Int. J. Appl. Comput. Math
DOI 10.1007/s40819-016-0272-6

SHORT COMMUNICATION

On the Convergence of Iterative Methods and Pseudoinverse Approaches in Global Meshless Collocation

Pankaj K. Mishra[1] · **Sankar K. Nath**[1]

Abstract In the context of numerical approximation of partial differential equations, meshless methods are recent developments, which have been reported to be relatively straightforward, yet provide better convergence and accuracy as compared to the conventional mesh-based approaches, for some specific problems like stress–strain analysis and modeling in a deforming media. Among several proposed schemes, strong-form collocation schemes using radial basis functions are easy-to-program, require no mesh in the domain or at the boundary, avoid numerical integration, and have similar formulations for all dimensions. Some of the radial basiss functions like the Gaussian, multiquadric and inverse-multiquadric are infinitely smooth. The standard global meshless formulations based on such radial basis functions are often found to lead towards ill-conditioned systems of linear equations. For such formulations, we investigate the degree of ill-conditioning against degrees of freedom for different radial basis functions. Also, the convergence of four iterative methods and pseudoinverse approaches has been tested for the solution of such ill-conditioned systems. In order to compute the pseudoinverse, two different approaches, i.e., singular value decomposition and full rank Cholesky factorization have been used. A set of numerical experiments, performed here, demonstrate that pseudoinverse approach based on singular value decomposition performs better than the iterative methods. Although, pseudoinverse computation via full rank Cholesky factorization is faster, it is not recommended in global meshless collocation schemes due to poor convergence at relatively large degrees of freedom.

Keywords Radial basis function · Ill-conditioning · Generalized inverse · Iterative methods · Meshless methods

Mathematics Subject Classification 65 · 68

✉ Sankar K. Nath
nath@gg.iitkgp.ernet.in

Pankaj K. Mishra
pkmishra@gg.iitkgp.ernet.in

1 Department of Geology and Geophysics, IIT Kharagpur, Kharagpur, India

Published online: 02 November 2016

Introduction

Radial basis functions (RBFs) which were initially introduced for surface fitting of scattered data [14], have become very popular for approximation of geometrically complex functions [3]. In 1990, RBFs were used as basis functions for numerical solution of partial differential equations (PDEs) using collocation method [15]. Successively, RBF collocation methods (RBFCM) were applied for modeling various phenomena in structural mechanics [9], quantum mechanics [5], fluid dynamics [6], transport phenomena [22,23], and radiative transfer mechanism [17]. Global collocation methods for large systems may involve solving ill-conditioned system of linear equations [1,2,11]. The ill-conditioning might occur either due to inappropriate shaping of the basis, e.g., flat RBF or due to very large and dense coefficient matrix for large degrees of freedom or inappropriate positioning of the nodes. Several approaches have been proposed to deal with such ill-conditioned systems namely replacement of global solvers with LU decomposition scheme, use of variable shape parameters [16], domain decomposition [18], moving least square iteration, and RBF-QR approach etc. Most of the techniques available to deal with ill-conditioned problems in RBFCM implementation are limited to small degrees of freedom, i.e., few hundreds in unit square [10]. A detailed analysis of various meshless methods based on global approximation of radial basis functions can be found in [7,8]. Here we perform numerical experiments to assess the efficacy of RBFCM using different basis functions like multiquadric (MQ), inverse multiquadric (IMQ) and Gaussian RBF (G-RBF) on a wide range of problem sizes, namely from 9 nodes up to 4096 nodes in unit square domain, focusing on some conventional approaches to deal with the ill-conditioned system. In order to deal with such ill-conditioning due to large and dense coefficient matrix in the system of linear equations, the generalized inverse using Moore–Penrose pseudoinverse technique has been adopted. In addition, four different iterative methods *viz.* conjugate gradient with stabilizer (*cgs*), biconjugate gradient method (*bicg*), biconjugate gradient with stabilizer (*bicgs*) and least square method (*lsqr*), have also been tested for comparison purposes. These iterative methods have been adopted from the MATLAB regularization toolbox $v4.1$—2007 [13].

It has been found that LSQR performs better than the other three approaches, therefore, a more detailed comparison has been drawn between LSQR and pseudoinverse (PINV) approaches in terms of the execution time, the condition number variation, and the RMS error convergence. As an attempt to make the algorithm faster, a fast approach for pseudoinverse computing via full rank Cholesky factorization has been tested with global meshless collocation algorithms. All the experiments have been performed using a desktop computer with Intel core i3 processor, 4 GB RAM, and Ubuntu 64-bit OS.

Test Case

Generalized Poisson's equation is an important mathematical representation used for modelling many physical phenomena like electrostatics, steady state thermal diffusion, potential distributions etc. In this investigation we consider the numerical solution of 2D generalized Poisson's equation with different boundary conditions and closed form solutions. The generalized Poisson equation is given by,

$$\nabla[\sigma(x, y)\nabla V(x, y)] = -f(x, y) \quad (x, y) \in \Omega. \tag{1}$$

and the Dirichlet boundary conditions are,

$$V(x, y) = 0 \quad (x, y) \in \Gamma, \tag{2}$$

where $\Omega \subset \mathbf{R}^2$ stands for computational domain and $\Gamma \subset \mathbf{R}^1$ for the boundary. The Coefficient $\sigma(x, y)$ and the source term $f(x, y)$ are chosen depending upon the considered problem type.

