

Reproducing Kernel Element Method for Galerkin Solution of Elastostatic Problems

Mario J. Juha ¹

Received: 24-feb-2012, Accepted: 21-sep-2012

Available online: 30-nov-2012

MSC: 65N30

Abstract

The Reproducing Kernel Element Method (RKEM) is a relatively new technique developed to have two distinguished features: arbitrary high order smoothness and arbitrary interpolation order of the shape functions. This paper provides a tutorial on the derivation and computational implementation of RKEM for Galerkin discretizations of linear elastostatic problems for one and two dimensional space. A key characteristic of RKEM is that it does not require mid-side nodes in the elements to increase the interpolatory power of its shape functions, and contrary to meshless methods, the same mesh used to construct the shape functions is used for integration of the stiffness matrix. Furthermore, some issues about the quadrature used for integration are discussed in this paper. It is hoped that this may attract the attention of mathematicians.

Key words: Elasticity, convergence, RKEM, continuity, Galerkin methods

¹ Doctor en Ingeniería Civil, mario.juha@uac.edu.co, Docente Investigador, Universidad Autónoma del Caribe

Método del elemento reproductor del núcleo para soluciones de problemas elasto-estáticos.

Resumen

El Método del Elemento Reproductor del Núcleo (RKEM) es una técnica relativamente nueva desarrollada para tener dos propiedades distinguidas: suavidad de orden arbitrario y funciones de interpolación de orden arbitrario. Este artículo provee un tutorial acerca de la deducción e implementación de RKEM en discretizaciones de Galerkin en problemas elasto-estáticos en una y dos dimensiones. Una característica clave de RKEM es que no requiere de nodos intermedios en los lados de los elementos para incrementar el poder de interpolación de sus funciones de forma, y contrario a los métodos sin malla, la misma malla usada para construir las funciones de forma es usada para la integración de la matriz de rigidez. Además, algunas cuestiones acerca de la cuadratura usada para integración son discutidas en el artículo. Se espera que esto pueda atraer la atención de matemáticos.

Palabras claves: Elasticidad, convergencia, RKEM, continuidad, método de Galerkin

1 Introduction

One of the most widely used computational methods in engineering today is the Finite Element Method (FEM). While FEM has a wide range of uses, it does have some limitations, as any approximation method does. One of the limitations is the ability to provide arbitrary smooth global interpolation in higher dimension [1]. One of the key characteristic of RKEM is that can be used in geometry representation as have been shown in [2]. This open a wonderful opportunity to introduce the method in bio-sciences. But first, we need to show that the method can be use to solve differential equations, as well as, geometry representation. The finite element method may not be used to solve problem that require higher order of continuity, because in one way or another, is always difficult to construct higher order interpolation functions [3] to calculate stresses and strains. Thus, such quantities are not very well represented within an element, and various strategies are used to smooth these results. For these reason, it would be advantageous to have a method to generate higher-order smooth interpolation. The aim of this paper is to provide

the details of the implementation of the RKEM, including its programming in 1D. The details of the method are first explicitly outlined for one-dimensional problems and next for a two-dimensional problem. The essential concepts will be described in the context of a one-dimensional and two-dimensional problems in elastostatic. Also, we want to show the mathematics involved in the derivation of the discrete system of equations using an RKEM interpolation in the Galerkin weakform. Its hopes that this attract the attention of mathematicians, physicist and engineers working on continuum mechanics.

2 Theoretical Background

2.1 Interpolation Using RKEM Shape Functions

An important application area for RKEM is the interpolation of a set of points. Curve fitting and interpolations are extremely related each other. According to Piegł [4], we can distinguish two types of fitting, *interpolations and approximation*. Methods like finite element and finite differences uses the concept of interpolation, on the other hand, most of the popular meshfree methods uses the concept of approximation. This difference has an enormous impact in the application of the methods. In this section we introduce the interpolation capability of the reproducing kernel element method.

2.1.1 Concept of Shape Function in RKEM A shape function is the name given to a collection of functions used to interpolate or approximate a data set. The successful of an interpolation/approximation depends of the shape functions chosen. In general we want a method capable of representing a function at some point just using the information in its vicinity. A number of ways to construct shape functions have been proposed and we can find many papers about them. The RKEM shape function is closely related to the *finite integral representation*, that according to Liu [5] is categorized as:

- Finite integral representation methods.

$$f(x) = \int_{x_1}^{x_2} f(\xi) \kappa(x - \xi) d\xi$$

where $\kappa(x - \xi)$ is known as the kernel, x , ξ , x_1 and x_2 are points over the real line.

Furthermore the RKEM shape function enjoys some distinguished features [6]:

1. The smoothness of the global basis functions solely determined by that of the kernel function.
2. The global basis functions of RKEM have the Kronecker delta property at the associated nodes, provided that some conditions on the support size of the kernel function are met.

To construct the RKEM shape functions we combine the so-called global partition polynomials [6] with compactly supported functions defined through kernel functions of special forms. A kernel function $\kappa(z; x)$ has the property that it is non-zero only when $\|z\|_2 < \rho$, where $z = x_0 - x$ and x_0 is the point where the kernel is centered. The positive number ρ represents the support size of the kernel function with respect to its first argument. According to [1], the kernel function have the general form presented in Eq. (1).

$$\kappa(z; x) := \frac{1}{\rho^d} w\left(\frac{z}{\rho}\right) b(x) \quad (1)$$

where d is the spatial dimension, x is the point at which the kernel is evaluated, w is compactly-supported smooth window function, and $b(x)$ is a normalization factor [7, 6]. The requirements on the window function are

$$\begin{cases} \text{supp}(w) \subset B_1 \\ w(x) > 0 & \text{for } \|x\|_2 < 1 \\ w \in C^l(\mathbb{R}^d) & \text{for some } l \geq 1 \end{cases} \quad (2)$$

where, $B_1 = \{x \in \mathbb{R}^d \mid \|x\|_2 \leq 1\}$ is the unit ball. Now, using the finite integral representation method together with the global partition polynomials and Eq. (1) and (2) the RKEM interpolant ($\mathcal{I}f$) is given by [6],

$$\mathcal{I}f(x) = \sum_{e \in \Lambda_E} \left[\int_{\Omega_e} \kappa(x - y; x) dy \left(\sum_{i \in \Lambda_e} \psi_{e,i}(x) f(x_{e,i}) \right) \right] \quad (3)$$

