Finite integration method for solving multi-dimensional partial differential equations

M. Li\textsuperscript{b,*}, C.S. Chen\textsuperscript{b,c}, Y.C. Hon\textsuperscript{b,d}, P.H. Wen\textsuperscript{a,b}

\textsuperscript{a} School of Engineering and Materials Science, Queen Mary, University of London, London E1 4NS, UK
\textsuperscript{b} College of Mathematics, Taiyuan University of Technology, Taiyuan, China
\textsuperscript{c} Department of Mathematics, University of Southern Mississippi, Hattiesburg, MS 39406, USA
\textsuperscript{d} Department of Mathematics, City University of Hong Kong, Hong Kong, China

Based on the recently developed Finite Integration Method (FIM) for solving one-dimensional ordinary and partial differential equations, this paper extends the technique to higher dimensional partial differential equations. The main idea is to extend the first order finite integration matrices constructed by using either Ordinary Linear Approach (OLA) (uniform distribution of nodes) or Radial Basis Function (RBF) interpolation (uniform/random distributions of nodes) to higher order integration matrices. Using standard time integration techniques, such as Laplace transform, we have shown that the FIM is capable for solving time-dependent partial differential equations. Illustrative numerical examples are given in two-dimension to compare the FIM (FIM-OLA and FIM-RBF) with the finite difference method and point collocation method to demonstrate its superior accuracy and efficiency.

1. Introduction

Mathematical models in terms of partial differential equations (PDEs) have commonly been used to describe a wide variety of physical phenomena such as sound, heat, electrostatics, electrodynamics, fluid flow, and elasticity. Under various boundary conditions, it is very rare that these models can be solved in closed form solutions. Numerical methods are unavoidable for seeking approximate solutions to simulate the dynamics and characteristics of the models. Due to the advance of computational methods, these kinds of numerical approximation can usually be achieved inexpensively to high accuracy together with a reliable bound on the error between the analytical solution and its numerical approximation. There are many numerical techniques available for solving differential equations \cite{1-5} including the Finite Element Method (FEM) and Boundary Element Method (BEM). In the last decade, the development of the Radial Basis Functions (RBFs) as a truly meshless method has drawn attention from many researchers. In particular, the use of multiquadric radial basis function (MQ-RBF) in \cite{6-8} has shown the superior convergence of the method in comparing with FEM and BEM. Li et al. \cite{9} later compared the accuracy of the RBFs with another two meshless methods, the Method of Fundamental Solutions and Dual Reciprocity Method. Numerical results indicated that these meshless methods provide a similar optimal accuracy for solving both elliptic and parabolic equations in 2D.

\* Corresponding author.
E-mail address: liming01@tyut.edu.cn (M. Li).

http://dx.doi.org/10.1016/j.apm.2015.03.049
0307-904X/© 2015 Elsevier Inc. All rights reserved.
Recently, Wen et al. [10] developed a Finite Integration Method (FIM) for solving differential equation in 1D and demonstrated its applications to nonlocal elasticity problems [11]. It has been shown that the FIM gives higher degree of accuracy than the Finite Difference Method (FDM) and Point Collocation Method (PCM). In this paper, the FIM is further extended to solve multi-dimensional partial differential equations. In this paper, two-dimensional partial differential equations are given for static and dynamic problems as illustrative examples. Similar to the FDM and the PCM, a finite number of points, known as field points, are distributed in the computational domain. The field points are generated either uniformly (grid) along the independent coordinate or randomly in the domain. The integration matrix of the first order is obtained by the direct integration with either OLA approximation or RBFs interpolation. Based on these first order integration matrices, any finite integration matrix with multi-layer integration can easily be obtained. To compare with other numerical methods, the PCM and analytical solutions are used. For two-dimensional time-dependent problems, Laplace transform has been applied to remove the time-dependent variable. The differential equation is then solved in the Laplace space with given initial and boundary conditions. To demonstrate the accuracy and efficiency of the FIM, several numerical examples are given.

2. The FIM for one-dimensional problems

Numerical quadrature rule based on Ordinary Linear Approach (OLA) is the simplest computational scheme for integration [10]. Starting from one-dimension problem, an integral of a given function \( U(x) \) can be written as

\[
U(x) = \int_0^x u(\xi) d\xi.
\]

Applying the linear interpolation technique to (1), we have

\[
U(x_k) = \int_0^{x_k} u(\xi) d\xi = \sum_{i=1}^{k} a_{ki} u(x_i),
\]

where, using trapezoidal rule,

\[
a_{ii} = 0,
\]

\[
a_{ki} = \begin{cases} 0.5\Delta, & i = 1, \\ \Delta, & i = 2,3,\ldots,k-1, \\ 0.5\Delta, & i = k, \\ 0, & i > k, \end{cases}
\]

and \( x_i = \Delta (i - 1), \Delta = b/(N - 1), i = 1,2,\ldots,N \) are nodal points in \([0,b]\), and \( x_1 = 0, x_N = b \). Note that (2) can be written in a matrix form as

\[
\mathbf{U} = \mathbf{A} \mathbf{u},
\]

where \( \mathbf{U} = [U_1, U_2, \ldots, U_N]^T \), \( \mathbf{u} = [u_1, u_2, \ldots, u_N]^T \), the first order integration matrix

\[
\mathbf{A} = (a_{ki}) = \Delta \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 1 & 1/2 & 0 & 0 \\ 1/2 & 1 & 1 & 1/2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1/2 & 1 & 1 & 1 & 1/2 \end{pmatrix}_{N \times N},
\]

and \( U_i = U(x_i), u_i = u(x_i) \) are the values of integration and the integral function respectively at each nodes. Thereafter, consider a multi-integral for one-dimensional problem

\[
U^{(2)}(x) = \int_0^x \int_0^\xi u(\zeta) d\zeta d\xi, \quad x \in [0,b].
\]

Applying the above OLA technique again for integral function \( U^{(2)}(x) \), we have

\[
U^{(2)}(x_k) = \int_0^{x_k} \int_0^\xi u(\zeta) d\zeta d\xi = \sum_{i=1}^{k} \sum_{j=1}^{i} a_{ki} a_{lj} u(x_i) = \sum_{i=1}^{k} a_{ki} U^{(2)}(x_i).
\]

