Comput Mech (2011) 48:689–697
DOI 10.1007/s00466-011-0613-6

ORIGINAL PAPER

The finite point method for the p-Laplace equation

Mehdi Tatari · Maryam Kamranian · Mehdi Dehghan

Received: 16 January 2011 / Accepted: 29 May 2011 / Published online: 28 June 2011

Abstract In this paper, the finite point method (FPM) is presented for solving the 2D, nonlinear, elliptic p-Laplace or p-harmonic equation. The FPM is a truly meshfree technique based on the combination of the moving least squares approximation on a cloud of points with the point collocation method to discretize the governing equation. The lack of dependence on a mesh or integration procedure is an important feature, which makes the FPM simple, efficient and applicable to solve nonlinear problems. Applications are demonstrated through illustrative examples.

Keywords Meshfree method · Finite point method · p-Laplace equation · Injection molding · MLS approximation

1 Introduction

1.1 The model investigated

The p-Laplace operator

$$\Delta_p u := \operatorname{div}(|\nabla u|^{p-2}\nabla u), \tag{1.1}$$

has been regarded as a counterpart to the Laplace operator ($p = 2$) for the nonlinear phenomena. The p-Laplace equation, $\Delta_p u = 0$, plays an important role in the modeling of many phenomena in several areas such as glaciology, non-Newtonian rheology or edge-preserving image deblurring [22].

The motivation for this work is the numerical simulation of injection molding, a process of industrial relevance whereby molten polymer is driven into a cavity (the mold) in order to manufacture small plastic parts. If the polymer viscosity obeys a power law and the mold is thin compared to its planar dimensions, the classical mathematical model of the injection molding is the Hele-Shaw approximation [1,12].

As mentioned in [4–6], if we consider the isothermal Hele-Shaw flows, which physically arise whenever the fluid viscosity does not depend on temperature, the problems are related to solve the following 2D, non-linear, elliptic equation

$$\operatorname{div}(|\nabla u|^{\gamma}\nabla u) = 0, \tag{1.2}$$

which is [22] a p-Laplace equation of index $p = \gamma + 2$. The solution yields the pressure distribution $u(x, y)$ in the filled region of the mould Ω with boundary Γ. The boundary of the domain is divided in to three parts: *injection gate*, *walls*, and *front*. As is said in [4] the exponent γ completely characterizes the polymer rheology, and is typically $\gamma \approx 1/2$. If the pressure profile p_{in} is set along the injection gate by the injection machine, the boundary conditions are

$$\begin{cases} u = p_{in}, & \mathbf{x} \in \Gamma_{in} \text{ (injection)}, \\ \frac{\partial u}{\partial n} = 0, & \mathbf{x} \in \Gamma_{w} \text{ (walls)}, \\ u = 0, & \mathbf{x} \in \Gamma_{f} \text{ (front)}. \end{cases} \tag{1.3}$$

From this pressure field, the average planar velocity $\vec{v}$ can be computed and the location of the advancing front can be

M. Tatari · M. Kamranian
Department of Mathematical Sciences,
Isfahan University of Technology,
84156-83111 Isfahan, Iran
e-mail: mtatari@cc.iut.ac.ir

M. Kamranian
e-mail: m.kamranian@math.iut.ac.ir

M. Dehghan (✉)
Department of Applied Mathematics,
Faculty of Mathematics and Computer Sciences,
Amirkabir University of Technology,
No. 424 Hafez Avenue, Tehran, Iran
e-mail: mdehghan@aut.ac.ir; mdehghan.aut@gmail.com

updated:

$$\vec{v} = -|\nabla u|^{\gamma} \nabla u.$$

The boundary conditions (1.3) are the most usual in the commercial software, also the following nonlinear boundary conditions are proper [4]

$$\begin{cases} -|\nabla u|^{\gamma} \frac{\partial u}{\partial n} = q_{in}, & \mathbf{x} \in \Gamma_{in} \text{ (injection)}, \\ \frac{\partial u}{\partial n} = 0, & \mathbf{x} \in \Gamma_{w} \text{ (walls)}, \\ u = 0, & \mathbf{x} \in \Gamma_{f} \text{ (front)}, \end{cases} \tag{1.4}$$

where q_{in} is the profile of the velocity field $\vec{v}$ along the injection segment and it is a known function. We will refer to (1.3) and (1.4) as Dirichlet-injection and Neumann-injection boundary conditions, respectively.

The numerical simulation of the Hele-Shaw flow requires some methods [22] for solving equation (1.2) at every time step with some techniques to advance the front to its new position, until the mould domain has been completely filled [4,5]. Once the pressure is known by solving the above equations, the velocity at each point in the moving front is computed as

$$\vec{v} = (v_1, v_2) = -|\nabla u|^{\gamma} \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right). \tag{1.5}$$

The main purpose of this article is computing the pressure u from the p-Laplace equation, and we do not concern to update the location of the front. For more details of modeling the flow of an injected fluid in a mould see Section 3 of [4].

This problem features a nonlinear, unsteady, and free-boundary flow confined inside a potentially irregular 2D geometry, and therefore is well suited to meshfree discretizations. In [5], an alternative, meshfree framework was proposed for solving this problem combining the method of asymmetric radial basis function (RBF) collocation for pressure with the level sets for capturing the front motion. In [6] the authors were linearized the equation and applied a meshfree technique based on RBF approximation introduced by G. Fasshauer which allows to solve it in the framework of Kansa method. In [22] the meshfree local Petrov Galerkin (MLPG) technique is developed for solving the p-Laplace equation.

1.2 A brief review of meshfree methods

Meshfree methods have become quite popular for solving PDEs appear in physics and engineering. The motivation is to cut down modeling costs in the industrial applications by avoiding the labor intensive step of mesh generation. These methods are particularly attractive in problems with moving interfaces since no remeshing is necessary.

A family of meshfree methods is based on smooth particle hydrodynamic procedures [24,32]. A second class of meshfree methods derived from the generalized [19,31] finite difference (GFD) techniques [8]. Here the approximation around each point is typically defined in terms of Taylor series expansions and the discrete equations are found using the point collocation. Among a third class of meshfree techniques we find the so called diffuse element (DE) method [25], the element free Galerkin (EFG) method [3,11] and the reproducing kernel particle (RKP) method [20,21]. These three methods use the local interpolations for defining the approximate field around a point in terms of the values in adjacent points, whereas the discretized system of equations is typically obtained by integrating the Galerkin variational form over a suitable background grid.

