A study on convergence and complexity of reproducing kernel collocation method

Hsin-Yun Hu† and Chiu-Kai Lai
Department of Mathematics, Tunghai University, Taichung 407, Taiwan R.O.C.

Jiun-Shyan Chen
Department of Civil and Environmental Engineering, University of California, Los Angeles, CA 90095, U.S.A.

(Received June 1, 2009, Accepted August 14, 2009)

Abstract. In this work, we discuss a reproducing kernel collocation method (RKCM) for solving 2nd order PDE based on strong formulation, where the reproducing kernel shape functions with compact support are used as approximation functions. The method based on strong form collocation avoids the domain integration, and leads to well-conditioned discrete system of equations. We investigate the convergence and the computational complexity for this proposed method. An important result obtained from the analysis is that the degree of basis in the reproducing kernel approximation has to be greater than one for the method to converge. Some numerical experiments are provided to validate the error analysis. The complexity of RKCM is also analyzed, and the complexity comparison with the weak formulation using reproducing kernel approximation is presented.

Keywords: reproducing kernel approximation; convergence; complexity; strong form collocation method.

1. Introduction

The finite element method (FEM) (Ciarlet 1978) dominates the development of numerical methods for several decades due to its deep mathematical theory and wide applications. The meshfree methods (Belytschko et al. 1996) is a new class of computational method and they have received much attention in the last decade. Meshfree methods inherit most of the advantages of FEM, in the mean while, overcoming a major disadvantage in the discretization where no mesh is needed; the discretization in meshfree methods merely rely on a set of particle. Many meshfree methods were developed in recent years, such as smooth particle hydrodynamics (Gingold and Monagahan 1977), diffuse element method (DEM) (Nayroles et al. 1992), element free Galerkin (EFG) method (Belytschko et al. 1994), partition of unity method (PUM) (Melenk and Babuska 1996), HP-clouds (Duarte and Oden 1996), reproducing kernel particle method (RKPM) (Liu et al. 1995, Chen et al. 1996, Chen et al. 2000, Chen et al. 2003), natural element method (NEM)

† Assistant Professor, Corresponding Author, E-mail: huhy@thu.edu.tw
(Sukumar et al. 1998), meshless local Petrov-Galerkin method (MLPG) (Atluri and Zhu 2000), radial basis collocation method (RBCM) (Kansa 1992a, 1992b), multiple level set method (MLSM) (Zhang et al. 2008), among others. These methods are more effective than FEM in formulating adaptive refinement schemes, and in solving problems with cracks, large deformation, multi-scale features (Rojek and Onate 2008), and shocks.

The development of meshfree methods can be traced back from two branches, one based on weak Galerkin form, for example, DEM, EFG, PUM, RKPM, NEM, MLPG, MLSM, and the other based on strong collocation form, such as RBCM. In the weak form approach (Chen et al. 2003, Bernardi and Maday 1997, Han and Meng 2001) for 2nd order PDE, the approximation is in the Sobolev $H^1$ space, and it yields a symmetric discrete system when test and trial functions are constructed with the same approximation space. Using approximation functions with compact support, the resulting discrete system is stable like finite element method and it exhibits an algebraic convergence if the approximation functions have polynomial reproducibility (Chen et al. 1996, Chen et al. 2003). However, the need of quadrature rules in the domain integration consumes a higher computational cost. Alternatively, strong form collocation method (Hu and Li 2006, Li et al. 2008) avoids the need of domain integration, and the approximation is constructed in the Sobolev $H^2$ space. A commonly used approximation function for the strong form collocation method is the infinite differentiable radial basis function (Hardy 1971, 1990), and this method is generally called the radial basis collocation method that exhibits exponential convergence rate (Kansa 1992b, Hu and Li 2006). However the method is overshadowed by its fully dense and ill-conditioned discrete system due to the non-local nature of radial basis functions. In this work, we discuss a reproducing kernel collocation method (RKCM), where the reproducing kernel shape functions with compact support are used as approximation functions. The method avoids the domain integration, and leads to well-conditioned discrete system of equations.

The objective of this work is to study the convergence and complexity of a strong form collocation method based on a reproducing kernel approximation, called the reproducing kernel collocation method RKCM (Hu and Chen 2009). Reproducing kernel approximation functions (Liu et al. 1995, Chen et al. 1996, Han and Meng 2001) are compactly supported and RKCM yields a well-conditioned discrete system similar to that in the FEM, but it does not require a domain integration. We first discuss the convergence of this method and obtain the necessary conditions for convergence. We then analyze the complexity of the approach and make comparison with the condition in RKPM.

The remaining part of this paper is organized as follows. An introduction to reproducing kernel (RK) shape function, their properties and approximation error and convergence are given in Section 2. In Section 3, the derivatives of RK shape function, the formulation of RKCM and its convergence rate are discussed, and numerical experiments are illustrated to validate the theoretical results. The computational complexity is addressed in Section 4. Conclusions are summarized in the last Section.

2. Reproducing kernel (RK) approximation

2.1 The RK shape function

We first describe the RK approximation in one-dimension. The multi-dimensional formation can be easily obtained with similar construction. Let a function $f(x)$ be approximated by
A study on convergence and complexity of reproducing kernel collocation method

\[ f^h(x) = \sum_{i=1}^{N_p} d_i \psi_i(x), \quad x \in \Omega \]  

(2.1)

where \( \Omega \subset \mathbb{R}, \psi_i(x) \) are shape functions centered at \( x_i \), and \( d_i \) are the coefficients to be sought. The construction of the shape functions is based on a set of particles

\[ S = \{x_i\}_{i=1}^{N_p} = \{x_1, x_2, \ldots, x_{N_p}\} \subseteq \Omega \cup \partial \Omega \]  

(2.2)

with maximal nodal distance \( h \), \( N_p \) is the number of particles, and the RK shape function is given as follows:

\[ \psi_i(x) = H^T(0)M^{-1}H(x-x_i) \phi_a(x-x_i) \]  

(2.3)

where

\[ M(x) = \sum_{i=1}^{N_p} H(x-x_i)H^T(x-x_i) \phi_a(x-x_i) \]  

(2.4)

\[ H^T(x-x_i) = [1, (x-x_i), (x-x_i)^2, \ldots, (x-x_i)^n] \]  

(2.5)

\[ H^T(0) = [1, 0, \ldots, 0] \]  

(2.6)

The vectors \( H(x-x_i) \) has dimension \( n+1 \), and \( M(x) \) is a moment matrix with dimension \( (n+1) \times (n+1) \). The function \( \phi_a(x-x_i) \) is called the kernel function, for example, the cubic B-spline

\[ \phi_a(z) = \begin{cases} 
\frac{2}{3} - 4z^2 + 4z^3, & 0 \leq z < \frac{1}{2} \\
\frac{4}{3} - 4z^2 + \frac{4z^3}{3}, & \frac{1}{2} \leq z < 1 \\
0, & z \geq 1 
\end{cases} \]  

(2.7)

where \( z = \frac{|x-x_i|}{a} \) and \( a \) is support size. For flexibility, we allow the support size to be dependent on \( l \). Another choice of kernel function is the quintic B-spline

\[ \phi_a(z) = \begin{cases} 
\frac{11}{20} - \frac{9z^2}{2} + \frac{81z^4}{4} - \frac{81z^5}{4}, & 0 \leq z < \frac{1}{3} \\
\frac{17}{40} + \frac{15z^2}{8} - \frac{63z^4}{4} + \frac{135z^5}{8} - \frac{243z^6}{8} + \frac{81z^7}{8}, & \frac{1}{3} \leq z < \frac{2}{3} \\
\frac{81}{40} - \frac{81z^2}{8} + \frac{81z^4}{4} - \frac{81z^5}{8} + \frac{81z^6}{40}, & \frac{2}{3} \leq z < 1 \\
0, & z \geq 1 
\end{cases} \]  