The Collocation Scheme

The collocation scheme proposed by Kansa [15], also termed as asymmetric or non-symmetric collocation method is adopted in the present synthesis. Let us assume an approximate solution of the problem defined in Eqs. (1)–(2) as $\tilde{V}$. In meshless collocation method the solution is written in terms of unknown coefficients and a basis functions. In general, there is no standard protocol for selecting RBF. However, several researchers followed the results from Frank's analyses [12] and used Hardy's Multiquadrics. In this paper we have used three different RBF, namely Gaussian, multiquadric and inverse multiquadric as given in Table 1. The $\varepsilon > 0$ is known as the shape parameter of the RBF. The approximate solution is therefore expressed as,

$$\tilde{V}(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k \phi(\mathbf{x}) = \alpha \Phi. \tag{3}$$

Equation (1) can also be expressed as,

$$\nabla \sigma(x, y) \nabla V(x, y) + \sigma(x, y) \nabla^2 V(x, y) = -f(x, y) \quad (x, y) \in \Omega. \tag{4}$$

Substituting the approximate solution given by Eq. (3) in the Eq. (4) we can write,

$$\nabla \sigma(x, y) \alpha \nabla \Phi + \sigma(x, y) \alpha \nabla^2 \Phi = -f(x, y) \quad (x, y) \in \Omega. \tag{5}$$

Similarly substituting the approximate solution given by Eq. (3) in Eq. (2) we can write,

$$\alpha \Phi = 0 \quad (x, y) \in \Gamma. \tag{6}$$

Eqs. (5) and (6) can be written in matrix form as,

$$\begin{bmatrix} \nabla \sigma(x, y) \nabla \Phi + \sigma(x, y) \nabla^2 \Phi \\ \Phi \end{bmatrix} \begin{bmatrix} \alpha \end{bmatrix} = \begin{bmatrix} -f(x, y) \\ 0 \end{bmatrix}, \tag{7}$$

Table 1 Mathematical expressions of some frequently used radial basis functions

Type of radial basis function	Expression
Gaussian	$\phi(r) = e^{-(\varepsilon r)^2}$
Inverse quadratic	$\phi(r) = \frac{1}{1+(\varepsilon r)^2}$
Multiquadric	$\phi(r) = \sqrt{1 + (\varepsilon r)^2}$
Inverse multiquadric	$\phi(r) = \frac{1}{\sqrt{1+(\varepsilon r)^2}}$
Polyharmonic spline	$\phi(r) = r^k, \ k = 1, 3, 5, \ldots$
Thin plate spline	$\phi(r) = r^2 \ln(r)$

or,

$$A\alpha = F, \tag{8}$$

where,

$$A = \begin{bmatrix} \nabla\sigma(x, y)\nabla\Phi + \sigma(x, y)\nabla^2\Phi \\ \Phi \end{bmatrix} \quad F = \begin{bmatrix} -f(x, y) \\ 0 \end{bmatrix} \tag{9}$$

Matrix A is termed as system matrix. The field variable $\tilde{V}$ can be computed by solving Eq. (8) for α and then substituting its value in Eq. (4).

Moore–Penrose Psuedoinverse

Matrix **G** is called the generalized inverse of a matrix **A** if it satisfies the following condition,

$$\mathbf{AGA} = \mathbf{A}. \tag{10}$$

Moore [19] and Penrose [20] independently proposed a unique psuedoinverse for non-invertible matrices by imposing three extra conditions in addition to condition given by Eq. (13),

1. General condition: $\mathbf{AGA} = \mathbf{A}$
2. Reflexive condition: $\mathbf{GAG} = \mathbf{G}$
3. Normalized condition $(\mathbf{AG})' = \mathbf{GA}$
4. reverse normalized condition $(\mathbf{GA})' = \mathbf{AG}$

If matrix **G** satisfies the first condition then **G** is the generalized inverse of matrix **A** and if **G** satisfies all the four conditions given above then it is unique and termed as Moore–Penrose psuedoinverse of **A**.

There are many approaches for computing the Moore–Penrose inverse of a matrix. Singular value decomposition (SVD) is one of the most frequently used method for this purpose. This method is the algorithm behind the in-built pseudoinvere computing function *"pinv"* in MATLAB. Although computing pseudoinverse via SVD is very precise, it is very time-intensive especially when the involved matrices are large. In 2005, Pierre Courrieu proposed an algorithm for fast computing of pseudoinverse using full rank Cholesky factorization [4]. In this paper, we test the performance of aforementioned approaches for computation of pseudoinverse in the context of solving system arising in global meshless collocation method. For the completeness of this paper, we list the MATLAB functions of these approaches in the following sections.