The finite point method (FPM) proposed in [7,26–29] is a truly meshfree procedure. The approximation around each point is obtained using the standard moving least squares techniques similarly as in DE and EFG methods. The discrete system of equations is obtained by sampling the governing differential equations at each point as in GFD methods. We refer the interested reader to [9,10,13–18,23] for more research works on meshfree methods.

1.3 The main aim and the organization of the paper

In this paper we use the FPM for solving the nonlinear p-Laplace equation. We believe that this approach could also be applied to other typical two dimensional PDEs involving the p-Laplace operator as the core nonlinearity, such as the Perona-Malik equation for the nonlinear (edge-preserving) image denoising [30], or the problem of finding the minimal surface resting on a given boundary.

The content of the paper is structured as follows. In Sect. 2, the basis of the FPM is described. In Sect. 3, the linearization and the numerical implementation are discussed. In order to confirm the validity of the approach, some problems are solved in Sect. 4. The convergence and accuracy of the new method are discussed too.

2 The finite point method

The FPM employs a weighted least squares technique to construct the meshfree approximation function and a point collocation procedure in order to discretize the governing partial differential equations (PDE).

To approximate a function $u(\mathbf{x})$ in the problem domain Ω, over a number of randomly located nodes $\{\mathbf{x}_i\}$, $i = 1, 2, \ldots, n$, the moving least squares approximation $u^h(\mathbf{x})$ of $u(\mathbf{x})$, $\forall \mathbf{x} \in \bar{\Omega}$, can be defined by

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}), \quad \forall \mathbf{x} \in \bar{\Omega}, \tag{2.1}$$

where $\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x})]$ is a complete monomial basis of order m and $\mathbf{a}(\mathbf{x})$ is a vector containing

coefficients $a_j(\mathbf{x})$, $j = 1, 2, \ldots, m$ which are functions of the space coordinates $\mathbf{x}$. For example for a 2-D problem,

$\mathbf{p}^T(\mathbf{x}) = [1, x, y]$, linear basis, $m = 3$,

$\mathbf{p}^T(\mathbf{x}) = [1, x, y, x^2, xy, y^2]$, quadratic basis, $m = 6$.

The unknown parameters $a_j(\mathbf{x})$ are determined at any point $\mathbf{x}$, by minimizing a functional $\mathcal{J}(\mathbf{x})$ defined by

$$\begin{aligned}\mathcal{J}(\mathbf{x}) &= \sum_{i=1}^{n} w(\mathbf{x}-\mathbf{x}_i)(u^h(\mathbf{x}_i)-u_i)^2 \\ &= \sum_{i=1}^{n} w(\mathbf{x}-\mathbf{x}_i)(\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x})-u_i)^2,\end{aligned} \tag{2.2}$$

where $w(\mathbf{x}-\mathbf{x}_i)$ is the weight function with compact support associated with node i, n is the number of nodes in $\bar{\Omega}$ for which the weight function $w(\mathbf{x}-\mathbf{x}_i) > 0$ and the parameters u_i are specified. Eq. (2.2) can be written as

$$\mathcal{J}(\mathbf{x}) = [\mathbf{P}.\mathbf{a}(\mathbf{x}) - \mathbf{u}]^T.\mathbf{W}.[\mathbf{P}.\mathbf{a}(\mathbf{x}) - \mathbf{u}], \tag{2.3}$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}^T(\mathbf{x}_1) \\ \mathbf{p}^T(\mathbf{x}_2) \\ \vdots \\ \mathbf{p}^T(\mathbf{x}_n) \end{bmatrix}_{n\times m}, \quad \mathbf{W} = \begin{bmatrix} w(\mathbf{x}-\mathbf{x}_1) & \cdots & \mathbf{0} \\ \cdots & \ddots & \cdots \\ \mathbf{0} & \cdots & w(\mathbf{x}-\mathbf{x}_n) \end{bmatrix}_{n\times n}.$$

The standard minimization of (2.3) with respect to $\mathbf{a}(\mathbf{x})$ can be obtained by setting the derivative of $\mathcal{J}$ with respect to $\mathbf{a}(\mathbf{x})$ equal to zero. The following linear system results:

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{u}, \tag{2.4}$$

where matrices $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are defined by

$$\mathbf{A}(\mathbf{x}) = \mathbf{P}^T\mathbf{W}\mathbf{P} = \sum_{i=1}^{n} w(\mathbf{x}-\mathbf{x}_i)\mathbf{p}(\mathbf{x}_i)\mathbf{p}^T(\mathbf{x}_i), \tag{2.5}$$

$$\begin{aligned}\mathbf{B}(\mathbf{x}) &= \mathbf{P}^T\mathbf{W} \\ &= [w(\mathbf{x}-\mathbf{x}_1)\mathbf{p}(\mathbf{x}_1), w(\mathbf{x}-\mathbf{x}_2)\mathbf{p}(\mathbf{x}_2), \ldots, w(\mathbf{x}-\mathbf{x}_n)\mathbf{p}(\mathbf{x}_n)].\end{aligned} \tag{2.6}$$

Solving for $\mathbf{a}(\mathbf{x})$ from Eq. (2.4) and substituting it into Eq. (2.1), the MLS approximation can be defined as

$$u^h(\mathbf{x}) = \sum_{i=1}^{n} \phi_i(\mathbf{x})u_i = \mathbf{\Phi}^T(\mathbf{x})\mathbf{u}, \quad \mathbf{x} \in \bar{\Omega}, \tag{2.7}$$

where

$$\mathbf{\Phi}^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}), \tag{2.8}$$

or for the shape function $\phi_i(\mathbf{x})$ associated with node i at a point $\mathbf{x}$ we have

$$\phi_i(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})(\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}))_{ji}. \tag{2.9}$$

The matrix $\mathbf{A}(\mathbf{x})$ is often called the moment matrix, it is of size $m \times m$. This matrix must be inverted whenever the MLS shape functions are to be evaluated. It can be seen that this is the case if and only if the rank of $\mathbf{P}$ equals m. A necessary condition for a well-defined MLS approximation is that at least m weight functions are non-zero (i.e. $n \geq m$) for each sample point $\mathbf{x} \in \Omega$ [33].