From a practical point of view we can cast Eq (3) into

$$\mathcal{I}f^h(x) = \sum_{I \in \Lambda_p} \Psi_I(x) f_I \quad (4)$$

where f_I correspond to nodal weights, Λ_p is the set of all the nodes in the domain and $\Psi_I(x)$ the RKEM shape function associated to node I and is given by

$$\Psi_I(x) := \sum_{k=1}^{\ell} \left(\sum_{j \in \Lambda_{e_k}} \frac{1}{\rho_{e_k,j}^d} w \left(\frac{x - x_{e_k,j}}{\rho_{e_k,j}} \right) \Delta V_{e_k,j} \right) b(x) \psi_{e_k,i_k}(x) \quad (5)$$

where ℓ is the number of elements sharing the node I , Λ_{e_k} is the set of all the nodes of element e_k and $b(x)$ is known as the normalizer and is equal to

$$b(x) := \left\{ \mathbf{A}_{e \in \Lambda_E} \left[\left(\sum_{j=1}^n \frac{1}{\rho_{e,j}^d} w \left(\frac{x - x_{e,j}}{\rho_{e,j}} \right) \Delta V_{e,j} \right) \right] \right\}^{-1} \quad (6)$$

where $\mathbf{A}_{e \in \Lambda_E}$ is the assembling operator [3]. Note that we have used nodal integration to discretize the integral in Eq. (3). In [1] is presented a methodology to construct the global partition polynomial for a triangular RKEM element and in [8] is presented a way to enrich the global partition polynomial with extra derivatives, both approximation have been used in this paper. Lu et al. [8] have presented a method of getting a better interpolation capability using the primary variable and its first derivative, as is shown in Eq. (7).

$$\mathcal{I}f^h(x) = \sum_I (\Psi_I(x)f_I + \Psi'_I(x)f'_I) \quad (7)$$

For the curve fitting case we can view the $f(x)$ like the interpolation, f_I like the actual point and we will need to calculate the f'_I . To understand how to come with the Eq. (4) we will present a brief example in one dimension and using linear elements.

Example. Consider the mesh show in Fig. 1, we assume that the domain is $\Omega = (0, 1)$. We want to interpolate a function using linear RKEM elements.

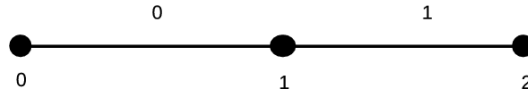


Figure 1: Spatial discretization with two elements and three nodes.

First we should expand Eq (3) in the following way:

$$u(x) = \left[\int_{\Omega_0} \frac{1}{\rho} \phi \left(\frac{y-x}{\rho} \right) dy \right] [\psi_{0,0}(x) \hat{u}_0 + \psi_{0,1}(x) \hat{u}_1] b_0(x) + \\ \left[\int_{\Omega_1} \frac{1}{\rho} \phi \left(\frac{y-x}{\rho} \right) dy \right] [\psi_{1,0}(x) \hat{u}_1 + \psi_{1,1}(x) \hat{u}_2] b_0(x)$$

where \hat{u}_i correspond to the value of the function to interpolate and evaluated at the node i . We can manipulate terms so that we can get the following representation of the same equation.

$$u(x) = \sum_{I=0}^2 \Psi_I \hat{u}_I$$

where,

$$\Psi_0(x) = \left[\int_{\Omega_0} \frac{1}{\rho} \phi \left(\frac{y-x}{\rho} \right) dy \right] [\psi_{0,0}(x) b_0(x)]$$

$$\Psi_1(x) = \left[\int_{\Omega_0} \frac{1}{\rho} \phi \left(\frac{y-x}{\rho} \right) dy \right] [\psi_{0,1}(x) b_0(x)] \\ + \left[\int_{\Omega_1} \frac{1}{\rho} \phi \left(\frac{y-x}{\rho} \right) dy \right] [\psi_{1,0}(x) b_0(x)]$$

$$\Psi_2(x) = \left[\int_{\Omega_1} \frac{1}{\rho} \phi \left(\frac{y-x}{\rho} \right) dy \right] [\psi_{1,1}(x) b_0(x)]$$

In Fig 2 we can see a pictorial definition of $\Psi_2(x)$. Note that in this case the function $\Psi_2(x)$ has a negative value. The normalizer $b_0(x)$ could be calculated in the following way:

$$b_0(x) = \left[\int_{\Omega_0} \frac{1}{\rho} \phi \left(\frac{y-x}{\rho} \right) dy + \int_{\Omega_1} \frac{1}{\rho} \phi \left(\frac{y-x}{\rho} \right) dy \right]^{-1}$$

Note that $b_0(x)$ is no more than the area below the window function evaluated at point x .

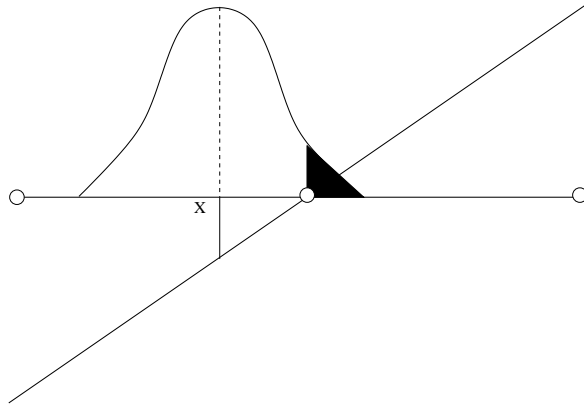
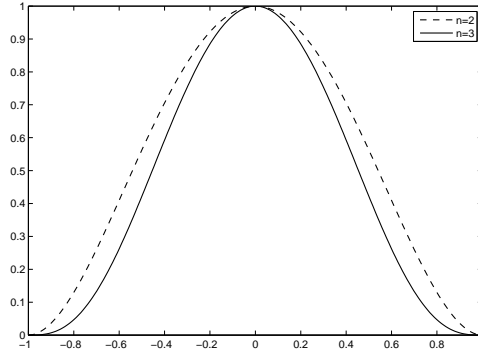


Figure 2: Window function and $\psi_{1,1}$ evaluated at point x .