(2.8)

Here we show how the RK shape function \( \psi_i(x) \) in (2.3) is obtained. RK shape function can be
constructed as the multiplication of two functions

\[ \psi_i(x) = C(x; x - x_i) \phi_i(x - x_i), \quad \forall x_i \in S \]  

(2.9)

where \( C(x; x - x_i) \) is a correction function and \( \phi_i(x - x_i) \) is kernel function. For reproduction of complete \( n \)th order polynomial, the correction function is formed by a set of polynomial basis as

\[ C(x; x - x_i) = \sum_{i=0}^{n} (x - x_i)^i b_i(x) = H^T(x - x_i) b(x) \]  

(2.10)

The coefficients \( b_i(x) \) can be obtained by satisfying the following reproducing conditions

\[ \sum_{i=1}^{N_p} \psi_i(x) x_i^i = x^i, \quad i = 0, 1, \ldots, n \]  

(2.11)

that is,

\[ \sum_{i=1}^{N_p} C(x; x - x_i) \phi_i(x - x_i) x_i^i = x^i, \quad i = 0, 1, \ldots, n \]  

(2.12)

where \( n \) is reproducing degree. Eq. (2.12) is equivalent to

\[ \sum_{i=1}^{N_p} C(x; x - x_i) \phi_i(x - x_i)(x - x_i)^i = \delta_{i0}, \quad i = 0, 1, \ldots, n \]  

(2.13)

where \( \delta_{ij} \) is Kronecker delta. Eq. (2.13) can be rewritten in a vector form

\[ \sum_{i=1}^{N_p} C(x; x - x_i) \phi_i(x - x_i) H(x - x_i) = H(0) \]  

(2.14)

Substituting (2.10) into (2.14), we obtain

\[ \sum_{i=1}^{N_p} H(x - x_i) H^T(x - x_i) \phi_i(x - x_i) b(x) = H(0) \]  

(2.15)

Denote above equation as

\[ M(x) b(x) = H(0) \]  

(2.16)

where \( M(x) \) is defined as

\[ M(x) = \begin{bmatrix} \sum_{i=1}^{N_p} \phi_i(x - x_i) & \sum_{i=1}^{N_p} (x - x_i) \phi_i(x - x_i) & \cdots & \sum_{i=1}^{N_p} (x - x_i)^n \phi_i(x - x_i) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N_p} (x - x_i)^n \phi_i(x - x_i) & \sum_{i=1}^{N_p} (x - x_i)^{n+1} \phi_i(x - x_i) & \cdots & \sum_{i=1}^{N_p} (x - x_i) \phi_i(x - x_i) \end{bmatrix} \]  

(2.17)

with dimension \((n + 1) \times (n + 1)\). From (2.16), we have the coefficient vector

\[ b(x) = M^{-1}(x) H(0) \]  

(2.18)
The correction function in (2.10) becomes
\[ C(x; x - x_i) = H^T(x - x_i)M^{-1}(x)H(0) = H^T(0)M^{-1}(x)H(x - x_i) \] (2.19)

Thus, we have the function approximation
\[ f^h(x) = \sum_{i=1}^{N_p} d_i \psi_i(x) = \sum_{i=1}^{N_p} d_i C(x; x - x_i) \phi_i(x - x_i) \]
\[ = \sum_{i=1}^{N_p} d_i H^T(0)M^{-1}(x)H(x - x_i) \phi_i(x - x_i) \] (2.20)

where
\[ \psi_i(x) = H^T(0)M^{-1}(x)H(x - x_i) \phi_i(x - x_i) \] (2.21)

This shape function \( \psi_i(x) \) is called the reproducing kernel (RK) shape function, and its properties are summarized as follows.
(a) The construction of RK shape function \( \psi_i(x) \) does not rely on a mesh, it only relies a set of particles.
(b) The RK shape function has the same compact support as the kernel function, and the smoothness and locality of the shape function is determined by the kernel function.
(c) The degree of completeness and consistency of the shape function is determined by the correction function.
(d) For the moment matrix \( M(x) \) to be nonsingular, any position \( x \in \Omega \) needs to be covered by at least \( n + 1 \) kernel functions. Let \( \kappa \) be the maximal overlapping number in domain, usually \( \kappa \geq n + 1 \). A suggestion for the choice of support size is \( a = (n + 1)h \), where \( h \) is maximal nodal distance.
(e) The RK shape function does not satisfy the Kronecker delta property, i.e., \( \psi_i(x_j) \neq \delta_{ij} \). This means that the coefficients are not equal to nodal values, i.e., \( f^h(x_i) \neq d_i \).

2.2 Convergence properties of RK approximation

In this section we study the convergence of function approximation by RK shape functions.
Assume \( f(x) = \sin \pi x \), where \( x \in [0, 1] = : \Omega \). The sine function can be approximated by
\[ f^h(x) = \sum_{i=1}^{N_p} d_i \psi_i(x) , \; x \in \Omega \] (2.22)
in which the cubic B-spline kernel function \( \phi_i(x - x_i) \) is adopted in the RK shape function. The set of nodal points \( \{ x_i \}_{i=1}^{N_p} \) is called the source points. We define another set of points, called the collocation points
\[ \Xi = \{ \xi_j \}_{j=1}^{N_s} = \{ \xi_1, \xi_2, \ldots, \xi_{N_s} \} \subseteq \Omega \cup \partial \Omega \] (2.23)
where \( N_s \) is the total number of collocation points. The set of collocation points may or may not equal to the set of source points used in the approximation in (2.22). To approximate the function \( f(x) \), we force the residuals of the function approximation to be zero at these collocation points:
\[ f^h(\xi_i) - f(\xi_i) = 0 , \; \forall \xi_i , \; i = 1, 2, \ldots, N_s \] (2.24)
It follows that

\[ f^h(\xi_i) = \sin \pi \xi_i, \quad \forall \xi_i \]  

(2.25)

And this can be written a linear system as

\[ f^h(\xi_1) = \sum_{i=1}^{N_p} d_i \psi_i(\xi_1) = \sin \pi \xi_1 \]  

(2.26)

\[ f^h(\xi_2) = \sum_{i=1}^{N_p} d_i \psi_i(\xi_2) = \sin \pi \xi_2 \]  

(2.27)

\[ \vdots \]

\[ f^h(\xi_{N_s}) = \sum_{i=1}^{N_p} d_i \psi_i(\xi_{N_s}) = \sin \pi \xi_{N_s} \]  

(2.28)

Denote above system of equations as follows

\[
\begin{bmatrix}
\psi_1(\xi_1) & \psi_2(\xi_1) & \ldots & \psi_{N_p}(\xi_1) \\
\psi_1(\xi_2) & \psi_2(\xi_2) & \ldots & \psi_{N_p}(\xi_2) \\
\vdots & \vdots & \ddots & \vdots \\
\psi_1(\xi_{N_s}) & \psi_2(\xi_{N_s}) & \ldots & \psi_{N_p}(\xi_{N_s})
\end{bmatrix}
\begin{bmatrix}
d_1 \\
d_2 \\
\vdots \\
d_{N_p}
\end{bmatrix}
= 
\begin{bmatrix}
\sin \pi \xi_1 \\
\sin \pi \xi_2 \\
\vdots \\
\sin \pi \xi_{N_s}
\end{bmatrix}
\]  

(2.29)

When \( N_s = N_p = N \), the coefficients \( d_i \) can be obtain by Gaussian Elimination with Backward Substitution (GE with BS), and then we obtain the approximation \( f^h(x) \). Since the RK approximation belongs to a local approximation, the support size of the RK shape function is selected as \( a = (n+1)h \), this leads to a band matrix on the left hand side of linear system (2.29). A band Gaussian Elimination with Backward Substitution can be introduced to solve the linear system. The detailed algorithms are given in Appendix.