The above multi-integral can also be written in a matrix form as

\[
\mathbf{U}^{(2)} = \mathbf{A}^{(2)} \mathbf{u} = \mathbf{A}^2 \mathbf{u},
\]

where
For two-dimensional problems, let us consider a uniform distribution of collocation points as shown in Fig. 1. Similar to (1), we define
\[
U_x(x, y) = \int_p^x u(\xi, y) d\xi, \quad U_x(x_k, y_k) = \int_p^{x_k} u(\xi, y_k) d\xi,
\]
and the total number of point is \(k = N_1(j - 1) + i\), where \(i\) and \(j\) denote the number of column and the number of row respectively. This numbering system is called the global number system. We can also express each nodal value of integration in (12) in a matrix form as
\[
U_x = A_x u,
\]
where integral nodal value \(U_x = [U_{x1}, U_{x2}, ..., U_{xM}]^T\), nodal value \(u = [u_1, u_2, ..., u_M]^T\) and \(M\) is the total number of collocation points \((M = N_1 \times N_2\) for grid shown in Fig. 1). For a rectangular domain, the first order integration matrix
\[
A_x = \begin{pmatrix}
A & 0 & \ldots & 0 \\
0 & A & 0 & \ldots \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & A
\end{pmatrix}_{N_1 N_2},
\]
in which, \(A\) is integration matrix for one-dimension given in (6) with dimension \(N_1 \times N_1\). Similarly, the integration along \(y\) axis is
which can be written in the matrix form as

\[ \mathbf{U}_f = \mathbf{A} \mathbf{u}, \]

in the local system for the collocation points, where \( k = N_2(i - 1) + j \). The first order integration matrix in the local system is

\[ \mathbf{A}' = \begin{pmatrix}
A & 0 & \ldots & 0 \\
0 & A & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & A
\end{pmatrix}, \]

in which \( A \) is the integration matrix for one-dimension integral given in (6) with dimension \( N_2 \times N_2 \). By a simple re-arrangement of the number of the nodes, (16) can be rewritten, in the global system, as

\[ \mathbf{U}_f = \mathbf{A} \mathbf{u}. \]

For the multi-integration in two-dimensional problem in a rectangular domain, we consider the following integral with respect to coordinate \( x \)

\[ U_x^{(2)}(x, y) = \int_0^x \int_0^y u(x, y) \, dx \, dy, \quad x_i \in [0, b_1], \ y \in [0, b_2], \]

and use the same procedure for one-dimension, one has

\[ U_x^{(2)}(x_i, y_k) = \int_0^y u(x_i, y) \, dy = \sum_{j=1}^{N_2} (a_{jk}) \mathbf{u}, \]

or in a matrix form

\[ \mathbf{U}_x^{(2)} = \mathbf{A}^2 \mathbf{u}, \]

where

\[ \mathbf{A}^2 = \mathbf{A} \mathbf{A} = \begin{pmatrix}
A^2 & 0 & \ldots & 0 \\
0 & A^2 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & A^2
\end{pmatrix}, \]

Similarly, one has multi-integration \( U_y^{(2)}(x, y) \) with respect to coordinate \( y \)

\[ U_y^{(2)}(x_k, y) = \int_0^y \int_0^x u(x, y) \, dx \, dy = \sum_{i=1}^{N_2} (a_{ij}) \mathbf{u}, \]

and

\[ \mathbf{U}_y^{(2)} = \mathbf{A}^2 \mathbf{u}. \]

This method can be extended to the higher order integrations, i.e.

\[ U_x^{(m)}(x_k, y_k) = \int_{\text{layers}}^{x_k} \ldots \int_{\text{layers}}^{x_k} u(x_1, y_1) \, dx_{1} \ldots \, dx_{m}, \quad U_y^{(m)}(x_k, y_k) = \int_{\text{layers}}^{y_k} \ldots \int_{\text{layers}}^{y_k} u(x_1, y_1) \, dy_{1} \ldots \, dy_{m}, \]

\[ x_i \in [0, b_1], \ y_k \in [0, b_2]. \]

Applying ordinary linear interpolation technique again for integral function \( U^{(m)}(x, y) \), we have

\[ U_x^{(m)}(x_k, y_k) = \int_{\text{layers}}^{x_k} \ldots \int_{\text{layers}}^{x_k} u(x_1, y_1) \, dx_{1} \ldots \, dx_{m} = \sum_{i=1}^{M} \sum_{j=1}^{M} (a_{ik}) \mathbf{u}(x_i, y_j), \]

or in a matrix form

\[ \mathbf{U}_x^{(m)} = \mathbf{A}^{(m)} \mathbf{u}, \]

where

\[ \mathbf{A}^{(m)} = \begin{pmatrix}
A^{(m)} & 0 & \ldots & 0 \\
0 & A^{(m)} & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & A^{(m)}
\end{pmatrix}, \]

and

\[ \mathbf{A}^{(m)} = \mathbf{A} \mathbf{A} = \begin{pmatrix}
A^{(m)} & 0 & \ldots & 0 \\
0 & A^{(m)} & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & A^{(m)}
\end{pmatrix}. \]
\begin{equation}
U^{(m)}_y(x_k, y_k) = \int_0^{\gamma_k} \int_0^{\eta_k} u(x_k, \eta_1) d\eta_1 d\eta_m - \sum_{i=1}^{M} \sum_{j=1}^{M} (a_{i,j})_y \cdots (a_{i,j})_y u(x_i, y_i) = \sum_{i=1}^{M} (a_{i,m}^{(m)})^T u_i,
\end{equation}

Again, it can also be written, in a matrix form, as
\begin{equation}
U^{(m)}_x = A^m_x u, \quad U^{(m)}_y = A^m_y u.
\end{equation}

In addition, this method can be extended to multi-layers integration with two coordinates \(x\) and \(y\) as follow:
\begin{equation}
U^{(mm)}(x_k, y_k) = \int_0^{\gamma_k} \int_0^{\gamma_k} \int_0^{\eta_k} \int_0^{\eta_k} u(\xi, \eta) d\xi_1 d\xi_m d\eta_1 d\eta_m \quad x_k \in [0, b_1], y_k \in [0, b_2],
\end{equation}

and the nodal values of the above integration are obtained in the matrix form as
\begin{equation}
U^{(mm)} = A^{mm}_x A^{mm}_y u.
\end{equation}

3. The FIM with radial basis functions

For uniform distribution of nodes (grid), the multi-layer integrations at each node can be obtained quite easily in a matrix form. However, in general case, if the nodes distribution is random, the algorithm OLA discussed in the Section 2 is not valid. In this case, interpolation schemes have to be introduced. Recently, the radial basis functions interpolation schemes and moving least squares method are very popular meshless methods. For example, the MQ-RBF was introduced by Hardy [7] for the interpolation of topographical surfaces in the early stage of radial bases function application. Note that \(u(x)\) in the domain \(\Omega\) can be interpolated over a number of randomly distributed nodes \(x_i = (x_i, y_i), i = 1, 2, \ldots, M\), as
\begin{equation}
u(x) = \sum_{i=1}^{M} R_i(x, x_i) x_i + \sum_{q=1}^{Q} P_q(x) \beta_q = R(x) z + P(x) \beta, \quad x \in \Omega,
\end{equation}

where \(R(x) = [R_1(x, x_1), R_2(x, x_2), \ldots, R_M(x, x_M)]\) is a set of radial basis functions centred at \(x = (x, y), z = [z_1, z_2, \ldots, z_M]^T\) and \(\beta = [\beta_1, \beta_2, \ldots, \beta_Q]^T\) are the coefficients to be determined. For uniqueness, the polynomial terms \(P_q(x)\) satisfies the following additional requirement
\begin{equation}
\sum_{i=1}^{M} P_q(x_i) x_i = 0, \quad q = 1, 2, \ldots, Q.
\end{equation}

