Comput Mech (2005) 36: 298–306
DOI 10.1007/s00466-005-0667-4

ORIGINAL PAPER

Xin Liu · G. R. Liu · Kang Tai · K. Y. Lam

Radial point interpolation collocation method (RPICM) for the solution of nonlinear poisson problems

Received: 22 April 2004 / Accepted: 9 February 2005 / Published online: 14 April 2005

Abstract This paper applies radial point interpolation collocation method (RPICM) for solving nonlinear Poisson equations arising in computational chemistry and physics. Thin plate spline (TPS) Radial basis functions are used in the work. A series of test examples are numerically analysed using the present method, including 2D Liouville equation, Bratu problem and Poisson-Boltzmann equation, in order to test the accuracy and efficiency of the proposed schemes. Several aspects have been numerically investigated, namely the enforcement of additional polynomial terms; and the application of the Hermite-type interpolation which makes use of the normal gradient on Neumann boundary for the solution of PDEs with Neumann boundary conditions. Particular emphasis was on an efficient scheme, namely Hermite-type interpolation for dealing with Neumann boundary conditions. The numerical results demonstrate that a good accuracy can be obtained. The *h*-convergence rates are also studied for RPICM with coarse and fine discretization models.

X. Liu (✉)
Department of Mechanics, Zhejiang University,
Hangzhou, 310027, P.R. China
E-mail: sdcxliu@zju.edu.cn, cfxinliu@hotmail.com

G. R. Liu · K. Tai
SMA-Fellow, Singapore-MIT Alliance, Singapore

G. R. Liu
Dept. of Mech. Eng, Centre for ACES,
National University of Singapore, Singapore

K. Tai
School of Mech. Prod. Eng,
Nanyang Technological University, Singapore

K. Y. Lam
Institute of High Performance Computing, Singapore

X. Liu
SMA-Research Fellow, Singapore-MIT Alliance, Singapore

Keywords RPICM · Hermite-type interpolation · Meshfree · Nonlinear poisson equation · Thin plate spline

1 Introduction

In recent years, research on meshless (meshfree) methods has made significant progress in science and engineering, especially in the area of computational mechanics [1]. Meshfree methods based on radial basis function (RBF) have a clear advantage over other meshfree methods due to its simplicity and stability in field variable interpolation [2–3]. However, traditional radial basis function methods result in fully-populated matrices [4–7], which limit its application in large scale practical engineering problems. Radial point interpolation method (RPIM) was proposed by Liu and Wang [2], and has been improved and applied with Galerkin-based formulations [3–4]. In [8], an efficient radial basis point interpolation collocation-based formulation, namely radial point interpolation collocation method (RPICM), has been presented and applied to solve 2-D linear elastic problem. In RPICM, the collocation scheme, which is simple and efficient to solve partial differential equations, has been adopted without the need of numerical integrations. In addition, RPICM uses RBFs in a locally support domain so that the system matrix becomes sparse and hence can be applied to more complicated problems.

However, the research results in [9–10] showed that the accuracy obtained by using direct collocation scheme is a bit poor especially on boundary. In addition, the collocation scheme, which has difficulties in dealing with Neumann boundary conditions, is very different from the Galerkin scheme that can deal with Neumann boundary conditions naturally. In [9], a Hermite-type interpolation scheme in generalized finite difference

method has been proposed to improve the accuracy of collocation-based approach for solving solid problems. In [10], the Hermite-type interpolation was successfully extended to compactly support radial basis function method. In our paper, the Hermite-type interpolation is adopted in RPICM in order to improve the accuracy for solving nonlinear Poisson equation with Neumann boundary conditions. Approximate field functions are carried out not only with the nodal values but also with the normal gradient on Neumann boundaries by taking advantage of the point interpolation method based on radial basis functions.

Solution of Poisson type differential equations can be obtained by finding an approximate particular solution to the forcing term followed by a boundary element method. Recently, Ramachandran proposed osculatory radial basis functions (RBFs), called osculatory interpolation, to find the approximate particular solution [12–13]. In traditional RBF interpolation, the gradient information is not used to construct a better approximation to solution function. However, in osculatory RBF interpolation, both the value of the function and its normal derivatives at the nodal points are utilized for constructing the approximate function, and it produces higher accuracy. In our paper, gradient information, called Hermite-type interpolation, is adopted only for these nodes on Neumann boundaries. Meanwhile, our method is based on point interpolation schemes, so it is local.

In this paper, the radial point interpolation collocation method (RPICM) is applied. The formulation for constructing shape functions based on radial point interpolation and Hermite radial point interpolation is described and formulated in Sect. 2 and 3. The detail collocation schemes are discussed in Sect. 4. In Sect. 5, the accuracy and simplicity of this presented approach is shown numerically by a series of test examples. We conclude with a summary in Sect. 6.

2 Radial point interpolation method

When MQ and Gaussian radial basis functions were adopted to construct field function in local support domain, and then it is applied for solving PDEs, *h*-convergence usually fails because it is heavily dependent on the choice of the shape parameter [11]. In view of this fact, the approximation of a function $u(\mathbf{x})$ will be constructed using TPS radial basis functions in this paper. It may be written as a linear combination of n radial basis functions, viz.,

$$u(\mathbf{x}) \cong \hat{u}(\mathbf{x}) = \sum_{i=1}^{n} a_i \phi(\|\mathbf{r} - \mathbf{r_i}\|) \tag{1}$$

$$\phi(\|\mathbf{r} - \mathbf{r_i}\|) = \|\mathbf{r} - \mathbf{r_i}\|^{2m} \log(\|\mathbf{r} - \mathbf{r_i}\|) \tag{2}$$

where n is the number of points in the support domain near $\mathbf{x}$, a_i are coefficients to be determined.
Here $\mathbf{r}$ is the distance between two points. In 2-D problems, we have

$$\|\mathbf{r} - \mathbf{r}_i\| = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{3}$$

The interpolations of a function at the kth point can have the form of

$$\begin{aligned} \hat{u}(\mathbf{x}_k) &= a_1 \phi(\|\mathbf{r}_k - \mathbf{r}_1\|) \\ &\quad + a_2 \phi(\|\mathbf{r}_k - \mathbf{r}_2\|) + \cdots + a_n \phi(\|\mathbf{r}_k - \mathbf{r}_n\|), \\ &\quad k = 1, 2, \ldots, n \end{aligned} \tag{4}$$