Here is a comparison of the operation counts for two Elimination methods mentioned above. For traditional Gaussian Elimination with Backward Substitution:

\[
M/D: \frac{N^3}{3} + \frac{N^2}{2} - \frac{5N}{6}
\]  

(2.30)

\[
A/S: \frac{N^3}{3} + \frac{N^2}{2} - \frac{5N}{6}
\]  

(2.31)

where the abbreviations M/D and A/S stand for multiplication/division and addition/subtraction, respectively. For band Gaussian Elimination with Backward Substitution:

\[
M/D: N(w^2 + w - 1) - \frac{2w^3}{3} + \frac{2w}{3}
\]  

(2.32)

\[
A/S: N(w^2 - 1) - \frac{2w^3}{3} + \frac{w^2}{2} + \frac{w}{6}
\]  

(2.33)
A study on convergence and complexity of reproducing kernel collocation method

The bandwidth of the band matrix is $2w - 1$. The operation counts under different dimensions $N = 10, 50, 100$ are listed in Table 1. Usually $w \ll N$, and the latter algorithm will be used for solving the linear system arising from RK approximation.

When $N_s > N_p$, the coefficients $d_i$ can be obtained by $QR$ factorization based on the Householder and Givens transformation or by singular value decomposition ($SVD$) with substitutions (Golub and Van Loan 1996).

In the followings we provide numerical tests for the RK approximation, wherein equally spaced source points and collocation points are chosen.

### Table 1: A comparison of the counts for two GE with BS

<table>
<thead>
<tr>
<th>$N$</th>
<th>M/D A/S</th>
<th>$w$</th>
<th>M/D A/S</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>430</td>
<td>3</td>
<td>94</td>
</tr>
<tr>
<td>50</td>
<td>44,150</td>
<td>5</td>
<td>1,370</td>
</tr>
<tr>
<td>100</td>
<td>343,300</td>
<td>7</td>
<td>5,276</td>
</tr>
</tbody>
</table>

![Fig. 1](image1.png) The approximated sine function and the error by using support size $a = 0.3$ and degree of basis functions $n = 2$

![Fig. 2](image2.png) Errors of the approximated function obtained by using support sizes $a = 0.4$ and $a = 0.5$, and degree of basis functions $n = 2$
The number of collocation points is approximately two times of the number of source points. The degree of basis function $n = 2$ is considered with different support sizes. The exact function, the approximated functions and errors are plotted in Figs. 1 and 2. If we consider much smaller support size for the quadratic basis function, we obtain a different result, see Fig. 3. We can see from the result in Fig. 3 that the support size of kernel function has to be greater than the maximal nodal distance to obtain a solution, otherwise, it leads to a singular moment matrix and yields a wrong solution. The numerical results show that the larger support sizes does not seem to improve much on accuracy, but it significantly increases the computational cost due to the larger bandwidth in the system of Eq. (2.29).

The main parameters for accuracy are the number of source points $Np$ and the degree of basis function $n$. We will discuss these issues in next section.

2.3 Convergence of RK approximation

We assume a quasi-uniform distribution of source points as defined below

$$c_0 \leq \frac{a_K}{a_J} \leq c_1, \forall K, J$$

(2.36)

where $a_K$ and $a_J$ denote the support sizes of kernel functions centered at source points $x_K$ and $x_J$, respectively, and $c_0$ and $c_1$ are generic constants. If function $f(x)$ is $n + 1$ order differentiable, then we have a local interpolation error (Chen et al. 2003, Han and Meng 2001) as follows

$$\left\| f(x) - f^h(x) \right\|_{0, \omega} \leq C a_1^n + \left| f(x) \right|_{n + 1, \omega_1}$$

(2.37)

where $\omega_1$ is the compact support of kernel function centered at $x_K$ with support size $a_K$. The norm $\left\| \cdot \right\|_{0, \omega}$ denotes the Sobolev zero norm, which is equal to the $L_2$-norm, and $n$ is the degree of basis function.
A study on convergence and complexity of reproducing kernel collocation method

Moreover, we have the following global interpolation error estimate.

**Lemma 1.** Let source points be in a quasi-uniform distribution. From the local interpolation error given in Eq. (2.37), we have the global interpolation error estimation

\[
\| f(x) - f^h(x) \|_{0, \Omega} \leq C \kappa \alpha^{n+1} |f(x)|_{n+1, \Omega}
\]

where \( \kappa \ll N_p \) is the maximal overlapping number in domain, and \( \alpha = \max_i a_i \), and \( C \) is a constant independent of parameters \( \kappa, \alpha \), and \( n \).

**Remark 1.** If we choose \( \alpha = (n+1)h \) as the support size of kernel function, then we have the estimation

\[
\| f(x) - f^h(x) \|_{0, \Omega} = O(\alpha^{n+1}) \approx O(h^{n+1})
\]

where \( n \geq 1 \). The convergence rate is proportional to power \( n+1 \) of nodal distance \( h \).

To validate the convergence rate, we use the reproducing degree \( n = 1, 2, 3 \), the number of source points \( N_p = 6, 8, \ldots, 20 \) and the number of collocation points \( N_s = 12, 16, \ldots, 40 \). Equally spaced collocation points and source point are adopted. We define \( h = 1/N_p \) and choose \( \alpha = (n+1)h \). The result of errors in \( L_2 \)-norm with various levels of refinement is plotted in Fig. 4. Here \( r \) denotes the numerical order of convergence rate. Compared with the error estimate in Eq. (2.39), there exists a super-convergence behavior when the degree of basis function \( n = 3 \).

**3. Reproducing kernel collocation method (RKCM)**

In this section, the RK approximation described in Section 2 is used for solving partial differential equations under a strong form collocation framework, and we call it the reproducing kernel collocation method (RKCM). The higher order derivatives of RK shape functions are first introduced, and the implementation scheme of RKCM and convergence properties are discussed.
3.1 Derivatives of RK shape function

The RK shape function is constructed by the multiplication of correction and kernel functions as
\[ \psi_l(x) = C(x;x-x_i)\phi_l(x-x_i) \]  
(3.1)

The first and second order derivatives can be obtained by product rule
\[ \psi_{l,i}(x) = C_{,i}(x;x-x_i)\phi_l(x-x_i) + C(x;x-x_i)\phi_{l,i}(x-x_i) \]  
(3.2)
\[ \psi_{l,ii}(x) = C_{,ii}(x;x-x_i)\phi_l(x-x_i) + 2C_{,i}(x;x-x_i)\phi_{l,ii}(x-x_i) \]
\[ + C(x;x-x_i)\phi_{l,ii}(x-x_i) \]  
(3.3)

where the correction function and its derivatives are given as
\[ C(x;x-x_i) = H^T(0)M^{-1}(x)H(x-x_i) \]  
(3.4)
\[ C_{,i}(x;x-x_i) = H^T(0)M^{-1}(x)H(x-x_i) + H^T(0)M^{-1}(x)H_{,i}(x-x_i) \]  
(3.5)
\[ C_{,ii}(x;x-x_i) = H^T(0)M^{-1}(x)H(x-x_i) + 2H^T(0)M^{-1}(x)H_{,i}(x-x_i) \]
\[ + H^T(0)M^{-1}(x)H_{,ii}(x-x_i) \]  
(3.6)