From (31) and (32), we have
\begin{equation}
R_0 x + P_0 \beta = u, \quad P_0^T x = 0,
\end{equation}

where
\begin{equation}
R_0 = \begin{bmatrix}
R_1(x_1, x_1) & R_2(x_1, x_2) & \cdots & R_M(x_1, x_M) \\
R_1(x_2, x_1) & R_2(x_2, x_2) & \cdots & R_M(x_2, x_M) \\
\vdots & \vdots & \ddots & \vdots \\
R_1(x_M, x_1) & R_2(x_M, x_2) & \cdots & R_M(x_M, x_M)
\end{bmatrix}, \quad P_0 = \begin{bmatrix}
P_1(x_1) & P_2(x_1) & \cdots & P_Q(x_1) \\
P_1(x_2) & P_2(x_2) & \cdots & P_Q(x_2) \\
\vdots & \vdots & \ddots & \vdots \\
P_1(x_M) & P_2(x_M) & \cdots & P_Q(x_M)
\end{bmatrix},
\end{equation}

and \(u = [u(x_1), u(x_2), \ldots, u(x_M)]^T\). Eq. (33) gives
\begin{equation}
\beta = (P_0^T R_0^{-1} P_0)^{-1} P_0^T R_0^{-1} u, \quad \beta = R_0^{-1} (1 - P_0 (P_0^T R_0^{-1} P_0)^{-1} P_0^T R_0^{-1} P_0)^{-1} P_0^T R_0^{-1} u.
\end{equation}

Substituting the coefficients \(z\) and \(\beta\) from (35) into (31), we have
\begin{equation}
u(x) = [R(x) R_0^{-1} (1 - P_0 (P_0^T R_0^{-1} P_0)^{-1} P_0^T R_0^{-1} P_0)^{-1} P_0^T R_0^{-1} P_0^T R_0^{-1} + P(x) (P_0^T R_0^{-1} P_0)^{-1} P_0^T R_0^{-1} P_0^T R_0^{-1}] u = \sum_{i=1}^{M} \phi_i(x) u_i,
\end{equation}

where \(\phi_i(x)\) is the shape function. From (36), we have
\begin{equation}
\frac{\partial u}{\partial \mathbf{x}} = \sum_{i=1}^{M} \bar{R}_i x(x_i) x_i + \sum_{q=1}^{Q} P_q(x) \beta_q = \sum_{i=1}^{N} \bar{\phi}_i(x) u_i \quad \text{or} \quad u_x = D u,
\end{equation}
\[
\frac{\partial u}{\partial y} = \sum_{i=1}^{M} R_{yi}(x) z_i + \sum_{q=1}^{Q} P_{qy}(x) \beta_q = \sum_{i=1}^{N} \phi_{iy}(x) u_i \quad \text{or} \quad u_y = D_y u. \tag{38}
\]

Furthermore,
\[
\int u(x) dx = \sum_{i=1}^{M} R_{yi}(x) z_i + \sum_{q=1}^{Q} P_{qy}(x) \beta_q + f_0(y) = \sum_{i=1}^{M} \phi_{yi}(x) u_i + f_0(y).
\]

and
\[
\int u(x) dy = \sum_{i=1}^{M} R_{xi}(x) z_i + \sum_{q=1}^{Q} P_{qy}(x) \beta_q + g_0(x) = \sum_{i=1}^{M} \phi_{xi}(x) u_i + g_0(x).
\]

where
\[
\begin{align*}
\bar{R}_{ai}(x) &= \int R_i(x) dx, \quad \bar{P}_{qi}(x) = \int P_q(x) dx, \quad \bar{R}_{yi}(x) = \int R_i(x) dy, \quad \bar{P}_{qi}(y) = \int P_q(x) dy, \\
\bar{\phi}_{ai} &= \int \phi_i(x) dx, \quad \bar{\phi}_{yi} = \int \phi_i(x) dy,
\end{align*}
\]

and \(f_0(y)\) and \(g_0(x)\) are arbitrary functions. Therefore, the integration matrices of the first order are
\[
A_x = \begin{pmatrix}
\bar{\phi}_{x1}(X_1) & \bar{\phi}_{x2}(X_1) & \cdots & \bar{\phi}_{xM}(X_1) \\
\bar{\phi}_{x1}(X_2) & \bar{\phi}_{x2}(X_2) & \cdots & \bar{\phi}_{xM}(X_2) \\
\vdots & \vdots & \ddots & \vdots \\
\bar{\phi}_{x1}(X_M) & \bar{\phi}_{x2}(X_M) & \cdots & \bar{\phi}_{xM}(X_M)
\end{pmatrix}
\]

and
\[
A_y = \begin{pmatrix}
\bar{\phi}_{y1}(X_1) & \bar{\phi}_{y2}(X_1) & \cdots & \bar{\phi}_{yM}(X_1) \\
\bar{\phi}_{y1}(X_2) & \bar{\phi}_{y2}(X_2) & \cdots & \bar{\phi}_{yM}(X_2) \\
\vdots & \vdots & \ddots & \vdots \\
\bar{\phi}_{y1}(X_M) & \bar{\phi}_{y2}(X_M) & \cdots & \bar{\phi}_{yM}(X_M)
\end{pmatrix}
\]

For multi-layer integration, the integration matrix can be obtained from (30). Note that the double layer integration can be obtained analytically in the following algorithm as
\[
\int \int u(x) dx dy = \sum_{i=1}^{M} \bar{R}_{ai}(x) z_i + \sum_{q=1}^{Q} \bar{P}_{qi}(x) \beta_q + x f_0(y) + y f_1(y) = \sum_{i=1}^{M} \bar{\phi}_{ai}(x) u_i + x f_0(y) + f_1(y).
\]

and
\[
\int \int u(x) dy dx = \sum_{i=1}^{M} \bar{R}_{ai}(x) z_i + \sum_{q=1}^{Q} \bar{P}_{qi}(x) \beta_q + y g_0(x) + g_1(x) = \sum_{i=1}^{M} \bar{\phi}_{ai}(x) u_i + y g_0(x) + g_1(x),
\]

where
\[
\bar{R}_{ai}(x) = \int \int R_i(x) dx dy, \quad \bar{P}_{qi}(x) = \int \int P_q(x) dx dy,
\]

and
\[
\bar{\phi}_{ai} = \int \int \phi_i(x) dx dy, \quad \bar{\phi}_{yi} = \int \int \phi_i(x) dy dx,
\]