The function interpolation can be expressed in a matrix form as follows:

$$\hat{\mathbf{u}}^e = \mathbf{\Phi a} \tag{5}$$

$$\hat{\mathbf{u}}^e = [\, \hat{u}(\mathbf{x}_1) \quad \cdots \quad \hat{u}(\mathbf{x}_k) \cdots \hat{u}(\mathbf{x}_n) \,]^{\mathrm{T}} \tag{6}$$

$$\mathbf{a} = [\, a_1 \quad \cdots \quad a_i \quad \cdots \quad a_n \,]^{\mathrm{T}} \tag{7}$$

Thus, the unknown coefficients vector is found to be

$$\mathbf{a} = \mathbf{\Phi}^{-1} \hat{\mathbf{u}}^e \tag{8}$$

The form of the approximation function can be obtained as follows:

$$\hat{u}(\mathbf{x}) = \boldsymbol{\varphi a} = \boldsymbol{\varphi} \mathbf{\Phi}^{-1} \hat{\mathbf{u}}^e = \boldsymbol{\psi} \hat{\mathbf{u}}^e \tag{9}$$

$$\boldsymbol{\psi} = [\, \phi(\|\mathbf{r} - \mathbf{r}_1\|) \quad \phi(\|\mathbf{r} - \mathbf{r}_2\|) \quad \cdots \quad \phi(\|\mathbf{r} - \mathbf{r}_n\|) \,] \tag{10}$$

where the matrix of shape functions can be expressed as follows

$$\boldsymbol{\psi} = \boldsymbol{\varphi} \mathbf{\Phi}^{-1} = [\, \psi_1 \quad \cdots \quad \psi_i \quad \cdots \quad \psi_n \,] \tag{11}$$

in which $\psi_1 (i = 1, \ldots, n)$ are shape functions for points in the support domain, which satisfy.

$$\psi_i(x_j) = \begin{cases} 1, & \mathrm{j} = \mathrm{i} \\ 0, & \mathrm{j} \neq \mathrm{i} \end{cases} \tag{12}$$

$$\mathbf{\Phi} = \begin{bmatrix} \phi(\|\mathbf{r}_1 - \mathbf{r}_1\|) & \cdots & \phi(\|\mathbf{r}_1 - \mathbf{r}_\mathrm{i}\|) & \cdots & \phi(\|\mathbf{r}_1 - \mathbf{r}_\mathrm{n}\|) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \phi(\|\mathbf{r}_\mathrm{i} - \mathbf{r}_1\|) & \cdots & \phi(\|\mathbf{r}_\mathrm{i} - \mathbf{r}_\mathrm{i}\|) & \cdots & \phi(\|\mathbf{r}_\mathrm{i} - \mathbf{r}_\mathrm{n}\|) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \phi(\|\mathbf{r}_\mathrm{n} - \mathbf{r}_1\|) & \cdots & \phi(\|\mathbf{r}_\mathrm{n} - \mathbf{r}_\mathrm{i}\|) & \cdots & \phi(\|\mathbf{r}_\mathrm{n} - \mathbf{r}_\mathrm{n}\|) \end{bmatrix} \tag{13}$$

Whether $\mathbf{\Phi}^{-1}$ exists and how to assure its existence may depend on the choice of the local support domain. Enough number of nodes and good performance arrangement of nodes in the local support domain near collocation point $\boldsymbol{x}$ must be constructed for this collocation scheme. In our examples in Sect. 5, the number of nearest points in the support domain will be chosen to be 9, 16, 25 or 36 in order to observe its influence on accuracy and stability.

3 Hermite radial point interpolation method

The approximation of a function $u(\mathbf{x})$ may be written as a linear combination of radial basis functions at the n nodes within support domain near $\mathbf{x}$ and its normal derivatives at the n_b nodes on Neumann boundaries:

$$u(\mathbf{x}) \cong \hat{u}(\mathbf{x}) = \sum_{i=1}^{n} a_i \phi_i + \sum_{j=1}^{n_b} b_j \frac{\partial \phi_j^b}{\partial n} + G(\mathbf{x}) \tag{14}$$

$$\phi_i = \phi(\|\mathbf{x} - \mathbf{x}_i\|),\ \phi_j^b = \phi(\|\mathbf{x} - \mathbf{x}_j^b\|),$$

$$\frac{\partial \phi_j^b}{\partial n} = l_j^x \frac{\partial \phi_j^b}{\partial x} + l_j^y \frac{\partial \phi_j^b}{\partial y} \tag{15}$$

Constant term:

$$G(\mathbf{x}) = g_0 \tag{16}$$

Linear polynomial:

$$G(\mathbf{x}) = g_0 + g_1 x + g_2 y \tag{17}$$

Square polynomial:

$$G(\mathbf{x}) = g_0 + g_1 x + g_2 y + g_3 x^2 + g_4 xy + g_5 y^2 \tag{18}$$

a_i are coefficients which correspond to radial basis ϕ_i of function, b_j are coefficients which correspond to normal derivative of radial basis ϕ_j of function at the points on Neumann boundaries, and $g_0, g_1, g_2, \ldots$ are the coefficients of the additional unknown polynomial. ϕ is the radial basis. l_j^x, l_j^y are the elements of normal vector at the jth point on Neumann boundaries.

The coefficients a_i and b_j in Equation (14) can be determined by enforcing that the function interpolations pass through all n nodes within the support domain and the normal derivatives' interpolations of function pass through n_b nodes on Neumann boundaries.