The moment matrix and its derivatives are given as
\[ M(x) = \sum_{i=1}^{NP} H(x-x_i)H^T(x-x_i)\phi_l(x-x_i) \]  
(3.7)
\[ M_{,i}(x) = \sum_{i=1}^{NP} \{ H_{,i}(x-x_i)H^T(x-x_i)\phi_l(x-x_i) + H(x-x_i)H_{,i}(x-x_i)\phi_l(x-x_i) \]
\[ + H(x-x_i)H^T(x-x_i)\phi_{l,i}(x-x_i) \} \]  
(3.8)
\[ M_{,ii}(x) = \sum_{i=1}^{NP} \{ H_{,ii}(x-x_i)H^T(x-x_i)\phi_l(x-x_i) + 2H_{,i}(x-x_i)H_{,i}(x-x_i)\phi_l(x-x_i) \]
\[ + 2H_{,i}(x-x_i)H^T(x-x_i)\phi_{l,ii}(x-x_i) + H(x-x_i)H_{,ii}(x-x_i)\phi_l(x-x_i) \]
\[ + 2H(x-x_i)H^T(x-x_i)\phi_{l,ii}(x-x_i) + H(x-x_i)H^T(x-x_i)\phi_{l,ii}(x-x_i) \} \]  
(3.9)

In the followings, we use the implicit differentiation to obtain the derivatives of \( M^{-1}(x) \), i.e., \( M_{,i}^{-1}(x) \) and \( M_{,ii}^{-1}(x) \). Since \( M(x)M^{-1}(x) = I \), where \( I \) denotes an identity matrix. Taking differentiation on both sides, we obtain
\[ \frac{d}{dx}(M(x)M^{-1}(x)) = \frac{d}{dx}I \]  
(3.10)

and then by product rule
\[ M_{,i}(x)M^{-1}(x) + M(x)M_{,ii}^{-1}(x) = 0 \]  
(3.11)
A study on convergence and complexity of reproducing kernel collocation method

where \( \mathbf{0} \) is a zero matrix, it follows that

\[
M_{x}^{-1} = -M^{-1}(x)M_{x}(x)M^{-1}(x)
\]  
(3.12)

In a similar manner, we may obtain the second derivative

\[
\frac{d}{dx}(M_{x}^{-1}) = \frac{d}{dx}(-M^{-1}(x)M_{x}(x)M^{-1}(x))
\]
(3.13)

and then

\[
M_{xx}^{-1} = -M^{-1}(x)(M_{xx}(x)M^{-1}(x) + 2M_{x}(x)M_{x}^{-1}(x))
\]
(3.14)

In section 4, we will discuss the complexity of the shape function and its derivatives based on the equations given in Eq. (3.1) to Eq. (3.14). For the construction of the shape function and its derivatives, we first form the matrices, then the correction function and derivatives, and finally the shape function and derivatives can be constructed.

Denote \( \psi_{l}^{(1)}(x) = \psi_{l,x}(x) \) and \( \psi_{l}^{(2)}(x) = \psi_{l,xx}(x) \), we have the bounds (Chen et al. 2003, Han and Meng 2001)

\[
\left| \psi_{l}(x) \right|_{\infty} \leq C_{1}
\]
(3.15)

\[
\left| \psi_{l}^{(l)}(x) \right|_{\infty} \leq C_{2}a^{l}, \quad l = 1, 2, \ldots,
\]
(3.16)

where \( C_{1} \) and \( C_{2} \) are generic constants. In the followings we will provide the bounds for the high order derivatives of a function \( v = v(x) \), which is approximated by the linear combination of RK shape function

\[
v(x) = \sum_{l=1}^{N_p} b_{l} \psi_{l}(x)
\]
(3.17)

Denote finite dimensional space

\[
V = \text{span}\{ \psi_{1}(x), \psi_{2}(x), \ldots, \psi_{N_p}(x) \} \subset H^{2}(\Omega)
\]
(3.18)

We need the following inverse inequalities to prove the convergence.

**Lemma 2.** Let the source points be in a quasi-uniform distribution. For \( \forall v \in V \), there exist the inequalities

\[
\| v \|_{L^{2}(\Omega)} \leq C_{1}\kappa^{\frac{1}{2}}a^{-\frac{3}{2}}n^{-\frac{1}{2}}\| v \|_{L^{2}(\Omega)}, \quad l = 1, 2, 3, \ldots,
\]
(3.19)

\[
\| v \|_{H^{l}(\Omega)} \leq C_{2}\kappa^{\frac{1}{2}}a^{-\frac{3}{2}}n^{-\frac{1}{2}}\| v \|_{H^{l}(\Omega)}, \quad l = 1, 2, 3, \ldots,
\]
(3.20)

\[
\| v \|_{L^{2}(\Gamma)} \leq C_{3}\kappa^{\frac{1}{2}}a^{d(l+1)}n^{2(l+1)}\| v \|_{L^{2}(\Gamma)}, \quad l = 1, 2, 3, \ldots,
\]
(3.21)

where \( \kappa \) is the maximal overlapping number in domain, \( a \) is the maximal support size, \( \Gamma = \partial \Omega \) denotes the boundary of \( \Omega \), and \( C_{l} \) are generic constants.
3.2 Implementation scheme for RKCM

Consider an elliptic problem

\[ L_d u = f, \text{ in } \Omega \]  (3.22)

\[ B_d u = g, \text{ on } \Gamma \]  (3.23)

where \( L_d \) and \( B_d \) denote the operators in the domain and on the boundary, respectively.

The collocation method is the minimization of discrete least-squares functional (Hu and Li 2006, Li et al. 2008)

\[ \hat{E}(v) = \min_{v \in \mathcal{V}} \hat{E}(v) \]  (3.24)

where \( \hat{E}(v) \) is a quadrature version of least-squares functional defined as

\[ \hat{E}(v) = \frac{1}{2} \int_{\Omega} (L_d v - f)^2 d\Omega + \frac{1}{2} \int_{\Gamma} (B_d v - g)^2 d\Gamma \]  (3.25)

here \( \hat{\int} \) denotes the quadrature version of \( \int \). A set of collocation points \( \Xi = \{ \xi_1, \xi_2, ..., \xi_{\text{Ns}} \} \) with maximal spacing \( h = \max_{\xi_i \in \Omega} (\xi_i) \) is used. The parameter \( h \) has to satisfy the following relationship (Hu and Li 2006)

\[ h \approx o (h^2) \]  (3.26)

to ensure the error from quadrature rules to be bounded as follows

\[ \left| \left( \hat{\int} - \int \right) G(x) \right| = o(1) \rightarrow 0 \]  (3.27)

Here the integrand \( G(x) \) can be \( L_d v(x) \) or \( B_d v(x) \). Consequently we may obtain an optimal solution \( u_h(x) \) as discussed below.