and \(f_i(y)\) and \(g_i(x), i = 0, 1\) are arbitrary functions. Therefore, the integration matrices of multi- integration matrix are as follows:
\[
B_x = \begin{pmatrix}
\bar{\phi}_{x1}(X_1) & \bar{\phi}_{x2}(X_1) & \cdots & \bar{\phi}_{xM}(X_1) \\
\bar{\phi}_{x1}(X_2) & \bar{\phi}_{x2}(X_2) & \cdots & \bar{\phi}_{xM}(X_2) \\
\vdots & \vdots & \ddots & \vdots \\
\bar{\phi}_{x1}(X_M) & \bar{\phi}_{x2}(X_M) & \cdots & \bar{\phi}_{xM}(X_M)
\end{pmatrix}
\]

and
\[ B_y = \begin{pmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_M(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_M(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_M) & \phi_2(x_M) & \cdots & \phi_M(x_M) \end{pmatrix}_{M \times M} \]  

(50)

It is noteworthy that these two integration matrices can replace integration matrix \( A_x \) and \( A_y \) in (43) and (44) respectively. Three types of radial basis functions, i.e. MQ, linear, and Thin-Plate Splines (TPS) are observed in this paper. Their derivatives and integrals in equations above are presented in the Appendix. For the \( m \)th order derivative, we have the relationship between higher order derivative matrix and first order derivative matrix as

\[ \frac{\partial^m}{\partial x^m} \frac{\partial^2 u}{\partial y^2} = D_x^m D_y^2 u, \]  

(51)

which can be used by the Point Collocation Method (PCM) directly. Unlike the OLA in Section 2, the integral matrices and the derivative matrices with any order using radial basis functions, \( A^{(m)} \), \( B \) and \( D^{(m)} \), are full ranked matrices.

4. The FIM for multi-dimensional problems

The FIM described in Sections 2 and 3 is readily extendable to solving higher dimensional problems. For illustration, consider the following two-dimensional partial differential equation

\[ \begin{align*}
\alpha_1(x,y) \frac{\partial^2 u}{\partial x^2} + \alpha_2(x,y) \frac{\partial^2 u}{\partial y^2} + \alpha_3(x,y) u &= b(x,y), \quad x \in \Omega, \\
\Lambda [u(x,y)] &= h(x,y), \quad x \in \partial \Omega,
\end{align*} \]  

(52)

where \( \Lambda \) is a boundary operator, \( \alpha_1(x,y), \alpha_2(x,y), \alpha_3(x,y), b(x,y) \) and \( h(x,y) \) are given functions. \( u \) is generally referred as potential, which represents the transversal displacement of a membrane. \( \Omega \) and \( \partial \Omega \) are simple connected domain and its boundary respectively. Integrating twice in (52) with respect to coordinates \( x \) and \( y \) respectively, one has

\[ \begin{align*}
\int \int \left[ \alpha_1(x,y) \frac{\partial^2 u}{\partial x^2} + \alpha_2(x,y) \frac{\partial^2 u}{\partial y^2} + \alpha_3(x,y) u \right] dx \, dy \\
= \int \int b(x,y) dx \, dy + x f_0(y) + f_1(y) + y g_0(x) + g_1(x),
\end{align*} \]  

(53)

where \( f_0(y) \), \( f_1(y) \), \( g_0(x) \) and \( g_1(x) \) are unknown one-dimensional functions. Using the technique of integration by part, we have

\[ \begin{align*}
\int \int \left[ \alpha_1 u - 2 \int u \frac{\partial \alpha_1}{\partial x} dx + \int \left[ \frac{\partial^2 u}{\partial x^2} dx \right] dy \\
+ \int \int \left[ \alpha_3 u dx \, dy \ight] = \int \int b(x,y) dx \, dy + x f_0(y) + f_1(y) + y g_0(x) + g_1(x).
\end{align*} \]  

(54)

By using integration matrix mentioned in the previous sections, we have

\[ \left[ \alpha_1^2 \alpha_1 + \alpha_2^2 \alpha_2 - 2 \alpha_1 \alpha_2 \alpha_3 + \alpha_3^2 \alpha_3 \right] u = \left[ \alpha_1^2 \alpha_1 b + \alpha_2^2 \alpha_2 f_0 + \alpha_3 g_0 \right] + \left[ \alpha_1 g_1 + \alpha_3 f_1 \right], \]  

(55)

where \( X = \{x_1, \ldots, x_M\} \), \( Y = \{y_1, \ldots, y_M\} \), \( f_0 = \left[ f_0, f_0, \ldots, f_0 \right]^T \), \( f_1 = \left[ f_1, f_1, \ldots, f_1 \right]^T \), \( g_0 = \left[ g_0, g_0, \ldots, g_0 \right]^T \), \( g_1 = \left[ g_1, g_1, \ldots, g_1 \right]^T \), \( p \) and \( r \) are numbers of point to be used for interpolation of functions \( f(y) \) and \( g(x) \) respectively, \( \Psi_x \) and \( \Psi_y \) are matrices of one-dimensional shape function with respect to coordinates \( x \) and \( y \) respectively, and

\[ \alpha_l = \begin{pmatrix} \alpha_l(x_1) & 0 & \cdots & 0 \\ 0 & \alpha_l(x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_l(x_M) \end{pmatrix}, \quad l = 1, 2, 3. \]  

(56)

Integral functions \( f_0(y), f_1(y), g_0(x) \) and \( g_1(x) \) can be interpolated in terms of the nodal values in the following procedure:

1. Determine the regions of functions \( f(y) \) and \( g(x) \), i.e. \( [y_1, y_r], [x_1, x_p] \), and uniformly distributed points in these regions as shown in Fig. 2;

2. Determine one-dimensional shape function matrices \( \Psi_x \) and \( \Psi_y \);

(a) Ordinary Linear Approximation (OLA)
(b) By using linear interpolation, one has
Therefore, the matrices of shape function are

\[
\begin{bmatrix}
0 & 0 & \frac{\bar{y}_n - \bar{y}}{\bar{y}_n - \bar{y}_{n-1}} & \frac{\bar{y}_n - \bar{y}_{n-1}}{\bar{y}_n - \bar{y}} & 0 & 0 \\
0 & 0 & \frac{\bar{x}_n - \bar{x}}{\bar{x}_n - \bar{x}_{n-1}} & \frac{\bar{x}_n - \bar{x}_{n-1}}{\bar{x}_n - \bar{x}} & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \frac{\bar{y}_n - \bar{y}}{\bar{y}_n - \bar{y}_{n-1}} & \frac{\bar{y}_n - \bar{y}_{n-1}}{\bar{y}_n - \bar{y}} & 0 & 0 \\
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
0 & 0 & \frac{\bar{x}_n - \bar{x}}{\bar{x}_n - \bar{x}_{n-1}} & \frac{\bar{x}_n - \bar{x}_{n-1}}{\bar{x}_n - \bar{x}} & 0 & 0 \\
0 & 0 & \frac{\bar{y}_n - \bar{y}}{\bar{y}_n - \bar{y}_{n-1}} & \frac{\bar{y}_n - \bar{y}_{n-1}}{\bar{y}_n - \bar{y}} & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \frac{\bar{x}_n - \bar{x}}{\bar{x}_n - \bar{x}_{n-1}} & \frac{\bar{x}_n - \bar{x}_{n-1}}{\bar{x}_n - \bar{x}} & 0 & 0 \\
\end{bmatrix}
\]