Pseudoinverse Using Singular Value Decomposition

The following function has been created using an intrinsic function of MATLAB named 'pinv'.

```
function [x] = pseudosvd(A, b)
% 'pinv' is the MATLAB's intrinsic function for
% computing pseudoinverse
invA = pinv(A);
x = invA*b;
end
```

Pseudoinverse Using Full Rank Cholesky Factorization

The following function has been created using the algorithm and the function given in [4].

```
function x = pseudogeninv(A,b)
invA = geninv(A);
x = invA*b;
%------------
function invG = geninv(G)
% Returns the Moore-Penrose inverse of the argument
% Transpose if m < n
[m,n]=size(G); transpose=false;
if m<n
transpose=true;
A=G*G';
n=m;
else
A=G'*G;
end
% Full rank Cholesky factorization of A
dA=diag(A); tol= min(dA(dA>0))*1e-9;
L=zeros(size(A));
r=0;
for k=1:n
r=r+1;
L(k:n,r)=A(k:n,k)-L(k:n,1:(r-1))*L(k,1:(r-1))';
% Note: for r=1, the substracted vector is zero
if L(k,r)>tol
L(k,r)=sqrt(L(k,r));
if k<n
L((k+1):n,r)=L((k+1):n,r)/L(k,r);
end
else
r=r-1;
end
end
L=L(:,1:r);
% Computation of the generalized inverse of G
M=inv(L'*L);
if transpose
Y=G'*L*M*M*L';
else
Y=L*M*M*L'*G';
end
end
```

Numerical Test 1

For this numerical test, we consider the following generalized Poisson equation with single variable coefficient and null Dirichlet boundary conditions. we select the following set up in accordance with [25].

$$\nabla[\sigma(x, y)\nabla V(x, y)] = -f(x, y) \quad (x, y) \in \Omega, \tag{11}$$

$$V(x, y) = 0 \quad (x, y) \in \Gamma, \tag{12}$$

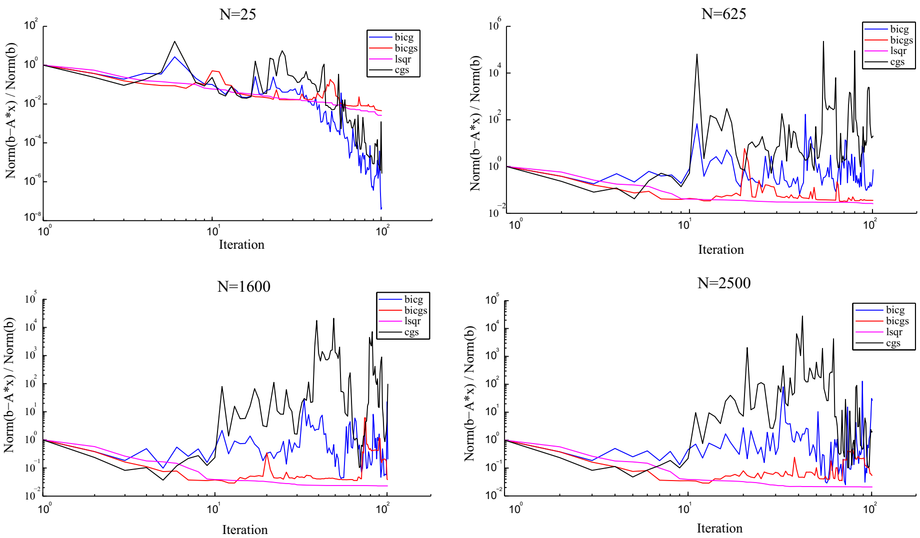

Fig. 1 convergence plots against iteration number for four iterative approaches; biconjugate gradient method (bicg), biconjugate gradient method with stabilizer (bicgs), least square method (lsqr), and conjugate gradient method (cgs) for the degrees of freedoms 25, 625, 1600, and 2500

where,

$$\sigma(x, y) = 1 + xy^2, \tag{13}$$

and

$$f(x) = y^4 - y^3 + 4y^3x - 4y^4x + 2y - 2y^2 - 2x^2y + 6x^2y^2 + 2x^3y - 6x^3y^2 + 2x - 2x^2. \tag{14}$$

The closed form solution of the problem, defined in Eqs. (11)–(14) is given by,

$$V(x, y) = xy(1 - x)(1 - y). \tag{15}$$

The meshless collocation scheme based on global approximation of radial basis functions, described earlier, has been used to solve the generalized Poisson equation with varying degrees of freedom in the spatial domain. In this test we consider a Poisson equation having single variable coefficient and a smooth closed form solution as given by Eqs. (11)–(15). First of all, four different iterative methods *viz.* conjugate gradient method with stabilizer (cgs), biconjugate gradient method (bicgs), biconjugate gradient method with stabilizer (bicgstab) and least square method (lsqr) are applied to solve the system of linear equations. Figure 1 exhibits the convergence plots against iteration number for these four iterative approaches with different degrees of freedoms, i.e., 25, 625, 1600, and 2500. It can be seen that for small problem sizes, i.e., with less number of nodes in the domain, all four iterative methods exhibit good convergence. As the degrees of freedom increases, the systems of linear equation become more and more ill-conditioned thereby affecting the convergence of these approaches significantly. At larger degrees of freedom, *cgs* and *bicg* diverge whereas *bicgs* and *lsqr* perform comparatively better. The least square method (*lsqr*) provides better convergence among all the considered iterative methods. Next, a comparative study of least square method with 'Moore–Penrose pseudoinverse via singular value decomposition (SVD)' is

