



# Reproducing kernel element method. Part IV: Globally compatible $C^n$ ( $n \geq 1$ ) triangular hierarchy

Daniel C. Simkins Jr.<sup>a</sup>, Shaofan Li<sup>a,\*</sup>, Hongsheng Lu<sup>b</sup>, Wing Kam Liu<sup>b</sup>

<sup>a</sup> *Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720, USA*

<sup>b</sup> *Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA*

Received 17 June 2003; received in revised form 24 September 2003; accepted 2 December 2003

## Abstract

In this part of the work, a globally compatible  $C^n(\Omega)$  triangular element hierarchy is constructed in the framework of reproducing kernel element method (RKEM) for arbitrary two dimensional domains. In principle, the smoothness of the globally conforming element can be made arbitrarily high ( $n \geq 1$ ). The triangle interpolation field can interpolate the derivatives of an unknown function up to arbitrary  $m$ th order, ( $I^m$ ), and it can reproduce complete  $k$ th order polynomials with  $k \geq m$ . This is the first interpolation hierarchical structure that has ever been constructed with both minimal degrees of freedom and higher order smoothness and continuity over discretizations of a multiple dimensional domain. The performance of the newly constructed compatible element is evaluated in solving several Kirchhoff plate problems.

© 2004 Elsevier B.V. All rights reserved.

*Keywords:* Finite element methods; Meshfree methods; Triangle elements; Kirchhof plates

## 1. Introduction

In the early series of this work [6,7], a hybrid meshfree/finite element method is proposed, and it is termed reproducing kernel element method (RKEM). The main advantage and application of this new method is to build smooth interpolation field on any mesh of an arbitrary domain, so that it can be used in: (1) general curved surface fitting in multiple dimension with high demand in smoothness; (2) Galerkin weak formulations that contain higher order derivatives, such as simulations of plate and shell structures and computations of gradient elasticity and gradient plasticity.

The main technical ingredient of RKEM is combining meshfree reproducing kernel shape functions [5,8,9] with a special partition of unity—the *global partition polynomials* to form a high quality “*ultra finite element*” interpolant that possesses arbitrary smoothness and satisfies higher order Kronecker- $\delta$  property. Apparently, to construct such globally conforming shape functions on an arbitrary, multi-dimensional

\* Corresponding author. Tel.: +1-510-642-5362; fax: +1-510-643-8928.  
E-mail address: [li@ce.berkeley.edu](mailto:li@ce.berkeley.edu) (S. Li).

mesh is not a trivial task. In this part of the work, a detailed technical account on how to construct RKEM shape functions on general triangle meshes is presented, which includes (1) how to construct the so-called *global partition polynomials* on general triangle meshes, and (2) how to program RKEM computer code.

**2. Globally compatible RKEM triangle**

The general structure of an RKEM formulation is a hybrid or a combination between the so-called global partition polynomials and the meshfree reproducing kernel functions. Precisely speaking, an RKEM interpolation field is defined as

$$\mathcal{I}f(\mathbf{x}) = \sum_{e \in \mathcal{A}_E} \left[ \int_{\Omega_e} \mathcal{K}_\rho(\mathbf{x} - \mathbf{y}; \mathbf{x}) \, d\mathbf{y} \left( \sum_{i \in \mathcal{A}_e} \psi_{e,i}(\mathbf{x}) f_{e,i} \right) \right], \tag{2.1}$$

where  $f(\mathbf{x})$  is the unknown function,  $\mathcal{K}_\rho$  is the meshfree kernel function, and  $\psi_{e,i}(\mathbf{x})$  are the so-called global partition polynomials, and  $\mathcal{I}$  is the interpolation operator. The set  $\mathcal{A}_E$  is the index set for all the elements, and the set  $\mathcal{A}_e$  is the index set for the nodal points in an element. For detailed discussions on RKEM formulation, readers are referred to Parts I and II of this work [6,7].

We first present a general formulation for a particular family of global partition polynomials suitable for use in generating RKEM triangular elements. For convenience, we only consider its applications in solving scalar type partial differential equations. Specifically, we shall present numerical solution of thin plate problems by using the newly proposed elements.

Unlike the construction of high order finite element shape functions on triangle meshes, the RKEM interpolant on a triangle mesh does not need any extra degrees of freedom either on the sides of a triangle or in the interior of a triangle. The requirement on degrees of freedom is absolutely minimum. Each node of a triangle element has a number of nodal degrees of freedom (DOF), which are related to the primary unknown and its various derivatives. For each DOF, we have to construct an associated global partition polynomial. One of the objectives of this paper is to show how to construct these global partition polynomials.

Using multiple index notation, one may write the global partition polynomials for each element as

$$\psi_{e,1}(\mathbf{x}) = \begin{bmatrix} \psi_1^{(00)} \\ \psi_1^{(10)} \\ \psi_1^{(01)} \\ \vdots \end{bmatrix} \quad \psi_{e,2}(\mathbf{x}) = \begin{bmatrix} \psi_2^{(00)} \\ \psi_2^{(10)} \\ \psi_2^{(01)} \\ \vdots \end{bmatrix} \quad \psi_{e,3}(\mathbf{x}) = \begin{bmatrix} \psi_3^{(00)} \\ \psi_3^{(10)} \\ \psi_3^{(01)} \\ \vdots \end{bmatrix}, \tag{2.2}$$

where  $e \in \mathcal{A}_E$  and 1, 2, 3 represents the three nodes in an triangle, such that the primary scalar variable can be interpolated by the formula

$$\mathcal{I}_{\text{fem}} w(x, y) = [\psi_1^T \quad \psi_2^T \quad \psi_3^T] \mathbf{w}_l, \quad \forall \mathbf{x} \in \Omega_e, \tag{2.3}$$

where the superscript T denotes transpose, and  $\mathbf{w}_l$  is a vector of all the nodal unknowns:

$$\mathbf{w}_I = \begin{bmatrix} w(x_1, y_1) \\ w_{,x}(x_1, y_1) \\ w_{,y}(x_1, y_1) \\ \vdots \\ w(x_2, y_2) \\ w_{,x}(x_2, y_2) \\ w_{,y}(x_2, y_2) \\ \vdots \\ w(x_3, y_3) \\ w_{,x}(x_3, y_3) \\ w_{,y}(x_3, y_3) \\ \vdots \end{bmatrix} \quad (2.4)$$

It may be noted that in (2.2) the conventional multiple index notation is used (see [6,7] for explanation).

There are two systematic approaches that one can take to construct the global partition polynomials, each having advantages and disadvantages. The first is to work completely in the geometric (physical) domain, and determine the global partition polynomials directly. We call this approach the direct approach. In the second approach one first finds parametric global partition polynomials and then map a general triangle to a unit triangle (parent), as shown in Fig. 1. We call the second approach the parametric approach. We demonstrate the first approach with the T12P3I(4/3) triangle element, the parametric approach is demonstrated in construction of T9P2I1 and T18P4I2 triangle elements. The relative merits will be noted as we proceed.

### 2.1. Direct approach

Following Bell [1], we define an interpolation field using as many terms from Pascal’s triangle, Fig. 2, as are necessary to match the total number of unknowns in an element. In the event that the number of unknowns does not coincide with a complete set of monomial terms from Pascal’s triangle, symmetric combinations are used to approximate higher order terms. Examples will be shown in later sections. We depart from Bell [1], however, in that we do not define any side nodes, we only have corner nodes and all unknowns are at the corner nodes. This eliminates the need for static condensation, while it keeps the requirement on the number of degrees of freedom at each node absolutely minimal. It may be noted that the inter-element non-conformities will be assuaged by the meshfree reproducing kernel function. Therefore,

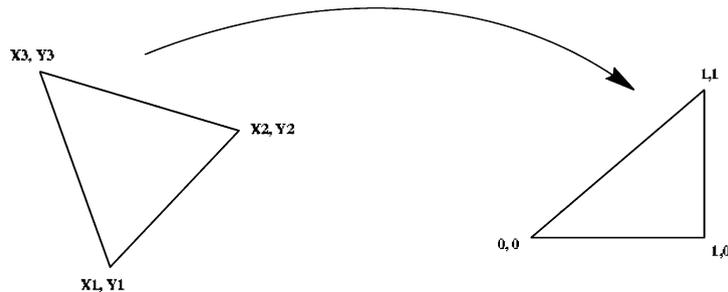


Fig. 1. Arbitrary triangle and its map into a unit triangle.

$$\begin{array}{cccccccc}
 & & & & & & & 1 \\
 0 & & & & & & & \\
 1 & & & & x & & y & \\
 2 & & & x^2 & & xy & & y^2 \\
 3 & & x^3 & & x^2y & & xy^2 & & y^3 \\
 4 & x^4 & & x^3y & & x^2y^2 & & xy^3 & & y^4 \\
 5 & x^5 & & x^4y & & x^3y^2 & & x^2y^3 & & xy^4 & & y^5
 \end{array} \tag{2.5}$$

Fig. 2. Pascal's triangle.

this is a globally compatible interpolation field. Its continuity is up to the smoothness of the compact supported window function embedded in meshfree reproducing kernel (see: discussions in [6,7,9]).