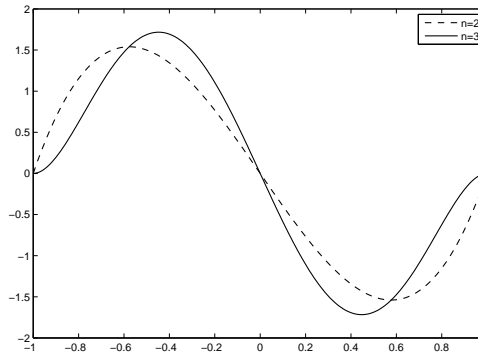
2.1.2 Window function In the definition of the RKEM shape functions we introduced the concept of window function. Now we will expand further this concept. A window function is simply a continuous function that has a value of zero outside its support. An example of window function is the *conical window function* shown below

$$w(x) = \begin{cases} (1 - x^2)^n & -1 \leq x \leq +1 \\ 0 & |x| > 1 \end{cases} \quad (8)$$

where n represent the degree of continuity. We say that the window function is of class C^n . In Fig. 3 is shown a plot of the window function and its first derivative.



(a) Window function.



(b) Window function first derivatives.

Figure 3: Window function and its derivative for $w(x) = (1 - x^2)^n$.

2.2 RKEM in One Dimension

This section present a very detailed procedure of the RKEM applied to the elastostatic differential equation in 1D. We will implement the L2P1I0 element [1] that was described briefly in the previous example. The characteristic of this element is that only reproduce linear polynomials.

2.2.1 Discrete Equations in 1D Consider the following one-dimensional problem on the domain $0 \leq x \leq 1$:

$$Eu_{,xx} + b = 0 \quad \text{in } \Omega = (0, 1) \quad (9)$$

where $u(x)$ is the displacement, E is the Young's modulus, and b is the body force per unit volume. The following specific boundary conditions are chosen:

$$Eu_{,x} = \bar{t} \quad (x = \Gamma_t) \quad (10)$$

$$u = \bar{u} \quad (x = \Gamma_u) \quad (11)$$

In a more mathematical terminology Eq. (10) is a Neumann boundary condition [3] and Eq. (11) is a Dirichlet boundary condition. To obtain the discrete equations it is first necessary to use a weak form of the equilibrium equation and boundary conditions. The following weak form is used:

$$\int_0^1 u_{,x} Eu_{,x} dx - \int_0^1 u b dx - u\bar{t} \Big|_{\Gamma_t} = 0 \quad (12)$$

then the equilibrium Eq. (9) and boundary conditions (10) and (11) are satisfied. Note that due the RKEM interpolation has the Kronecker delta property we do not need to use a constraint Galerkin weak form, that is, we can impose condition (11) directly like in finite element analysis. In order to obtain the discrete equations from the weak form, the approximate solution $u(x)$ is constructed according to what was presented in previous sections. For the displacement u^h , we have:

$$u^h(x) = \sum_I^n \Psi_I u_I \quad (13)$$

where n is the number of nodes used in the support domain of the point at x for constructing the RKEM shape function $\Psi_I(x)$ and u_I are nodal weights. By using Eq (13) and taking derivatives respect to x we obtain,

$$u_{,x}^h(x) = \sum_I^n \Psi_{I,x} u_I \quad (14)$$

Now, replacing Eq (13), (14) in (12) yields to the following discrete system of equations:

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad (15)$$

where

$$\mathbf{K}_{IJ} = \int_0^1 \Psi_{I,x} E \Psi_{J,x} dx \quad (16)$$

$$\mathbf{f} = \int_0^1 \Psi_I b dx + \Psi_I \bar{t} \Big|_{\Gamma_t} \quad (17)$$

To get a solution of (15) we need to impose the Dirichlet boundary conditions to the system in order to avoid rigid body motion. There is a variety of ways to do this. The two most popular methods are the penalty approach and static condensation [9]. In this work we used the second approach. Due that the RKEM interpolation has the Kronecker delta property, we can set the boundary conditions directly. These equations are assembled by integrating over the domain of the problem using Gauss quadrature. To evaluate the integrals in (16) and (17), it is necessary to define integration cells over the domain of the problem, in RKEM we have two options, we can use directly the RKEM mesh or to use an integration background mesh. As we explained in previous sections, the RKEM shape functions are piecewise rational shape functions, therefore, we need to use many Gauss point to evaluate the integrals such that we can obtain a well conditioned, non singular system of equations (15). To solve this system of equation we used a factorization process that lead to the Gauss elimination method. Once the cells and corresponding quadrature points are defined, the discrete equations are assembled by looping over each quadrature point, x_Q . In the assembly of the discrete equation (16), at each quadrature point the nodes whose domain of influence contain the point x_Q are first determined. These are referred to as support nodes for the quadrature point. The shape function (5) and its shape function derivatives corresponding to each of these support nodes are calculated and assembled. A similar procedure is used at the Neuman boundary conditions to evaluate the applied traction contribution to the force vector (17).

2.3 RKEM in Two Dimension

This section present a very detailed procedure of the RKEM applied to the elastostatic differential equation in 2D. We will implement the T9P2I1 element [1], whose main characteristic is that uses three corner nodes to interpolate

a second order polynomial and its first derivative. The essential concepts outlined in previous sections do not change for the RKEM method in two or three dimensions. The scalar variable x simply becomes a vector $\mathbf{x}^T = [x \ y]$ in all the equations stated previously. There are, however, some differences and extensions which are not so trivial and consequently need further explanation. These include the use of radial window functions, the procedure to evaluate the coverage, calculating of the derivatives of shape functions and normalizer, assembly of discrete system of equations.

2.3.1 Radial Window Function In two dimensions a node's domain of influence covers an area in the domain. The choice of the shape of this domain is arbitrary. However, circular domain or square domains have been used most frequently. In this work we have used the circular domains for all the computation. In RKEM the radius of the circle can not be arbitrary, it need to be a function of the mesh [10]. The scheme for the circular domain is shown in Fig. 4.

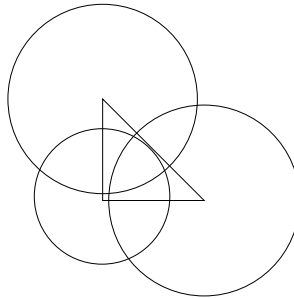


Figure 4: Domain of influence in two dimension using circular domains. The triangle corresponds to the domain of the problem and the vertexes are the nodes.