The interpolations of the function at the kth point have the form:

$$\hat{u}_k = \hat{u}(\mathbf{x}_k) = \sum_{i=1}^{n} a_i \phi_{ki} + \sum_{j=1}^{n_b} b_j \frac{\partial \phi_{kj}^b}{\partial n} + G(\mathbf{x}_k), \quad k = 1, 2, \ldots, n \tag{19}$$

$$\phi_{kj}^b = \phi\left(\left\|\mathbf{x}_k - \mathbf{x}_j^b\right\|\right), \quad \phi_{kj}^b = \phi\left(\left\|\mathbf{x}_k - \mathbf{x}_j^b\right\|\right) \tag{20}$$

The interpolations of the normal derivatives of function at the mth point on the Neumann boundaries have the form:

$$\frac{\partial \hat{u}_m^b}{\partial n} = \frac{\partial \hat{u}^b(\mathbf{x}_m)}{\partial n} = \sum_{i=1}^{n} a_i \frac{\partial \phi_{mi}}{\partial n} + \sum_{j=1}^{n_b} b_j \frac{\partial}{\partial n}\left(\frac{\partial \phi_{mj}^b}{\partial n}\right) + \frac{\partial G(\mathbf{x}_m)}{\partial n}, m = 1, 2, \ldots, n_b \tag{21}$$

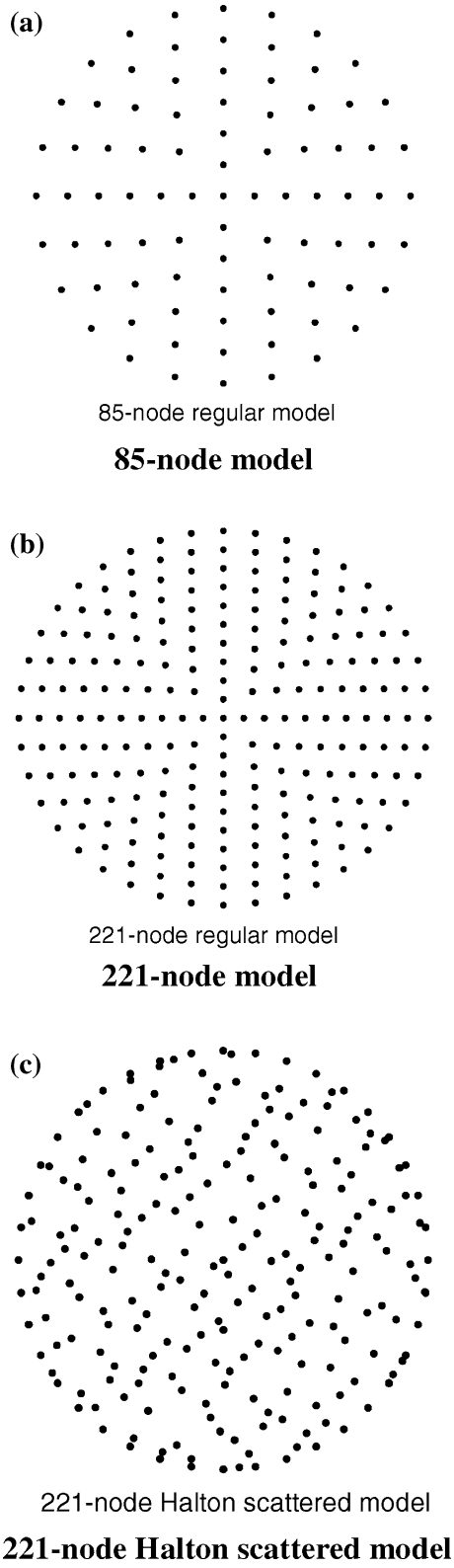


Fig. 1a–c Quasi-Uniform node discrezation model and scattered point model for unit circle solution domain for example 1 and 2. **a** 85-node model. **b** 221-node model. **c** 221 node Halton scattered model

In addition, the additional polynomial terms have to satisfy an extra requirement that guarantees unique approximation of the function, and the following constraints are usually imposed (see, [15] for the details):

$$\sum_{k=1}^{n} a_k = 0 \tag{22}$$

$$\sum_{k=1}^{n} a_k x_k = 0,\ \sum_{k=1}^{n} a_k y_k = 0 \tag{23}$$

$$\sum_{k=1}^{n} a_k x_k^2 = 0,\ \sum_{k=1}^{n} a_k x_k y_k = 0,\ \sum_{k=1}^{n} a_k y_k^2 = 0 \tag{24}$$

For constant additional term, only one constraint equation (22) is enforced. For linear additional term, three constraint equations (22–23) are enforced. For quadratic additional terms, six constraint equations (22–24) are enforced.

The interpolations of function and normal derivatives on the Neumann boundaries can be expressed by matrix formulations as follows:

$$\hat{\mathbf{u}}^e = \mathbf{\Psi a} \tag{25}$$

where $\hat{\mathbf{u}}^e$ is the vector that collects all variables of the nodal function values at the n nodes in the support domain and all variables of normal derivatives of the nodal function at the n_b nodes on the Neumann boundaries in the support domain.

$$\hat{\mathbf{u}}^e = \left[\ \hat{u}_1 \ \cdots \ \hat{u}_k \ \cdots \frac{\partial \hat{u}_1^b}{\partial n} \ \cdots \ \frac{\partial \hat{u}_m^b}{\partial n} \ \cdots \ \frac{\partial \hat{u}_{n_b}^b}{\partial n} \ \ 0 \ \right]^{\mathrm{T}} \tag{26}$$

The coefficients vector $\mathbf{a}$ is defined as

$$\mathbf{a} = [a_1 \ \cdots \ a_i \ \cdots \ a_n \ \ b_1 \ \cdots \ b_j \ \cdots \ b_{n_b} \ \ c_0 \ \cdots]^{\mathrm{T}} \tag{27}$$

Thus the unknown coefficients vector

$$\mathbf{a} = \mathbf{\Psi}^{-1}\hat{\mathbf{u}}^e \tag{28}$$

Finally, the approximation form of function can be obtained as follows:

$$\hat{\mathbf{u}}(\mathbf{x}) = \mathbf{\Phi a} = \mathbf{\Phi\Psi}^{-1}\hat{\mathbf{u}}^e = \boldsymbol{\psi}\hat{\mathbf{u}}^e \tag{29}$$