The element interpolation provided by the global partition polynomials can be written as a vector equation

$$w(x,y) = \mathbf{\Phi}^T(x,y)\mathbf{c}, \tag{2.6}$$

where

$$\mathbf{c}^T := [c_1 \ c_2 \ c_3 \ c_4 \ c_5 \ c_6 \ \dots]^{1 \times n_{\text{dof}}},$$

$$\mathbf{\Phi}^T(x,y) := [1 \ x \ y \ x^2 \ xy \ y^2 \ \dots]^{1 \times n_{\text{dof}}} \tag{2.7}$$

and  $n_{\text{dof}}$  is the total number of degrees of freedom in an element.

We can determine the coefficients  $\mathbf{c}$  in terms of the nodal unknowns  $\mathbf{w}_I$ , by solving a set of linear equations

$$\begin{aligned}
 w(x_1, y_1) &= \mathbf{\Phi}^T(x_1, y_1)\mathbf{c} \\
 w_x(x_1, y_1) &= \mathbf{\Phi}_{,x}^T(x_1, y_1)\mathbf{c} \\
 w_y(x_1, y_1) &= \mathbf{\Phi}_{,y}^T(x_1, y_1)\mathbf{c} \\
 &\vdots \\
 w(x_2, y_2) &= \mathbf{\Phi}^T(x_2, y_2)\mathbf{c} \\
 w_x(x_2, y_2) &= \mathbf{\Phi}_{,x}^T(x_2, y_2)\mathbf{c} \\
 w_y(x_2, y_2) &= \mathbf{\Phi}_{,y}^T(x_2, y_2)\mathbf{c} \\
 &\vdots \\
 w(x_3, y_3) &= \mathbf{\Phi}^T(x_3, y_3)\mathbf{c} \\
 w_x(x_3, y_3) &= \mathbf{\Phi}_{,x}^T(x_3, y_3)\mathbf{c} \\
 w_y(x_3, y_3) &= \mathbf{\Phi}_{,y}^T(x_3, y_3)\mathbf{c} \\
 &\vdots
 \end{aligned}$$

Denote the resulting coefficient matrix  $\mathbf{A}$ , then

$$\mathbf{c} = \mathbf{A}^{-1}\mathbf{w}_I \tag{2.8}$$

and

$$w(x,y) = \mathbf{\Phi}^T\mathbf{A}^{-1}\mathbf{w}_I, \tag{2.9}$$

where

$$\begin{aligned} \mathbf{A}^T &= [A_{ij}^T]^{n_{\text{dof}} \times n_{\text{dof}}} \\ &= [\Phi(x_1, y_1), \Phi_x(x_1, y_1), \Phi_y(x_1, y_1), \dots; \Phi(x_2, y_2), \Phi_x(x_2, y_2), \Phi_y(x_2, y_2), \dots; \Phi(x_3, y_3), \\ &\quad \Phi_x(x_3, y_3), \Phi_y(x_3, y_3), \dots]. \end{aligned} \tag{2.10}$$

The desired nodal shape functions can now be determined by comparing Eqs. (2.3) and (2.9).

The direct approach has as its advantage that no mapping is required. This allows any number of DOF per node, e.g. the T12P3I(4/3) triangle. The disadvantage of the direct approach is that the  $\mathbf{A}$  matrix is likely too cumbersome to invert analytically, resulting in a numerical inversion for every element. While on modern computers inversion of  $12 \times 12$  or  $18 \times 18$  matrices are pedestrian, the cost may not be insignificant when they are done for many elements. Furthermore, the  $\mathbf{A}$  matrix may even be singular for certain orientation of a given triangle, which will result in an obvious disaster.

### 2.2. Parametric approach

We now consider the parametric approach. The only difference in this approach from the first approach is that the geometric (physical) triangle is mapped to a parent triangle.

We choose the parent triangle to be the unit triangle with vertices  $(0, 0)$ ,  $(1, 0)$  and  $(1, 1)$  in the  $s-t$  parametric coordinate system (see Fig. 1). The reason for this choice of unit triangle will be explained below.

The area coordinate shape functions for this triangle are:

$$N_1(s, t) = 1 - s, \tag{2.11}$$

$$N_2(s, t) = s - t, \tag{2.12}$$

$$N_3(s, t) = t. \tag{2.13}$$

Let the transformed nodal unknowns be denoted  $\tilde{\mathbf{w}}_I$ , which is related to the original unknowns  $\mathbf{w}_I$  by

$$\tilde{\mathbf{w}}_I = \Lambda \mathbf{w}_I.$$

Since  $w$  is a scalar, it is unchanged by the transformation. However, derivatives must be transformed since the interpolation over the parent triangle correctly interpolates  $\tilde{w}$ ,  $\tilde{w}_{,s}$ ,  $\tilde{w}_{,t}$ , etc. We compute the transformation through the use of Eqs. (2.11)–(2.13) and note that the Jacobian is constant.

In terms of the nodal coordinates, the Jacobian and its inverse are

$$\mathbf{J} = \begin{bmatrix} \frac{ds}{dx} & \frac{ds}{dy} \\ \frac{dt}{dx} & \frac{dt}{dy} \end{bmatrix} = \frac{1}{x_{21}y_{32} - y_{21}x_{32}} \begin{bmatrix} y_{32} & x_{23} \\ y_{12} & x_{21} \end{bmatrix}, \tag{2.14}$$

$$\mathbf{J}^{-1} = \begin{bmatrix} \frac{dx}{ds} & \frac{dx}{dt} \\ \frac{dy}{ds} & \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} x_{21} & x_{32} \\ y_{21} & y_{32} \end{bmatrix}, \tag{2.15}$$

where the notation  $x_{ij} := x_i - x_j$ , and  $y_{ij} := y_i - y_j$ ,  $i, j = 1, 2, 3$ .

We can now write down an expression for  $\Lambda$ . First, partition the matrix into blocks that act only on values at a particular node, let

$$\Lambda := \begin{bmatrix} \lambda_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_2 \end{bmatrix}, \tag{2.16}$$

where  $\mathbf{0}$  is a matrix of all 0's, and the  $\lambda_i$  are matrices that operate on the function value and derivatives at node  $i$ . Explicit expressions will be given in the two parametric examples below.

An interpolation field is now computed for the unit triangle using Eqs. (2.6)–(2.8), but all variables are expressed in the  $s$ – $t$  parametric coordinate system. The interpolation can be written as a vector equation

$$\tilde{w}(s, t) = \Phi^T(s, t)\mathbf{c}, \tag{2.17}$$

where

$$\mathbf{c}^T := [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad \dots],$$

$$\Phi^T(s, t) := [1 \quad s \quad t \quad s^2 \quad st \quad t^2 \quad \dots].$$

For the unit triangle, we can determine the coefficients  $\mathbf{c}$  in terms of the nodal unknowns  $\tilde{\mathbf{w}}_I$ , by solving the set of linear equations

$$\begin{aligned} \tilde{w}(0, 0) &= \Phi^T(0, 0)\mathbf{c} \\ \tilde{w}_{,s}(0, 0) &= \Phi^T_{,s}(0, 0)\mathbf{c} \\ \tilde{w}_{,t}(0, 0) &= \Phi^T_{,t}(0, 0)\mathbf{c} \\ &\vdots \\ \tilde{w}(1, 0) &= \Phi^T(1, 0)\mathbf{c} \\ \tilde{w}_{,s}(1, 0) &= \Phi^T_{,s}(1, 0)\mathbf{c} \\ \tilde{w}_{,t}(1, 0) &= \Phi^T_{,t}(1, 0)\mathbf{c} \\ &\vdots \\ \tilde{w}(1, 1) &= \Phi^T(1, 1)\mathbf{c} \\ \tilde{w}_{,s}(1, 1) &= \Phi^T_{,s}(1, 1)\mathbf{c} \\ \tilde{w}_{,t}(1, 1) &= \Phi^T_{,t}(1, 1)\mathbf{c} \\ &\vdots \end{aligned}$$

Denote the resulting coefficient matrix  $\tilde{\mathbf{A}}$ , then

$$\mathbf{c} = \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{w}}_I \tag{2.18}$$

and define global partition polynomials in the parent domain by

$$\tilde{w} = \begin{bmatrix} \tilde{\psi}_1^T & \tilde{\psi}_2^T & \tilde{\psi}_3^T \end{bmatrix} \tilde{\mathbf{w}}_I, \tag{2.19}$$

$$\tilde{w}(s, t) = \Phi^T \tilde{\mathbf{A}}^{-1} \tilde{\mathbf{w}}_I. \tag{2.20}$$

The  $\tilde{\psi}_i$  can be determined by comparing Eqs. (2.19) and (2.20). Since the interpolation is in the parent domain, we have to transform the derivatives back to the geometric domain, so the results are

$$w(s, t) = \Lambda^{-1} \Phi^T \tilde{\mathbf{A}}^{-1} \tilde{\mathbf{w}}_I = \Lambda^{-1} \Phi^T \tilde{\mathbf{A}}^{-1} \Lambda \mathbf{w}_I = \Lambda^{-1} \begin{bmatrix} \tilde{\psi}_1^T & \tilde{\psi}_2^T & \tilde{\psi}_3^T \end{bmatrix} \Lambda \mathbf{w}_I. \tag{2.21}$$

The desired nodal global partition polynomials are determined from Eq. (2.21).