The window function definition in 2D used in this work will be:

$$w(\mathbf{x} - \mathbf{x}_I) = w(r_I) \quad (18)$$

where $w(r_I)$ is given by Eq. (8) with x replaced by r_I ; and r_I are given by

$$r_I = \frac{\|x - x_I\|}{\rho_I} \quad (19)$$

The requirement is that the argument in the normalizer be non-zero everywhere in the domain, and thus invertible, which is necessary to compute the shape function.

2.4 Quadrature

The computation of the stiffness and load vectors requires integration of the shape functions and products of derivatives of shape functions. The RKEM functions are piecewise rational functions that are able to reproduce polynomials. As can be seen in the plots of the shape function in Fig. 5 the functions are globally continuous and smooth. In Fig. 6 we can see typical entries to integrate in the weak form. Note that due to the strong oscillation this kind of functions are difficult to integrate exactly. Even though the RKEM shape functions are not polynomials, Gauss quadrature is used anyway. We used the work presented in [11] to generate Gauss points for a triangular domain. Currently, 64 quadrature points per element are used to integrate the T9P2I1 element. Still, the problem of finding an optimal quadrature rule to integrate exactly the weak form is an open research area.

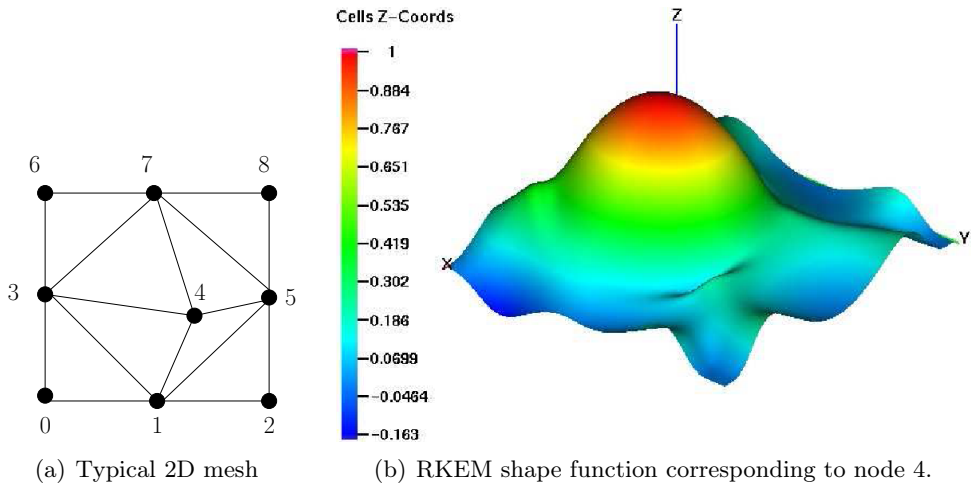


Figure 5: RKEM mesh and and global shape function for node 4.

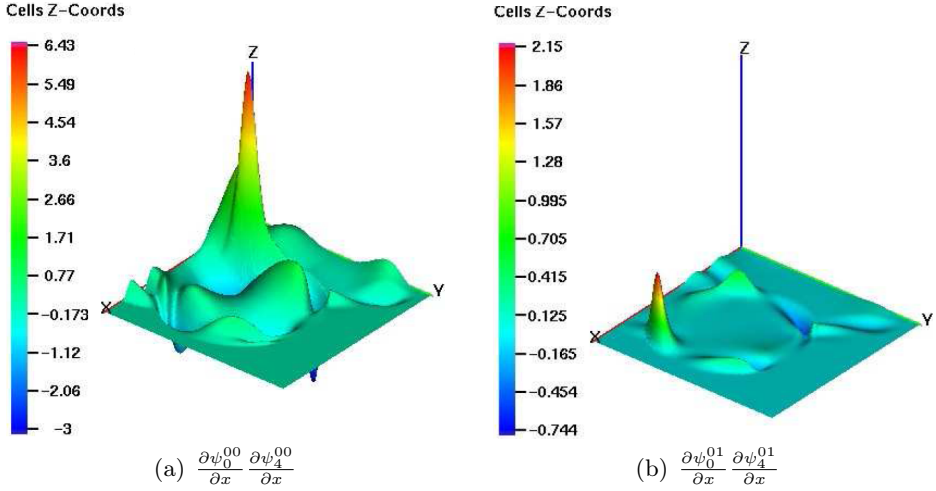


Figure 6: Typical elements to integrate in the stiffness matrix. We have used the T9P2I1 element.

2.5 Discrete Equations in 2D

The discrete equations in two dimensions are very similar to those of the one-dimensional formulation. Line integrals in x become surface integrals in x and y and there are some other subtle differences.

2.5.1 Formulation The partial differential equation and boundary condition for a 2D solid mechanics problem can be written in the form:

$$L^T \sigma + b = 0 \quad \text{in } \Omega \quad (20)$$

$$u = \bar{u} \quad \text{in } \Gamma_u \quad (21)$$

$$\sigma n = \bar{t} \quad \text{in } \Gamma_t \quad (22)$$

where σ is the stress tensor for 2D solid and L is the differential operator over σ . The other terms were explained in the previous section. The Galerkin weak form for the problem stated by Eq. (20), (21) and (22) can be given by

$$\int_{\Omega} \delta (Lu)^T D (Lu) d\Omega - \int_{\Omega} \delta u^T b d\Omega - \int_{\Gamma_t} \delta u^T \bar{t} d\Gamma = 0 \quad (23)$$