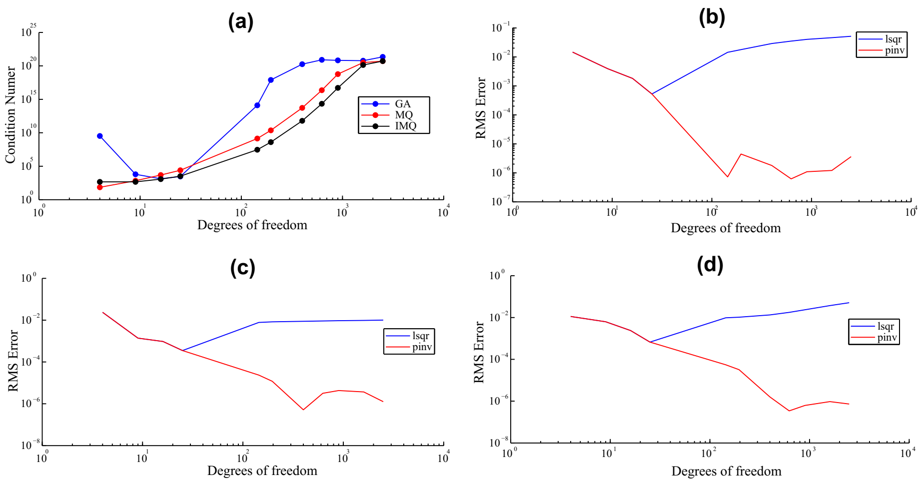

Fig. 2 **a** Variation of the condition number of the coefficient matrix with the degrees of freedom using global approximation of the three different radial basis functions viz. the Gaussian (GA), multiquadric (MQ), and inverse multiquadric (IMQ). The RMS error convergence of the meshless collocation algorithm with least square method (lsqr), Moore–Penrose pseudoinverse (pinv), and pseudoinverse using full rank Cholesky factorization (geninv) for **b** GA, **c** MQ, and **d** IMQ radial basis functions respectively

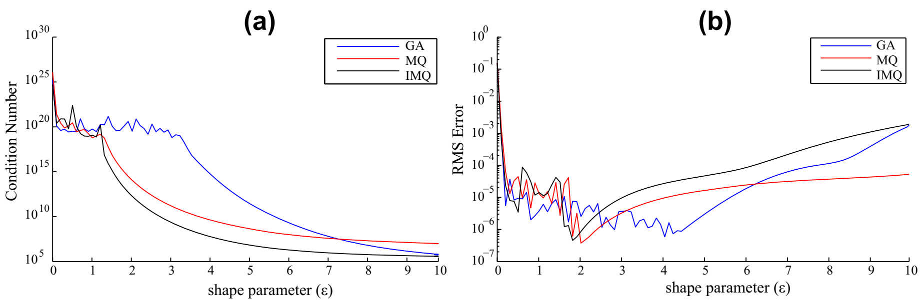

Fig. 3 **a** The variations in the condition numbers with shape parameters and **b** the variation in the RMS error with shape parameter, for the Gaussian (GA), multiquadric (MQ) and inverse multiquadric (IMQ) radial basis function

undertaken to bring out the efficacy of pseudoinverse in meshless methods. Figure 2a shows the variation of the condition number of the coefficient matrix with the degrees of freedom using global approximation of the three different radial basis functions *viz.* the Gaussian (GA), multiquadric (MQ), and inverse multiquadric (IMQ). Figure 2b–d shows the RMS error convergence of the meshless collocation algorithm with least square method (lsqr) and Moore–Penrose pseudoinverse via SVD (*pinv*) for GA, MQ and IMQ radial basis functions respectively. The variations in the condition numbers with shape parameters have been shown in Fig. 3a. The RMS error variations have been shown in Fig. 3b. The condition number is relatively very large at small values of shape parameters and significantly small at large shape parameters. The RMS error, on the other hand, is minimum for a specific value of the shape parameter which is termed as “optimal” shape parameter for a specific problem type. The elapsed CPU times at various degrees of freedom have been shown in Fig. 4. As the coef-

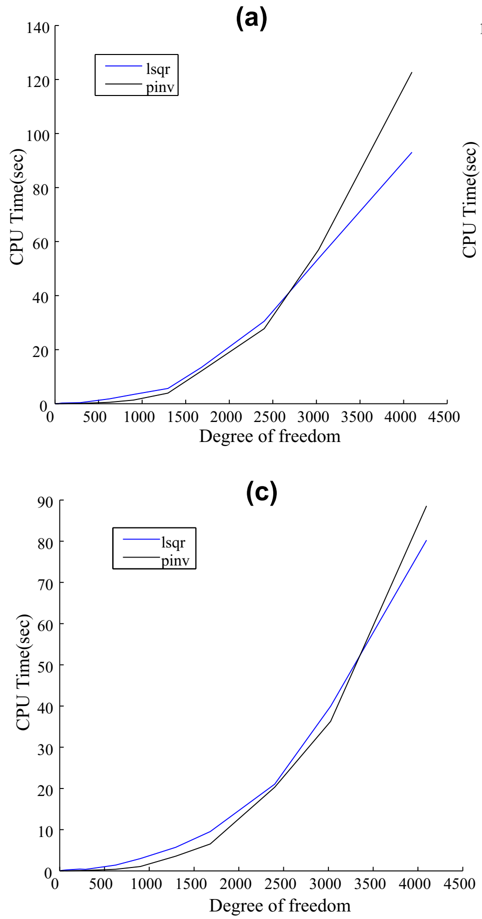

Fig. 4 Elapsed CPU times for different degrees of freedoms in the numerical test 1 using **a** multiquadric, **b** inverse multiquadric, and **c** the Gaussian radial basis function

ficient matrix becomes larger, the elapsed CPU time for '*pinv*' becomes more than that of '*lsqr*'. The data, generated in this numerical test, has been given in Tables 2, 3 and 4.