The reason for the particular choice of unit triangle is that the linear system Eq. (2.18) is singular for other common choices of unit triangle, and this choice is reasonably well-conditioned.

The parametric approach requires all the derivatives of a given order be DOF's at the nodes so that the transformation to the parent domain and back can succeed. This means that there is a restriction on the number of DOF per node: *all derivatives of a given order must be present so that the transformations of the derivatives in  $\Lambda$  work properly*. In particular, the T12P3I(4/3) element cannot be implemented in the parametric approach because of this restriction. Nonetheless, the advantages of the parametric approach are that the matrix  $\tilde{\mathbf{A}}$  is inverted once and the global partition polynomials in the parent domain can be determined analytically, and hence it is much more efficient. Second, the parent domain is chosen so  $\tilde{\mathbf{A}}$  is invertible. Since any non-degenerate triangle has an invertible map to the parent triangle, this method is guaranteed to work, regardless of element shape or orientation, at least in determining the global partition functions.

### 2.3. Regularity requirement on meshes

As discussed in Part II of this work [6], there is a requirement on the aspect ratio and gradation of the triangles that can be in a mesh suitable for RKEM computations. The issue lies in the necessity of the RKEM shape functions having enough overlap at every point to ensure a partition of unity, but not to overlap too much and lose the Kronecker- $\delta$  properties. This is illustrated in Fig. 3. In the triangular meshes considered in this paper, the requirements translate into the radius of support for node  $i$  ( $\rho_i$ ) satisfying

$$\frac{1}{2} \max_{j \in A_i} d_{ij} \leq \rho_i \leq \min_{j \in A_i} d_{ij}, \tag{2.22}$$

where  $A_i$  is the set of all node indices sharing an edge with node  $i$ , and  $d_{ij}$  is the distance from node  $i$  to node  $j$ . Larger values of  $\rho_i$  tend to work better. All examples in the work used

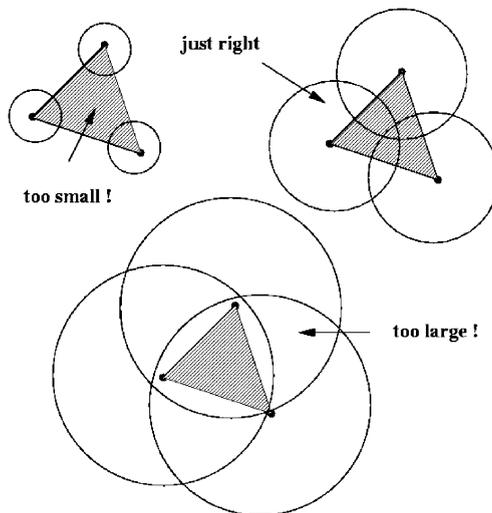


Fig. 3. How to choose support size?

$$\rho_i = 0.95 \min_{j \in \mathcal{A}_i} d_{ij}.$$

We call the meshes satisfying the condition (2.22) *quasi-uniform meshes*.

### 3. Element I: the T9P2I1 triangle

The first triangle element constructed is the T9P2I1 element (see Fig. 4). In Fig. 4, the solid circle at each nodal point represents the degree of freedom corresponding to the unknown function at the node, and the hollow circle at each nodal point represents the first derivatives of the unknown function at the nodal point. Therefore, each nodal point has 3 degrees of freedom, the element has 9 degrees of freedom (T9), it can reproduce complete second order polynomials (P2), and it interpolates an unknown function up to the first order derivatives (I1).

The parametric approach outlined in Section 2.2 will be applied to generate the global partition polynomials. Further, this construction is minimal in the sense that it only requires 3 degrees of freedom at each nodal point, displacement and first derivatives, to reproduce the complete quadratic polynomials. As mentioned before, the smoothness or continuity of the global RKEM shape function is determined by the continuity of the window function in the reproducing kernel. In most examples presented in this paper, a fifth order spline is used as the window function of the meshfree kernel. Therefore, the element is a global  $C^4$  compatible element.

#### 3.1. Global partition polynomials

This element has 9 degrees of freedom, so we construct the global partition polynomials by using the following parametric variables:

$$\mathbf{c}^T := [c_1 \ c_2 \ c_3 \ c_4 \ c_5 \ c_6 \ c_7 \ c_8 \ c_9],$$

$$\Phi^T(s, t) := [1 \ s \ t \ s^2 \ st \ t^2 \ s^3 \ s^2t + st^2 \ t^3].$$

Note that a complete set of cubic monomials have cardinality 10, but we only have nine unknowns, so two terms of the cubic monomials have been summed in a symmetric way. Quadratic polynomials are

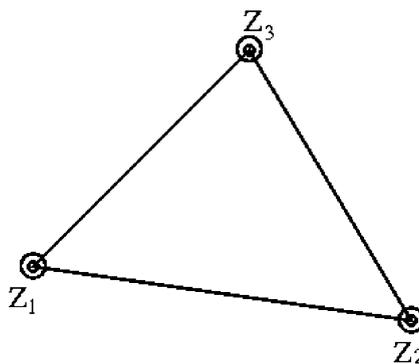


Fig. 4. The 9 degrees of freedom triangle: T9P2I1.

reproduced. The linear system Eq. (2.18) is a set of nine linear equations. The transformation matrix Eq. (2.16) now has diagonal block matrices that are  $3 \times 3$

$$\lambda_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & x_{,s} & x_{,t} \\ 0 & y_{,s} & y_{,t} \end{bmatrix}.$$

The final results for the parent domain global partition polynomials are:

$$\tilde{\psi}_1^{(00)} = 2s^3 - 3s^2 + 1, \quad (3.23)$$

$$\tilde{\psi}_1^{(10)} = s^3 - 2s^2 + s, \quad (3.24)$$

$$\tilde{\psi}_1^{(01)} = \frac{1}{2}(st^2 + s^2t - t^2 - 3st) + t, \quad (3.25)$$

$$\tilde{\psi}_2^{(00)} = 2t^3 - 3t^2 - 2s^3 + 3s^2, \quad (3.26)$$

$$\tilde{\psi}_2^{(10)} = \frac{1}{2}(t^2 + st - st^2 - s^2t) + s^3 - s^2, \quad (3.27)$$

$$\tilde{\psi}_2^{(01)} = t^3 - \frac{1}{2}(st^2 + s^2t + 3t^2 - 3st), \quad (3.28)$$

$$\tilde{\psi}_3^{(00)} = 3t^2 - 2t^3, \quad (3.29)$$

$$\tilde{\psi}_3^{(10)} = \frac{1}{2}(st^2 + s^2t - t^2 - st), \quad (3.30)$$

$$\tilde{\psi}_3^{(01)} = t^3 - t^2. \quad (3.31)$$

### 3.2. Global RKEM shape functions

Using nodal integration to integrate meshfree kernel function (2.1), the global RKEM interpolation field is constructed based on the following formula,

$$\mathcal{I}^h f(\mathbf{x}) = \mathbf{A} \left[ \sum_{e \in \mathcal{A}_E} \mathcal{K}_\rho(\mathbf{x} - \mathbf{x}_{e,j}; \mathbf{x}) \Delta V_{e,j} \left( \sum_{i \in \mathcal{A}_e} \psi_{e,i}(\mathbf{x}) f(\mathbf{x}_{e,i}) \right) \right], \quad (3.32)$$

where  $\Delta V_{e,j}$  is the nodal integration weight, which can be easily assigned for each nodal point based on *Lobatto quadrature rule* (see [4,6]).

The shapes and the profiles of the global RKEM shape functions of T9P2I1 triangle element is displayed in Fig. 5. It is important to note that each plot is scaled for visibility, thus relative magnitudes cannot be determined from the plots. The shape of the global interpolation function depends upon the mesh and node, whether it is on the boundary or it is in the interior of the domain. These shape functions plotted in Fig. 5 are computed for the middle node of the mesh shown in Fig. 12.