For the displacement vector we have

$$u^h = \begin{Bmatrix} u \\ v \end{Bmatrix}^h = \sum_I^n \begin{bmatrix} \psi_I^{00} & \psi_I^{10} & \psi_I^{01} & 0 & 0 & 0 \\ 0 & 0 & 0 & \psi_I^{00} & \psi_I^{10} & \psi_I^{01} \end{bmatrix} \begin{Bmatrix} u_I \\ u'_I \\ u''_I \\ v_I \\ v'_I \\ v''_I \end{Bmatrix} = \sum_I^n \Psi_I u_I \quad (24)$$

where Ψ_I is the matrix of shape functions. By using Eq. (24), the product of Lu^h (which gives the strains) becomes

$$\begin{aligned} Lu^h &= L \sum_I^n \Psi_I u_I = \sum_I^n \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \psi_I^{00} & \psi_I^{10} & \psi_I^{01} & 0 & 0 & 0 \\ 0 & 0 & 0 & \psi_I^{00} & \psi_I^{10} & \psi_I^{01} \end{bmatrix} u_I \\ &= \sum_I^n \begin{bmatrix} \psi_{I,x}^{00} & \psi_{I,x}^{10} & \psi_{I,x}^{01} & 0 & 0 & 0 \\ 0 & 0 & 0 & \psi_{I,y}^{00} & \psi_{I,y}^{10} & \psi_{I,y}^{01} \\ \psi_{I,y}^{00} & \psi_{I,y}^{10} & \psi_{I,y}^{01} & \psi_{I,x}^{00} & \psi_{I,x}^{10} & \psi_{I,x}^{01} \end{bmatrix} u_I = \sum_I^n B_I u_I \end{aligned} \quad (25)$$

where B_I is the *strain matrix* for node I . Substituting Eq. (24) and (25) into (23), we have

$$Ku = f$$

where

$$K_{IJ} = \int_{\Omega} B_I^T D B_J d\Omega \quad (26)$$

$$f_I = \int_{\Omega} \Psi_I^T b d\Omega + \int_{\Gamma_t} \Psi_I^T \bar{t} d\Gamma \quad (27)$$

$$D = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \quad \text{for plane strain} \quad (28)$$

λ and μ are the **Lame parameters** [3]

3 Results and Discussions

3.1 One-Dimensional Numerical Example

Consider the implementation of the RKEM method in one dimension for a problem in linear elastostatic. Figure 7 shows the example problem, a 1D bar of unit length subjected to a linear body force of magnitude x .

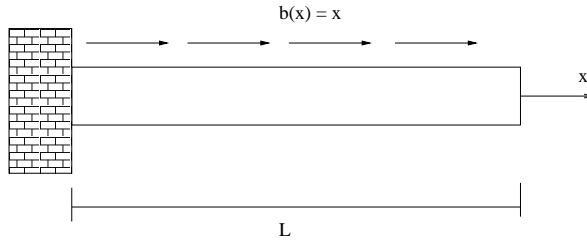


Figure 7: The one-dimensional example problem. A bar of unit length subjected to a linear body force and fixed at the point $x = 0$.

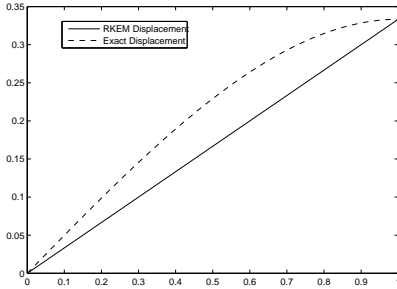
The displacement of the bar is fixed at the left end, and the right end is traction free. The bar has a constant cross sectional area of unit value, and modulus of elasticity E . The equilibrium equation and boundary conditions for this problem are given by:

$$(S) = \begin{cases} Eu_{,xx} + x = 0 & 0 < x < 1 \\ u(0) = 0 \\ u(1) = \frac{1}{3} \end{cases} \quad (29)$$

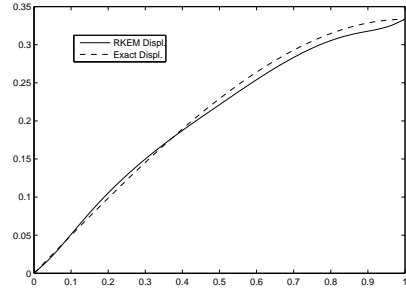
The exact solution to the above problem is given by:

$$u(x) = \frac{1}{E} \left[\frac{1}{2}x - \frac{x^3}{6} \right] \quad (30)$$

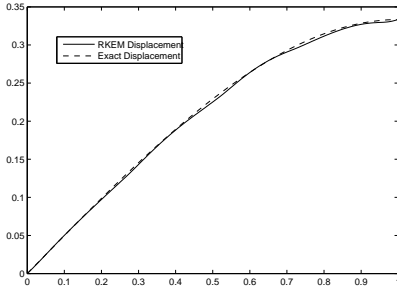
We shall use this solution to examine the quality of the RKEM solution. Figure 8 is a comparison of the RKEM solution to the exact solution for the displacement along the bar. These results were obtained using two to nine nodes along the length of the bar and 16 integration points per element.



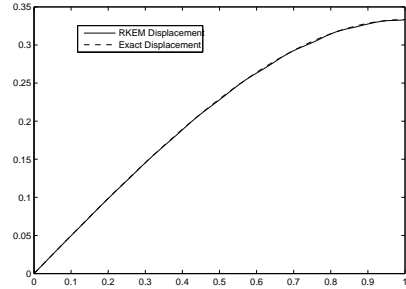
(a) 1 element in the domain



(b) 2 element in the domain



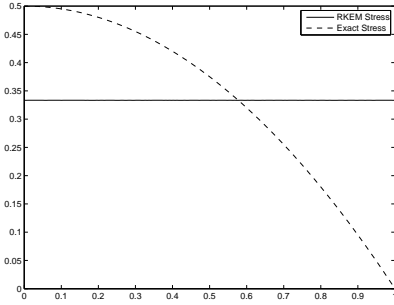
(c) 4 element in the domain



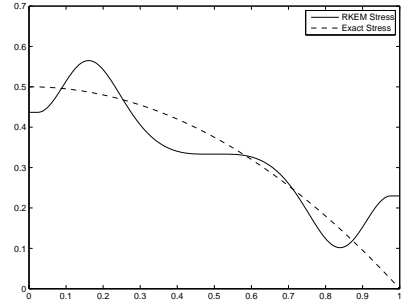
(d) 8 element in the domain

Figure 8: Comparison of RKEM and exact results for displacement for one-dimensional problem

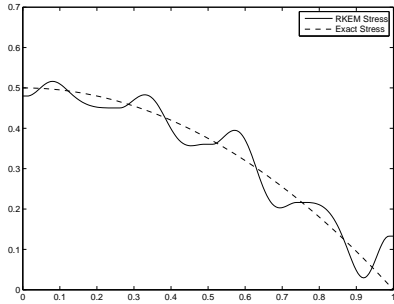
The displacement field calculated with RKEM is nearly exact and continuous. Note that we can generate a globally continuous function with just two nodes. It should be noted that these results were plotted using 200 sampling points along the bar. In Fig. 9 we can observe the solution in the derivative. Note that, unlike finite element, our solution in derivative is continuous. Remember that for the case of linear finite element the stresses are approximated by a constant value along the element, thus we have jumps between elements. The L2P1I0 element used in this example is not that important for practical application, but was chosen because it is easy to show how the RKEM method works. In Section 2.3 we use a higher order element in 2D.