The minimization of discrete functional (3.24) leads to solving the following system of equations:

\[ L_d \left\{ \sum_{j=1}^{Np} b_j \psi_j(\xi_i) \right\} = f(\xi_i), \; \xi_i \in \Omega \]  (3.28)

\[ B_d \left\{ \sum_{j=1}^{Np} b_j \psi_j(\xi_j) \right\} = g(\xi_j), \; \xi_j \in \Gamma \]  (3.29)

They can be rewritten as follows

\[ Fx = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{Np} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = r \]  (3.30)
The entries of matrix $F$ and the components of $r$ are given by

$$
[F_1]_{ij} = L_d \psi_i(\xi_j), \quad [r_1]_i = f(\xi_i), \quad i = 1, 2, \ldots, N_t
$$

(3.31)

$$
[F_2]_{ij} = B_d \psi_i(\xi_j), \quad [r_2]_j = g(\xi_j), \quad j = N_t + 1, \ldots, N_s
$$

(3.32)

where $i = 1, 2, \ldots, N_p, N_t$ is the number of collocation points in the domain, and $N_s$ is the total number of collocation points in the domain and on the boundaries. The matrix $F$ has dimension $N_s \times N_p$, $r$ is a known vector with dimension $N_s$. We have an over-determined system when $Ns > Np$, and the Cholesky decomposition with substitution (Burden and Faires 2005) can be adopted on its normal equation to find coefficients $b_k$ and to obtain an optimal solution $u_k(x) \in V$.

### 3.3 Convergence of RBCM

In this section let us discuss the convergence properties of RKCM. Consider the following linear operators in one-dimension

$$
L_d = \frac{d^2}{dx^2}
$$

(3.33)

$$
B_d = \alpha \frac{d}{dx} + \beta
$$

(3.34)

where $\alpha \geq 0, \beta > 0$. Define a norm

$$
\|v\|_H^2 = \|v\|_{L_2, \Omega}^2 + \alpha v_x^2 + \beta v_{xx}^2
$$

(3.35)

The minimization of functional Eq. (3.24) can be described equivalently in the following variational problem

$$
\hat{B}(u, v) = \hat{F}(v)
$$

(3.36)

where

$$
\hat{B}(u, v) = \int_\Omega u_x v_x d\Omega + \int_\Gamma (\alpha u_x + \beta)(\alpha v_x + \beta) dl
$$

(3.37)

$$
\hat{F}(v) = -\int_\Omega f v_x d\Omega + \int_\Gamma g(\alpha v_x + \beta) dl
$$

(3.38)

**Lemma 3.** Suppose there exist two inequalities (Hu and Chen 2009),

$$
\hat{B}(u, v) \leq C_1 \|v\|_{H^1}^2, \quad \forall v \in V
$$

(3.39)

$$
\hat{B}(v, v) \geq C_2 \|v\|_{H^1}^2, \quad \forall v \in V
$$

(3.40)

Then, there exists an optimal estimate

$$
\|u - u_k\|_H \leq C_1 \min_{v \in V} \|u - v\|_H
$$

(3.41)

where $C_i$ are constants independent of $\alpha, h, n$ and $N_p$. 

Lemma 4. Assume that the solution $u(x)$ is sufficiently smooth, the global interpolation error estimate for solution of RKCM is given as follows
\[
\|u - u_h\|_{L^2(\Omega)} \leq C \kappa a^{n+1} \|u\|_{H^1(\Omega)} \tag{3.42}
\]
where $I \geq 1$, and $C$ is a constant independent of $\kappa, a$ and $n$.

Theorem 1. Let the conditions in Lemmas 2, 3 and 4 hold. There exists the error bound for the solution of RKCM as follows (Hu and Chen 2009)
\[
\|u - u_h\|_{H^2} \leq C \{ \|u - v\|_{H^2(\Omega)} + \kappa \|u - v\|_{H^1(\Gamma)} + \beta \|u - v\|_{L^2(\Gamma)} \}
\leq C \kappa a^{n+1} \|u\|_{H^1(\Omega)} \tag{3.43}
\]
where $\kappa$ is the overlapping number, $a$ is the maximal support size and $n$ is the degree of basis function, and $C$ is a constant.

According to Theorem 1, there exists the following convergence behavior
\[
\|u - u_h\|_{H^2} = O(1), \text{ when } n = 1 \tag{3.44}
\]
This shows that the error does not converge when the linear basis function $n = 1$ is adopted. Specifically, the degree of basis function has to be greater than one, $n > 1$, for convergence. Moreover, we have
\[
\|u - u_h\|_{H^2} = O(a^1) \approx O(h^1), \text{ when } n = 2 \tag{3.45}
\]
\[
\|u - u_h\|_{H^2} = O(a^2) \approx O(h^2), \text{ when } n = 3 \tag{3.46}
\]
The error of solution under $H$-norm is equivalent to in Sobolev two norm, it follows that the convergence behavior in Sobolev zero norm will be
\[
\|u - u_h\|_{0, \Omega} = O(a^{1+\frac{1}{2}}) \approx O(h^\frac{1}{2}), \text{ when } n = 2 \tag{3.47}
\]
\[
\|u - u_h\|_{0, \Omega} = O(a^{2+\frac{1}{2}}) \approx O(h^\frac{3}{2}), \text{ when } n = 3 \tag{3.48}
\]
Two numerical tests will be given in the next section to validate these theoretical results.

3.4 Numerical experiments

Consider a Poisson problem in domain $\Omega = \{x|0 < x < 1\}$ with pure Dirichlet boundary conditions
\[
\begin{align*}
  u''(x) &= f(x), \text{ in } \Omega \\
  u(0) &= 0, \text{ on } \Gamma_0 \\
  u(1) &= 0, \text{ on } \Gamma_1 \tag{3.49}
\end{align*}
\]
where $u''(x) = \frac{d^2 u(x)}{dx^2}$, $f(x) = -\pi^2 \sin \pi x$, $\Gamma_0$ denotes endpoint $x = 0$, and $\Gamma_1$ denotes endpoint $x = 1$. The analytical solution is $u(x) = \sin \pi x$.

We use the reproducing degree $n = 1, 2, 3$, the number of source points $N_p = 6, 8, ..., 20$, and
A study on convergence and complexity of reproducing kernel collocation method

...the number of collocations is four times of the number of source points, i.e., $N_s = 24, 32, \ldots, 80$. Equally spaced collocation points and source points are used, and the kernel function $\phi_a(x-x_i)$ is chosen as the quintic B-spline defined in Eq. (2.8). We define $h = 1/Np$ and choose the support size $a = (n+1)h$. The errors of solution in $L^2$ norm with various levels of refinement are plotted in Fig. 5.

In the second test, we consider a problem with different boundary conditions as follows

\begin{align*}
    u''(x) &= f(x), \text{ in } \Omega \\
    u(0) &= 0, \text{ on } \Gamma_0 \\
    u'(1) &= -\pi, \text{ on } \Gamma_1
\end{align*}

(3.50)

where $u'(x) = du(x)/dx$. The exact solution is the same as previous example. The same reproducing degree, collocation points and source points as those used in the previous example are employed in this experiment. The test results are illustrated in Fig. 6.

We can see from Figs. 5 and 6 that the error, indeed, does not converge when linear basis function $n=1$ is used. This is consistent with our theoretical results. The rates of convergence for $n=2, 3$ are reasonably consistent with the theoretical results in Eqs. (3.47) and (3.48). This convergence behavior is different from the that in the reproducing kernel particle method (RKPM) (Liu et al. 1995, Chen et al. 1996), in which the linear basis function is sufficient for convergence. Compared to weak formulation such as RKPM, RKCM based on strong form approach is rather simple in forming the discrete equation and it avoids domain integration. However, the second order derivatives are needed in RKCM. In the next section, we analyze the computational complexity of RKCM.
4. Computational complexity of RKCM

In computations, the CPU time is proportional to the total operations in addition (A), subtraction (S), multiplication (M) and division (D) computations. In this section we discuss the operations counts for forming a system of Eq. (3.30) in RKCM and solution of the linear system.