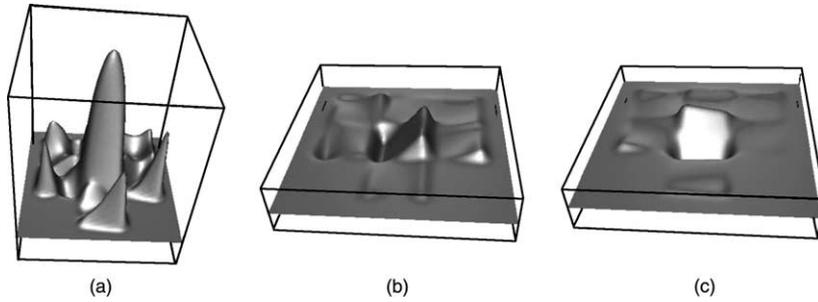


Fig. 5. The global shape functions of T9P2I1 element: (a)  $\Psi_I^{(00)}$ , (b)  $\Psi_I^{(10)}$ , (c)  $\Psi_I^{(01)}$ .

**4. Element II: the T12P3I(4/3) triangle element**

In the second example, we use the direct approach to construct the so-called T12P3I(4/3) element (see Fig. 6). In this element, each nodal point has 4 degrees of freedom: a nodal value of a unknown scalar function,  $w_I$ , represented by the solid circle, the nodal values of the two first derivatives of the unknown function,  $\frac{\partial w}{\partial x}|_I$  and  $\frac{\partial w}{\partial y}|_I$ , which are represented by the hollow circle, and the nodal value of the second order mixed derivative  $\frac{\partial^2 w}{\partial x \partial y}|_I$ , which is represented by a cross at the nodal point in Fig. 6.

Therefore, there are 12 degrees of freedom in this element (T12); it can reproduce complete third order polynomials (P3); since the interpolation order is greater than one, and is below two, we denote the interpolation order as  $I(4/3)$ .

*4.1. Global partition polynomials*

This element has 12 degrees of freedom, so we define the associated global partition polynomials by choosing the following vectors:

$$\mathbf{c}^T := [c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}], \tag{4.33}$$

$$\Phi^T(x, y) := [1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3, x^2(x^2 + xy + y^2), y^2(x^2 + xy + y^2)], \tag{4.34}$$

so that

$$w(x, y) = \Phi^T(x, y)\mathbf{c} = \Psi^T(x, y)\mathbf{w}_I. \tag{4.35}$$

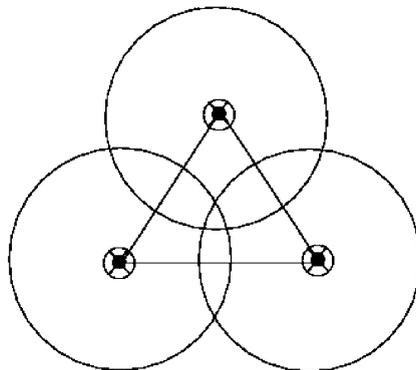


Fig. 6. The 12 degrees of freedom triangle element: T12P3I(4/3).

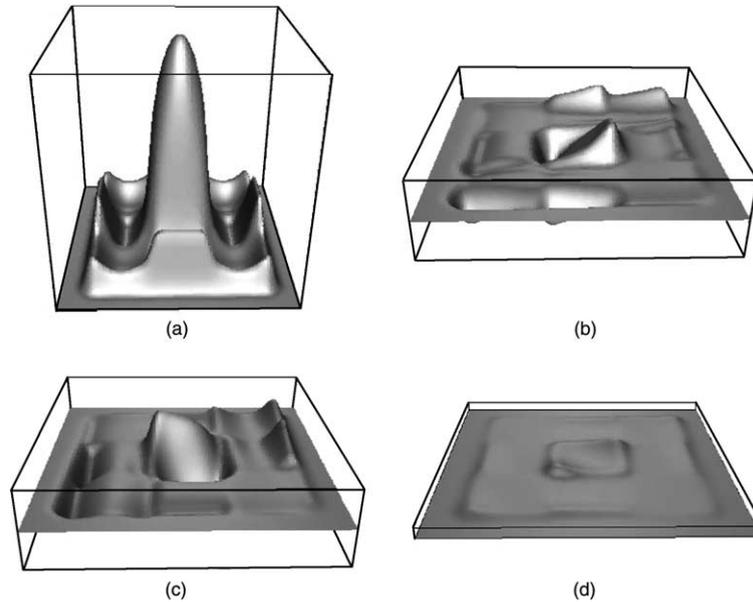


Fig. 7. The global shape functions of T12P3I(4/3) element: (a)  $\Psi_I^{(00)}$ , (b)  $\Psi_I^{(10)}$ , (c)  $\Psi_I^{(01)}$ , (d)  $\Psi_I^{(11)}$ .

The vectors  $\mathbf{c}$  and  $\mathbf{w}_I$  are related by

$$\mathbf{c} = \mathbf{A}^{-1}\mathbf{w}_I, \quad (4.36)$$

where the matrix  $\mathbf{A}$  is given in Eq. (4.37).

The T12P3I(4/3) element can reproduce complete cubic polynomials. The linear system Eq. (2.8) is a set of 12 equations that may be too cumbersome to invert analytically, so they are inverted numerically in this work.

#### 4.2. Global RKEM shape functions

In this section we provide plots of the global RKEM shape functions and their derivatives. There are not explicit expressions for the global partition polynomials, since they are determined by numerically solving a linear system at each evaluation point. The same caveats as in Section 3.2 apply. These shape functions were all computed for the middle node of the mesh in Fig. 12 (Fig. 7).

### 5. Element III: the T18P4I2 triangle

The third example is the so-called T18P4I2 element. It is an 18 degree of freedom triangle (T18) that can reproduce complete fourth order polynomials (P4) and interpolates the derivatives of an unknown function up to the second order (I2). At each nodal point, it has 6 degrees of freedom: nodal function value, first, and second derivatives (see Fig. 8).

Again, the parametric methodology outlined in Section 2 is applied to generate RKEM shape functions that globally interpolate quartic polynomials. In our construction, a fifth

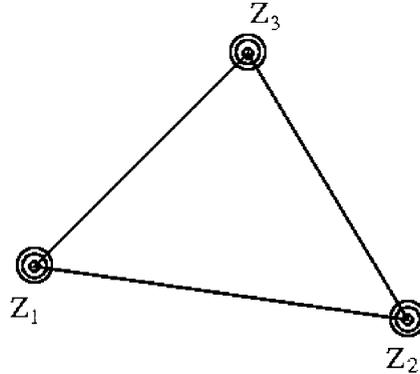


Fig. 8. The 18 degrees of freedom triangle: T12P4I2 element.

$$\mathbf{A}^T = \begin{pmatrix}
 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 & x_1^3 & x_1^2y_1 & x_1y_1^2 & y_1^3 & x_1^2(x_1^2 + x_1y_1 + y_1^2) & y_1^2(x_1^2 + x_1y_1 + y_1^2) \\
 0 & 1 & 0 & 2x_1 & y_1 & 0 & 3x_1^2 & 2x_1y_1 & y_1^2 & 0 & x_1(4x_1^2 + 3x_1y_1 + 2y_1^2) & y_1^2(2x_1 + y_1) \\
 0 & 0 & 1 & 0 & x_1 & 2y_1 & 0 & x_1^2 & 2x_1y_1 & 3y_1^2 & x_1^2(x_1 + 2y_1) & y_1(4y_1^2 + 3x_1y_1 + 2x_1^2) \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2x_1 & 2y_1 & 0 & 3x_1^2 + 4x_1y_1 & 3y_1^2 + 4x_1y_1 \\
 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 & x_2^3 & x_2^2y_2 & x_2y_2^2 & y_2^3 & x_2^2(x_2^2 + x_2y_2 + y_2^2) & y_2^2(x_2^2 + x_2y_2 + y_2^2) \\
 0 & 1 & 0 & 2x_2 & y_2 & 0 & 3x_2^2 & 2x_2y_2 & y_2^2 & 0 & x_2(4x_2^2 + 3x_2y_2 + 2y_2^2) & y_2^2(2x_2 + y_2) \\
 0 & 0 & 1 & 0 & x_2 & 2y_2 & 0 & x_2^2 & 2x_2y_2 & 3y_2^2 & x_2^2(x_2 + 2y_2) & y_2(4y_2^2 + 3x_2y_2 + 2x_2^2) \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2x_2 & 2y_2 & 0 & 3x_2^2 + 4x_2y_2 & 3y_2^2 + 4x_2y_2 \\
 1 & x_3 & y_3 & x_3^2 & x_3y_3 & y_3^2 & x_3^3 & x_3^2y_3 & x_3y_3^2 & y_3^3 & x_3^2(x_3^2 + x_3y_3 + y_3^2) & y_3^2(x_3^2 + x_3y_3 + y_3^2) \\
 0 & 1 & 0 & 2x_3 & y_3 & 0 & 3x_3^2 & 2x_3y_3 & y_3^2 & 0 & x_3(4x_3^2 + 3x_3y_3 + 2y_3^2) & y_3^2(2x_3 + y_3) \\
 0 & 0 & 1 & 0 & x_3 & 2y_3 & 0 & x_3^2 & 2x_3y_3 & 3y_3^2 & x_3^2(x_3 + 2y_3) & y_3(4y_3^2 + 3x_3y_3 + 2x_3^2) \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2x_3 & 2y_3 & 0 & 3x_3^2 + 4x_3y_3 & 3y_3^2 + 4x_3y_3
 \end{pmatrix} \tag{5.37}$$

order spline is used as the window function. Therefore, the construction example shown in this paper is actually global  $C^4$  in terms of smoothness. To achieve all these properties, 18 is the minimal number of degrees of freedom that one needs to construct such an interpolant.