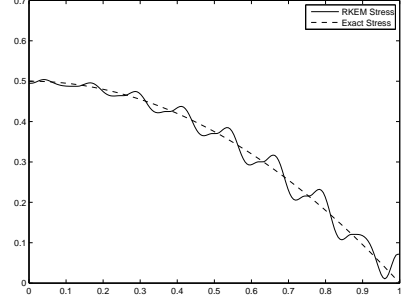
(a) 1 element in the domain



(b) 2 element in the domain



(c) 4 element in the domain



(d) 8 element in the domain

Figure 9: Comparison of RKEM and exact results for stress for one-dimensional problem

3.2 RKEM Program Description

1. Set up nodal coordinates and mesh.
2. Calculate radius of support and check quasi-uniformity condition [10].
3. Set bounding boxes for integration (background mesh).
4. Set up integration points, weights, and Jacobian for each cell.
5. Loop over integration points.

- (a) Calculate normalizer and its derivative at point x_G .
 - (b) Calculate shape functions and its derivatives at point x_G
 - (c) Assemble contributions to K matrix (16) and force vector (17)
6. Apply boundary condition. Static condensed out the matrix K .
 7. Solve for nodal parameters u_I

The latter provides a flow chart which outlines the essential steps of the program. In the first step of the program, the nodal coordinates and the mesh are set up. The nodes are spaced uniformly along the length of the bar, which is not necessary but simplifies the program. Then in the second step the radius of support is set up such that is equal to 0.95 times the minimum distance between elements connecting a node. With the information gotten in the previous step we must check the quasi-uniformity condition. We tested different quadratures and 16 Gauss points per cell is the best option. Note that in this program we loop over all the Gauss points, but this is not necessary because as we know the RKEM shape functions has a compact support and just the nodes inside its influence need to be accounted for. But we loop for all the nodes to clarify the program. In one dimension this is not a serious issue but in multidimensional problem it could be cumbersome. After we assembled the matrix and vector and applied the boundary condition we solve the system of equation using LU decomposition. Because we have the Kronecker delta property, the nodal values calculated are the real nodal values.

3.3 Two-Dimensional Numerical Examples

In this section, two classical problems in linear elastostatic will be described and solve using a RKEM program written in C++ and compiled using pgCC. These examples serves to illustrate the accuracy of the RKEM method in calculating the displacement and stress fields in two dimensions.

3.3.1 Timoshenko Beam Problem Consider the equations of the two dimensional linear isotropic elasticity theory on the domain illustrated in Fig. 10. The boundary conditions are given in [12], where P is a given constant and $I = 2c^3/3$.

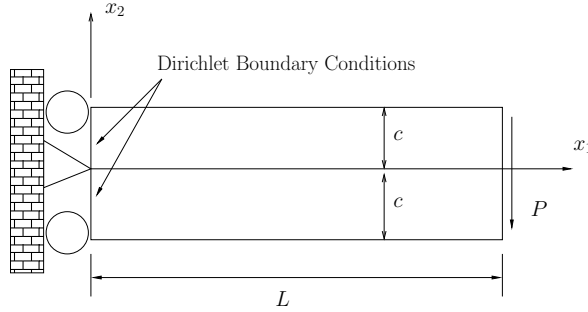


Figure 10: Domain for plane stress elasticity problem.

The problem was solved for the plane stress case with $E = 1$, $\nu = 0.3$, $c = 0.125$, $L = 1$, and $P = -1$. The regular RKEM meshes used to solve the Galerkin weakform (23) are shown in Fig. 11. In the series of Fig. 12 - 13 we can see the good interpolation capability of the RKEM. Note also that we are not using any post process to deal with the stress plot.

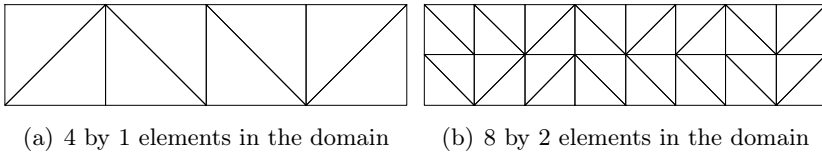
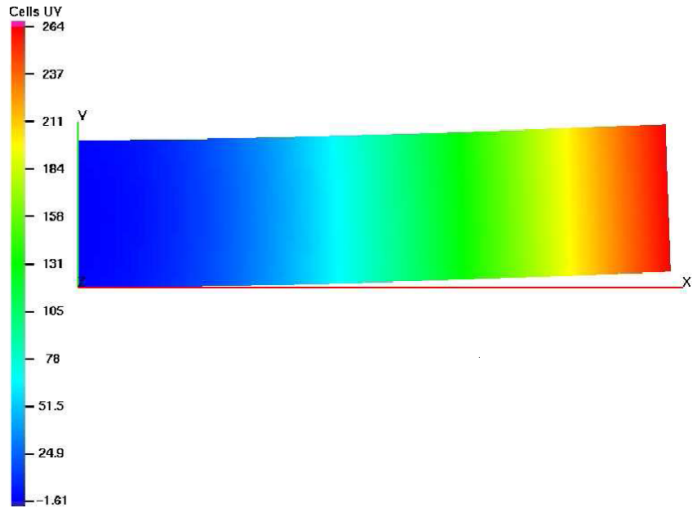
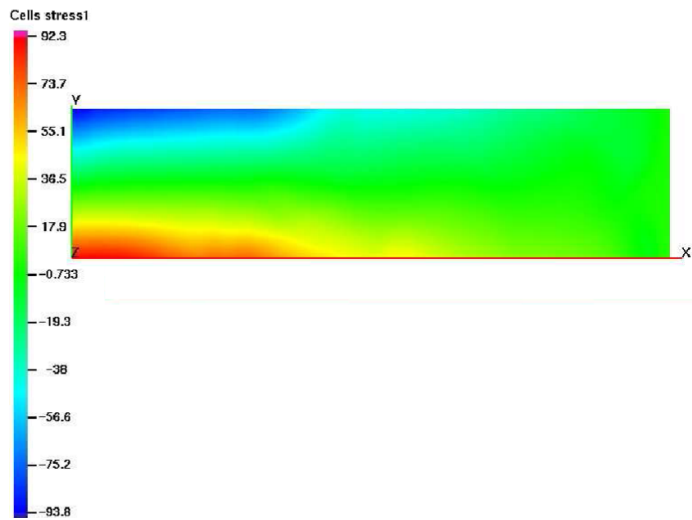


Figure 11: Series of meshes used to solve the Timoshenko beam problem.