Numerical Test 2

For this numerical test, we consider the following generalized Poisson equation with two variable coefficient and null Dirichlet boundary conditions.

$$\frac{\partial}{\partial x}\left(\sigma_1(x, y)\frac{\partial}{\partial x}V(x, y)\right) + \frac{\partial}{\partial y}\left(\sigma_2(x, y)\frac{\partial}{\partial y}V(x, y)\right) = f(x, y), \tag{16}$$

where,

$$\sigma_1(x, y) = 2 - x^2 - y^2, \quad \sigma_1(x, y) = e^{x-y}, \tag{17}$$

Table 2 The output data from numerical test 1 using multiquadric radial basis function, where 'grid' is total number of nodes in the domain, 'Cond(A)' is condition number of the system matrix, 'E_{lsqr}' and 'E_{pinv}' are RMS error for the corresponding degrees of freedom, and 'T_{lsqr}' and 'T_{pinv}' are elapsed CPU times

Grid	Cond(A)	E_{LSQR}	E_{PINV}	T_{LSQR} (s)	T_{PINV} (s)
3 × 3	5.462389e+02	7.462048e−03	7.462048e−03	0.017190	0.014480
5 × 5	9.486098e+03	1.229512e−03	1.229512e−03	0.035038	0.021289
9 × 9	5.556616e+05	8.330284e−05	6.969761e−05	0.233432	0.024695
15 × 15	9.786292e+07	9.891521e−04	2.428474e−05	0.319333	0.081860
17 × 17	2.282450e+08	2.042705e−03	1.792220e−05	0.366024	0.097768
25 × 25	5.719549e+10	8.091326e−03	3.558741e−06	1.785787	0.498712
30 × 30	1.853098e+12	8.634835e−03	6.173954e−07	3.422583	1.309519
36 × 36	4.587563e+13	8.801420e−03	1.769110e−07	5.653921	3.921776
41 × 41	2.052002e+15	8.818715e−03	1.862631e−06	13.459742	12.105087
49 × 49	1.235882e+17	8.692244e−03	2.189240e−06	30.532397	27.755595
55 × 55	2.600771e+19	8.660601e−03	1.526058e−06	53.796464	57.001199
64 × 64	3.481882e+21	8.524119e−03	1.567361e−06	93.057866	122.725873

Table 3 The output data from numerical test 1 using inverse multiquadric radial basis function, where 'grid' is total number of nodes in the domain, 'Cond(A)' is condition number of the system matrix, 'E_{lsqr}' and 'E_{pinv}' are RMS error for the corresponding degrees of freedom, and 'T_{lsqr}' and 'T_{pinv}' are elapsed CPU times

Grid	Cond(A)	E_{LSQR}	E_{PINV}	T_{LSQR} (s)	T_{PINV} (s)
3 × 3	4.552021e+02	6.287141e−03	6.287141e−03	0.019338	0.095246
5 × 5	3.278429e+03	6.679342e−04	6.679342e−04	0.037225	0.059405
9 × 9	4.547333e+05	1.879437e−04	8.006780e−05	0.320914	0.025570
15 × 15	2.429614e+09	4.605594e−03	9.617913e−06	0.330483	0.070297
17 × 17	1.520203e+10	6.914031e−03	4.323024e−06	0.527138	0.107767
25 × 25	2.183591e+14	9.617107e−03	3.436355e−07	1.504586	0.803488
30 × 30	5.191176e+16	9.594382e−03	6.234800e−07	3.120213	1.588092
36 × 36	1.132854e+20	9.906616e−03	3.172328e−06	6.020479	3.717254
41 × 41	5.645700e+20	9.956196e−03	7.405996e−07	13.528421	11.007850
49 × 49	7.643773e+20	1.125763e−02	1.181245e−06	30.178800	19.428191
55 × 55	1.030282e+22	1.198814e−02	6.964089e−07	40.303699	47.157390
64 × 64	9.389660e+21	1.371020e−02	1.645707e−06	86.811549	95.896649

and,

$$f(x, y) = -16x(1-x)(3-2y)e^{x-y} + 32y(1-y)(3x^2 - y^2 - x - 2). \tag{18}$$

The closed form solution is given by,

$$V(x, y) = 16x(1-x)y(1-y). \tag{19}$$

In numerical test 1, it has been observed that computing pseudoinverse via SVD gets relatively costlier as the problem becomes large. In this test, we consider another method for computing

Table 4 The output data from numerical test 1 using the Gaussian radial basis function, where 'grid' is total number of nodes in the domain, 'Cond(A)' is condition number of the system matrix, 'E_{lsqr}' and 'E_{pinv}' are RMS error for the corresponding degrees of freedom, and 'T_{lsqr}' and 'T_{pinv}' are elapsed CPU times