(b) Radial basis functions

In this case, two one-dimensional shape function matrices with the radial bases function interpolation \( \Phi_x \) and \( \Phi_y \) are

\[
\begin{bmatrix}
\psi_1(x) & \psi_2(x) & \psi_3(x) & \psi_4(x) & \psi_5(x) & 0 & 0 \\
0 & 0 & \frac{\bar{x}_n - \bar{x}}{\bar{x}_n - \bar{x}_{n-1}} & \frac{\bar{x}_n - \bar{x}_{n-1}}{\bar{x}_n - \bar{x}} & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \frac{\bar{x}_n - \bar{x}}{\bar{x}_n - \bar{x}_{n-1}} & \frac{\bar{x}_n - \bar{x}_{n-1}}{\bar{x}_n - \bar{x}} & 0 & 0 \\
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
\psi_1(y) & \psi_2(y) & \psi_3(y) & \psi_4(y) & \psi_5(y) & 0 & 0 \\
0 & 0 & \frac{\bar{y}_n - \bar{y}}{\bar{y}_n - \bar{y}_{n-1}} & \frac{\bar{y}_n - \bar{y}_{n-1}}{\bar{y}_n - \bar{y}} & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \frac{\bar{y}_n - \bar{y}}{\bar{y}_n - \bar{y}_{n-1}} & \frac{\bar{y}_n - \bar{y}_{n-1}}{\bar{y}_n - \bar{y}} & 0 & 0 \\
\end{bmatrix}
\]

in which \( \psi_1(x) \) and \( \psi_1(y) \) are shape functions in one dimensional case as shown in Fig. 2. In (55), we have \( M \) nodal unknowns of \( u \), \( 2q \) unknowns of \( f_0 \), \( f_1 \) and \( 2r \) unknowns of \( g_0 \), \( g_1 \). For a rectangular plate with uniform distribution of nodes \( (N_1 \times N_2) \), obviously one has \( 2p + 2r \) nodes located on the boundary. By selecting \( p = N_1 - 1 \) and \( r = N_2 - 1 \) for uniform distribution of
node, there are \( N_1 \times N_2 + 2(N_1 + N_2 - 2) \) linear system of equations to determine all unknowns, i.e. \( u, f_0, f_1, g_0, \) and \( g_1 \). In fact, the number of boundary points to determine four one-dimensional integral functions is arbitrary. The number of points \( (L) \) on the boundary should be greater than or equal to \( 2(p + r) \). If \( L = 2(p + r) \), the standard Gaussian solver can be used directly. Otherwise, the Singular Value Decomposition (SVD) \([12]\) scheme should be introduced.

For partial differential equation with constants coefficients in \([52]\), i.e. \( \alpha_1 = \alpha_2 = 1 \), using double layer integrations in \((49)\) and \((50)\), we have

\[
[B_b + B_e + B_2 \phi_3] u = B_b B_e \phi + X \phi_3 f_0 + \Psi_l f_1 + Y \Psi_2 g_0 + \Psi_3 g_1.
\]

(63)

Under this circumstance, the first order integration matrices \( A_k \) and \( A_r \) are not necessary.

5. The FIM for Time-dependent problems

For two-dimensional dynamics, the governing equation is written, in time domain, as

\[
\alpha_1 \frac{\partial^2 u}{\partial x^2} + \alpha_2 \frac{\partial^2 u}{\partial y^2} + A(x) \frac{\partial u}{\partial t} + B(x) \frac{\partial^2 u}{\partial t^2} + \alpha_3 u = b(x, t), \quad t > 0, \quad x \in \Omega,
\]

(64)

where \( \alpha_1(x), \alpha_2(x), \alpha_3(x), A(x), B(x), \) and \( b(x, t) \) are given functions. In addition, the boundary conditions and initial conditions are specified as

\[
\begin{align*}
&u(x, 0) = d_1(x), \quad \dot{u}(x, 0) = d_2(x), \quad x \in \Omega, \\
&\Lambda[u(x, t)] = e(x, t), \quad x \in \partial \Omega,
\end{align*}
\]

(65)

where \( d_1(x), d_2(x), \) and \( e(x, t) \) are given functions. By applying the Laplace transform to \((64)\) with consideration of the initial conditions, one obtains

\[
\alpha_1 \frac{\partial^2 \tilde{u}}{\partial x^2} + \alpha_2 \frac{\partial^2 \tilde{u}}{\partial y^2} + A[s\tilde{u} - d_1] + B[s^2 \tilde{u} - sd_1 - d_2] + \alpha_3 \tilde{u} = \tilde{b},
\]

(66)

where the Laplace transform is defined as

\[
\tilde{f} = \int_0^\infty f(t)e^{-st}dt.
\]

(67)

From \((66)\), we have

\[
\alpha_1 \frac{\partial^2 \tilde{u}}{\partial x^2} + \alpha_2 \frac{\partial^2 \tilde{u}}{\partial y^2} + \tilde{E}\tilde{u} = \tilde{h},
\]

(68)

where

\[
\tilde{E}(x, s) = [A(x)s + B(x)s^2 + \alpha_3(x)], \\
\tilde{h}(x, s) = [A(x)d_1(x) + B(x)s d_1(x) + d_2(x)].
\]

(69)

The numerical procedure of solving \((68)\) is the same as solving \((52)\) in Section 4. Applying the integration operation twice on \((68)\), we have

\[
\left[ A_2^2 \lambda_1 + A_2^2 \lambda_2 - 2A_2 A_2 (A_2 \lambda_1 (x_{1,x} + A_2 \lambda_2 (x_{1,y} + A_2 \lambda_1 (x_{1,x}) + A_2 \lambda_2 (x_{1,y}) + \tilde{E}) \right] u = A_2^2 \lambda_2 \tilde{h} + X \phi_0 f_0 + \phi_1 y_0 + Y \phi_2 g_0 + \phi_3 g_1,
\]

(70)

where

\[
\tilde{E}(x, s) = \begin{pmatrix}
\tilde{E}(x_1, s) & 0 & \cdots & 0 \\
0 & \tilde{E}(x_2, s) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \tilde{E}(x_M, s)
\end{pmatrix},
\]

