEM. Table 2 indicates that our method is much faster than the BEM. When 1,024 boundary nodes are used, the CPU time required in the BEM is about twenty-six times that in our method. Therefore, the computing speed and efficiency of our method are higher than those of the BEM.



**Table 2 CPU time (in seconds)**

<i>N</i>	32	64	128	256	512	1,024
BEM	0.094	0.359	0.796	4.274	31.839	182.426
Our method	0.031	0.063	0.141	0.624	3.885	6.958



### 4.2 Groundwater flow problem

This example involves a groundwater flow problem related to percolation in gently sloping beaches (see Figure 4). This real world physical problem concerns groundwater pass through a sand beach where the seepage face between the saturated area and dry area approaches the upper surface of the sand beach, which is expected to be nearly horizontal [1]. Since the domain of interest is the saturated region, the aim in solving this problem is to determine the number and positions of the separation points.

The mathematical model of this problem can be formulated as

$$\begin{cases} \Delta u = 0, & \text{in } \Omega = (0,1) \times (0,1), \\ u(0, x_2) = 0, & u(1, x_2) = f(1, x_2), \quad 0 \leq x_2 \leq 1, \\ q(x_1, 0) = 0, & 0 \leq x_1 \leq 1, \\ u \leq f, & q \leq 0, \quad (u - f)q = 0, \quad (x_1, x_2) \in \Gamma_R = \{0 \leq x_1 \leq 1, x_2 = 1\}, \end{cases}$$

where  $f(x_1, x_2) = \sin(12x_1) - 2$  depicts the surface profile.

When 32 boundary nodes are used on each side of the domain, Figure 5 depicts numerical results of our method and the BEM [11] for  $u$  on  $\Gamma_R$ . Although no analytical solutions exist for this physical problem, we can see that the results of our method are in excellent agreement with those of the BEM.

**Table 3 CPU time (in seconds) for the groundwater flow problem**

<i>N</i>	32	64	128	256	512	1,024
BEM	0.188	0.328	2.012	14.274	93.069	673.173
Our method	0.047	0.078	0.156	0.499	2.043	8.814

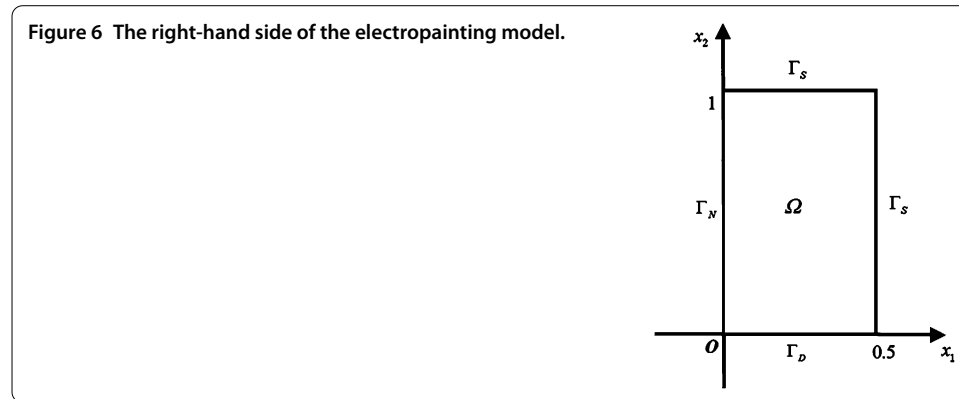


Table 3 gives the CPU time for various numbers of boundary nodes. We can find that the CPU time required in our method is much less than that required in the BEM.