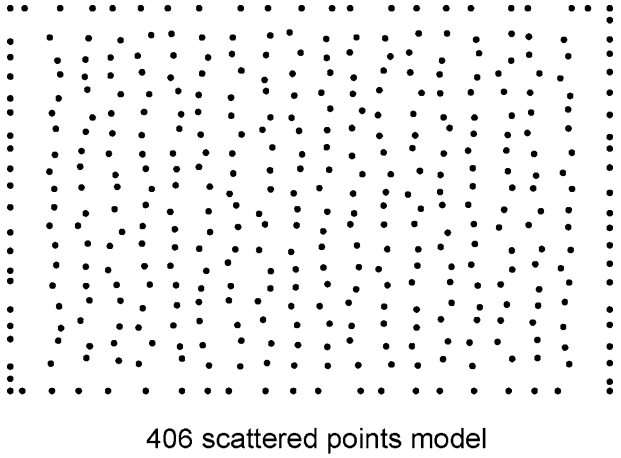


Fig. 2 406-node scattered points model for Example 3

The matrix of radial basis, its normal derivatives and linear additional terms is defined by

$$\mathbf{\Phi} = \left[\phi_1 \cdots \phi_i \cdots \phi_n \frac{\partial \phi_1^b}{\partial n} \cdots \frac{\partial \phi_m^b}{\partial n} \cdots \frac{\partial \phi_{n_b}^b}{\partial n} 1 \ \ x \ \ y\right] \tag{30}$$

The matrix of shape functions can be expressed as follows

$$\boldsymbol{\psi} = \mathbf{\Phi\Psi}^{-1} = \left[\psi_1 \ \cdots \ \psi_i \ \cdots \ \psi_n \ \ \psi_1^H \ \cdots \ \psi_m^H \ \cdots \ \psi_{n_b}^H\right] \tag{31}$$

Here $\psi(i = 1, 2, \ldots, n)$ and $\psi_j^H(j = 1, 2, \ldots, n_b)$ are shape functions.

Finally, the function $u(\mathbf{x})$ can be expressed as follows:

$$\hat{u} = \sum_{k=1}^{n} \psi_k \hat{u}_k^e + \sum_{j=1}^{n_b} \psi_j^H \frac{\partial \hat{u}_j^e}{\partial n} \tag{32}$$

4 Collocation schemes

Let us consider the following general second-order partial differential equation, given by

$$\Psi(u) = A(x,y)\frac{\partial^2 u}{\partial x^2} + 2B(x,y)\frac{\partial^2 u}{\partial x \partial y} + C(x,y)\frac{\partial^2 u}{\partial y^2} - \Phi\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) \quad \text{in } \Omega \tag{33}$$

This PDE can be solved using point interpolation collocation method when appropriate boundary conditions are imposed.

- **When Φ is linear in u**

If the expression Φ is linear in *u, i.e.*, i

$$\Phi = D(x,y)\frac{\partial u}{\partial x} + E(x,y)\frac{\partial u}{\partial y} + F(x,y)u + G(x,y) \tag{34}$$

then the equation would be a linear PDE.
together with the general boundary:
Neumann boundary condition:

$$L_{b1}(u) = \mathbf{n}^{\mathrm{T}} \cdot \nabla u + \bar{g}_n = 0 \quad \text{on } \Gamma_{b1} \tag{35}$$

Dirichlet boundary condition:

$$u - \bar{u} = 0 \quad \text{on } \Gamma_{b2} \tag{36}$$

The coefficients *A, B, C, D, E, F* and *G* may all depend upon x and y.

Assuming that there are N_d internal (domain) points and $N_b = N_{b1} + N_{b2}$ boundary points, where N_{b1} are Neumann boundary points and N_{b2} are Dirichlet boundary points.

In general, the location of the collocation points can be different from the location of nodes in the discretised model. However, for the sake of simplicity, collocation points are the same as the nodes of the model.

Table 1 quasi-uniform 85-node coarse model with Quadratic additional polynomial ($m=6$)

	results in [12]	9 nodes in SD	16 nodes in SD	25 nodes in SD
u at $r=0.0$	0.1127	No convengence	0.1561	0.1194
u at $r=0.7$	0.5964	–	0.6219	0.5994
u at $r=0.9$	0.8589	–	0.8720	0.8609
$\frac{\partial u}{\partial n}$ at $r=1.0$	1.4332	–	1.2836	1.3960

Table 2 quasi-uniform 221-node fine model with linear additional polynomial ($m=4$)

	results in [12]	9 nodes in SD	16 nodes in SD	25 nodes in SD
u at $r=0.0$	0.1127	0.3079	No convengence	0.3917
u at $r=0.7$	0.5964	0.6861	–	0.7398
u at $r=0.9$	0.8589	0.8889	–	0.9075
$\frac{\partial u}{\partial n}$ at $r=1.0$	1.4332	1.1387	–	0.9339

Table 3 quasi-uniform 221-node fine model with Quadratic additional polynomial ($m=6$)

	results in [12]	9 nodes in SD	16 nodes in SD	25 nodes in SD
u at $r=0.0$	0.1127	0.1167	0.1437	0.1254
u at $r=0.7$	0.5964	0.5765	0.6098	0.6026
u at $r=0.9$	0.8589	0.8597	0.8630	0.8620
$\frac{\partial u}{\partial n}$ at $r=1.0$	1.4332	1.4404	1.3943	1.3954

The following N_d equations are satisfied in internal domain nodes:

$$\Psi_i = \Psi(\hat{u}_i) = A\frac{\partial^2 \hat{u}_i}{\partial x^2} + B\frac{\partial^2 \hat{u}_i}{\partial x \partial y} + C\frac{\partial^2 \hat{u}_i}{\partial y^2} - \left(D\frac{\partial \hat{u}_i}{\partial x} + E\frac{\partial \hat{u}_i}{\partial y} + F\hat{u}_i + G(x_i, y_i)\right) = 0 \quad \text{in } \Omega \tag{37}$$

Table 4 Halton scattered 221-node fine model with Quadratic additional polynomial ($m=6$)

	results in [12]	9 nodes	16 nodes	25 nodes	30 nodes
u at $r=0.0$	0.1127	0.1052	0.3861	0.1018	0.1103
u at $r=0.7$	0.5964	0.5938	0.6676	0.5857	0.5231
u at $r=0.9$	0.8589	0.8583	0.8809	0.8520	0.8579
$\frac{\partial u}{\partial n}$ at $r=1.0$	1.4332	–	1.2342	1.4922	1.4414

Table 5 quasi-uniform 85-node coarse model with Quadratic additional polynomial ($m=6$)

	results in [12]	9 nodes in SD	16 nodes in SD	25 nodes in SD
u at $r=0.0$	1.3176	No Convengence	1.3139	1.3161
u at $r=0.7$	1.1559	–	1.1526	1.1550
u at $r=0.9$	1.0568	–	1.0548	1.0562
$\frac{\partial u}{\partial n}$ at $r=1.0$	−0.5900	–	−0.5668	−0.5823