(71)

\[
\tilde{h}(x, s) = \begin{pmatrix}
\tilde{h}(x_1, s) & 0 & \cdots & 0 \\
0 & \tilde{h}(x_2, s) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \tilde{h}(x_M, s)
\end{pmatrix}.
\]

If \( \lambda_1 = \lambda_2 = -B = 1, A = -a, d_1 = d_2 = b = 0 \), Eq. \((70)\) becomes

\[
A_2^2 \lambda_2 - (a + s)A_2^2 \lambda_2 \tilde{u} = X \phi_0 f_0 + \phi_1 y_0 + Y \phi_2 g_0 + \phi_3 g_1,
\]

(72)

By using double layer integration matrix, \((72)\) becomes

\[
(B_2 + B_e - (a + s)B_2 \phi_3) u = X \phi_0 f_0 + \phi_1 y_0 + Y \phi_2 g_0 + \phi_3 g_1,
\]

(73)

All unknowns including nodal values of potential \( u \) and four one-dimensional functions \( f_0, f_1 \) and \( g_0, g_1 \) can be obtained numerically in the Laplace transform space for each specified Laplace parameter \( s_k \). If \( K + 1 \) samples in the transformation space \( s_k, k = 0, 1, \ldots, K \), are selected in the Laplace transform domain, the transformed variables i.e. \( \tilde{u}(x, s_k) \), are evaluated by the numerical procedure above. Then \( u(x, t) \) in the time domain can be determined by Laplace inversion techniques. A simple and accurate method proposed by Durbin \([13]\) is adopted in this paper. The formula of Durbin inversion scheme is given by
\[
\begin{align*}
    f(t) &= \frac{2e^{\pi t}}{T} \left[ -\frac{1}{2} \hat{f}(\eta) + \sum_{k=0}^{N} \text{Re}(\hat{f}(\eta + 2k\pi i/T) e^{2k\pi i/T}) \right] 
\end{align*}
\]

where \( \hat{f}(s_k) \) denotes the transformed variable in the Laplace domain, the parameter of the Laplace transform is chosen as: \( s_k = \eta + 2k\pi i/T \) \((i = \sqrt{-1})\). There are two free normalised parameters in \( s_k: \eta \) and \( T \). The selection of parameters \( T \) depends on the observing period in the time domain.

6. Numerical examples

Example 6.1. We firstly consider the following partial differential equation

\[
\begin{align*}
    \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + ku &= \sin \pi x \sin \pi y, \quad (x, y) \in \Omega, \\
    u(x, y) &= 0, \quad (x, y) \in \partial \Omega,
\end{align*}
\]

where \( \Omega \cup \partial \Omega = [0, 1] \times [0, 1] \).

For the OLA, uniform distribution of nodes is selected \((N_1 = N_2 = 21, M = 441, q = r = N_1)\). For radial basis functions approach in Eqs. (61–63), three radial basis functions are considered, i.e.

1. MQ: \( R(r) = \sqrt{c^2 + r^2} \);
2. Linear (R): \( R(r) = r \);
3. Thin-Plate Splines (TPS): \( R(r) = r^2 \ln r \).

Double layer integration scheme given in (73) is used. For the choice of the free parameter \( c \) in [10], it was observed that a stable and high accurate solution can be obtained when \( c \leq 1.8 \) for one-dimensional problems. It is remarked here that a theoretical justification on the choice of this free parameter \( c \) in MQ was recently given by Luh [14]. In this paper, we simply select this parameter as \( c = 1/N_1 \) and the number of polynomial terms is \( Q = 6 \) for all examples in this section. The analytical solution of (75) is given by

\[
    u^*(x, y) = \frac{\sin \pi x \sin \pi y}{k^2 - 2\pi^2}.
\]

The average relative error is defined as

\[
    \varepsilon = \frac{1}{M} \sum_{i=1}^{M} \left| \frac{u_i - u_i^*}{u^*_{\text{max}}} \right|, \quad u^*_{\text{max}} = \frac{1}{k^2 - 2\pi^2}.
\]

To demonstrate the accuracy of the proposed method, we compare the numerical results with the Point Collocation Method (PCM) and the Finite Difference Method (FDM). The logarithms of the average error \( \varepsilon \) to base 10 varied with two kinds of node distributions as shown in Fig. 3, i.e. uniform and randomly distributions (441 nodes for uniform grid and 102 nodes for random grid). Numerical results of average relative error are shown in Fig. 4 and Fig. 5 using these two distributions of node respectively. Due to the singularity at \( k = 2\pi^2 \), the relative errors are increased significantly near this point for each case. In addition, the accuracy of the FIM is higher than that of the PCM and FDM in most cases. However, the accuracy is also different for using other interpolation schemes in the FIM. The OLA is the simplest method, but suffers the low accuracy using the FIM as shown in Fig. 4. Also in the selection of radial basis functions, the MQ and the TPS are apparently superior to others. However, there is a free parameter in the MQ which has significant effect on the accuracy and has to be selected carefully. Fig. 5 shows the logarithms of the average error \( \varepsilon \) with irregular distribution of nodes in the domain. The conclusion is similar to that for the uniform node distribution. The accuracy of the PCM is the lowest and very close to that by the FIM with linear RBF. For the irregular node distribution, the method of OLA is not available. To observe the effect of the selection of shape parameter \( c \) in MQ, the results using two solvers, i.e. SVD and Gaussian algorithms, are presented in Fig. 6 for the case \( k = 10 \). It is shown that the differences between these two solvers are small. In addition, we can conclude that the reasonable degree of accuracy can be obtained for \( c < 0.35 \) for two-dimensional problems using FIM. Comparing the computational effort, the CPU times used by PCM, FDM and FIM are very close as there are no any special function need to be calculated in the system matrix and also the Gaussian linear algebraic equation solver is used.

Example 6.2. In this example, consider the following partial differential equation

\[
\begin{align*}
    &(x(1-x) \frac{\partial^2 u}{\partial x^2} + y(1-y) \frac{\partial^2 u}{\partial y^2} = -4xy(1-x)(1-y), \quad (x, y) \in \Omega, \\
    &u(x, y) = 0, \quad (x, y) \in \partial \Omega,
\end{align*}
\]
Fig. 3. Distributions of nodes: (a) uniform; (b) random.

Fig. 4. The logarithms of the average errors versus parameter $k$ for different method (regular distribution).