Grid	Cond(A)	E_{LSQR}	E_{PINV}	T_{LSQR} (s)	T_{PINV} (s)
3 × 3	6.236811e+03	3.926078e−03	3.926078e−03	0.052828	0.016133
5 × 5	2.931602e+03	5.259066e−04	5.259066e−04	0.096394	0.047093
9 × 9	9.297106e+08	9.561455e−04	1.188569e−05	0.232991	0.021570
15 × 15	2.324896e+19	4.202737e−03	9.642844e−07	0.435192	0.075517
17 × 17	1.024460e+20	4.988621e−03	1.927309e−06	0.359292	0.108808
25 × 25	7.927389e+20	6.411139e−03	6.217996e−07	1.406606	0.390521
30 × 30	6.605889e+20	6.973757e−03	1.094752e−06	2.982883	1.063036
36 × 36	1.296275e+22	7.473213e−03	1.796719e−06	5.710581	3.554613
41 × 41	2.559521e+21	7.848528e−03	2.683546e−06	9.535559	6.517664
49 × 49	1.272206e+22	8.465665e−03	3.533894e−06	21.044453	20.291193
55 × 55	4.167489e+21	8.886912e−03	4.300104e−06	39.945304	36.280128
64 × 64	9.350845e+21	9.594764e−03	5.595720e−06	80.261718	88.565587

pseudoinverse using full rank Cholesky factorization which has been proven to be faster than computing pseudoinverse via SVD [4]. Also, since only '*lsqr*' shows relatively better convergence among all four iterative methods (as shown in numerical test 1), we only consider '*lsqr*', '*pinv*' and pseudoinverse using full rank Cholesky factorization '*geninv*'. Figure 5a, b shows the variation in the condition number and RMS error at different shape parameters, for three radial basis functions. Figure 5c shows the elapsed CPU time for various degrees of freedom in this test using inverse multiquadric radial basis function. It was observed 'geninv' is significantly faster than 'pinv'. However, the accuracy of 'geninv' is very poor at larger degrees of freedom which can be seen in Fig. 6.

Numerical Test 3

Analysis of radial basis collocation method for non-local boundary conditioned can be found in [21] and [26]. For this numerical test, we consider the following simplified Poisson equation with mixed boundary conditions.

$$\nabla V(x, y) = f(x, y), \quad (x, y) \in [0, 1] \times [0, 1]. \tag{20}$$

where,

$$f(x, y) = -5.4x^3.$$

The boundary conditions are,

$$\begin{aligned}
&\frac{\partial}{\partial n} V(x, y) = 0, \quad (x, y) \in \{0 \le x \le 1, y = 0\} \cup \{0 \le x \le 1, y = 1\}, \\
&V(x, y) = 0.1, \quad (x, y) \in \{x = 1, 0 \le y \le 1\}, \\
&V(x, y) = 1.0, \quad (x, y) \in \{x = 0, 0 \le y \le 1\}.
\end{aligned}$$

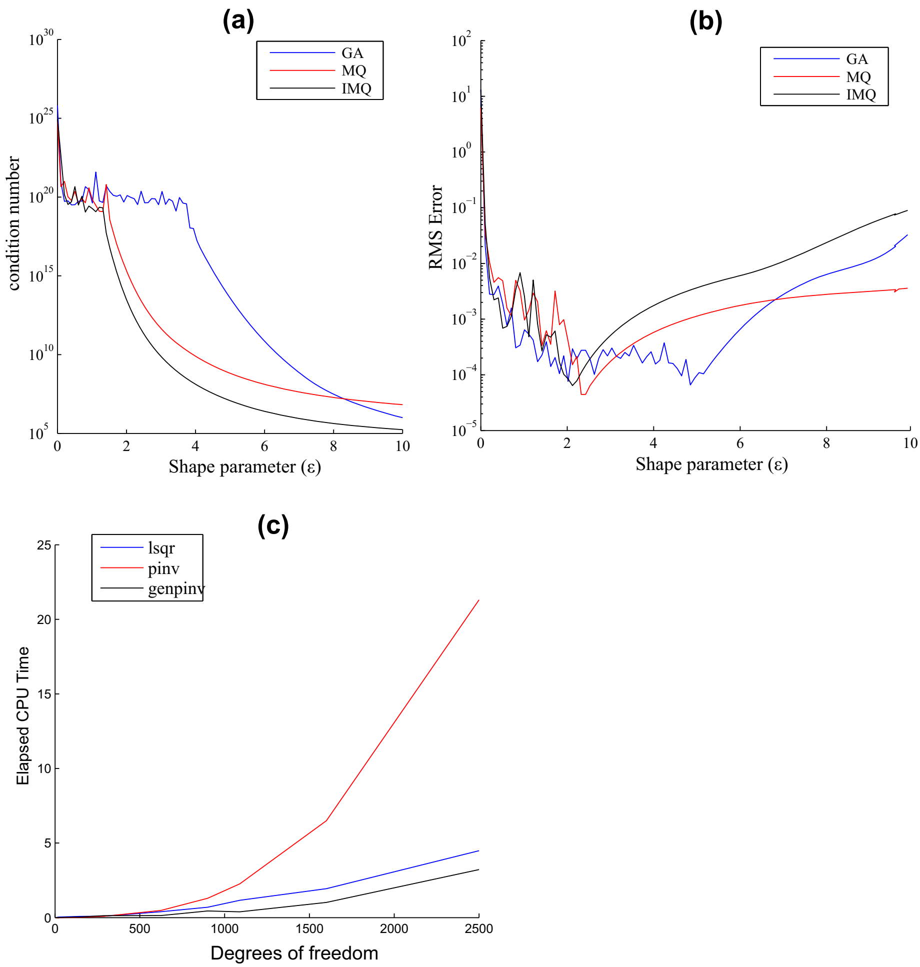

Fig. 5 **a** The variations in the condition numbers with shape parameters and **b** the variation in the RMS error with shape parameter, for the Gaussian (GA), multiquadric (MQ) and inverse multiquadric (IMQ) radial basis function. **c** Elapsed CPU time for various degrees of freedom using inverse multiquadric radial basis functions