### 5.1. Global partition polynomials

The element has 18 degrees of freedom,

$$\mathbf{c}^\dagger := [c_1 \ c_2 \ c_3 \ c_4 \ c_5 \ c_6 \ c_7 \ c_8 \ c_9 \ c_{10} \ c_{11} \ c_{12} \ c_{13} \ c_{14} \ c_{15} \ c_{16} \ c_{17} \ c_{18}],$$

$$\Phi^\dagger(s, t) := [1 \ s \ t \ s^2 \ st \ t^2 \ \dots \ s^4 \ s^3t \ s^2t^2 \ st^3 \ t^4 \ s^4(s+t) \ s^2t^2(s+t) \ t^4(s+t)],$$

where  $\dots$  stand for the complete cubic terms. Again, note the symmetric combinations of quintic terms to achieve 18 terms.

The linear system Eq. (2.18) now is a set of 18 linear equations. The transformation matrix Eq. (2.16) now has diagonal block matrices that are  $6 \times 6$

$$\lambda_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_{,s} & x_{,t} & 0 & 0 & 0 \\ 0 & y_{,s} & y_{,t} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{,s}^2 & x_{,t}^2 & 2x_{,s}x_{,t} \\ 0 & 0 & 0 & y_{,s}^2 & y_{,t}^2 & 2y_{,s}y_{,t} \\ 0 & 0 & 0 & x_{,s}y_{,s} & x_{,t}y_{,t} & x_{,s}y_{,t} + x_{,t}y_{,s} \end{bmatrix}.$$

The final results for the parent domain global partition polynomials are:

$$\tilde{\psi}_1^{(00)} = -6st^3 + 3t^3 - 6s^2t^2 + 3st^2 + 3s^2t^2(t+s) - 6s^4(t+s) + 12s^3t - 6s^2t + 15s^4 - 10s^3 + 1, \quad (5.38)$$

$$\tilde{\psi}_1^{(10)} = -3st^3 + \frac{3t^3}{2} - 3s^2t^2 + \frac{3st^2}{2} + \frac{3s^2t^2(t+s)}{2} - 3s^4(t+s) + 6s^3t - 3s^2t + 8s^4 - 6s^3 + s, \quad (5.39)$$

$$\tilde{\psi}_1^{(01)} = 3st^3 - \frac{3t^3}{2} + 3s^2t^2 - \frac{3st^2}{2} - \frac{3s^2t^2(t+s)}{2} + 2s^3t - 3s^2t + t, \quad (5.40)$$

$$\tilde{\psi}_1^{(20)} = -\frac{st^3}{2} + \frac{t^3}{4} - \frac{s^2t^2}{2} + \frac{st^2}{4} + \frac{s^2t^2(t+s)}{4} - \frac{s^4(t+s)}{2} + s^3t - \frac{s^2t}{2} + \frac{3s^4}{2} - \frac{3s^3}{2} + \frac{s^2}{2}, \quad (5.41)$$

$$\tilde{\psi}_1^{(02)} = \frac{st^3}{2} - \frac{t^3}{4} + s^2t^2 - \frac{5st^2}{4} + \frac{t^2}{2} - \frac{s^2t^2(t+s)}{4}, \quad (5.42)$$

$$\tilde{\psi}_1^{(11)} = st^3 - \frac{t^3}{2} + s^2t^2 - \frac{st^2}{2} - \frac{s^2t^2(t+s)}{2} + s^3t - 2s^2t + st, \quad (5.43)$$

$$\tilde{\psi}_2^{(00)} = 21t^4 + 12st^3 - 22t^3 - 6s^2t^2 + 6st^2 - 6t^4(t+s) + 6s^4(t+s) - 12s^3t + 6s^2t - 15s^4 + 10s^3, \quad (5.44)$$

$$\tilde{\psi}_2^{(10)} = -4st^3 + t^3 - 9s^2t^2 + 6st^2 + 3s^2t^2(t+s) - 3s^4(t+s) + 6s^3t - 3s^2t + 7s^4 - 4s^3, \quad (5.45)$$

$$\tilde{\psi}_2^{(01)} = 11t^4 - 9t^3 - 9s^2t^2 + 6st^2 - 3t^4(t+s) + 3s^2t^2(t+s) - 2s^3t + 3s^2t, \quad (5.46)$$

$$\tilde{\psi}_2^{(20)} = st^3 - \frac{t^3}{2} + s^2t^2 - \frac{st^2}{2} - \frac{s^2t^2(t+s)}{2} + \frac{s^4(t+s)}{2} - s^3t + \frac{s^2t}{2} - s^4 + \frac{s^3}{2}, \quad (5.47)$$

$$\tilde{\psi}_2^{(02)} = 2t^4 - 2t^3 - 2s^2t^2 + 2st^2 - \frac{t^4(t+s)}{2} + \frac{s^2t^2(t+s)}{2}, \quad (5.48)$$

$$\tilde{\psi}_2^{(11)} = st^3 - t^3 - 2s^2t^2 + 2st^2 + s^3t - s^2t, \quad (5.49)$$

$$\tilde{\psi}_3^{(00)} = -21t^4 - 6st^3 + 19t^3 + 12s^2t^2 - 9st^2 + 6t^4(t+s) - 3s^2t^2(t+s), \quad (5.50)$$

$$\tilde{\psi}_3^{(10)} = st^3 + \frac{t^3}{2} + 6s^2t^2 - \frac{9st^2}{2} - \frac{3s^2t^2(t+s)}{2}, \quad (5.51)$$

$$\tilde{\psi}_3^{(01)} = 10t^4 + 3st^3 - \frac{17t^3}{2} - 6s^2t^2 + \frac{9st^2}{2} - 3t^4(t+s) + \frac{3s^2t^2(t+s)}{2}, \quad (5.52)$$

$$\tilde{\psi}_3^{(20)} = -\frac{st^3}{2} + \frac{t^3}{4} - \frac{s^2t^2}{2} + \frac{st^2}{4} + \frac{s^2t^2(t+s)}{4}, \tag{5.53}$$

$$\tilde{\psi}_3^{(02)} = -\frac{3t^4}{2} - \frac{st^3}{2} + \frac{5t^3}{4} + s^2t^2 - \frac{3st^2}{4} + \frac{t^4(t+s)}{2} - \frac{s^2t^2(t+s)}{4}, \tag{5.54}$$

$$\tilde{\psi}_3^{(11)} = -\frac{t^3}{2} - 2s^2t^2 + \frac{3st^2}{2} + \frac{s^2t^2(t+s)}{2}. \tag{5.55}$$

*5.2. Global shape functions*

The global RKEM shape functions are plotted in Fig. 9. The same caveats as in Section 3.2 apply. According to the proposed construction procedure, as the order of the interpolant increases, the support size of the global shape function remain the same, as long as the mesh remains the same. This is in contrast with the popular moving least square reproducing kernel interpolant (MLS) [3,5,8,9]. For the MLS interpolant, as the reproducing capacity increases, the support size of the kernel function increases as well.

