Enriched Reproducing Kernel Particle Approximation for Simulating Problems Involving Moving Interfaces

Pierre Joyot, Jean Trunzler, Francisco Chinesta

Enriched Reproducing Kernel Particle Approximation for Simulating Problems Involving Moving Interfaces

Pierre Joyot[1], Jean Trunzler[1], and Fransisco Chinesta[2]

[1] LIPSI-ESTIA, Technopole Izarbel, 64210 Bidart, France
{j.trunzler,p.joyot}@estia.fr
[2] LMSP, 151 Bd. de l'Hôpital, 75013 Paris, France
francisco.chinesta@paris.ensam.fr

Summary. In this paper we propose a new approximation technique within the context of meshless methods able to reproduce functions with discontinuous derivatives. This approach involves some concepts of the reproducing kernel particle method (RKPM), which have been extended in order to reproduce functions with discontinuous derivatives. This strategy will be referred as Enriched Reproducing Kernel Particle Approximation (E-RKPA). The accuracy of the proposed technique will be compared with standard RKP approximations (which only reproduces polynomials).

Key words: Meshless methods, discontinuous derivatives, enriched approximation, reproducing kernel particle method

1 Introduction

Meshless methods are an appealing choice for constructing functional approximations (with different degrees of consistency and continuity) without a mesh support. Thus, this kind of techniques seem to be specially appropriated for treating 3D problems involving large displacements, due to the fact that the approximation is constructed only from the cloud of nodes whose positions evolve during the material deformation. In this manner neither remeshing nor fields projections are a priori required.

Other important point lies in the easy introduction of some known information related to the problem solution within the approximation functional basis. For this purpose, different reproduction conditions are enforced in the construction of the approximation functions. This approach has been widely used in the context of the moving least squares approximations involved in the diffuse meshless techniques [1] as well as in the element free Galerkin method [3]. Very accurate results were obtained for example in fracture mechanics by introducing the crack tip behavior into the approximation basis [4].

In this work we propose a numerical strategy, based on the reproducing kernel particle techniques, able to construct approximation functions with discontinuous derivatives on fixed or moving interfaces. This problem was treated in the context of the partition of unity by Kronggauz et al. [7]. In our approach the size of the discrete system of equations remains unchanged because no additional degrees of freedom are introduced related to the enrichment. However, the fact of enriching the approximation implies a bigger moment matrix that can result ill conditioned when the enrichment is applied in the whole domain. To circumvent this difficulty local enrichments seem more appropriate. This paper focuses on local enrichments with particular reproduction conditions.

The starting point of our development is the reproducing kernel particle approximation (RKPA). The RKP approximation was introduced by Liu et al. [10] for enforcing some degree of consistency to standard smooth particle approximations, i.e. they proved that starting from a SPH (smooth particle hydrodynamics) approximation [5] it is possible to enhance the kernel function for reproducing a certain degree of polynomials. We have extended or generalized this procedure in order to reproduce any function, and more concretely, functions involving discontinuous derivatives. The question of the local enrichment will be then addressed.

2 Enriched Functional Approximations

2.1 Reproducing Conditions for Enriched Shape Function

The approximation of a function $u(\boldsymbol{x})$ is defined by

$$u(\boldsymbol{x}) = \sum_{I=1}^{NP} \psi_I(\boldsymbol{x}) u(\boldsymbol{x}_I) \tag{2.1}$$

where $u(\boldsymbol{x}_I)$ are the nodal values of the approximated function and NP the number of nodes used to discretize the domain Ω. $\psi_I(\boldsymbol{x})$ is the shape function which can be written in the general form:

$$\psi_I(\boldsymbol{x}) = C\phi(\boldsymbol{x} - \boldsymbol{x}_I) \tag{2.2}$$

where ϕ is the kernel function which has a compact support. Consequently $\psi_I(\boldsymbol{x})$ will be non zero only for a small set of nodes.

We represent by $\Lambda(\boldsymbol{x})$ the set of nodes whose supports include the point $\boldsymbol{x}$. Thus, we can write Eq. (2.1) as

$$u(\boldsymbol{x}) = \sum_{\lambda \in \Lambda(\boldsymbol{x})} C\phi(\boldsymbol{x} - \boldsymbol{x}_\lambda) u(\boldsymbol{x}_\lambda) \tag{2.3}$$

In the following, we use the simplified notation:

$$u(\boldsymbol{x}) = \sum_{\lambda} C\phi_{\lambda} u(\boldsymbol{x}_{\lambda}) \tag{2.4}$$

In the RKPM context Liu and al. [10] define C as the correction function used to ensure the reproduction conditions. Thus, any linear combination of the functions used to define the reproduction conditions can be approximated exactly. Usually the reproduction conditions are imposed to ensure that the approximation can reproduce polynomials up to a specified degree that represents the approximation order of consistency. In the present work, we want to reproduce a function consisting of a polynomial part and others additional functions used to include known information about the approximated field, as for example discontinuous normal derivatives across fixed or moving interfaces.