Table 6 quasi-uniform 221-node fine model with linear additional polynomial ($m=4$)

	results in [12]	9 nodes in SD	16 nodes in SD	25 nodes in SD
u at $r=0.0$	1.3176	1.2801	1.5672	1.2610
u at $r=0.7$	1.1559	1.1347	1.2376	1.1183
u at $r=0.9$	1.0568	1.0491	1.0719	1.0428
$\frac{\partial u}{\partial n}$ at $r=1.0$	−0.5900	−0.5067	−0.4482	−0.4325

Table 7 quasi-uniform 221-node fine model with Quadratic additional polynomial ($m=6$)

	results in [12]	9 nodes in SD	16 nodes in SD	25 nodes in SD
u at $r=0.0$	1.3176	1.3158	1.3153	1.3155
u at $r=0.7$	1.1559	1.1588	1.1544	1.1546
u at $r=0.9$	1.0568	1.05632	1.0562	1.0560
$\frac{\partial u}{\partial n}$ at $r=1.0$	−0.5900	−0.5872	−0.58405	−0.5802

The following N_{b1} equations are satisfied on Neumann boundary Γ_{b1}:

$$\mathbf{n}^{\mathrm{T}} \cdot \nabla \hat{u}_i + \bar{g}_n = 0, \quad i = 1, \ldots, N_{b1} \tag{38}$$

The following N_{b2} equations are satisfied on Dirichlet boundary Γ_{b2}:

$$\hat{u}_i - \bar{u} = 0, \ i = 1, \ldots, N_{b2} \tag{39}$$

In collocation method, two equations will be imposed at all Neumann boundary nodes: one equation resulting from the known Neumann boundary condition, and one from the PDEs (i.e., the same as the interior of the domain).

$\hat{u}_i$ can be obtained by equation (9) or (32). Its derivatives can be obtained by the following equations:

Table 8 Halton scattered 221-node fine model with Quadratic additional polynomial ($m=6$)

	results in [12]	9 nodes in SD	16 nodes in SD	25 nodes in SD	30 nodes in SD
u at $r=0.0$	1.3176	1.3172	1.2821	1.3173	1.3169
u at $r=0.7$	1.1559	1.1554	1.1484	1.1562	1.0541
u at $r=0.9$	1.0568	1.0565	1.0543	1.0572	1.0565
$\frac{\partial u}{\partial n}$ at $r=1.0$	−0.5900	–	−0.5677	−0.5921	-0.5871

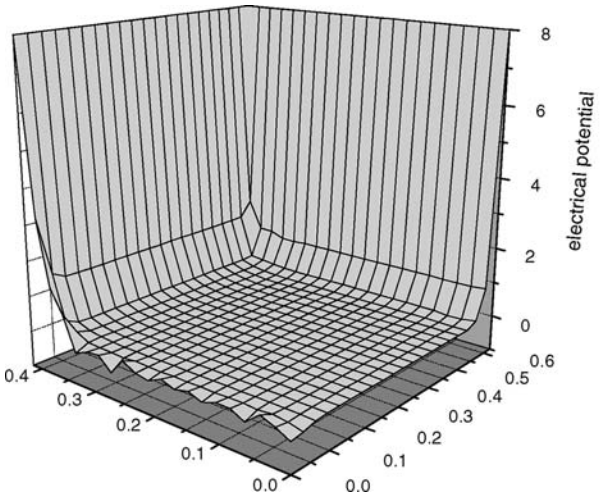
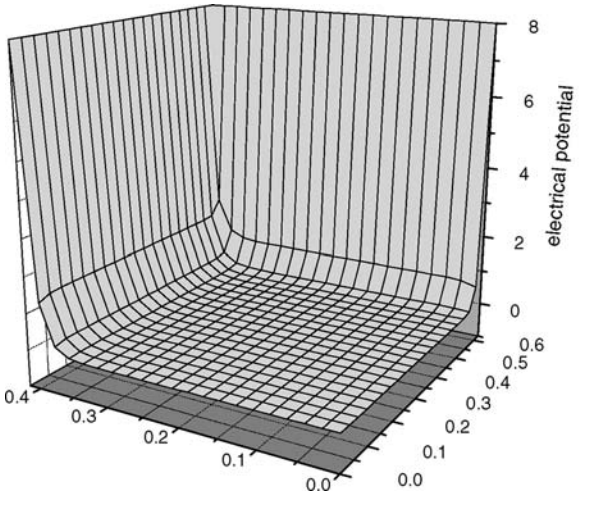


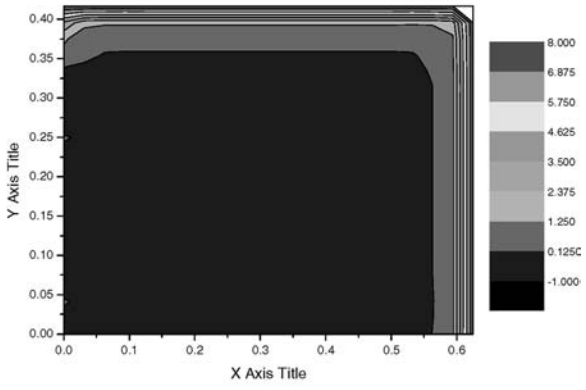
Fig. 3 The solution for 2D Poisson-Boltzmann equation with 21×21 uniform model TPS + linear additional polynomial without Hermite interpolation on Neumann boundaries 25 nearest points within every support domain