The smoothness of the shape functions $\phi_i(\mathbf{x})$ is determined by that of the basis functions and of the weight functions. If $w(\mathbf{x}-\mathbf{x}_i) \in C^k(\Omega)$ and $p_j(\mathbf{x}) \in C^l(\Omega), i = 1, 2, \ldots, n, j = 1, 2, \ldots, m$, then $\phi_i(\mathbf{x}) \in C^{min(k,l)}(\Omega)$. The partial derivatives of $\phi_i(\mathbf{x})$ are obtained as

$$\phi_{i,k}(\mathbf{x}) = \sum_{j=1}^{m}(p_{j,k}(\mathbf{A}^{-1}\mathbf{B})_{ji} + p_j(\mathbf{A}^{-1}\mathbf{B}_{,k} + \mathbf{A}^{-1}_{,k}\mathbf{B})_{ji}), \tag{2.10}$$

and

$$\begin{aligned}\phi_{i,kl}(\mathbf{x}) = \sum_{j=1}^{m}\Big(&p_{j,kl}(\mathbf{A}^{-1}\mathbf{B})_{ji} + p_j(\mathbf{A}^{-1}_{,l}\mathbf{B}_{,k} + \mathbf{A}^{-1}\mathbf{B}_{,kl} \\ &+\mathbf{A}^{-1}_{,kl}\mathbf{B} + \mathbf{A}^{-1}_{,k}\mathbf{B}_{,l})_{ji} \\ &+p_{j,k}(\mathbf{A}^{-1}\mathbf{B}_{,l} + \mathbf{A}^{-1}_{,l}\mathbf{B})_{ji} \\ &+ p_{j,l}(\mathbf{A}^{-1}\mathbf{B}_{,k} + \mathbf{A}^{-1}_{,k}\mathbf{B})_{ji}\Big),\end{aligned} \tag{2.11}$$

where, $(\,)_{,k}$ and $(\,)_{,kl}$ denote $\partial(\,)/\partial x_k$ and $\partial^2(\,)/\partial x_{kl}$, respectively. Also $\mathbf{A}^{-1}_{,k} = (\mathbf{A}^{-1})_{,k}$ represents the derivative of the inverse of $\mathbf{A}$ with respect to x_k, which is given by

$$\mathbf{A}^{-1}_{,k} = -\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1}, \tag{2.12}$$

where

$$\mathbf{A}_{,k}(\mathbf{x}) = \sum_{i=1}^{n} w_{,k}(\mathbf{x}-\mathbf{x}_i)\mathbf{p}(\mathbf{x}_i)\mathbf{p}^T(\mathbf{x}_i). \tag{2.13}$$

So the first order and the second order partial derivatives of $u^h(\mathbf{x})$ are obtained by

$$u^h_{,k}(\mathbf{x}) = \sum_{i=1}^{n} \phi_{i,k}(\mathbf{x})u_i, \quad \mathbf{x} \in \bar{\Omega}, \tag{2.14}$$

and

$$u^h_{,kl}(\mathbf{x}) = \sum_{i=1}^{n} \phi_{i,kl}(\mathbf{x})u_i, \quad \mathbf{x} \in \bar{\Omega}, \tag{2.15}$$

respectively.

When the meshfree approximation functions are constructed, the FPM uses a point collocation technique to discretize the governing equation. The point collocation approach gives rise to a linear system of equations, the solution of which provides the nodal parameters at the nodes. Once the nodal parameters are computed, the unknown solution at each node can be computed from Eq. (2.7).

Let Ω be the solution domain of a boundary value problem and $\Gamma = \Gamma_t \cup \Gamma_u$ be the boundary. Assume that the problem is governed by the following set of equations:

$$\begin{aligned} &A(u) = 0, && \text{in } \Omega,\\ &B(u) = 0, && \text{on } \Gamma_t \text{ (Neumann's condition)},\\ &u - u_p = 0, && \text{on } \Gamma_u \text{ (Dirichlet's condition)}, \end{aligned} \tag{2.16}$$

where A and B are the linear differential operators, u is the unknown function and u_p is the prescribed value of u in Γ_u.

The discretized system of equations in the FPM is found by substituting Eq. (2.7) into Eq. (2.16) and collocating the differential equation at each point in the analysis domain. This gives the set of equations

$$\begin{aligned} &A(u^h(\mathbf{x}_j)) = 0, && j = 1, \ldots, N,\\ &B(u^h(\mathbf{x}_j)) = 0, && j = N+1, \ldots, N+N_t,\\ &u^h(\mathbf{x}_j) - u_j = 0, && j = N+N_t+1, \ldots, n, \end{aligned} \tag{2.17}$$

in which N, N_t, and N_u are the number of collocation points located in domain Ω and on the boundary Γ_t and Γ_u, respectively and $n = N + N_t + N_u$. The system of Eqs. (2.17) leads to a system of algebraic equations of the form

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{2.18}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{u}$ is the vector collecting the point parameters u_i, $i = 1, \ldots, n$, where n is the total number of nodes located in the interior of Ω and its boundary and $\mathbf{f}$ is a vector of the known forces acting at the points.

If the differential operator A be nonlinear, a nonlinear system of equations will result. In this case, we use a predictor–corrector scheme based on the solution of the successive linear systems.

3 The discretization and numerical implementation

We start with Eq. (1.2) which can be simplified to

$$\frac{\partial}{\partial x}\left(|\nabla u|^{p-2}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(|\nabla u|^{p-2}\frac{\partial u}{\partial y}\right) = 0. \tag{3.1}$$

For convenience Eq. (3.1) can be rewritten as

$$|\nabla u|^{p-2}\Delta u + \frac{\partial |\nabla u|^{p-2}}{\partial x}\frac{\partial u}{\partial x} + \frac{\partial |\nabla u|^{p-2}}{\partial y}\frac{\partial u}{\partial y} = 0. \tag{3.2}$$

To apply the scheme we consider n nodal points $\{\mathbf{x}_j\}_{j=1}^n$ on the domain of the problem and its boundary as MLS nodal points. Substituting (2.7) into (3.2) and using the collocation at each interior node $\mathbf{x}_j$ [i.e. $\mathbf{x}_j \in \text{int}\,(\Omega)$], result in

$$|\nabla u_j|^{p-2}\Delta u_j + \frac{\partial |\nabla u_j|^{p-2}}{\partial x}\frac{\partial u_j}{\partial x} + \frac{\partial |\nabla u_j|^{p-2}}{\partial y}\frac{\partial u_j}{\partial y} = 0, \tag{3.3}$$

where

$$u_j = \sum_{i=1}^{n}\phi_i(\mathbf{x}_j)u_i,$$

$$\frac{\partial u_j}{\partial x} = \sum_{i=1}^{n}\frac{\partial \phi_i}{\partial x}(\mathbf{x}_j)u_i, \quad \frac{\partial u_j}{\partial y} = \sum_{i=1}^{n}\frac{\partial \phi_i}{\partial y}(\mathbf{x}_j)u_i,$$