Fig. 5. The average errors versus parameter $k$ for different methods (irregular distribution of nodes).
where $\Omega \cup \partial \Omega = [0, 1] \times [0, 1]$. The analytical solution is given by $u^*(x, y) = xy(1 - x)(1 - y)$. The average relative error is defined as

$$
\varepsilon = \frac{1}{M} \sum_{i=1}^{M} \frac{|u_i - u'_i|}{|u'_{\text{max}}|}, \quad u'_{\text{max}} = u^*(0.5, 0.5) = \frac{1}{16}.
$$

(79)

In this example, we chose $N_1 = N_2 = N$ and $p = r = N$. Similar to the previous example, the shape parameter $c$ of MQ is selected as $c = 1 / N$. For the RBF approach, single integration matrix in (70) is used. The average errors $\varepsilon$ for various number of collocation point are shown in Table 1. Among these algorithms, the accuracy of OLA is the lowest and PSF of radial basis function is the highest.

**Example 6.3.** Consider the following partial differential equation with a unit circle domain

$$
\begin{align*}
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= -12xy, \quad (x, y) \in \Omega, \\
u(x, y) &= 0, \quad (x, y) \in \partial \Omega,
\end{align*}
$$

(80)

where $\Omega \cup \partial \Omega = \{(x, y) : x^2 + y^2 \leq 1\}$.

The analytical solution is given by $u^*(x, y) = xy(1 - x^2 - y^2)$. For implementation, three nodal densities with node numbers $M = 93, 343$ and $747$ and the node numbers for determining four one-dimensional functions $p = r = 11, 21$ and $31$ are given respectively. The distribution of node in a unit circle domain is shown in Fig. 7 for $M = 343$. To solve the linear system of equations, the SVD is used [12]. In this example, TPS is employed. It is because TPS has no shape parameter and its degree of accuracy is very close to that by MQ. The numerical solutions are plotted in Fig. 8 and the exact solution is given for comparison. The average error is defined as

$$
\varepsilon = \frac{1}{m} \sum_{i=1}^{m} |u(x_i, y_i) - u^*(x_i, y_i)|,
$$

(81)

where the test points are $\{(x_i, y_i) : x_i = y_i = (i - 1)/(m - 1) \cos \pi/4, i = 1, 2, ..., m, m = 21\}$. The average errors $\varepsilon$ using various number of collocation points $M$ are shown in Table 2.

**Example 6.4.** Finally, we consider the following scale wave equation with initial and boundary conditions

$$
\begin{align*}
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= a \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial t^2} = 0, \quad t > 0, \quad (x, y) \in \Omega, \\
u(x, y, 0) &= \hat{u}(x, y, 0) = 0, \quad (x, y) \in \Omega \cup \partial \Omega, \\
u_x(0, y, t) &= u_x(1, y, t) = u(x, 0, t) = 0, \quad 0 \leq x \leq 1, 0 \leq y \leq 1, \\
u_y(x, 1, t) &= H(t), \quad 0 \leq x \leq 1,
\end{align*}
$$

(82)

where $\Omega \cup \partial \Omega = [0, 1] \times [0, 1]$. $H(t)$ is Heaviside function, $a$ is the damping factor. Therefore, the boundary condition of the last equation of Eq. (82) in the Laplace transform domain is $\hat{u}_y(x, 1, s) = 1/s$. Uniform distribution of node is considered using $N_1 = N_2 = 20$ and $p = r = 20$ for one-dimensional functions. The number of samples in the Laplace space is $K = 100$. In this

![Log(ε) vs c](image)

**Fig. 6.** The logarithms of the average error versus shape parameter $c$ for two solvers for the case $k = 10$. 

example, two free parameters in (74) are selected as \( \eta = 5 \) and \( T = 20 \). The displacements using MQ and TPS at two locations \( (x = 1 \) and \( x = 0.5 \) if \( y = 0.5 \)) are presented in Fig. 9 without damping \( (\alpha = 0) \). Double layer integration scheme in (73) is used.

Good agreement has been observed for the time-dependent displacement with analytical solution. In this case, the analytical solution can be presented by a periodic function, at the top of the plate, as

\[
\begin{align*}
    u^*(1, t) &= \begin{cases} 
    t, & 0 \leq t \leq 2, \\
    4 - t, & 2 \leq t \leq 4,
    \end{cases} \\
    u^*(1, t + 4) &= u^*(1, t), \\
    \text{and} \\
    u^*(0.5, t) &= \begin{cases} 
    0, & 0 \leq t \leq 0.5, \\
    t - 0.5, & 0.5 \leq t \leq 1.5, \\
    1, & 1.5 \leq t \leq 2.5, \\
    3.5 - t, & 2.5 \leq t \leq 3.5, \\
    0, & 3.5 \leq t \leq 4,
    \end{cases} \\
    u^*(0.5, t + 4) &= u^*(0.5, t).
\end{align*}
\] (83)

and

\[
\begin{align*}
    u^*(1, t) &= \begin{cases} 
    t, & 0 \leq t \leq 2, \\
    4 - t, & 2 \leq t \leq 4,
    \end{cases} \\
    u^*(1, t + 4) &= u^*(1, t), \\
    \text{and} \\
    u^*(0.5, t) &= \begin{cases} 
    0, & 0 \leq t \leq 0.5, \\
    t - 0.5, & 0.5 \leq t \leq 1.5, \\
    1, & 1.5 \leq t \leq 2.5, \\
    3.5 - t, & 2.5 \leq t \leq 3.5, \\
    0, & 3.5 \leq t \leq 4,
    \end{cases} \\
    u^*(0.5, t + 4) &= u^*(0.5, t).
\end{align*}
\] (84)
Considering the damping effect $a = 0.5$, the displacements using MQ and TPS for two points are presented in Fig. 10. In this case, there is no close form solution available. However, the results using these two RBFs are too close to see the difference. It is apparent that the solutions at these two locations tend to the values of static value due to the effect of damping when $t > 11$.

### 7. Conclusion

In this paper, the Finite Integration Method (FIM) with ordinary linear approach and radial basis functions interpolation was extended to solve multi-dimensional differential equations. Coupled with the technique of Laplace transform, we demonstrated that the method can be applied to solve time-dependent partial differential equation. Compared with the

<table>
<thead>
<tr>
<th>M</th>
<th>93</th>
<th>343</th>
<th>747</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>0.009361</td>
<td>0.003909</td>
<td>0.000467</td>
</tr>
</tbody>
</table>

Table 2: Average errors for various node numbers.

Fig. 9. Variations of displacement on the top and middle of the plate versus time without damping.

Fig. 10. Variations of displacement on the top and middle of the plate versus time with damping effect.
Point Collocation Method (PCM) and the finite difference method, the proposed FIM performs much superior in accuracy and stability. For the FIM with radial basis functions interpolation, the use of randomly distributed nodes in the domain allows solving problems under irregular domains. In addition, FIM should be workable for arbitrary shape of the boundary. However, there are still some issues to deal with arbitrary shape boundary such as the determination of functions $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial^2 u}{\partial y^2}$, particularly for the problem with domain containing holes. Obviously the application of the FIM for solving two and three dimensional more complicated elasticity problems and time fractional diffusion equations [15] will be investigated in our future work.