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{n} \phi_j \hat{u}_j^e,\ \frac{\partial \hat{u}(\mathbf{x})}{\partial x} = \sum_{j=1}^{n} \frac{\partial \phi_j}{\partial x} \hat{u}_j^e, \frac{\partial^2 \hat{u}(\mathbf{x})}{\partial x^2} = \sum_{j=1}^{n} \frac{\partial^2 \phi_j}{\partial x^2} \hat{u}_j^e\ \frac{\partial \hat{u}(\mathbf{x})}{\partial y} = \sum_{j=1}^{n} \frac{\partial \phi_j}{\partial y} \hat{u}_j^e,\ \frac{\partial^2 \hat{u}(\mathbf{x})}{\partial y^2} = \sum_{j=1}^{n} \frac{\partial^2 \phi_j}{\partial y^2} \hat{u}_j^e, \frac{\partial^2 \hat{u}(\mathbf{x})}{\partial x \partial y} = \sum_{j=1}^{n} \frac{\partial^2 \phi_j}{\partial x \partial y} \hat{u}_j^e \tag{40}$$

$$\hat{u}_i = \hat{u}(\mathbf{x}_i),\ \frac{\partial \hat{u}_i}{\partial x} = \frac{\partial \hat{u}(\mathbf{x}_i)}{\partial x},\ \frac{\partial^2 \hat{u}_i}{\partial x^2} = \frac{\partial^2 \hat{u}(\mathbf{x}_i)}{\partial x^2}$$
$$\frac{\partial \hat{u}_i}{\partial y} = \frac{\partial \hat{u}(\mathbf{x}_i)}{\partial y},\ \frac{\partial^2 \hat{u}_i}{\partial y^2} = \frac{\partial^2 \hat{u}(\mathbf{x}_i)}{\partial y^2},\ \frac{\partial^2 \hat{u}_i}{\partial x \partial y} = \frac{\partial^2 \hat{u}(\mathbf{x}_i)}{\partial x \partial y} \tag{41}$$

Fig. 4 The solution for 2D Poisson-Boltzmann equation with 21×21 uniform model TPS + linear additional polynomial with Hermite interpolation on Neumann boundaries 9 nearest points within every support domain





• When Φ is nonlinear in u

For the nonlinear case, Newton-Raphson iteration scheme has to be adopted in numerical tests. Initial value vector $\hat{\mathbf{u}}^0$ should be assumed for the beginning of solution.

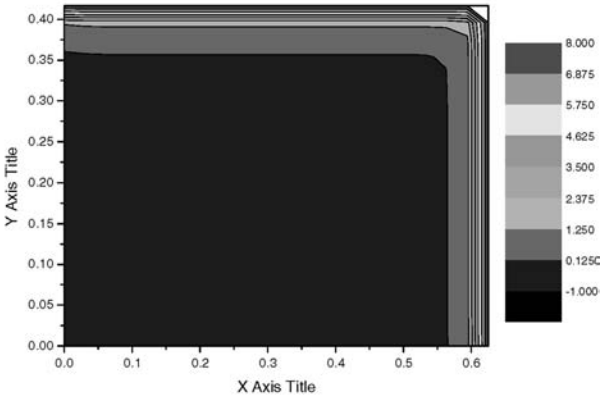
$$\Psi_i^k = \Psi(\hat{u}_i^k) = A\frac{\partial^2 \hat{u}_i^k}{\partial x^2} + B\frac{\partial^2 \hat{u}_i^k}{\partial x \partial y} + C\frac{\partial^2 \hat{u}_i^k}{\partial y^2} - \Phi\left(x_i, y_i, \hat{u}_i^k, \frac{\partial \hat{u}_i^k}{\partial x}, \frac{\partial \hat{u}_i^k}{\partial y}\right) = 0 \text{ in } \Omega \tag{42}$$

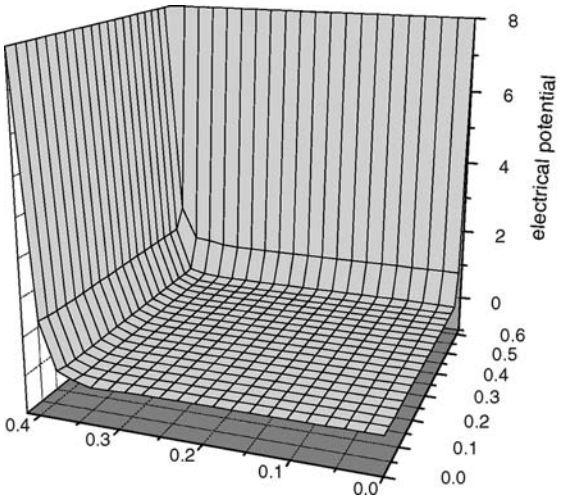
$$\mathbf{n}^{\mathrm{T}} \cdot \nabla \hat{u}_i^k + \bar{g}_n = 0,\ i = 1, \ldots, N_{b1} \tag{43}$$

$$\hat{u}_i^k - \bar{u} = 0,\ i = 1, \ldots, N_{b2} \tag{44}$$

$$\Psi_i^{k+1} = \Psi_i(\hat{\mathbf{u}}^{k+1}) = \Psi_i(\hat{\mathbf{u}}^k + \Delta\hat{\mathbf{u}}^k) = \Psi_i(\hat{\mathbf{u}}^k) + \frac{\partial \Psi_i^k}{\partial \hat{\mathbf{u}}^k} \bullet \Delta\hat{\mathbf{u}}^k + \mathrm{O}\left((\Delta\hat{\mathbf{u}}^k)^2\right) \tag{45}$$

$$\left[\frac{\partial \Psi_i^k}{\partial \hat{\mathbf{u}}^k}\right] \Delta\hat{\mathbf{u}}^k = -\left\{\Psi_i(\hat{\mathbf{u}}^k)\right\}, \Delta\hat{\mathbf{u}}^k = -\left[\frac{\partial \Psi_i^k}{\partial \hat{\mathbf{u}}^k}\right]^{-1} \left\{\Psi_i(\hat{\mathbf{u}}^k)\right\} \tag{46}$$





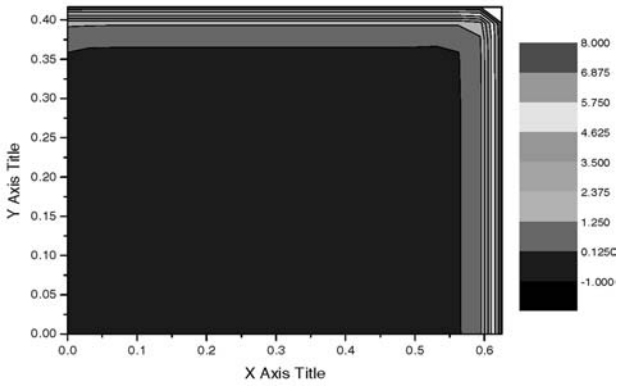


Fig. 5 The solution for 2D Poisson-Boltzmann equation with 21×21 uniform model TPS + linear additional polynomial with Hermite interpolation on Neumann boundaries 25 nearest points within every support domain

$$\hat{\mathbf{u}}^{k+1} = \hat{\mathbf{u}}^{k} + \Delta\hat{\mathbf{u}}^{k},\ \left\|\Delta\hat{\mathbf{u}}^{k}\right\| \le 10^{-4} \tag{47}$$

5 Numerical simulations

In this section, several examples on nonlinear Poisson equations are numerically analysed by TPS-RPICM. Several different results are obtained by using different additional polynomials and different exponentials. In addition, particular attention should be paid to example 3, where Hermite-type interpolation schemes on Neumann boundaries are adopted in order to improve numerical accuracy.