The linear system Eq. (3.30) consists of an $N_s \times N_p$ matrix $F$ and a vector $r$ with dimension $N_s$. Note that the main entries in matrix $F$ are $\psi_{i,s}(\xi)$ in the domain, and $\psi_0(\xi)$ or $\psi_{e,s}(\xi)$ on the boundaries. First, we calculate the operation counts for forming RK shape function $\psi_i(\xi)$ and its derivatives $\psi_{i,e}(\xi)$ and $\psi_{i,s}(\xi)$. The derivation was given in Section 3.1, the formation of RK shape function involves considerable vector-vector and vector-matrix multiplications. We will start with the operation count for the eлементary operations and then the count for the construction of RK shape functions.

4.1 Elementary operations

Let $\mathbb{R}$ denote the set of real values, $\mathbb{R}^s$ denote the set of real vectors, and $\mathbb{R}^{s \times s}$ denote the set of $s \times s$ matrices.

(a) Dot product:
If $x, y \in \mathbb{R}^s$, where $x = (x_1, x_2, \ldots, x_s)^T$, $y = (y_1, y_2, \ldots, y_s)^T$ operation counts of the dot product, $x \cdot y = x^T y = \sum_{i=1}^{s} x_i y_i$, are given below.

![Fig. 6 Convergence behavior for Poisson's problem with Dirichlet and Neumann boundary conditions](image)
A study on convergence and complexity of reproducing kernel collocation method

(b) Scale-Matrix multiplication:
If \( A \in \mathbb{R}^{s \times s} \), \( \gamma \in \mathbb{R} \) where \( A = [a_{ij}] \), then operation count for the scale-matrix multiplication, \( \gamma \cdot A = \gamma \cdot [a_{ij}], i,j = 1,2,\ldots,s \), is given as follows:

\[
\begin{array}{c|c|c}
M/D & A/S \\
\hline
\text{Forming } \gamma \cdot A & s^2 & 0 \\
\end{array}
\]

(c) Matrix-Vector multiplication:
If \( \mathbf{A} \in \mathbb{R}^{s \times s} \) and \( \mathbf{x} = (x_1, x_2, \ldots, x_s)^T \), then each component of \( \mathbf{Ax} \), i.e., \( (\mathbf{Ax})_i = \sum_{j=1}^{s} a_{ij} x_j \), can be viewed as a dot product. There are \( s \) dot products in matrix-vector multiplication, and the operation count is:

\[
\begin{array}{c|c|c}
M/D & A/S \\
\hline
\text{Forming } \mathbf{Ax} & s^2 & s^2 - s \\
\end{array}
\]

(d) Matrix-Matrix multiplication:
If \( \mathbf{A}, \mathbf{B} \in \mathbb{R}^{s \times s} \), where \( \mathbf{A} = [a_{ij}] \) and \( \mathbf{B} = [b_{ij}] \), then each entry in \( \mathbf{AB} \), i.e., \( (\mathbf{AB})_{ik} = \sum_{j=1}^{s} a_{ij} b_{kj} \), can be viewed as a dot product. There are \( s^2 \) entries in matrix-matrix multiplication, and the total counts are obtained as:

\[
\begin{array}{c|c|c}
M/D & A/S \\
\hline
\text{Forming } \mathbf{AB} & s^3 & s^3 - s^2 \\
\end{array}
\]

4.2 Operation counts of RK shape function and its derivatives

In section 2.1, we showed that for the moment \( \mathbf{M}(x) \) matrix to be invertible in constructing the RK shape function, any position \( x \in \Omega \) has to be covered by at least \( n+1 \) kernel functions. Let \( \kappa \) be the overlapping number, we have the relationship

\[
n + 1 \leq \kappa \ll Np \tag{4.1}
\]

Before the construction of shape function, we need to form moment matrices, obtain the inversions, and form the correction functions. Let \( s = n + 1 \), then the vectors \( \mathbf{H}(x - x_i) \) and \( \mathbf{H}(0) \) have dimension \( s \), the moment matrices have dimensions \( s \times s \), and we the following operation counts:
For computing $M^j_{s}(x)$ and $M^j_{s,s}(x)$, the operation counts obtained following Eqs. (3.12) and (3.14) are given below:

After forming the moment matrices, we then construct the correction functions in Eqs. (3.4)-(3.6) and shape functions in Eqs. (3.1)-(3.3). The operation counts for these computations are given in Tables 4 and 5, respectively.

Consequently, the total operation counts for each RK shape function calculation at a give location are given as follows:

$$\text{M/D: } s^3 + (2\kappa + 1)s^2 + s + 1$$  \hspace{1cm} (4.2)

$$\text{A/S: } s^3 + (\kappa - 2)s^2 + s - 1$$  \hspace{1cm} (4.3)

where $s \leq \kappa \ll Np$. The total operation counts for each first derivative $\psi_{i,x}(x)$ are
A study on convergence and complexity of reproducing kernel collocation method

For construction of a second derivative, the operation counts are

\[ (4.4) \]

\[ (4.5) \]

In one-dimension, \( s = n + 1 \). For the case \( s \approx \kappa \), the operation counts for the RK shape function and derivatives under different reproducing degree are listed in Table 6.

Table 6 Operation counts for RK shape functions with different reproducing degrees

<table>
<thead>
<tr>
<th>(M/D)</th>
<th>( \psi_t(x) )</th>
<th>( \psi_{i,s}(x) )</th>
<th>( \psi_{i,xx}(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n = 1 )</td>
<td>( 3s^3 + s^2 + s + 1 )</td>
<td>( 11s^3 + 4s^2 + 3s + 2 )</td>
<td>( 26s^3 + 12s^2 + 6s + 4 )</td>
</tr>
<tr>
<td>( n = 2 )</td>
<td>( 9s^2 - 2s^2 + s + 1 )</td>
<td>( 7s^3 - 5s^2 + s + 1 )</td>
<td>( 16s^3 - 11s^2 + s + 12 )</td>
</tr>
<tr>
<td>( n = 3 )</td>
<td>( 94 (M/D) )</td>
<td>( 39 (A/S) )</td>
<td>( 98 (A/S) )</td>
</tr>
<tr>
<td>( n = 4 )</td>
<td>( 213 (M/D) )</td>
<td>( 782 (M/D) )</td>
<td>( 1884 (M/D) )</td>
</tr>
<tr>
<td>( n = 5 )</td>
<td>( 99 (A/S) )</td>
<td>( 373 (A/S) )</td>
<td>( 864 (A/S) )</td>
</tr>
</tbody>
</table>

For construction of a second derivative \( \psi_{i,xx}(x) \), the operation counts are

\[ (4.4) \]

\[ (4.5) \]

\[ (4.6) \]

\[ (4.7) \]

In general, the cost of forming the linear system under a strong formulation is still much less than that for a weak formulation since each non-zero entry of the matrix and vector of the weak formulation requires a domain integration.