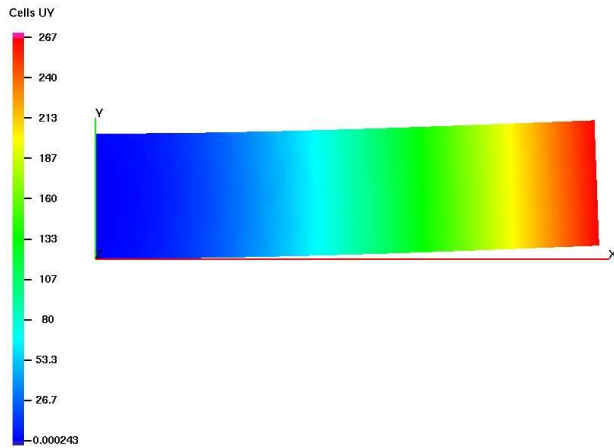


(a) Displacements in x_2 -direction for the 4 by 1 element in the domain.

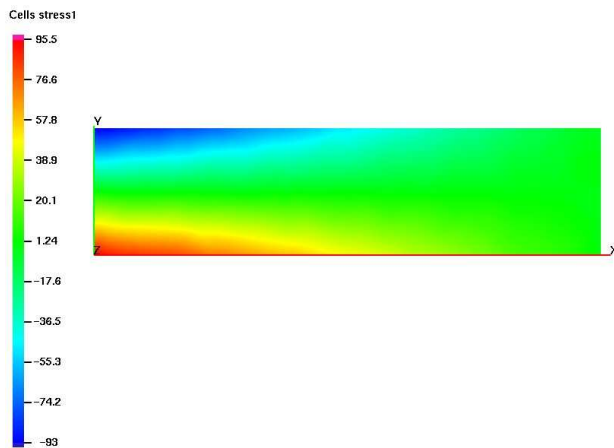


(b) Stresses in the x_1 -direction for the 4 by 1 element in the domain

Figure 12: Solution for the 4 by 1 series.



(a) Displacements in x_2 -direction for the 8 by 2 element in the domain.



(b) Stresses in the x_1 -direction for the 8 by 2 element in the domain

Figure 13: Solution for the 8 by 2 series.

3.3.2 Elasticity Problem The resolution of the problem of the 2D linear elasticity problem represented in Fig. 14 is considered in this section. The same problem was used in [13] to test the imposition of boundary conditions in mesh free methods. The resolution with the RKEM method is considered next in order to analyze the behavior of the imposition of essential boundary conditions. We used two different meshes (14×14 and 28×28 nodes) and a finer mesh is used for the representation of the solution. These meshes are shown in Fig 15(a) and Fig 15(b). Figure 16 and Fig. 17 shows the solution obtained. As observed in the last section, the prescribed displacements are directly imposed, i.e. the value of the corresponding nodal coefficients is set to the prescribed value. In both cases, the RKEM approximation at the boundary allows the exact enforcement of the prescribed displacement. Note that we have wiggles close to the boundary, that is because the RKEM does not enforce strongly the boundary conditions. That is, they are exactly at the nodes but between them, the solution is a piecewise global continuous polynomial. This effect is minimized when you refine the model, see Fig. 15(b). Also note that, we have not used any post process to smooth the results. After we calculated the nodal unknowns we use the interpolation property of RKEM to calculate the displacement using a finer distribution of points. For the integration of the weak form (Eq. (23)) we have used 64 Gauss points per elements. We used the RKEM mesh like the integration mesh. We need to use a high order rule because of the difficulty of the integration of the shape functions.

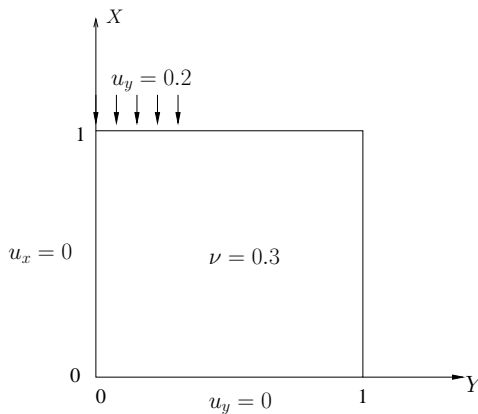
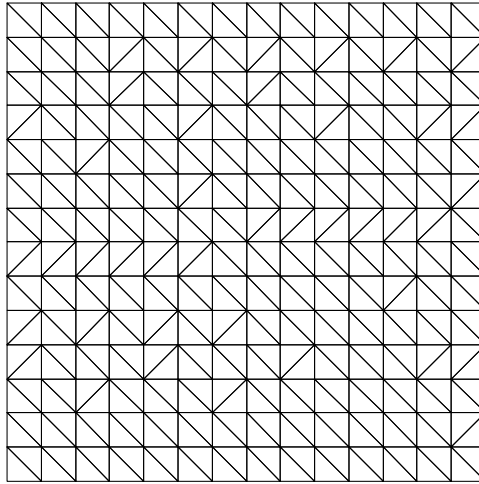
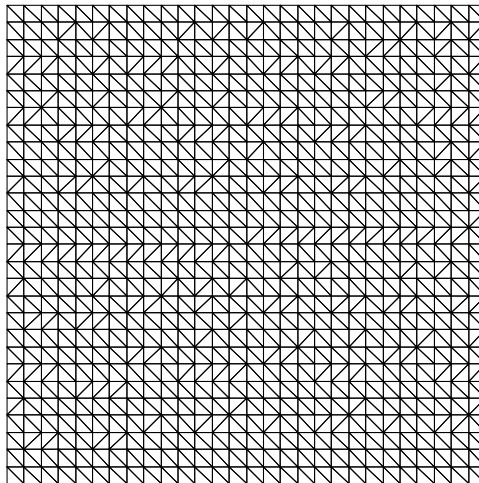


Figure 14: Problem statement.