### 4.3 An electropaint process

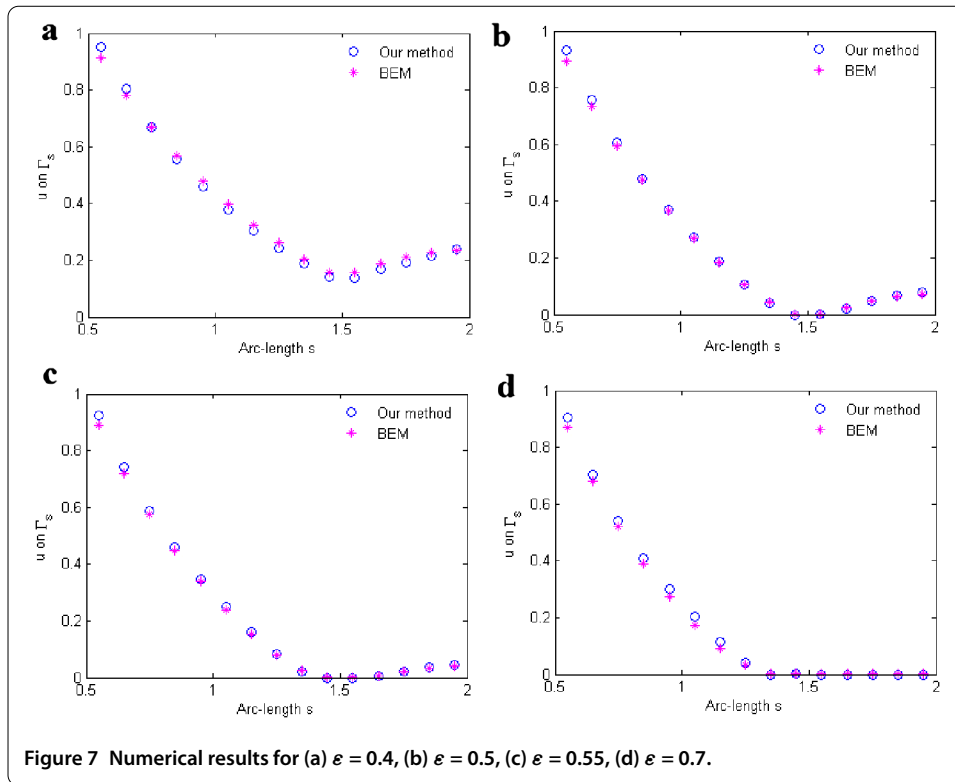
This example involves the simulation of the electropaint process [3, 4]. This problem describes the coating process of a metal surface with paint and thus is important in painting the car body. This realistic industrial problem can be modeled as the following boundary value problem:

$$\begin{cases} \Delta u = 0, & \text{in } \Omega = (0, 0.5) \times (0, 1), \\ u = 1, & \text{on } \Gamma_D = \{0 \leq x_1 \leq 0.5, x_2 = 0\}, \\ q = 0, & \text{on } \Gamma_N = \{x_1 = 0, 0 \leq x_2 \leq 1\}, \\ u \geq 0, \quad q \geq -\varepsilon, \quad u(q + \varepsilon) = 0, & \text{on } \Gamma_R = \Gamma - \Gamma_D - \Gamma_N, \end{cases}$$

where  $u$  is the electric potential and  $\varepsilon$  is the critical current necessary for paint deposition. Figure 6 depicts the sketched geometric configuration. In this problem,  $\Gamma_R$  can be considered as the workpiece to be painted. On  $\Gamma_R$ , the painted subregions are unknown in advance. Therefore, we need to decide the painted subregions and the corresponding paint thickness.

For various  $\varepsilon$ , Figure 7 gives the numerical results of our method and the BEM [10] when 480 boundary nodes are used. Here, the results are shown as plots over the arc-length  $s$ , and  $s = 0, s = 0.5, s = 1.5$  and  $s = 2$  correspond to the vertexes  $(0, 0), (0.5, 0), (0.5, 1)$  and  $(0, 1)$ , respectively. We can observe that the results of our method match those of the BEM. Besides, Table 4 displays the CPU time. Again, the CPU time required in our method is much less than that required in the BEM.

Paint distributions for various  $\varepsilon$  are depicted in Figure 8. Figure 7(a) and Figure 8(a) indicate that the workpiece is completely painted with the vertex  $(0.5, 1)$  receiving the least amount of paint for a small  $\varepsilon$ . If  $\varepsilon$  is increased, Figures 7(b)-(c) and Figures 8(b)-(c) indicate that the paint film near the vertex  $(0.5, 1)$  becomes thinner and eventually unpainted. With



**Table 4 CPU times (in seconds) for the electropainting problem**

$N$	120	240	480	960	1,920
BEM	1.404	4.867	20.499	91.403	462.795
Our method	0.047	0.078	0.281	1.528	9.501

a higher  $\epsilon$ , Figure 7(d) and Figure 8(d) indicate that both the vertex (0.5,1) and the top boundary are unpainted. Therefore, although no analytical solutions exist for this realistic industrial problem, the reliability of our method can be assessed based on the reasonable physical characteristics obtained. Additionally, we can see that the results of our method match those given in [10, 11].