$$\Delta u_j = \sum_{i=1}^{n}\left(\frac{\partial^2 \phi_i}{\partial x^2}(\mathbf{x}_j) + \frac{\partial^2 \phi_i}{\partial y^2}(\mathbf{x}_j)\right)u_i,$$

$$|\nabla u_j|^{p-2} = \left[\left(\sum_{i=1}^{n}\frac{\partial \phi_i}{\partial x}(\mathbf{x}_j)u_i\right)^2 + \left(\sum_{i=1}^{n}\frac{\partial \phi_i}{\partial y}(\mathbf{x}_j)u_i\right)^2\right]^{(p-2)/2},$$

$$\begin{aligned}\frac{\partial |\nabla u_j|^{p-2}}{\partial x} &= (p-2)\left[\left(\sum_{i=1}^{n}\frac{\partial \phi_i}{\partial x}(\mathbf{x}_j)u_i\right)^2 + \left(\sum_{i=1}^{n}\frac{\partial \phi_i}{\partial y}(\mathbf{x}_j)u_i\right)^2\right]^{(p-2)/2-1}\\ &\quad\times\left[\left(\sum_{i=1}^{n}\frac{\partial \phi_i}{\partial x}(\mathbf{x}_j)u_i\right)\left(\sum_{i=1}^{n}\frac{\partial^2 \phi_i}{\partial x^2}(\mathbf{x}_j)u_i\right) + \left(\sum_{i=1}^{n}\frac{\partial \phi_i}{\partial y}(\mathbf{x}_j)u_i\right)\left(\sum_{i=1}^{n}\frac{\partial^2 \phi_i}{\partial y \partial x}(\mathbf{x}_j)u_i\right)\right],\end{aligned}$$

$$\begin{aligned}\frac{\partial |\nabla u_j|^{p-2}}{\partial y} &= (p-2)\left[\left(\sum_{i=1}^{n}\frac{\partial \phi_i}{\partial x}(\mathbf{x}_j)u_i\right)^2 + \left(\sum_{i=1}^{n}\frac{\partial \phi_i}{\partial y}(\mathbf{x}_j)u_i\right)^2\right]^{(p-2)/2-1}\\ &\quad\times\left[\left(\sum_{i=1}^{n}\frac{\partial \phi_i}{\partial x}(\mathbf{x}_j)u_i\right)\left(\sum_{i=1}^{n}\frac{\partial^2 \phi_i}{\partial x \partial y}(\mathbf{x}_j)u_i\right) + \left(\sum_{i=1}^{n}\frac{\partial \phi_i}{\partial y}(\mathbf{x}_j)u_i\right)\left(\sum_{i=1}^{n}\frac{\partial^2 \phi_i}{\partial y^2}(\mathbf{x}_j)u_i\right)\right].\end{aligned}$$

Thus Eq. (3.3) is a nonlinear algebraic equation on the unknowns u_i. We use an iteration procedure based on linearizing Eq. (3.3) according to

$$\begin{aligned}&|\nabla u_j^{(k)}|^{p-2}\Delta u_j^{(k+1)} + \frac{\partial |\nabla u_j^{(k)}|^{p-2}}{\partial x}\frac{\partial u_j^{(k+1)}}{\partial x}\\ &\quad + \frac{\partial |\nabla u_j^{(k)}|^{p-2}}{\partial y}\frac{\partial u_j^{(k+1)}}{\partial y} = 0,\end{aligned} \tag{3.4}$$

where the superscript$^{(k)}$ refers to the iteration number. Putting

$$g_j^{(k)} = |\nabla u_j^{(k)}|^{p-2}, \tag{3.5}$$

Eq. (3.4) is simplified to

$$g_j^{(k)}\Delta u_j^{(k+1)} + \frac{\partial g_j^{(k)}}{\partial x}\frac{\partial u_j^{(k+1)}}{\partial x} + \frac{\partial g_j^{(k)}}{\partial y}\frac{\partial u_j^{(k+1)}}{\partial y} = 0, \tag{3.6}$$

or

$$\sum_{i=1}^{n}\left[g_j^{(k)}\left(\frac{\partial^2\phi_i}{\partial x^2}(\mathbf{x}_j)+\frac{\partial^2\phi_i}{\partial y^2}(\mathbf{x}_j)\right)+\frac{\partial g_j^{(k)}}{\partial x}\frac{\partial\phi_i}{\partial x}(\mathbf{x}_j)+\frac{\partial g_j^{(k)}}{\partial y}\frac{\partial\phi_i}{\partial y}(\mathbf{x}_j)\right]u_i^{(k+1)}=0,\quad \mathbf{x}_j\in\mathrm{int}(\Omega). \tag{3.7}$$

Using Eqs. (1.3) [or (1.4)] for those nodes lie on the boundary Γ_w (walls), we have

$$\sum_{i=1}^{n}\frac{\partial\phi_i}{\partial n}(\mathbf{x}_j)u_i^{(k+1)}=0,\quad \mathbf{x}_j\in\Gamma_w. \tag{3.8}$$

To impose the Dirichlet boundary conditions on Γ_f (front) we write

$$\sum_{i=1}^{n}\phi_i(\mathbf{x}_j)u_i^{(k+1)}=0,\quad \mathbf{x}_j\in\Gamma_f. \tag{3.9}$$

Also the Dirichlet boundary conditions in injection gate Γ_{in} in (1.3) are enforced directly as

$$\sum_{i=1}^{n}\phi_i(\mathbf{x}_j)u_i^{(k+1)}=p_{in}(\mathbf{x}_j),\quad \mathbf{x}_j\in\Gamma_{in}, \tag{3.10}$$

and the nonlinear Neumann boundary conditions in injection gate Γ_{in} in (1.4) are imposed by the following equations:

$$\sum_{i=1}^{n}\left(g_j^{(k)}\frac{\partial\phi_i}{\partial n}(\mathbf{x}_j)\right)u_i^{(k+1)}=-q_{in}(\mathbf{x}_j),\quad \mathbf{x}_j\in\Gamma_{in}. \tag{3.11}$$