(a) Mesh with 14×14 nodes and 392 T9P2I1 elements.



(b) Mesh with 28×28 nodes and 1568 T9P2I1 elements.

Figure 15: RKEM meshes used to solve the elasticity problem.

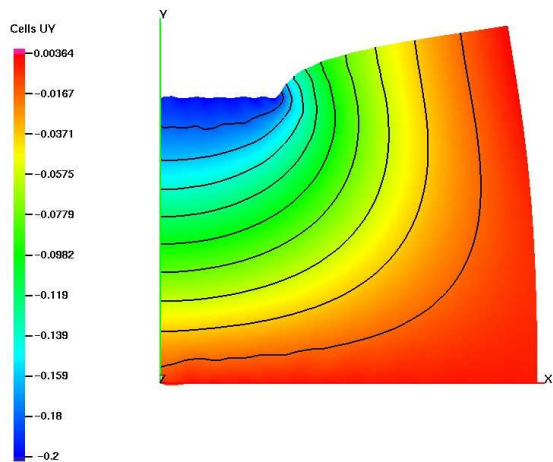


Figure 16: Solution using the mesh showed in Fig 15(a). Contour plot of the vertical displacement.

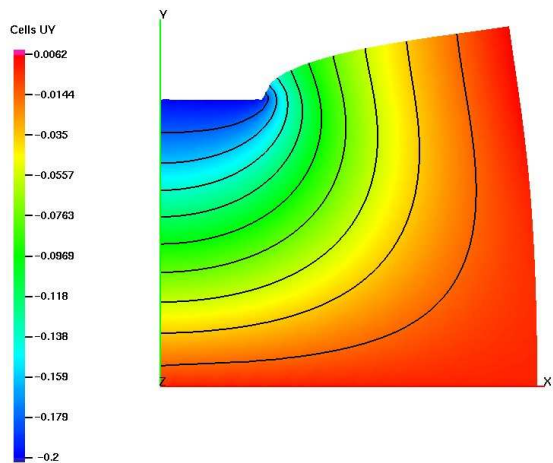


Figure 17: Solution using the mesh showed in Fig 15(b). Contour plot of the vertical displacement.

4 Conclusion

The details of RKEM and its numerical implementation have been presented, with specific emphasis on a one and two dimensional formulation. The RKEM interpolant and its use in a Galerkin weak form were described, including examples in elastostatic. The numerical results show the accuracy of the method for the displacement field. Furthermore, it show a better approximation property than a linear FEM element. We showed that the L2P1I0 element is globally continuous. The extension of the method to two dimension was described, and two simple 2D problems in linear elastostatics were presented and solved with the method and using the T9P2I1 element. The results showed the accuracy of the RKEM method in computing the stresses in the problem as well as the displacements. We have shown that we do not need any special treatment of the boundary conditions, then the computational implementation of the Dirichlet boundary conditions is straightforward. Still there are some issues about the computational time, specially for 2D or 3D problems, the integration of the weak form and the programming of the method. However, the potential of the RKEM for certain classes of problems is exciting and demands more research. The aim of this work has been to provide the reader with an introduction to the RKEM method, and to facilitate experimentation with the method independently.

5 Acknowledgment

The author wish to give his gratitude to Professor Daniel Simkins for given me the opportunity to study RKEM. Also, the author wish to give thanks to Dr. Nathan Collier for sharing his knowledge and programming expertise.

References

- [1] D. Simkins, S. Li, H. Lu, W. Kam Liu, “ Reproducing kernel element method. Part IV: Globally compatible triangular hierarchy”, *Computer Methods in Applied Mechanics and Engineering*, vol. 193, n.o 12-14, pp. 1013-1034, mar. 2004. Referenced in 72, 74, 75, 78, 80
- [2] D. C. S. Jr, N. Collier, M. Juha, L. Whitenack, “A Framework For Studying The RKEM Representation of Discrete Point Sets”, in *Meshfree Methods for Partial*

- Differential Equations IV*, M. Griebel y M. A. Schweitzer, Eds. Springer Berlin Heidelberg, 2008, pp. 301-314. Referenced in 72
- [3] T. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000. Referenced in 72, 75, 79, 84
 - [4] L. Piegl, W. Tiller, *The NURBS book*. New York, NY, USA: Springer-Verlag New York, Inc., 1997. Referenced in 73
 - [5] G. Liu, *Mesh Free Methods: Moving Beyond the Finite Element Method*, 1.a ed. CRC Press, 2002. Referenced in 73
 - [6] W. K. Liu, W. Han, H. Lu, S. Li, y J. Cao, “Reproducing kernel element method. Part I: Theoretical formulation”, *Computer Methods in Applied Mechanics and Engineering*, vol. 193, n.o 12-14, pp. 933-951, mar. 2004. Referenced in 74
 - [7] S. Li, H. Lu, W. Han, W. Liu, D. Simkins, “Reproducing kernel element method Part II: Globally conforming Im/Cn hierarchies”, *Computer Methods in Applied Mechanics and Engineering*, vol. 193, n.o 12-14, pp. 953-987, mar. 2004. Referenced in 74
 - [8] H. Lu, S. Li, D. C. Simkins Jr., W. Kam Liu, J. Cao, “Reproducing kernel element method Part III: Generalized enrichment and applications”, *Computer Methods in Applied Mechanics and Engineering*, vol. 193, n.o 12-14, pp. 989-1011, mar. 2004. Referenced in 75
 - [9] K.-J. Bathe, *Finite element procedures*. Prentice Hall, 1995. Referenced in 80
 - [10] N. Collier y D. C. Simkins, “The quasi-uniformity condition for reproducing kernel element method meshes”, *Computational Mechanics*, vol. 44, n.o 3, pp. 333-342, mar. 2009. Referenced in 81, 87
 - [11] A. H. Stroud, D. Secrest, *Gaussian quadrature formulas*. Prentice-Hall, 1966. Referenced in 82
 - [12] S. Timoshenko y J. N. Goodier, *Theory of elasticity*. McGraw-Hill, 1969. Referenced in 88
 - [13] S. Fernández-Méndez y A. Huerta, “Imposing essential boundary conditions in mesh-free methods”, *Computer Methods in Applied Mechanics and Engineering*, vol. 193, n.o 12-14, pp. 1257-1275, mar. 2004. Referenced in 92