### 4.3 Operation counts for solving a linear system

We take the second problem in Section 3.3, the Poisson problem with Dirichlet and Neumann boundary conditions, as an example for demonstration. From the collocation equations in Eqs. (3.28) and (3.29), we have

\[ \sum_{j=1}^{N_p} b_i \psi_{i,xx}(\xi) = f(\xi), \quad \forall \xi \in \Omega \]  \[ (4.8) \]

\[ \sum_{j=1}^{N_p} b_i \psi_i(\xi) = g_D(\xi), \quad \xi \in \Gamma_0 \]  \[ (4.9) \]

\[ \sum_{j=1}^{N_p} b_i \psi_{i,xx}(\xi) = g_N(\xi), \quad \xi \in \Gamma_1 \]  \[ (4.10) \]

where \( \Omega = \Omega \cup \Gamma_0 \cup \Gamma_1 \), and \( N_p \) is total number of collocation points. We express the above
The operation counts for forming the matrix $F$ are given below:

\[ M/D: Np(s^3 + (2\kappa + 1)s^2 + s + 1) + Np(3s^3 + (8\kappa + 4)s^2 + 3s + 2) + Np(Ns - 2)(6s^3 + (20\kappa + 12)s^2 + 6s + 4) \]  
\[ A/S: Np(s^3 + (\kappa - 2)s^2 + s - 1) + Np(3s^2 + (4\kappa - 5)s^2 + s + 1) + Np(Ns - 2)(6s^3 + (10\kappa - 11)s^2 + s + 12) \]  

Besides, we need $Ns$ functional evaluation in vector $r$. Under the condition in Eq. (3.26), the number of collocation points can be chosen about two to four times the number of source points in computation. The system in Eq. (4.11) is over-determined, and we use the following algorithm to find coefficient vector $x$.

**Algorithm based on Cholesky $LL^T$:**

Step 1: Compute $F^T F$ and $z = F^T r$

Step 2: Cholesky decomposition on $F^T F = LL^T$

Step 3: Solve $Ly = z$ by backward substitution

Step 4: Solve $L^T x = y$ by forward substitution

The operation counts for obtaining the coefficient vector $x$ are given as follows

\[ M/D: \frac{Ns \cdot Np^2}{2} + Ns \cdot Np + \frac{Np^3}{6} + \frac{Np^2}{2} \]  
\[ A/S: \frac{Ns \cdot Np^2}{6} + \frac{Np}{2} \]  

### 4.4 Comparison of operation counts for weak and strong formulations

The total cost to obtain a solution by using RKCM consists of two parts: forming a linear system in Eq. (4.11) and the solution of the linear system.

Taking the Poisson problem discussed above as an example, we consider $Ns = 4 \times Np$ and $\kappa \approx s = n + 1$, where $n > 1$, then the operation count for forming a linear system Eq. (4.11) of RKCM is

\[ \text{op: } 104 Np^2 \cdot s^3 - 38 Np \cdot s^3 \approx O(100 s^3 \cdot Np^2) \]  

The operation count for the solution of the linear system is

\[ \text{op: } \frac{12}{6} Np^3 + 5 Np^2 \approx O(2 Np^3) \]  

where op denotes one M/D plus one A/S, and $2 < s \ll Np$. In total, the operation count is proportional to $Np^3$. For $F$ with a small dimension $Np \leq 50 \cdot s^3$, more CPU time is on forming the linear system. On the other hand, for $F$ with a large dimension $Np > 50 \cdot s^3$, most computation time are on solving the linear system with Cholesky decomposition.
Let's now consider the case for RKPM, the operation count for forming a linear system is
\[ \text{op: } 11Np^2 \cdot s^3 + 6Np \cdot s^3 + \ldots + Np^2 \text{ domain integrations} \]  
(4.17)

Each domain integration consumes a much more CPU when higher order quadrature rules are used. If each domain integral consists of \( Np^2 \) local cell integration using Gaussian quadrature rules, the operation count in Eq. (4.17) becomes
\[ \text{op: } 11Np^2 \cdot s^3 + 6Np \cdot s^3 + \ldots + \mu Np^2 Np^2 \]
\[ \approx O(10s^3 \cdot Np^2) + O(\mu Np^4) \]  
(4.18)

where the number \( \mu \) is dependent on the order of Gaussian quadrature rules used. According to Eq. (4.18), CPU for forming a linear system is far beyond the CPU for solving the linear system. If Cholesky decomposition is used to solve the linear system of equations arising from RKPM (Steps 2–4 only), the operation count is
\[ \text{op: } \frac{Np^2}{6} + \frac{Np^2}{2} + \frac{Np^2}{2} \approx O\left(\frac{1}{6} Np^3\right) \]  
(4.19)

Finally we have a comparison of operation counts for RKPM and RKCM as
\[ \frac{O(\mu Np^4) + O\left(\frac{1}{6} Np^3\right)}{\text{RKPM}} > \frac{O(100s^3 \cdot Np^2) + O(2Np^3)}{\text{RKCM}} \]  
(4.20)

Overall, the total computational cost for weak formulation is higher than that for the strong formulation for larger systems.

5. Conclusions

The employment of reproducing kernel approximation under weak formulation (RKPM) has been widely used to solve PDEs, however, the method needs domain integration and special treatment of Dirichlet boundary conditions. Alternatively, non-local basis functions such as RBFs have been used as the approximation functions for solving PDEs under strong formulation, leading to a full and ill-conditioned system. To resolve these difficulties, we have introduced the compactly supported reproducing kernel approximation to the solution of PDEs under the strong collocation form (RKCM). This approach avoids domain integration and yields a banded and thus a much well-conditioned discrete system.

An important result in this study is to show that the degree of basis functions in the reproducing kernel approximation of RKCM has to be at least 2 for convergence. This is different from the case for RKPM, and the numerical experiments confirm this theoretical results. RKCM is as stable as the one based on weak formulation, such as FEM or RKPM. Further, the computational complexity of RKCM and RKPM are also analyzed and compared. By the operation counts of the two methods discussed in Section 4, we conclude that the RKCM is a simple and effective meshfree method for 2nd order PDEs that exist in many engineering and scientific problems.
Acknowledgements

The support for this work by National Science Council (Taiwan R.O.C.) is gratefully acknowledged.

References


A study on convergence and complexity of reproducing kernel collocation method


**Appendix**

Operation counts for solving fully densed and banded linear systems

Consider the following linear system of equations.

\[ E_i: a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = a_{i,n+1} \]
\[ E_2: a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = a_{2,n+1} \]
\[ \vdots \]
\[ E_n: a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = a_{n,n+1} \]

(a) Solution of a fully densed system (Algorithm A)

The solution procedures of a linear system by using Gaussian Elimination with Backward Substitution are summarized as follows.