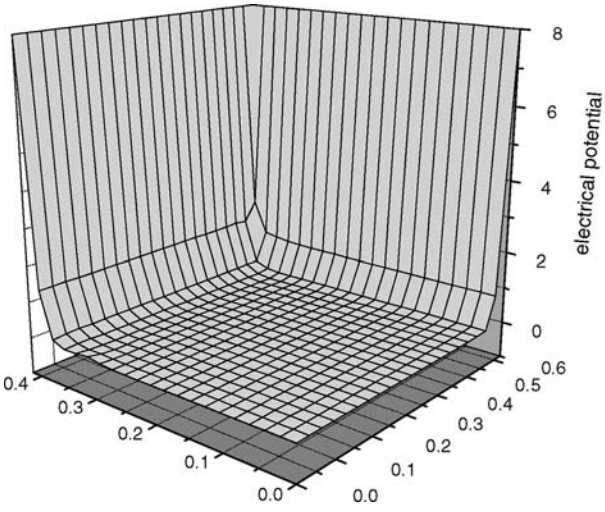


Fig. 6 The solution for 2D Poisson-Boltzmann equation with 21×21 uniform model TPS + linear additional polynomial with Hermite interpolation on Neumann boundaries 35 nearest points within every support domain

• Example 1: Liouville equation.

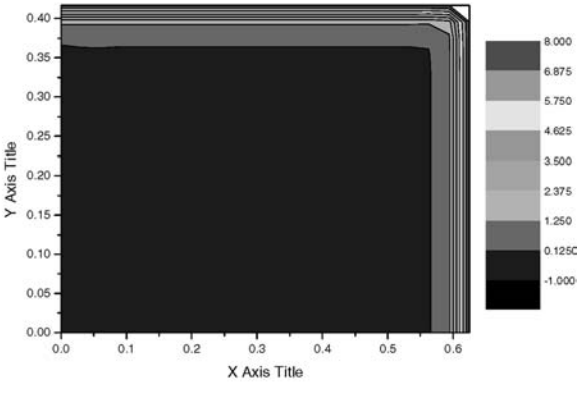
$$\nabla^2 u = \Phi = \frac{\lambda^2}{8} e^{-u}, \quad \lambda^2 = 40 \tag{48}$$

The solution domain is a unit circle in Figure 1 and Dirichlet condition of $u=1$ on the boundary is employed.

In this case, all these numerical results obtained with coarse and fine discrete models using TPS radial basis with linear and quadratic additional polynomials have been listed in Tables 1–4.

Table 1 is for TPS ($m = 6$) with quadratic polynomial with 85-node quasi-uniform coarse model; Table 2 is for TPS ($m = 4$) with linear polynomial with 221-node quasi-uniform fine model;

Table 3 is for TPS ($m = 6$) with quadratic polynomial with 221-node quasi-uniform fine model; Table 4 is for TPS ($m = 6$) with quadratic polynomial with 221-node Halton scattered fine model. From these results, it can be observed that the accuracy obtained using TPS with quadratic additional polynomial is higher than that obtained using TPS with linear additional polynomial. The results are better using more points in local support domain. For coarse model, less point in



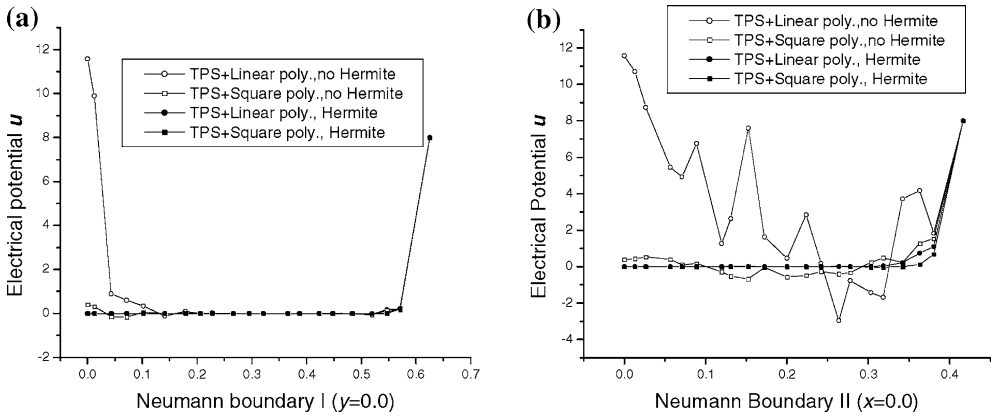


Fig. 7 The solution for 2D Poisson-Boltzmann equation with 406 scattered points model TPS + linear and square additional polynomial without and with Hermite-type interpolation on Neumann boundaries 25 nearest points within every support domain

local support domain lead the result to fail, so more points will be needed in its support domain. For fine model, however, good results can be obtained even with less point in support domain. In addition, the results obtained using Halton scattered points model is stable, and it shows that our method is suitable for scatter points model.

• Example 2: 2D Bratu problem

$$\nabla^2 u = \Phi = -\lambda e^u, \lambda = \frac{1}{e} \tag{49}$$

The solution domain and boundary condition are the same as in Example 1. The same coarse and fine discrete models as Example 1 are adopted to obtain the numerical results listed in Tables 5–8.