The closed form solution of this problem is given by,

$$V(x, y) = 1 - 0.9x^3.$$

As we have seen in earlier tests, all of the three radial basis functions, considered here, show similar characteristics. Hence, we choose only one of them (inverse multiquadric), for this test for better visualization. Figure 7a shows the variation in the condition number of the system matrix with shape parameter. The observation here is, at large values of the shape parameters, the problem becomes better conditioned. However, if we see Fig. 7b, c, the errors do not decrease at large values of the shape parameter. This phenomenon is called the "uncertainity principle" in radial basis interpolation and its applications. According to this uncertainty principle, "It is impossible to construct radial basis functions which guarantee good stability and small errors at the same time" [24]. However, in 2016, Fasshauer has shown the exceptions

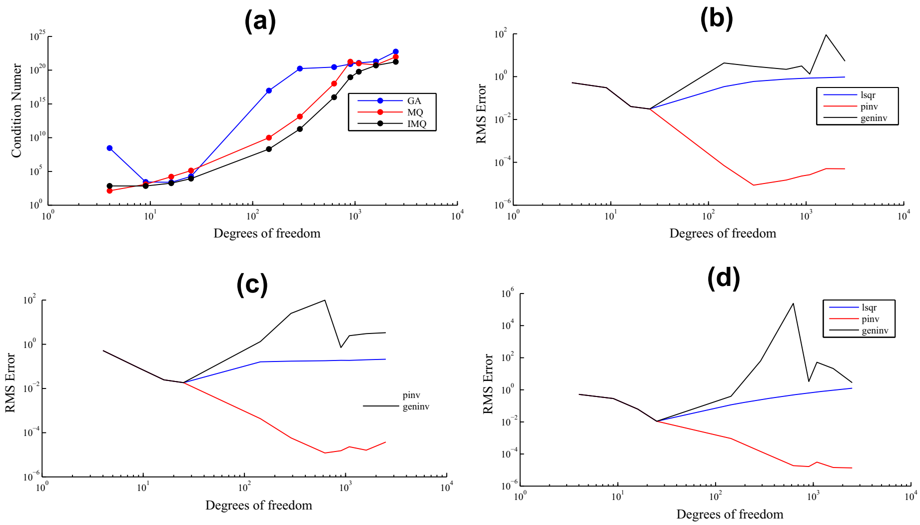

Fig. 6 **a** Variation of the condition number of the coefficient matrix with the degrees of freedom using global approximation of the three different radial basis functions viz. the Gaussian (GA), multiquadric (MQ), and inverse multiquadric (IMQ). The RMS error convergence of the meshless collocation algorithm with least square method (lsqr), Moore–Penrose pseudoinverse (pinv), and pseudoinverse using full rank Cholesky factorization (geninv) for **b** GA, **c** MQ, and **d** IMQ radial basis functions respectively

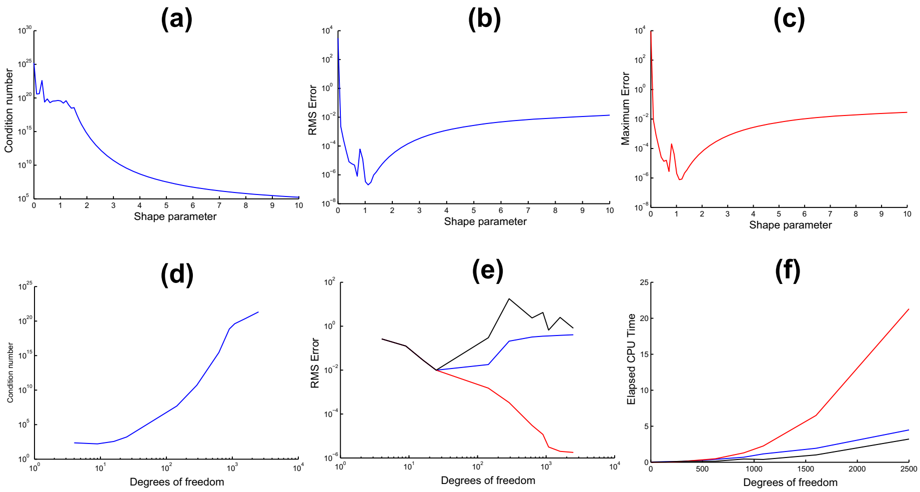

Fig. 7 **a** The condition number variation with shape parameter, **b** the RMS error variation with shape parameter, **c** maximum error variation with shape parameter, **d** condition number variation with degrees of freedom, **e** RMS error convergence, and **f** elapsed CPU time for various degrees of freedoms

of this uncertainty principle [7]. Figure 7d shows the condition number variation with degrees of freedom. Figure 7e, f shows the RMS error convergence and corresponding elapsed CPU times.