**INPUT** Number of unknowns and equations \( n \); augmented matrix \( A = [a_{ij}] \), where \( 1 \leq i \leq n \) and \( 1 \leq j \leq n + 1 \)

**OUTPUT** Solution \( x_1, x_2, \ldots, x_n \) or a message that the linear system has no unique solution

**Step 1:** For \( i = 1, 2, \ldots, n \) do steps 2–4

**Step 2:** Let \( p \) be the smallest integer with \( i \leq p \leq n \) and \( a_{pi} \neq 0 \). If no integer \( p \) can be found, then OUTPUT (“no unique solution exists”); STOP

**Step 3:** If \( p \neq i \), then perform \((E_p) \leftrightarrow (E_i)\)

**Step 4:** For \( j = i + 1, i + 2, \ldots, n \) do steps 5–6

**Step 5:** Set \( m_j = a_{ji}/a_{ii} \)

**Step 6:** Perform \((E_j - m_jE_i) \rightarrow (E_j)\);

**Step 7:** If \( a_{nn} = 0 \), then OUTPUT (“no unique solution exists”); STOP

**Step 8:** Set \( x_n = a_{n,n+1}/a_{nn} \)

**Step 9:** For \( i = n - 1, n - 2, \ldots, 1 \) set \( x_i = \left[ a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}x_j \right]/a_{ii} \)

**Step 10:** OUTPUT \((x_1, x_2, \ldots, x_n)\); STOP

In the above, Step 5 requires \( n - i \) division operations. In Step 6, the replacement of \( E_j \) by \((E_j - m_jE_i)\) requires \( (n - i)(n - i + 1) \) multiplications. Therefore, each term of equations requires \( (n - i)(n - i + 1) \) subtractions. For each counter \( i = 1, 2, \ldots, n - 1 \), the subtotal operations required in Step 5 and Step 6 are

\[ M/D: (n - i) + (n - i)(n - i + 1) = (n - i)(n - i + 2) \]
Summing up the counts for each $i$, we have

$$A/S: \sum_{i=1}^{n-1} (n-i)(n-i+1)$$

In addition, step 8 requires 1 division, and Step 9 requires $n-i$ multiplications and $n-i-1$ additions. The numbers of these operations are

$$M/D: \sum_{i=1}^{n-1} (n-i)(n-i+2) = \frac{2n^3 + 3n^2 - 5n}{6}$$

$$A/S: \sum_{i=1}^{n-1} (n-i)(n-i+1) = \frac{n^3 - n}{3}$$

The total amount of arithmetic operations required in Algorithm A are

$$M/D: \frac{2n^3 + 3n^2 - 5n}{6} + \frac{n^2 + n}{2} = \frac{n^3}{3} + \frac{n^2}{2} - \frac{n}{3}$$

$$A/S: \frac{n^3 - n}{3} + \frac{n^2 - n}{2} = \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6}$$

For large $n$, $M/D$ and $A/S$ are in the order of $\frac{n^3}{3}$.

(b) Solution of a banded system (Algorithm B)

The solution procedures of a $n \times n$ linear system with bandwidth $2w-1$ using Gaussian Elimination with Backward Substitution are described as follows.

INPUT Number of unknowns and equations $n$; augmented matrix $A = [a_{ij}]$, where $1 \leq j \leq n$ and $1 \leq j \leq n+1$

OUTPUT Solution $x_1, x_2, \ldots, x_n$ or a message that the linear system has no unique solution

Step 1: For $i = 1, 2, \ldots, n$ do steps 2–4

Step 2: Let $p$ be the smallest integer with $i \leq p \leq n$ and $a_{pi} \neq 0$. If no integer $p$ can be found, then OUTPUT (“no unique solution exists”); STOP

Step 3: If $p \neq i$, then perform $(E_p) \leftrightarrow (E_i)$

Step 4: For $j = i+1, i+2, \ldots, i+w-1$ do steps 5–6 (Note that $i+w-1 \leq n$)

Step 5: Set $m_{ji} = a_{ji}/a_{ii}$

Step 6: Perform $(E_j - m_{ji} E_i) \rightarrow (E_j)$

Step 7: If $a_{nn} = 0$, then OUTPUT (“no unique solution exists”); STOP

Step 8: Set $x_n = a_{n,n+1}/a_{nn}$

Step 9: For $i = n-1, n-2, \ldots, 1$ set

$$x_i = \left[ a_{i,n+1} - \sum_{j=i+1}^{i+w-1} a_{ij} x_j \right] / a_{ii}$$
Step 10: OUTPUT \((x_1, x_2, \ldots, x_n)\); STOP

In the above, Step 5 requires \(w - 1\) (for \(i \leq n - w + 1\)) or \(n - i\) (for \(i \geq n - w + 2\)) division operations. In Step 6, the replacement of \(E_j\) by \((E_j - m_i)E_i\) requires \((w - 1)w\) (for \(i \leq n - w + 1\)) or \((n - i)(n - i + 1)\) (for \(i \geq n - w + 2\)) multiplications and \((w - 1)w\) (for \(i \leq n - w + 1\)) or \((n - i)\) (for \(i \geq n - w + 2\)) subtractions. For each counter \(i = 1, 2, \ldots, n - 1\), the operations required in Step 5 and Step 6 are

\[
\begin{align*}
M/D: \quad & \left\{ \begin{array}{l}
(w - 1) + (w - 1)w = (w - 1)(w + 1) = w^2 - 1, \quad \text{for } i \leq n - w + 1 \\
(n - i) + (n - i)(n - i + 1) = (n - i)(n - i + 2), \quad \text{for } i \geq n - w + 2
\end{array} \right. \\
A/S: \quad & \left\{ \begin{array}{l}
(w - 1)w, \quad \text{for } i \leq n - w + 1 \\
(n - i)(n - i + 1), \quad \text{for } i \geq n - w + 2
\end{array} \right.
\end{align*}
\]

Summing up the counts for each \(i\), we have

\[
\begin{align*}
M/D: \quad & \sum_{i=1}^{n-w+1} (w^2 - 1) + \sum_{i=n-w+2}^{n-1} (n - i)(n - i + 2) = n(w^2 - 1) - \frac{2w^3}{3} + \frac{w^2}{2} + \frac{w}{6} \\
A/S: \quad & \sum_{i=1}^{n-w+1} (w^2 - w) + \sum_{i=n-w+2}^{n-1} (n - i)(n - i + 1) = n(w^2 - w) - \frac{2w^3}{3} + \frac{w^2}{2} - \frac{w}{6}
\end{align*}
\]

In addition, Step 8 requires 1 division, and in Step 9 requires \(w - 1\) (for \(i \leq n - w + 1\)) or \(n - i\) (for \(i \geq n - w + 2\)) multiplications and \(w - 2\) (for \(i \leq n - w + 1\)) or \(n - i - 1\) (for \(i \geq n - w + 2\)) additions. The numbers of these operations are

\[
\begin{align*}
M/D: \quad & 1 + \sum_{i=1}^{n-w+1} w + \sum_{i=n-w+2}^{n-1} ((n - i) + 1) = nw - \frac{w^2}{2} + \frac{w}{2} \\
A/S: \quad & \sum_{i=1}^{n-w+1} (w - 1) + \sum_{i=n-w+2}^{n-1} ((n - i - 1) + 1) = n(w - 1) - \frac{w^2}{2} + \frac{w}{2}
\end{align*}
\]

The total amount of arithmetic operations required in Algorithm B are

\[
\begin{align*}
M/D: \quad & n(w^2 + w - 1) - \frac{2w^3}{3} + \frac{2w}{3} \quad (A.3) \\
A/S: \quad & n(w^2 - 1) - \frac{2w^3}{3} + \frac{w^2}{2} + \frac{w}{6} \quad (A.4)
\end{align*}
\]

For large \(n\), M/D is approximately \(n(w^2 + w - 1)\) and A/S is approximately \(n(w^2 - 1)\). Since for a sparse and banded linear system, \(w \ll n\), it is more effective to use Algorithm B.

For a special case in Algorithm B by letting \(w = n\), we recover the operation counts of Algorithm A:

\[
\begin{align*}
(A.3) & = n(n^2 + n - 1) - \frac{2n^3}{3} + \frac{2n}{3} = n^3 + n^2 - \frac{n}{3} = (A.1) \\
(A.4) & = n(n^2 - 1) - \frac{2n^3}{3} + \frac{n^2}{2} + \frac{n}{6} = \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6} = (A.2)
\end{align*}
\]