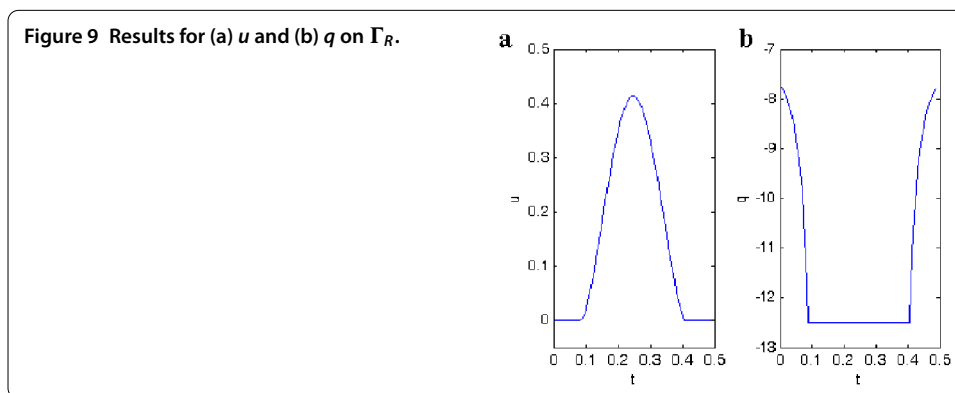
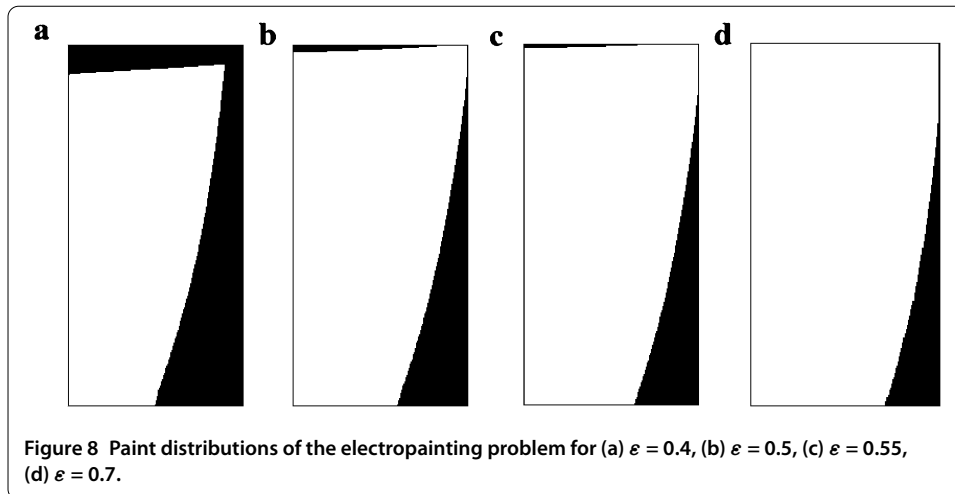
#### 4.4 Problem in a half-oval torus domain

The example considers a real world physical problem corresponding to the model investigated in [3, 9]. The model of the problem in the dimensionless variables is

$$\begin{cases} \Delta u = 0, & \text{in } \Omega = \{0.25x_1^2 + x_2^2 < 0.04, x_1^2 + 4x_2^2/9 > 0.01, x_2 \geq 0\}, \\ u = 1, & \text{on } \Gamma_D = \{0.25x_1^2 + x_2^2 < 0.04, x_2 \geq 0\}, \\ q = 0, & \text{on } \Gamma_N = \{0.1 < |x_1| < 0.4, x_2 = 0\}, \\ u \geq f, \quad q \geq g, \quad (u - f)(q - g) = 0, & \text{on } \Gamma_R = \{x_1^2 + 4x_2^2/9 > 0.01, x_2 \geq 0\}, \end{cases}$$

where  $f(x_1, x_2) = 0$  and  $g(x_1, x_2) = -12.5$ .

The numerical results for our method with  $N = 200$  are shown in Figure 9. Here, the results are shown as plots over the arc angle  $2\pi t$ ,  $t \in (0, 0.5)$ . We can see that the results are indistinguishable from those in [9, 11].



### 5 Conclusions

Meshless methods for partial differential equations with linear equality constraints are now well established. Unfortunately, presumably because nonlinearity and inequation simultaneously exist, the research on meshless methods for problems with nonlinear inequality constraints is scarce. In this paper, a meshless method is presented to deal with the numerical solution of partial differential equations with nonlinear inequality constraints. In this numerical method, the nonlinear inequality constraints are linearized naturally and efficiently, then the original nonlinear inequality problem is transformed into a sequence of linear equality problems. The meshless method in this paper inherently has some desirable numerical merits such as truly meshless, boundary-only and integration-free. Numerical examples are presented for some partial differential equations with nonlinear inequality constraints. The numerical results show that the present meshless method has the merits of good convergence rate, and higher computational accuracy and efficiency over the traditional mesh-based methods such as the BEM.