Equations (3.7)–(3.10) [or (3.7)–(3.9) and (3.11)] construct the following linear system of equations

$$\mathbf{K}^{(k)}\mathbf{u}^{(k+1)}=\mathbf{f},\quad k=0,1,2,\ldots, \tag{3.12}$$

where $\mathbf{u}^{(k)}=[u_1^{(k)},u_2^{(k)},\ldots,u_n^{(k)}]^T$, $K^{(k)}$ is a n by n matrix, must be updated in each step using the current $\mathbf{u}^{(k)}$ and $\mathbf{f}$ is a fixed right hand side. To start the iterations we use the solution of the Laplace equation ($p=2$) as an initial guess $\mathbf{u}^{(0)}$. Then we proceed with the iterations checking the convergence of the procedure by computing the residual at the interior collocation nodes

$$R^{(k+1)}(\mathbf{x}_j)=|\nabla u_j^{(k+1)}|^{p-2}\Delta u_j^{(k+1)}+\frac{\partial|\nabla u_j^{(k+1)}|^{p-2}}{\partial x}\frac{\partial u_j^{(k+1)}}{\partial x}+\frac{\partial|\nabla u_j^{(k+1)}|^{p-2}}{\partial y}\frac{\partial u_j^{(k+1)}}{\partial y}. \tag{3.13}$$

The residual of the Neumann boundary conditions is similarly computed

$$R^{(k+1)}(\mathbf{x}_j)=|\nabla u_j^{(k+1)}|^{p-2}\frac{\partial u_j^{(k+1)}}{\partial n}+q_{in}(\mathbf{x}_j). \tag{3.14}$$

We stop the iteration when $\max(|R^{(k+1)}(\mathbf{x}_j)|)\le\epsilon$, $j=1,2,\ldots,n$.

After the iterations end, we put $\mathbf{u}=\mathbf{u}^{(k+1)}$. Then the values of $u(\mathbf{x})$ at any point $\mathbf{x}\in\Omega$ can be approximated by Eq. (2.7) as:

$$u(\mathbf{x})\approx\sum_{i=1}^{n}\phi_i(\mathbf{x})u_i,\quad \mathbf{x}\in\bar{\Omega}. \tag{3.15}$$

Also the derivatives can be approximated using Eq. (2.14) as:

$$u_{,k}^h(\mathbf{x})\approx\sum_{i=1}^{n}\phi_{i,k}(\mathbf{x})u_i,\quad \mathbf{x}\in\bar{\Omega},$$

and the velocity field can easily be computed using Eq. (1.5).

Remark 3.1 As another criterion to check the convergence of the iteration scheme, it should be noted that Eq. (3.12) can be simplified to

$$\mathbf{u}^{(k+1)}=\mathbf{G}(\mathbf{u}^{(k)}),\quad k=0,1,2,\ldots,$$

in which $\mathbf{G}(\mathbf{u}^{(k)})=(\mathbf{K}^{(k)})^{-1}\mathbf{f}$. Therefore the proposed method can be regarded as a fixed point method with the iteration function $\mathbf{G}(\mathbf{u}^{(k)})$.

Let $\mathbf{u}^*$ be the fixed point for the function $\mathbf{G}:D\subseteq\mathbb{R}^n\to\mathbb{R}^n$, by the Taylor expansion of $\mathbf{G}$ at $\mathbf{u}^*$ we have

$$\mathbf{u}^*-\mathbf{u}^{k+1}=\mathbf{J}_\mathbf{G}(\mathbf{c}^{(k)})(\mathbf{u}^*-\mathbf{u}^k),\quad k=0,1,2,\ldots, \tag{3.16}$$

in which $\mathbf{J}_\mathbf{G}$ is the Jacobian matrix of $\mathbf{G}$ and $\mathbf{c}^{(k)}$ lies somewhere between $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(k+1)}$. Equation (3.16) implies that

$$\mathbf{u}^*-\mathbf{u}^{k+1}=(\mathbf{J}_\mathbf{G}(\mathbf{c}^{(k)}))^{k+1}(\mathbf{u}^*-\mathbf{u}^0),\quad k=0,1,2,\ldots.$$

Denote by $\rho(\mathbf{J}_\mathbf{G})$ the spectral radius of $\mathbf{J}_\mathbf{G}$. $\rho(\mathbf{J}_\mathbf{G}(\mathbf{c}^{(k)}))<1$ provides a necessary and sufficient condition for $(\mathbf{u}^*-\mathbf{u}^k)$ to converge to zero. Thus for convergence, it suffices to check that $\rho(\mathbf{J}_\mathbf{G})$ remains less than 1 in a suitable domain $D_0\subseteq D$. For more details see [2].

4 The numerical results

In this section, the FPM is applied to some examples, in order to validate the proposed method. The examples are chosen for comparison with the results of [6,22]. In both cases we have used the quartic spline weight function:

$$w(r)=\begin{cases}1-6r^2+8r^3-3r^4 & r\le 1,\\ 0 & r>1,\end{cases} \tag{4.1}$$

with $r=\frac{\|\mathbf{x}-\mathbf{x}_i\|}{d_i}$, where d_i is the support size of node i.

The numbers of boundary and interior nodes have stated in each case. Also in the proposed method, the second order approximation derivatives are needed. Thus, we have chosen $m=6$ in our computations.

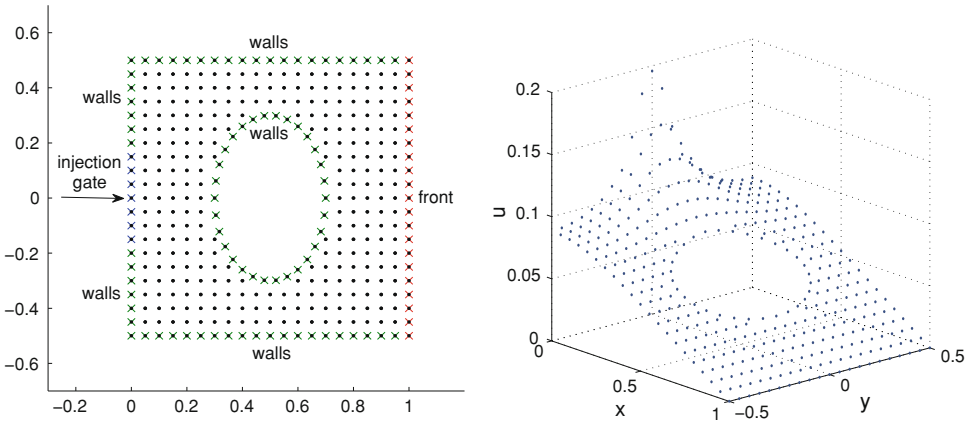

Fig. 1 Node distribution (*left*) and initial guess (*right*)