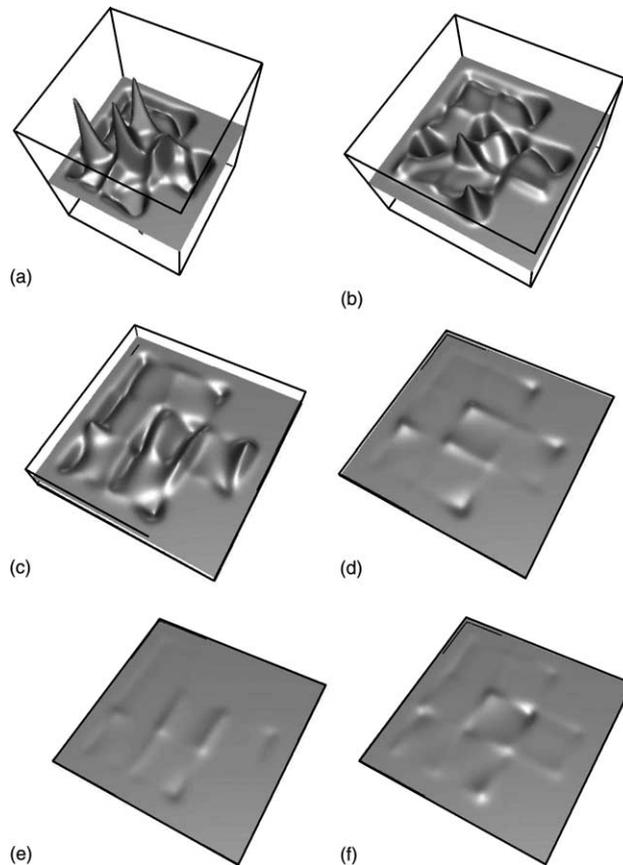


Fig. 9. The global shape functions of T18P4I2 element: (a)  $\Psi_i^{(00)}$ , (b)  $\Psi_i^{(10)}$ , (c)  $\Psi_i^{(01)}$ , (d)  $\Psi_i^{(20)}$ , (e)  $\Psi_i^{(02)}$ , (f)  $\Psi_i^{(11)}$ .

Consequently, the nature of the meshfree interpolation field becomes more and more non-local, whereas the nature of an RKEM interpolation field resembles to that of FEM interpolation field.

## 6. Implementation and computer programming

### 6.1. RKEM shape function pseudo code

The following is a pseudo-code sketch of the steps required to calculate an RKEM shape function. Of course, for a Galerkin solution, various derivatives are also required, they are computed via differentiation.

To compute the RKEM shape function for node  $I$ :

```

FOR  $i = 1$  TO number of elements
   $dv \leftarrow$  volume of element,
  FOR  $j = 1$  TO number of nodes per element
     $\phi \leftarrow$  compute window function
     $sum \leftarrow sum + \frac{1}{3} dv \phi$ 
  END FOR
END FOR

 $b \leftarrow \frac{1}{sum}$ 

FOR  $i = 1$  TO number of elements node  $I$  is a member
   $dv \leftarrow$  volume of element,
   $\psi \leftarrow$  nodal global partition polynomial
  FOR  $j = 1$  TO number of nodes per element
     $\phi \leftarrow$  compute window function
    FOR  $k = 1$  TO number of RKEM shape functions per node
       $\Psi(k) \leftarrow \Psi(k) + \frac{1}{3} \phi \frac{b}{\rho^2} dv \psi(k)$ 
    END FOR
  END FOR
END FOR

```

### 6.2. Quadrature

The computation of the stiffness and load vectors requires integration of the shape functions and products of shape functions, or their derivatives. The meshfree kernel functions are not polynomials, therefore the global RKEM shape functions are not polynomials, although they are able to reproduce polynomials based on their designed properties. As can be seen in the plots of the shape function in Section 5.2, the strong oscillation makes integration difficult.

Even though the RKEM shape functions are not polynomials, Gauss quadrature is used anyway. This may not be the best integration scheme, but it may be the most expedient. Higher order integration over triangles is presented in the work of Stroud and co-workers [2,11]. Currently, we are able to use 64 quadrature points per element to integrate T12P3I(4/3) elements, 576 quadrature points per element were used for the other two elements, though fewer may have been acceptable.

The main reason causing high order quadrature is probably the oscillatory nature of the RKEM shape functions. An efficient way to smooth the RKEM shape functions is to adopt the following partition of unity condition by using Gauss quadrature instead of using nodal integration to construct meshfree kernel functions,

$$\mathbf{A} \sum_{e \in A_E} \sum_{k \in A_{ge}} [\mathcal{K}_\rho(\mathbf{x} - \mathbf{x}_{kg}; \mathbf{x}) w_{kg}] \left( \sum_{i \in A_e} \psi_{e,i}(\mathbf{x}) \right) = 1, \quad (6.56)$$

where the set  $A_{ge}$  is the index set for the element quadrature points, and  $w_{kg}$  are the Gauss quadrature weights in an element. If Eq. (6.56) were used in construction meshfree reproducing kernel, the number of quadrature points might decrease to under 100 points per element while maintaining accurate computations.

### 6.3. Suggestions for computer coding and debugging

Here we share our experience in implementing the RKEM method. These are truly meant as suggestions and by no means do we assure the reader that our way is necessarily the best. We assume the reader is familiar with implementing the standard finite element method, as formulated in books such as [4] or [12].

First, the RKEM shape functions are truly global, though they do have compact support. Thus, assembly of the stiffness matrix involves more nodes than the ones connected to a particular element. The standard procedure of computing element stiffness matrices and assembling them is no longer as attractive. In the implementation used for this paper, the global stiffness matrix was assembled directly. Since the RKEM shape functions are compactly supported, there is a sparse structure to the stiffness matrix, but it is not as easy to determine apriori. For this reason, it is suggested that, initially, all loops should involve all nodes, shape functions that should not participate are guaranteed to be zero, and hence contribute nothing. Once an initial implementation works, a localization procedure should be implemented to reduce the runtime.

We found it convenient to have a data structure that is the dual of the connectivity matrix, i.e. we stored a list of element numbers that each node was a member of.

Some of us found our intuition lacking on what the shape functions should look like. This makes it harder to debug and makes unit testing of individual subprograms all the more important. Here are things that one can absolutely count on to give confidence in their code:

1. Verify that the global partition polynomials interpolate all the required terms. Be sure to test various element shapes and orientations.
2. Verify the RKEM interpolation properties at a random point in the domain. Verify that the Kronecker- $\delta$  properties are satisfied at all nodes. If not, check the radius of support.
3. Always verify the mesh satisfies the regularity conditions to ensure the Kronecker- $\delta$  properties.
4. Verify that the integration over the element actually integrates to the order expected. The very high order Gauss quadrature used involved hand entering many data points from published tables—an error-prone process.
5. Once these conditions are satisfied, one should have a working code. If not, start to look at other parts of the code.

## 7. Numerical examples

In this section, several numerical example problems are solved using the described method in order to assess its performance and robustness.

Since the Galerkin weak formulation of a Kirchhoff plate involves second derivatives, a best test for the proposed interpolant is solving various Kirchhoff plate problems for different geometries. Moreover, the boundary conditions of Kirchhoff plate problems involve interpolating boundary data of both the first

order derivative (slopes) and the second order derivative (curvatures), it provides a severe test to the newly proposed RKEM triangle interpolants. For more information on how to impose boundary conditions for finite element computation of thin plates, readers are referred to Hughes' finite element book [4, pp. 324–327].

We mainly consider three problems: the clamped rectangular Kirchhoff (thin) plate under uniform load, a simply supported rectangular thin plate under uniform load, and a clamped circular plate under uniform load.

The strong form of the equilibrium equation for a Kirchhoff plate is

$$\nabla^4 w = \frac{p}{D} \quad \forall (x, y) \in \Omega, \quad (7.57)$$

where  $\Omega$  is the domain of the problem. To solve this equation using a Galerkin method, we use the weak form. The general weak form reads as

$$\oint \hat{\mathbf{n}} \cdot \nabla(\nabla^2 w) \delta w \, dS - \oint (\nabla^2 w)(\nabla \delta w \cdot \hat{\mathbf{n}}) \, dS + \int (\nabla^2 w)(\nabla^2 \delta w) \, d\Omega = \int \frac{p}{D} \delta w \, d\Omega, \quad (7.58)$$

where  $\delta w$  is an arbitrary variation, and  $dS$  is differential boundary surface. In all the examples chosen,  $\frac{p}{D} = 1$ .

For a clamped plate, the essential boundary conditions are

$$\begin{aligned} w &= 0 \quad \forall (x, y) \in \partial\Omega, \\ w_{,n} &= 0 \quad \forall (x, y) \in \partial\Omega, \\ w_{,s} &= 0 \quad \forall (x, y) \in \partial\Omega. \end{aligned}$$

For a Galerkin solution, these boundary conditions require the arbitrary variation  $\delta w$  to also satisfy

$$\begin{aligned} \delta w &= 0 \quad \forall (x, y) \in \partial\Omega, \\ \delta w_{,n} &= 0 \quad \forall (x, y) \in \partial\Omega, \\ \delta w_{,s} &= 0 \quad \forall (x, y) \in \partial\Omega, \end{aligned}$$

hence, the two boundary integrals in Eq. (7.58) are zero. The weak form specific to a clamped plate then becomes

$$\int (\nabla^2 w)(\nabla^2 \delta w) \, d\Omega = \int \delta w \, d\Omega. \quad (7.59)$$

For a simply supported plate, the essential boundary conditions are

$$w = 0 \quad \text{on } \partial\Omega.$$

For a Galerkin solution, these boundary conditions require the arbitrary variation  $\delta w$  to also satisfy

$$\delta w = 0 \quad \text{on } \partial\Omega$$

hence, the first boundary integral in Eq. (7.58) is zero. Since there are no applied moments, this implies that  $\nabla^2 w = 0$ , so the second boundary integral in Eq. (7.58) is also zero. The weak form specific to this problem again becomes Eq. (7.59).