Table 5 is for TPS with quadratic polynomial ($m=6$) with 85-node quasi-uniform coarse model; Table 6 is for TPS with linear polynomial ($m=4$) with 221-node quasi-uniform fine model;

Table 7 is for TPS with quadratic polynomial with 221-node quasi-uniform fine model; Table 8 is for TPS with quadratic polynomial with 221-node Halton scattered fine model. From these results, similar conclusions as the previous example can be observed:

The accuracy obtained using TPS with quadratic additional polynomial is higher than that obtained using TPS with linear additional polynomial. The results are better using more points in local support domain. For coarse model, less point in local support domain leads the result to fail, so more points will be needed in its support domain. For fine model, however, good results can be obtained even with less point in support domain. In addition, the results obtained using Halton scattered points model is stable, and it shows that our method is suitable for scattered points model.

• Example 3: Poisson-Boltzmann equation [14].

$$\begin{aligned} &-\Delta u(x,y) + \kappa^2 \sinh(u(x,y)) = f, \\ &\Omega = \left[0,\frac{5}{8}\right] \times \left[0,\frac{5}{12}\right], \quad \kappa = 79.0,\ f = 0 \end{aligned} \tag{50}$$

Boundary conditions:

$$\begin{aligned} &u = 8.0, \quad \text{on } x = 5/8; u = 8.0, \quad \text{on } y = 5/12; \\ &\frac{\partial u}{\partial x} = 0, x = 0; \quad \frac{\partial u}{\partial y} = 0, y = 0 \end{aligned} \tag{51}$$

The solution domain for this problem is the rectangle: [0.0, 5/8]×[0.0, 5/12]. The electrical potential u can be approximated by radial basis point interpolation equation (4).

For uniformly distributed model, 21×21 nodes model has been employed in the calculations. The numerical results are obtained by TPS-RPICM with linear additional polynomials (m = 4). Three kind of the local support domain size are chosen by keeping 9, 25 and 35 nearest nodes in support domain respectively. When Hermite-type interpolation on Neumann boundaries is not adopted, the electrical potential obtained with 25 nearest points in every support domain is shown in Fig. 3. For 9 and 35 nearest points in every support domain, the computation fails before iterations are completed. It appears that the numerical stability and accuracy can be improved when quadratic additional polynomial is adopted. In this case, more nearest points within every support domain will lead to good accuracy especially for the numerical oscillation on Neumann boundaries. However, this conclusion does not apply for linear additional polynomial.

When Hermite-type interpolation on Neumann boundaries is adopted, the electrical potentials obtained with 9, 25 and 35 nearest points in every support domain are shown in Figs 4–6, respectively.

From the results, it is observed that the numerical oscillation on Neumann boundaries disappears when Hermite-type interpolation is adopted on Neumann boundaries. As a result, it can be concluded that the accuracy and stability have been improved with Hermite-type interpolation.

For scattered point model, 406 scattered points model in Fig. 2 has been employed in the calculations. The numerical results are obtained by TPS-RPICM with linear ($m = 4$) and square ($m = 6$) additional polynomials. The local support domain size is chosen in order to keep 25 nearest nodes in every SD. Figure 7a shows

the solution to the electrical potentials on Neumann boundary I for different additional polynomial without and with Hermite-type interpolations. Figure 7b shows the solution to the electrical potentials on Neumann boundary II for different additional polynomial without and with Hermite-type interpolations.

From the results, it is obvious that the numerical oscillation on boundary can be improved by the following two schemes:

(1) when quadratic polynomial is adopted instead of linear polynomial;
(2) when Hermite-type interpolation is adopted on Neumann boundaries.

The effectiveness is better using (2) than (1).

Conclusion

The results obtained by the present method demonstrate that accuracy is better using quadratic additional polynomial than using linear additional polynomial. The Hermite-type interpolation on Neumann boundary is very effective both for avoiding numerical oscillations on Neumann boundaries and for improving numerical stability during the calculations.

References

1. Liu GR (2002) Mesh free Methods, Moving beyond the Finite Element Method, CRC Press
2. Wang, J-G, Liu GR (2000). Radial point interpolation method for elastoplastic problems. Proc. of the 1st Int. Conf. On Structural Stability and Dynamics, Dec. 7–9, Taipei, Taiwan, pp. 703–708
3. Liu GR, Wang JG (2002) A point interpolation meshless method based on radial basis functions. Int J Numer Meth Eng 54:1623–1648
4. Wang JG, Liu GR, Lin P (2002) Numerical analysis of Biot's consolidation process by radial point interpolation method. Int J Solids Struct 39(6):1557–1573
5. Kansa EJ (1990) Multiquadrics: a scattered data approximation scheme with applications to computational fluid-dynamics. Comput Math Appl 19:147–161
6. Hon YC, Lu MW, et al. (1997) Multiquadric method for the numerical solution of a biphasic mixture model. Appl Math Comp 88:153–175
7. Golberg MA, Chen CS, Karur SR (1996) Improved multiquadrics approximation for partial differential equations. Eng Anal Boundary Elements, 18:9–17
8. Liu X, Liu GR, Tai K, Lam KY Radial Basis Point Interpolation Collocation Method For 2-d Solid Problem, Advances in Meshfree and X-FEM Methods. In: Liu GR (eds) proceedings of the 1st Asian Workshop on Meshfree methods. The 2nd International Conference on Structural Stability and Dynamics, Singapore, 16-18 December 2002. World Scientific, p 35–40
9. Liszka TJ, Duarte CA, Tworzydlo WW (1996) Hp-Meshless cloud method. Comput Meth Appl Mech Engrg 139:263–288
10. Zhang X, Song KZ, Lu MW, Liu X (2000) Meshless Methods based on collocation with radial basis function. Comput Mech 26 (4):333–343
11. Lee CK, Liu X, Fan SC (2003) Local multiquadric approximation for solving boundary value problems. Comput Mech 30:396–409
12. Balakrishnan K, Ramachandran P (2001) Osculatory interpolation in the method of fundamental solution for nonlinear Poisson problems. J Comput Phy 172:1–18
13. Ramachandran P, Balakrishnan K (2000) Radial basis functions as approximate particular solutions: review of recent progress. Eng Anal Boundary Elements 24:575–582
14. Arulanandam S, Li DQ (2000) Liquid transport in rectangular microchannels by electroosmotic pumping, Colloids and Surfaces A: Physicochemical and Engineering Aspects 161:89–102
15. Wu Z, Schaback R (1993) Local error estimates for radial basis function interpolation of scattered data. IMA J Numer Anal 13:13–27