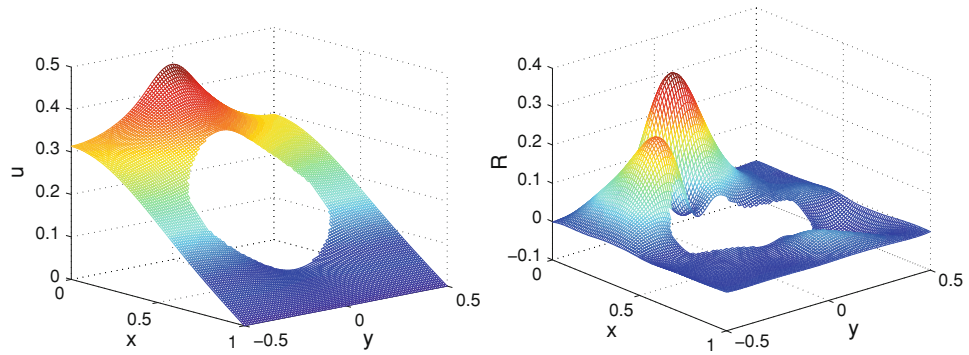

Fig. 2 FPM solution at a fine regular mesh (*left*) and residuals (*right*), it = 5

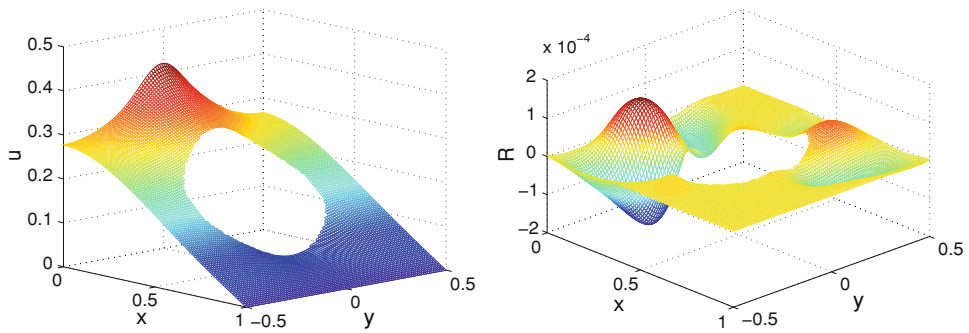

Fig. 3 FPM solution at a fine regular mesh (*left*) and residuals (*right*), it = 30

4.1 Problem 1

As a first numerical experiment we consider Eqs. (1.2) and (1.4) in a square box $[0, 1] \times [-1/2, 1/2]$ with an elliptical insert $\left(\frac{x-1/2}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1$, with $a = 0.2$ and $b = 0.3$. This problem has been taken from the literature [6]. The distribution of collocation nodes of such domain is shown in the left part of Fig. 1. Here we use $80 + 302$ boundary and interior collocation points. The exponent γ is 0.6 (which models polyethylene). The injection gate is the segment $x = 0, \ -\xi \le y \le \xi, \ \xi = 0.15$, highlighted with an arrow in Fig. 1.

Along it we have enforced the Neumann injection boundary condition. The profile q_{in} is

$$q_{in}(y) = 0.9(1 - 3(y/\xi)^2 + 2(|y|/\xi)^3), \tag{4.2}$$

such that the exact solution is smooth at both ends $(0, \pm\xi)$ of the injection segment. In order to start the iterations, we have used the solution of the Laplace equation on the same domain as an initial guess. Such an initial guess is shown in the right part of Fig. 1. However, using a constant function as an initial guess also leads to final result after more iterations. In the computations we put $d_i = 2.1 \times \delta$, in order to ensure the regularity of matrix $\mathbf{A}$ in MLS, in which $\delta = 0.05$ is an approximate distance between nodes. Figures 2 and 3 show the pressure function u at a fine regular mesh (left parts) and the corresponding residuals (right parts) after 5 and 30 iterations, respectively. Note that the values of u are calculated using Eq. (3.15). The residual of the nonlinear PDE drops to about 10^{-8} after 50 iterations. The condition number of the coefficient matrix $\mathbf{K}$ in Eq. (3.12) is about 10^3 and its sparseness structure is shown in the left part of Fig. 4. For

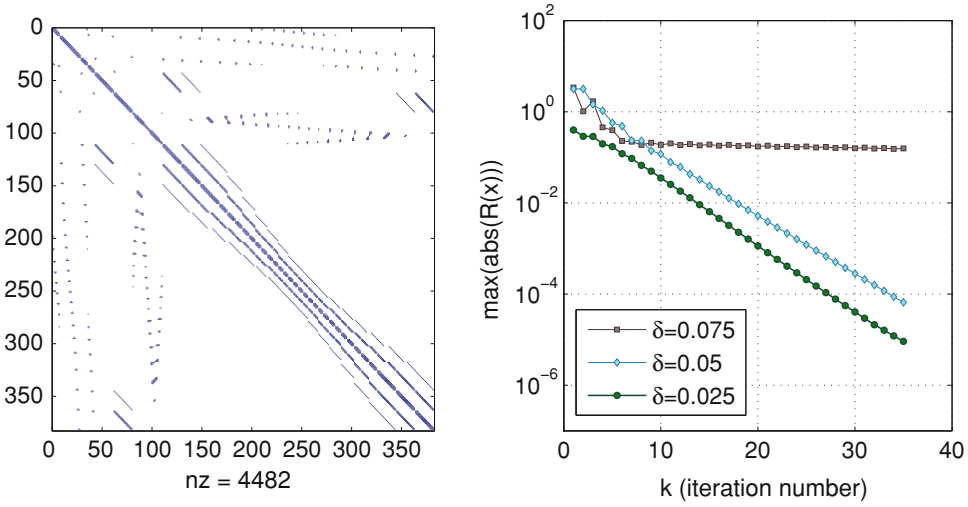

Fig. 4 Sparseness structure of coefficient matrix (*left*) and convergence of the iterative solution procedure for Eq. (3.4) (*right*)

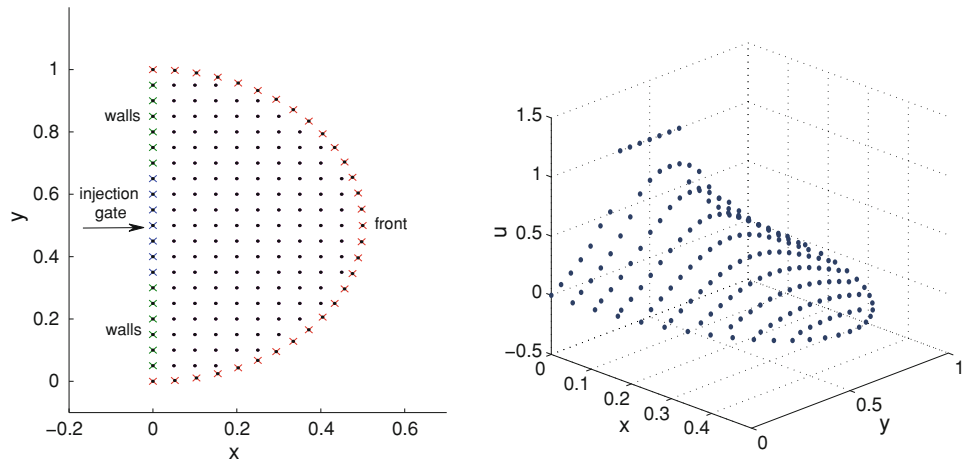

Fig. 5 Node distribution (*left*) and initial guess (*right*)