Appendix A.

(I) MQ

$$R_i(x) = \sqrt{c^2 + r^2}, \quad r = \sqrt{(x-x_i)^2 + (y-y_i)^2}, \quad (A1)$$

where $c$ is a free parameter. It is easy to obtain its first derivative as

$$R_{ix}(x, y) = \frac{x-x_i}{\sqrt{c^2 + r^2}}, \quad R_{iy}(x, y) = \frac{y-y_i}{\sqrt{c^2 + r^2}}. \quad (A2)$$

$$\bar{R}_{ix}(x, y) = \frac{x-x_i}{2} r + \frac{c^2 + (y-y_i)^2}{2} \ln \left( x-x_i + \sqrt{c^2 + r^2} \right). \quad (A3)$$

$$\bar{R}_{iy}(x, y) = \frac{y-y_i}{2} r + \frac{c^2 + (x-x_i)^2}{2} \ln \left( y-y_i + \sqrt{c^2 + r^2} \right). \quad (A4)$$

$$\tilde{R}_{ix}(x, y) = \frac{r^3}{6} + \frac{c^2 + (y-y_i)^2}{2} (x-x_i) \ln \left( x-x_i + \sqrt{c^2 + r^2} \right) - \frac{c^2 + (y-y_i)^2}{2} r, \quad (A5)$$

$$\tilde{R}_{iy}(x, y) = \frac{r^3}{6} + \frac{c^2 + (x-x_i)^2}{2} (y-y_i) \ln \left( y-y_i + \sqrt{c^2 + r^2} \right) - \frac{c^2 + (x-x_i)^2}{2} r. \quad (A6)$$

(II) Linear function

$$R_i(x) = r, \quad R_{ix}(x, y) = \frac{x-x_i}{r}, \quad R_{iy}(x, y) = \frac{y-y_i}{r}, \quad (A7)$$

$$\bar{R}_{ix}(x, y) = \frac{x-x_i}{2} r + \frac{c^2 + (y-y_i)^2}{2} \ln(x-x_i + r), \quad (A8)$$

$$\bar{R}_{iy}(x, y) = \frac{y-y_i}{2} r + \frac{c^2 + (x-x_i)^2}{2} \ln(y-y_i + r), \quad (A9)$$

$$\tilde{R}_{ix}(x, y) = \frac{r^3}{6} + \frac{(y-y_i)^2}{2} (x-x_i) \ln(x-x_i + r) - \frac{(y-y_i)^2}{2} r, \quad (A10)$$

$$\tilde{R}_{iy}(x, y) = \frac{r^3}{6} + \frac{(x-x_i)^2}{2} (y-y_i) \ln(y-y_i + r) - \frac{(x-x_i)^2}{2} r. \quad (A11)$$

(III) Thin-Plate Splines

$$R_i(x) = r^2 \ln r, \quad R_{ix}(x, y) = (x-x_i)(1 + 2 \ln r), \quad R_{iy}(x, y) = (y-y_i)(1 + 2 \ln r), \quad (A12)$$

$$\bar{R}_{ix}(x, y) = \left( y-y_i \right)^2 + \frac{(x-x_i)^2}{3} (x-x_i) \ln r - \frac{2}{3} \left( y-y_i \right)^2 + \frac{(x-x_i)^2}{6} (x-x_i) + \frac{2}{3} (y-y_i)^3 \tan^{-1} \frac{x-x_i}{y-y_i}, (A13)$$

$$\bar{R}_{iy}(x, y) = \left( x-x_i \right)^2 + \frac{(y-y_i)^2}{3} (y-y_i) \ln r - \frac{2}{3} \left( x-x_i \right)^2 + \frac{(y-y_i)^2}{6} (y-y_i) + \frac{2}{3} (x-x_i)^3 \tan^{-1} \frac{y-y_i}{x-x_i}, (A14)$$

$$\tilde{R}_{ix}(x, y) = \frac{1}{12} \left( (x-x_i)^4 + 6(x-x_i)^2 (y-y_i)^2 - 3(y-y_i)^4 \right) \ln r - \frac{13}{24} (x-x_i)^2 (y-y_i)^2 + \frac{2}{3} (x-x_i)(y-y_i)^3 \tan^{-1} \frac{x-x_i}{y-y_i}, (A15)$$
\[
\tilde{R}_c(x, y) = \frac{1}{12} ((y - y_i)^4 + 6(y - y_i)^2(x - x_i)^2 - 3(x - x_i)^4) \ln r - \frac{13}{24} (y - y_i)^2(x - x_i)^2 + \frac{2}{3} (y - y_i)(x - x_i)^3 \tan^{-1} \frac{y - y_i}{x - x_i}.
\]

(A16)

(IV) Polynomial Basis Functions
We select polynomial basis function as \((Q = 6)\)
\[
P_1 = 1, P_2 = x, P_3 = y, P_4 = x^2, P_5 = xy, P_6 = y^2.
\]

(A17)

Then, we have
\[
P_{1,x} = 0, P_{2,x} = 1, P_{3,x} = 0, P_{4,x} = 2x, P_{5,x} = y, P_{6,x} = 0.
\]

(A18)
\[
P_{1,y} = 0, P_{2,y} = 0, P_{3,y} = 1, P_{4,y} = 0, P_{5,y} = x, P_{6,y} = 2y.
\]

(A19)
\[
\tilde{P}_{x1} = x, \tilde{P}_{x2} = \frac{x^2}{2}, \tilde{P}_{x3} = xy, \tilde{P}_{x4} = \frac{x^3}{3}, \tilde{P}_{x5} = \frac{x^4}{2}, \tilde{P}_{x6} = xy^2,
\]

(A20)
\[
\tilde{P}_{y1} = y, \tilde{P}_{y2} = xy, \tilde{P}_{y3} = \frac{y^2}{2}, \tilde{P}_{y4} = x^2y, \tilde{P}_{y5} = \frac{xy^2}{2}, \tilde{P}_{y6} = \frac{y^3}{3},
\]

(A21)
\[
\tilde{P}_{x1} = \frac{x^2}{2}, \tilde{P}_{x2} = \frac{x^3}{6}, \tilde{P}_{x3} = \frac{x^2y}{2}, \tilde{P}_{x4} = \frac{x^4}{12}, \tilde{P}_{x5} = \frac{x^3y}{6}, \tilde{P}_{x6} = \frac{x^2y^2}{2},
\]

(A22)
\[
\tilde{P}_{y1} = \frac{y^2}{2}, \tilde{P}_{y2} = \frac{xy^2}{2}, \tilde{P}_{y3} = \frac{y^3}{6}, \tilde{P}_{y4} = \frac{x^2y^2}{2}, \tilde{P}_{y5} = \frac{xy^3}{6}, \tilde{P}_{y6} = \frac{y^4}{12}.
\]

(A23)

References