### 7.1. Clamped square plate

The first problem tested is a clamped unit square Kirchhoff plate, as depicted in Fig. 10. The deformed shape of the plate and the  $L_2$  convergence rate are depicted in Fig. 11.

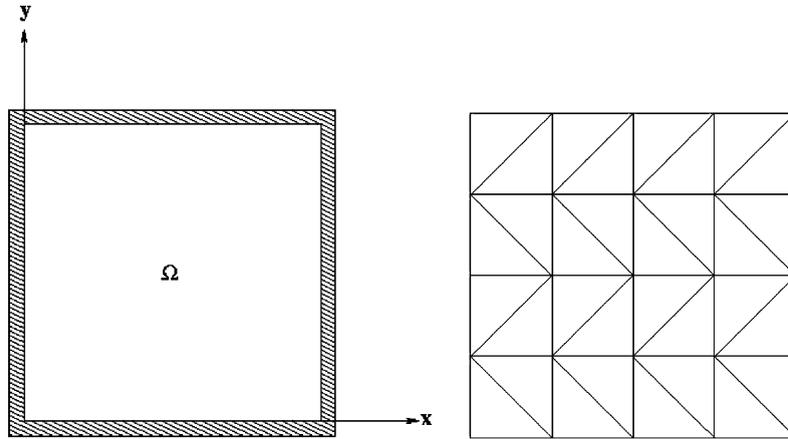


Fig. 10. Example 1: problem domain and triangle mesh.

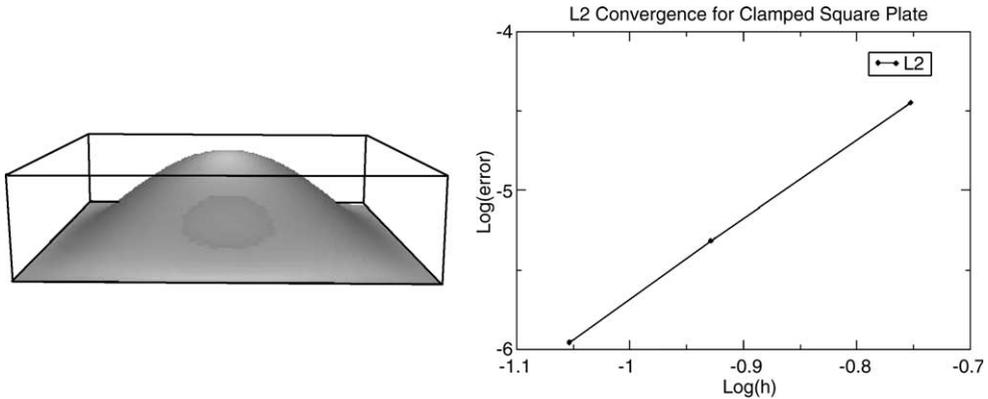


Fig. 11. Displaced shape and convergence of example 1.

The exact solution to the plate of dimensions  $a \times b$  is given in [13] as:

$$w(x,y) = \sum_{m=1}^M \sum_{n=1}^N w_{mn} \left(1 - \cos \frac{2m\pi x}{a}\right) \left(1 - \cos \frac{2n\pi y}{b}\right),$$

where the coefficients  $w_{mn}$  are computed using the method in [13].

### 7.2. Simply supported square plate

We next solve the simply supported unit square Kirchhoff plate, as depicted in Fig. 12.

The exact solution for a plate of dimension  $a \times b$  is given in [14] as:

$$w(x,y) = \frac{16p}{\pi^6 D} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{\sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b}}{mn[(m/a)^2 + (n/b)^2]^2} \quad (m, n = 1, 3, 5, \dots)$$

The  $L_2$ ,  $H_1$  and  $H_2$  errors were computed and are plotted along with the displaced shape in Fig. 13. The slopes of the lines as determined by regression are presented in Table 1.

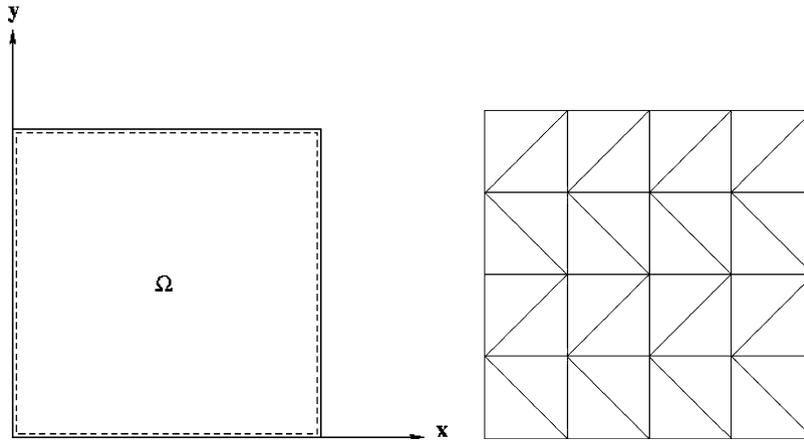


Fig. 12. Example 2: problem domain and triangle mesh.

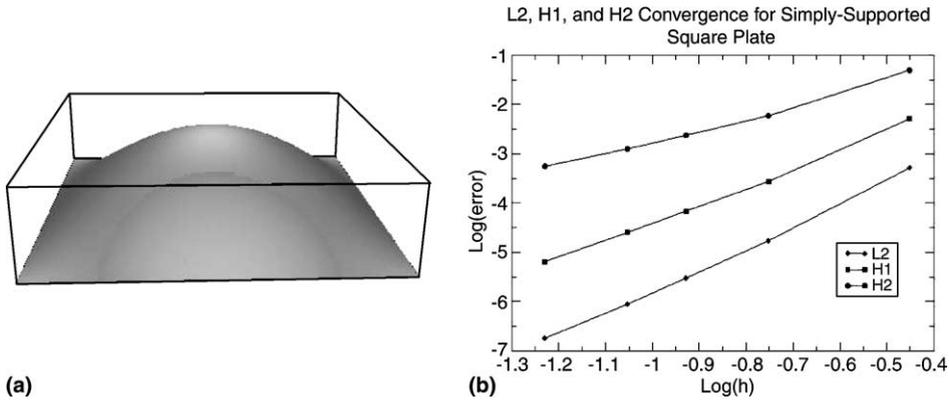


Fig. 13. Computation results for example 2: (a) deflection profile, and (b) convergence results.

Table 1  
Rates of convergence for simply supported square plate

| $L_2$  | $H_1$ | $H_2$  |
|--------|-------|--------|
| 4.4506 | 3.712 | 2.5041 |

### 7.3. Clamped circular plate

We now solve a clamped unit diameter circular Kirchhoff plate, as depicted in Fig. 14. The exact solution is given in [14] for a plate of radius  $a$  as

$$w(x, y) = \frac{P}{64D} (a^2 - x^2 - y^2)^2.$$

The deformed shape of the circular plate is juxtaposed with the convergence results depicted in Fig. 15.

The  $L_2$ ,  $H_1$  and  $H_2$  errors were computed and are plotted in Fig. 15. The slopes of the lines as determined by regression are presented in Table 2.

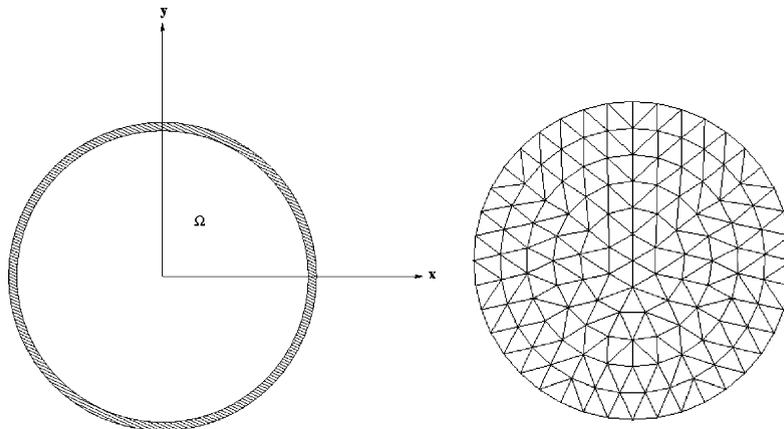


Fig. 14. Example 3: problem domain and triangle mesh.