the three–point collocation sets with $\delta = 0.025, 0.05, 0.075$, the maximum residual at the mesh points for the iterative procedure is shown in the right part of Fig. 4. It is observed that the maximum residual decreases linearly for the iterations for two cases $\delta = 0.025, 0.05$ and the numerical solution for $\delta = 0.05$ is as good as for the case of $\delta = 0.025$, while for the case $\delta = 0.075$, the maximum residual does not drop to zero. Moreover we have checked $\rho(\mathbf{J_G}) < 1$ as another criterion for convergence. For the iterative scheme $\rho(\mathbf{J_G})$ is about 0.8 for $\delta \leq 0.05$ while $\rho(\mathbf{J_G}) = 7.3$ for $\delta = 0.075$.

4.2 Problem 2

In this problem, Eqs. (1.2) and (1.3) are solved in a semi-disk $x^2 + (y - 0.5)^2 \leq 0.5^2,\ x \geq 0$ depicted in Fig. 5. The injection gate is $0.5 - \xi \leq y \leq 0.5 + \xi,\ x = 0,\ \xi = 0.15$. The node distribution of such domain is depicted in the left part of the Fig. 5. We use $50 + 137$ boundary and interior points, respectively. The pressure profile is $p_{in} = 1$ and the exponent γ is 0.6 as before.

The problem of this kind is investigated in [5]. All parameters are chosen as before. Figures 6 and 7 show the pressure function u at a fine regular mesh (left parts) and the corresponding residuals (right parts) after 5 and 30 iterations, respectively. However the difference is not apparent to the naked eye but $\|\mathbf{u}^{(30)} - \mathbf{u}^{(5)}\|_\infty$ is about 10^{-3}. As can be seen from Fig. 8, the maximum residual at the mesh points decreases linearly for the iterative process for the three collocation point sets with $\delta = 0.025, 0.05, 0.075$. Also in this case $\rho(\mathbf{J_G})$ is about 0.7 for the iterative scheme for $\delta \leq 0.05$, while $\rho(\mathbf{J_G}) = 0.92$ for the case of $\delta = 0.075$. Moreover it should be noted that in this case the residuals drop to zero faster than in the case of Nuemann boundary condition in the inlet gate. This should be expected, since in the case of Dirichlet boundary condition no linearization is needed and the boundary conditions have been enforced directly.

Also in [5] an oscillation appears in the solution at the injection gate while there is no oscillation in the numerical results obtained by the method proposed in the current paper.

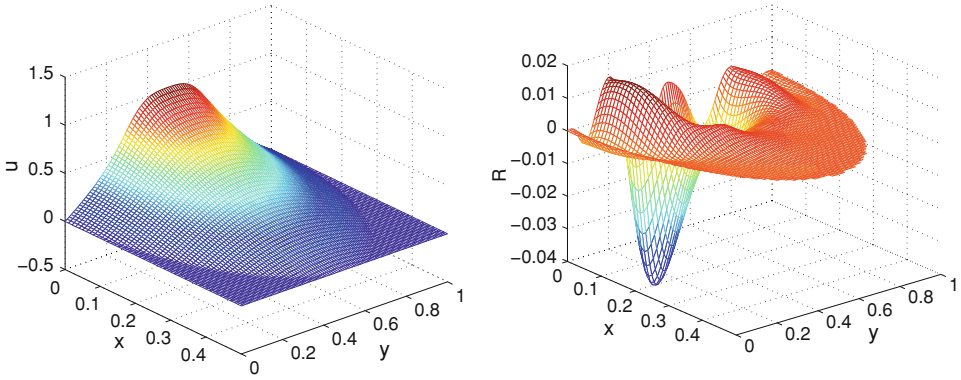

Fig. 6 FPM solution at a fine regular mesh (*left*) and residuals (*right*), it = 5

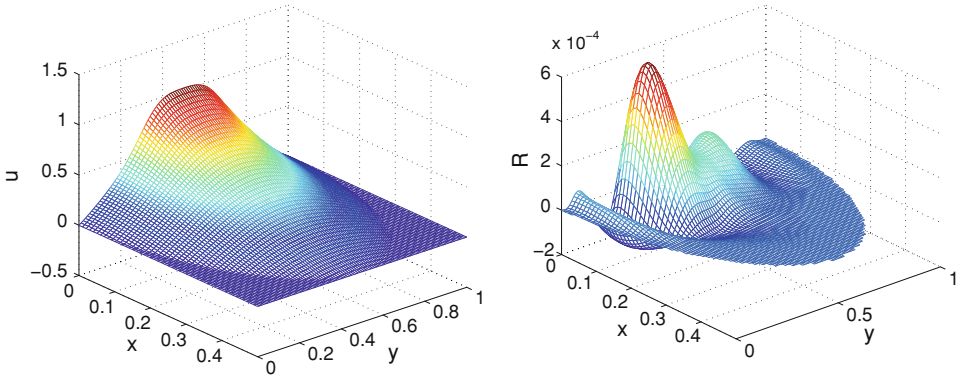

Fig. 7 FPM solution at a fine regular mesh (*left*) and residuals (*right*), it = 30

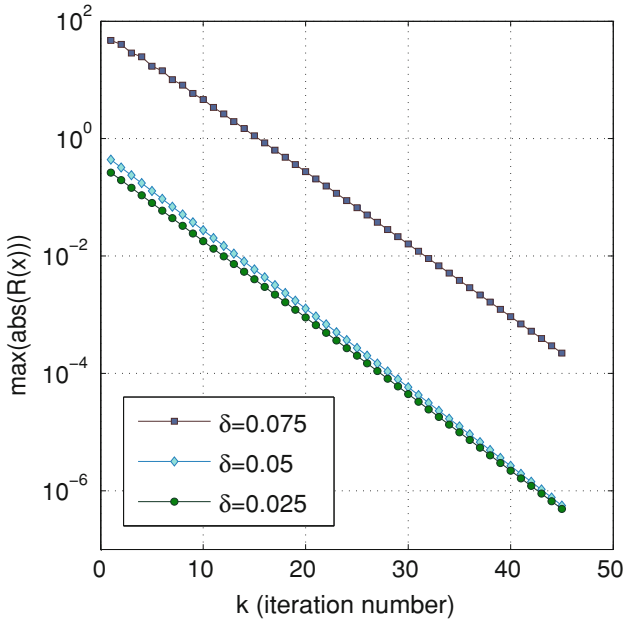

Fig. 8 Convergence of iterative solution procedure for Eq. (3.4)