#### Competing interests

The authors declare that they have no competing interests.

#### Authors' contributions

All authors contributed equally in writing this paper, and read and approved the final manuscript.

### Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 11471063), the Natural Science Foundation Project of CQ CSTC (No. cstc2014jcyjA00005) and the Talent Project of Chongqing Normal University (No. 14CSBJ04).

Received: 28 January 2015 Accepted: 20 May 2015 Published online: 10 June 2015

### References

- Aitchison, JM, Elliott, CM, Ockendon, JR: Percolation in gently sloping beaches. *IMA J. Appl. Math.* **30**, 269-287 (1983)
- Koksal, ME: An operator-difference method for telegraph equations arising in transmission lines. *Discrete Dyn. Nat. Soc.* **2011**, 561015 (2011)
- Aitchison, JM, Lacey, AA, Shillor, M: A model for an electropaint process. *IMA J. Appl. Math.* **33**, 17-31 (1984)
- Poole, PW, Aitchison, JM: Numerical model of an electropaint process with applications to the automotive industry. *IMA J. Math. Appl. Bus. Ind.* **8**, 347-360 (1997)
- Panzeca, T, Salerno, M, Terravecchia, S, Zito, L: The symmetric boundary element method for unilateral contact problems. *Comput. Methods Appl. Mech. Eng.* **197**, 2667-2679 (2008)
- Chen, JT, Hsiao, CC, Lee, YT: Study of free-surface seepage problem using hypersingular equations. *Commun. Numer. Methods Eng.* **23**, 755-769 (2007)
- Noor, MA: Projection iterative methods for extended general variational inequalities. *J. Appl. Math. Comput.* **32**, 83-95 (2010)
- Ryoo, CS: An approach to the numerical verification of solutions for variational inequalities using Schauder fixed point theory. *Bound. Value Probl.* **2014**, 235 (2014)
- Spann, W: On the boundary element method for the Signorini problem of the Laplacian. *Numer. Math.* **65**, 337-356 (1993)
- Zhang, SG, Zhu, JL: The boundary element-linear complementary method for the Signorini problem. *Eng. Anal. Bound. Elem.* **36**, 112-117 (2012)
- Zhang, SG: A projection iterative algorithm for the Signorini problem using the boundary element method. *Eng. Anal. Bound. Elem.* **50**, 313-319 (2015)
- Mukherjee, S, Mukherjee, YX: *Boundary Methods: Elements, Contours, and Nodes*. CRC Press, Boca Raton (2005)
- Wang, YC, Li, XL: A meshless algorithm with moving least squares approximations for elliptic Signorini problems. *Chin. Phys. B* **23**, 090202 (2014)
- Liu, GR: *Meshfree Methods: Moving Beyond the Finite Element Method*. CRC Press, Boca Raton (2009)
- Ren, YL, Li, XL: A meshfree method for Signorini problems using boundary integral equations. *Math. Probl. Eng.* **2014**, Article ID 490127 (2014)
- Zhang, JM, Masa, T, Yao, ZH: Singular and regular implementations of the hybrid boundary node method. *Tsinghua Sci. Technol.* **12**, 509-519 (2007)
- Tan, F, Wang, YH, Miao, Y: Regular hybrid boundary node method for biharmonic problems. *Eng. Anal. Bound. Elem.* **34**, 761-767 (2010)
- Li, XL: A meshless interpolating Galerkin boundary node method for Stokes flows. *Eng. Anal. Bound. Elem.* **51**, 112-122 (2015)
- Li, XL, Zhu, JL: Galerkin boundary node method for exterior Neumann problems. *J. Comput. Math.* **29**, 243-260 (2011)
- Li, XL: Implementation of boundary conditions in BIEs-based meshless methods: a dual boundary node method. *Eng. Anal. Bound. Elem.* **41**, 139-151 (2014)
- Li, XL, Li, SL: Meshless boundary node methods for Stokes problems. *Appl. Math. Model.* **39**, 1769-1783 (2015)
- Wang, JF, Wang, JF, Sun, FX, Cheng, YM: An interpolating boundary element-free method with nonsingular weight function for two-dimensional potential problems. *Int. J. Comput. Methods* **10**, 1350043 (2013)
- Li, F, Li, XL: The interpolating boundary element-free method for unilateral problems arising in variational inequalities. *Math. Probl. Eng.* **2014**, Article ID 518727 (2014)
- Li, XL: An interpolating boundary element-free method for three-dimensional potential problems. *Appl. Math. Model.* **39**, 3116-3134 (2015)

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)