Conclusion

Ill-conditioning problem is an open and serious issue in numerical solution by meshless collocation method using global approximation of RBFs [3]. In the present work, ill-conditioning due to large degrees of freedom as well as the performance of iterative methods and two pseudoinverse approaches have been addressed. The condition number of the system matrix increases approximately as $O(N^2)$ for all the three radial basis functions, considered here. The Iterative methods, tested here, do not converge for such ill-conditioned system of equations hence not recommended in global meshless collocation algorithms. The Moore–Penrose pseudoinverse approach works far better than iterative methods, however, its significantly time intensive at large degrees of freedoms. The pseudoinverse approch using Cholesky factorization makes the algorithm significantly faster, however, it shows poor convergence at larger degrees of freedom. Since the generalized Poisson's equation governs many processes in science and engineering including potential field distribution, the global radial basis collocation algorithm using Moore–Penrose pseudoinverse can be used for meshless modeling in science and engineering.

References

1. Beatson, R., Cherrie, J., Mouat, C.: Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration. Adv. Comput. Math. **11**(2–3), 253–270 (1999)
2. Beatson, R.K., Light, W.A., Billings, S.: Fast solution of the radial basis function interpolation equations: domain decomposition methods. SIAM J. Sci. Comput. **22**(5), 1717–1740 (2000)
3. Chen, W., Fu, Z., Chen, C.: Recent Advances in Radial Basis Function Collocation Methods. Springer, Berlin (2014)
4. Courrieu, P.: Fast computation of Moore–Penrose inverse matrices. Neural Inf. Process. Lett. Rev. **8**(2), 25–29 (2005)
5. Dehghan, M., Shokri, A.: A numerical method for two-dimensional Schrödinger equation using collocation and radial basis functions. Comput. Math. Appl. **54**(1), 136–146 (2007)
6. Demirkaya, G., Wafo, S.C., Ilegbusi, O.: Direct solution of Navier–Stokes equations by radial basis functions. Appl. Math. Model. **32**(9), 1848–1858 (2008)
7. Fasshauer, G., McCourt, M.: The uncertainty principle—an unfortunate misconception. In: Kernel-Based Approximation Methods Using MATLAB, pp. 199–201 (2016)
8. Fasshauer, G.F.: Meshfree Approximation Methods with MATLAB. World Scientific Publishing Co., Inc., River Edge (2007)
9. Ferreira, A., Roque, C., Jorge, R.: Static and free vibration analysis of composite shells by radial basis functions. Eng. Anal. Bound. Elem. **30**(9), 719–733 (2006)
10. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial basis functions. SIAM J. Sci. Comput. **33**(2), 869–892 (2011)
11. Fornberg, B., Wright, G.: Stable computation of multiquadric interpolants for all values of the shape parameter. Comput. Math. Appl. **48**(5–6), 853–867 (2004)
12. Franke, R.: A Critical Comparison of Some Methods for Interpolation of Scattered Data. Defense Technical Information Center (1979)
13. Hansen, P.C.: Regularization tools: a matlab package for analysis and solution of discrete ill-posed problems. Numer. Algorithms **6**(1), 1–35 (1994)
14. Hardy, Rolland L.: Multiquadric equations of topography and other irregular surfaces. J. Geophys. Res. **76**(8), 2156–2202 (1971)
15. Kansa, E.J.: Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations. Comput. Math. Appl. **19**(8–9), 147–161 (1990)
16. Kansa, E.J., Hon, Y.C.: Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations. Comput. Math. Appl. **39**(7–8), 123–137 (2000)
17. Kindelan, M., Bernal, F., González-Rodríguez, P., Moscoso, M.: Application of the RBF meshless method to the solution of the radiative transport equation. J. Comput. Phys. **229**(5), 1897–1908 (2010)

18. Ling, L., Kansa, E.J.: Preconditioning for radial basis functions with domain decomposition methods. Math. Comput. Model. **40**(13), 1413–1427 (2004)
19. Moore, E.H.: On the reciprocal of the general algebraic matrix. Bull. Am. Math. Soc. **26**(2), 394–395 (1920)
20. Penrose, R.: A generalized inverse for matrices. Math. Proc. Camb. Philos. Soc. **51**(3), 406–413 (1955)
21. Sajavicius, S.: Optimization, conditioning and accuracy of radial basis function method for partial differential equations with nonlocal boundary conditions—a case of two-dimensional Poisson equation. Eng. Anal. Bound. Elem. **37**(4), 788–804 (2013)
22. Sarler, B.: Towards a mesh-free computation of transport phenomena. Eng. Anal. Bound. Elem. **26**(9), 731–738 (2002)
23. Sarler, B.: From global to local radial basis function collocation method for transport phenomena. In: Leitão, V., Alves, C., Armando Duarte, C. (eds.) Advances in Meshfree Techniques, Computational Methods in Applied Sciences, vol. 5, pp. 257–282. Springer, Dordrecht (2007)
24. Schaback, R.: Error estimates and condition numbers for radial basis function interpolation. Adv. Comput. Math. **3**(3), 251–264 (2013)
25. Strang, G.: Computational Science and Engineering. Wellesley-Cambridge Press, Wellesley (2007)
26. ul Islam, S., Aziz, I., Ahmad, M.: Numerical solution of two-dimensional elliptic PDEs with nonlocal boundary conditions. Comput. Math. Appl. **69**(3), 180–205 (2015)