5 Conclusion

In this paper the numerical solution of the two-dimensional nonlinear, elliptic p-Laplace equation has been obtained using the FPM. The proposed technique reduces the given nonlinear PDE into a sequence of linear ones. These equations are solved using a simple iterative procedure that employs the solution of the Laplace equation with the same boundary conditions as an initial guess. The method is a truly meshfree method, which requires neither domain elements nor integrations and our numerical examples show good results, with the accuracy of solving the nonlinear PDE. The simplicity in the implementation shows the efficiency of the FPM. In this method we have found no oscillation near the injection gate that was appeared in some previous works. We believe that the new approach can also be applied to other typical two dimensional PDEs involving the p-Laplace operator as the core nonlinearity.

Acknowledgments The authors are very grateful to both reviewers for carefully reading the paper and for their comments and suggestions which have improved the paper.

References

1. Aronsson G, Janfalk U (1992) On Hele-Shaw flow of power-law fluids. Eur J Appl Math 3:343–366
2. Atkinson KE (1989) An introduction to numerical analysis. Wiley, New York
3. Belytschko T, Lu Y, Gu L (1994) Element free Galerkin methods. Int J Numer Meth Eng 37:229–256
4. Bernal F (2008) Meshless methods for elliptic and free-boundary problems. Ph. D. thesis, Universidad Carlos III De Madrid, Madrid
5. Bernal F, Kindelan M (2007) RBF meshless modeling of non-Newtonian Hele-Shaw flow. Eng Anal Bound Elem 31:863–874

6. Bernal F, Kindelan M (2009) A meshless solution to the p-Laplace equation. In: Ferreira AJM, Kansa EJ, Fasshauer GE, Leitão VMA (eds) Computational methods in applied sciences. Progress on meshless methods, Barcelona, Spain
7. Boroomand B, Najjar M, Oñate E (2009) The generalized finite point method. Comput Mech 44:173–190
8. Dehghan M (2006) Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices. Math Comput Simul 71:16–30
9. Dehghan M, Shokri A (2008) A numerical method for solution of the two-dimensional sine–Gordon equation using the radial basis functions. Math Comput Simul 79:700–715
10. Dehghan M, Ghesmati A (2010) Combination of meshless local weak and strong (MLWS) forms to solve the two dimensional hyperbolic telegraph equation. Eng Anal Bound Elements 34: 324–336
11. Dolbow J, Belytschko T (1998) An introduction to programming the meshless element free Galerkin method. Comput Meth Eng 5(3):207–241
12. Hieber CA, Shen SF (1979) A finite-element/finite difference simulation of the injection moulding filling process. J Non-Newtonian Fluid Mech 7:1–32
13. Kim DW, Liu WK (2006) Maximum principle and convergence analysis for the meshfree point collocation method. SIAM J Numer Anal 44(2):515–539
14. Kim DW, Yoon Y-C, Liu WK, Belytschko T (2007) Extrinsic meshfree approximation using asymptotic expansion for interfacial discontinuity of derivative. J Comput Phys 221:370–394
15. Kim DW, Liu WK, Yoon Y-C, Belytschko T, Lee S-H (2007) Meshfree point collocation method with intrinsic enrichment for interface problems. Comput Mech 40:1037–1052
16. Li S, Liu WK (2004) Meshfree particle methods. Springer, Berlin
17. Liu W, Jun S, Zhang Y (1995) Reproducing kernel particle method. Int J Numer Methods Fluids 20:1081–1106
18. Liu WK, Li S, Belytschko T (1997) Moving least squares reproducing kernel methods, Part I: methodology and convergence. Comput Methods Appl Mech Eng 143:113–154
19. Liszka T, Orkisz J (1975) The finite difference method at arbitrary irregular grids and its application in applied mechanics. Comp Struct 11:38–95
20. Liu WK, Jun S, Li S, Adee J, Belytschko T (1995) Reproducing kernel particle methods for structural dynamics. Int J Numer Meth Eng 5(3):1655–1679
21. Liu WK, Chen Y, Jun S, Chen JS, Belytschko T, Pan C, Uras RA, Chang CT (1996) Overview and applications of the reproducing kernel particle method. Arch Comput Meth Eng 3(1):3–80
22. Mirzaei D, Dehghan M (2010) MLPG approximation to the p-Laplace problem. Comput Mech 46:805–812
23. Mirzaei D, Dehghan M (2010) A meshless based method for solution of integral equations. Appl Numer Math 60:245–262
24. Monaghan JJ (1992) Smoothed particle hydrodynamics: some recent improvement and applications. Annu Rev Astron Phys 30:543–574
25. Nayroles B, Touzot G, Villon P (1992) Generalizing the FEM: diffuse approximation and diffuse elements. Comput Mech 10: 307–318
26. Oñate E, Idelsohn S, Zienkiewicz OC, Fisher T (1995) Finite point method for analysis of fluid flow problems. In: Proceedings of the 9th international conference on finite element methods in fluids, Venize, Italy, pp 15–21
27. Oñate E, Idelsohn S, Zienkiewicz OC, Taylor RL (1996) A finite point method in computational mechanics, applications to convective transport and fluid flow. Int J Num Meth Eng 39:3839–3866
28. Oñate E, Idelsohn S, Zienkiewicz OC, Taylor RL (1996) A stabilized finite point method for analysis of fluid mechanics's problems. Comput Meth Appl Eng 139:315–347
29. Oñate E, Idelsohn S (1998) A meshfree finite point method for advective–diffusive transport and fluid flow problems. Comput Mech 21:283–292
30. Perona P, Malik J (1990) Scale-space and edge detection using anistotropic diffusion. IEEE Trans Pattern Anal Mach Intell 12: 629–639
31. Perrone N, Kao R (1975) A general finite difference method for arbitrary meshes. Comp Struct 5:45–47
32. Randles PW, Libersky LD (1996) Smoothed particle hydrodynamics: some recent improvement and applications. Comp Meth Appl Mech Eng 139:375–408
33. Zuppa C (2003) Error estimates for moving least squares approximations. Bull Brazil Math Soc 34(2):231–249