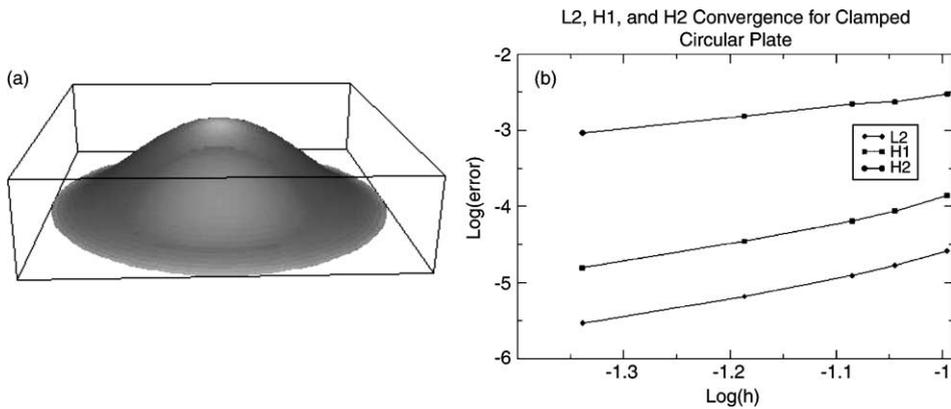


Fig. 15. Computation results for example 3: (a) deflection profile, and (b) convergence results.

Table 2  
Rates of convergence for clamped circular plate

| $L_2$  | $H_1$  | $H_2$  |
|--------|--------|--------|
| 2.6812 | 2.6495 | 1.4606 |

7.4. Simply supported triangular plate

Finally, we present the performance of the T12P3I(4/3) element in solving the unit height equilateral simply supported Kirchhoff plate with uniform loading. The problem geometry and mesh used are depicted in Fig. 16.

The exact solution for this problem with triangle height  $a$  is given in [1] as

$$w(x,y) = \frac{P}{64aD} \left[ x^3 - 3y^2x - a(x^2 + y^2) + \frac{4}{27}a^3 \right] \left( \frac{4}{9}a^2 - x^2 - y^2 \right). \tag{7.60}$$

The deformed shape of the triangular plate is juxtaposed with the convergence results depicted in Fig. 17.

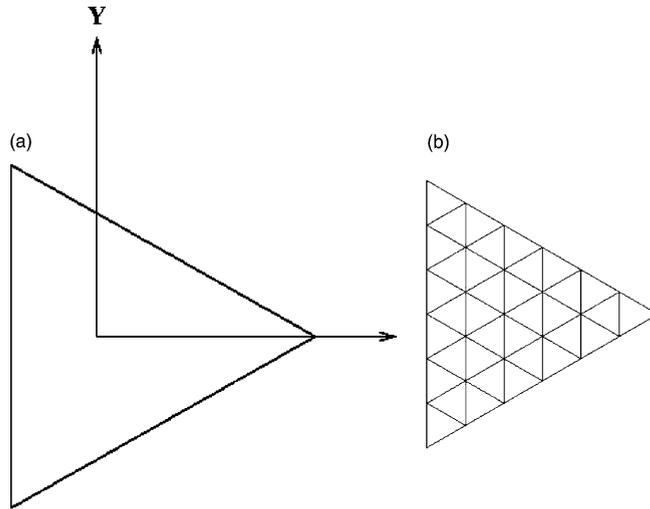


Fig. 16. Problem domain and mesh for simply supported triangular plate: (a) problem domain (b) 28 node.

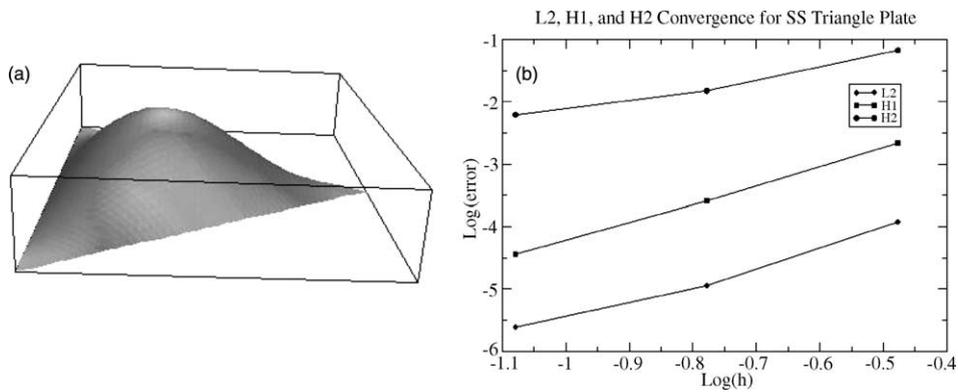


Fig. 17. Computation results for example 3: (a) deflection profile, and (b) convergence results.

Table 3  
Rates of convergence for clamped circular plate

| $L_2$  | $H_1$  | $H_2$  |
|--------|--------|--------|
| 2.8028 | 2.9498 | 1.7216 |

The  $L_2$ ,  $H_1$  and  $H_2$  errors were computed and are plotted in Fig. 17. The slopes of the lines as determined by regression are presented in Table 3.

### 8. Concluding remarks

In this paper, a globally conforming  $I^m/C^n/P^k$  triangle element interpolation hierarchy is constructed under the framework of RKEM.

Two systematic approaches are presented to construct the so-called global partition polynomials under various designs. The proposed interpolants have been used in numerical computations of several Kirchhoff plate problems. It has been shown that the newly constructed interpolants are very efficient and robust in the Galerkin type approximations that are involved with higher order derivatives in the weak forms or in the boundary conditions.

It may be observed that the higher order RKEM interpolants are very flexible in both construction and computation and they are suitable to many different meshes, structured or non-structured. As shown in Part III of this work [10], one can construct the optimal or minimal interpolation fields by utilizing incomplete interpolation sequence. Further study is needed to examine their performance in computations of shear deformable plate problems as well as linear and non-linear shell problems.

### Acknowledgements

The integration over triangles was aided by the use of the computer programs in STROUD.F90 assembled and maintained by John Burkardt.

This work is made possible by the support from NSF under grants CMS-0239130 to University of California (Berkeley), DMI-0115079 to Northwestern University.

### References

- [1] K. Bell, A refined triangular plate bending finite element, *Int. J. Numer. Methods Engrg.* 1 (1969) 101–122.
- [2] P.C. Hammer, O.J. Marlowe, A.H. Stroud, Numerical integration over simplexes and cones, *Math. Tables Other Aids Comp.* 10 (55) (1956) 130–137.
- [3] W. Han, X. Meng, Error analysis of the reproducing kernel particle method, *Comput. Methods Appl. Mech. Engrg.* 190 (2001) 6157–6181.
- [4] T.J.R. Hughes, *The Finite Element Method*, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1987.
- [5] S. Li, W.K. Liu, Moving least square reproducing kernel method (II) Fourier analysis, *Comput. Methods Appl. Mech. Engrg.* 139 (1996) 159–193.
- [6] S. Li, H. Lu, W. Han, W.K. Liu, D.C. Simkins, Reproducing kernel element. Part II: Global conforming  $I^m/C^n$  hierarchy, *Comput. Methods Appl. Mech. Engrg.*, in this issue.
- [7] W.K. Liu, W. Han, H. Lu, S. Li, J. Cao, Reproducing kernel element method. Part I: Theoretical formulation, *Comput. Methods Appl. Mech. Engrg.*, in this issue.
- [8] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, *Int. J. Numer. Meth. Fluids* 20 (1995) 1081–1106.
- [9] W.K. Liu, S. Li, T. Belytschko, Moving least square reproducing kernel method. Part I: Methodology and convergence, *Comput. Methods Appl. Mech. Engrg.* 143 (1997) 422–453.
- [10] H. Lu, S. Li, D.C. Simkins Jr., W.K. Liu, J. Cao, Reproducing kernel element method. Part III: Generalized enrichment and applications, *Comput. Methods Appl. Mech. Engrg.*, in this issue.
- [11] A.H. Stroud, Don Secrest, *Gaussian Quadrature Formulas*, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1966.
- [12] O.C. Zienkiewicz, R.L. Taylor, *The Finite Element Method*, vol. 1, fifth ed., Heinemann, Oxford, 2000.
- [13] R.L. Taylor, S. Govindjee, Solution of Clamped Rectangular Plate Problems, Technical Report: UCB/SEMM-2002/09, 2002.
- [14] A.C. Ugural, *Stresses in Plates and Shells*, second ed., McGraw-Hill, Boston, 1999.