Let $u(\boldsymbol{x})$ the function to be reproduced:

$$u(\boldsymbol{x}) = \sum_{\alpha} a_{\alpha} \boldsymbol{x}^{\alpha} + \sum_{j=1}^{ne} e_j \chi^j(\boldsymbol{x}). \tag{2.5}$$

where α is a multi-index used to represent the polynomial part of u, χ is the enrichment function and j a simple index that refers to the power of χ (we want to reproduce the enrichment function and its power up to the degree ne). Multi index notation is deeply described in [9].

First we consider the reproducing conditions for the polynomial part of u. When $|\alpha| = 0$, we obtain the partition of unity

$$\sum_{\lambda} C\phi_{\lambda} 1 = 1 \tag{2.6}$$

and for $|\alpha| \leq m, |\alpha| \neq 0$ (where m is the degree of the polynomial part)

$$\sum_{\lambda} C\phi_{\lambda} \boldsymbol{x}_{\lambda}^{\alpha} = \boldsymbol{x}^{\alpha} \tag{2.7}$$

Now, we consider the non-polynomial reproducing conditions:

$$\sum_{\lambda} C\phi_{\lambda} \chi^i(\boldsymbol{x}_{\lambda}) = \chi^i(\boldsymbol{x}) \quad 1 \leq i \leq ne \tag{2.8}$$

All these reproducing conditions can be written in the matrix form:

$$\sum_{\lambda} C\phi_{\lambda} \boldsymbol{R}(\boldsymbol{x}_{\lambda}) = \boldsymbol{R}(\boldsymbol{x}) \tag{2.9}$$

where $\boldsymbol{R}$ denotes the reproducing vector that consists of the polynomial part $\boldsymbol{R}_p$ and the non-polynomial one $\boldsymbol{R}_e$ ($\boldsymbol{R}^{\top}(\boldsymbol{x}) = \left[\boldsymbol{R}_p{}^{\top}(\boldsymbol{x}) \; \boldsymbol{R}_e{}^{\top}(\boldsymbol{x}) \right]$)

2.2 Direct Formulation of the Shape Functions

To construct the approximation the correction function must be defined. The choice of C lead to different formulations.

In the direct formulation, the correction function is defined by

$$C_{\mathtt{d}} = \boldsymbol{H}_{\mathtt{d}}^{\top}(\boldsymbol{x}, \boldsymbol{x}_\lambda, \boldsymbol{x}_\lambda - \boldsymbol{x})\boldsymbol{b}_{\mathtt{d}}(\boldsymbol{x}) \tag{2.10}$$

where $\boldsymbol{H}_{\mathtt{d}}$ consists of a polynomial part of degree m: $P_m(\boldsymbol{x}_\lambda - \boldsymbol{x})$, and the non-polynomial one $\boldsymbol{H}_{e_{\mathtt{d}}} = \left[\, \chi(\boldsymbol{x}_\lambda) - \chi(\boldsymbol{x}) \cdots \; (\chi(\boldsymbol{x}_\lambda) - \chi(\boldsymbol{x}))^{ne} \,\right]$:

$$\boldsymbol{H}_{\mathtt{d}}^{\top}(\boldsymbol{x}, \boldsymbol{x}_\lambda, \boldsymbol{x}_\lambda - \boldsymbol{x}) = \left[\, P_m(\boldsymbol{x}_\lambda - \boldsymbol{x}) \; \chi(\boldsymbol{x}_\lambda) - \chi(\boldsymbol{x}) \cdots \; (\chi(\boldsymbol{x}_\lambda) - \chi(\boldsymbol{x}))^{ne} \,\right] \tag{2.11}$$

From the definition of the correction function (2.10) the reproducing conditions (2.9) becomes

$$\left(\sum_\lambda \boldsymbol{R}(\boldsymbol{x}_\lambda) \boldsymbol{H}_d^{\top}(\boldsymbol{x}, \boldsymbol{x}_\lambda, \boldsymbol{x}_\lambda - \boldsymbol{x}) \phi_\lambda \right) \boldsymbol{b}_{\mathtt{d}}(\boldsymbol{x}) = \boldsymbol{R}(\boldsymbol{x}) \tag{2.12}$$

that can by written in the matrix form:

$$\boldsymbol{M}_{\mathtt{d}}(\boldsymbol{x}) \boldsymbol{b}_{\mathtt{d}}(\boldsymbol{x}) = \boldsymbol{R}(\boldsymbol{x}) \tag{2.13}$$

where the moment matrix $\boldsymbol{M}_{\mathtt{d}}(\boldsymbol{x})$ is defined by

$$\boldsymbol{M}_{\mathtt{d}}(\boldsymbol{x}) = \sum_\lambda \boldsymbol{R}(\boldsymbol{x}_\lambda) \boldsymbol{H}_{\mathtt{d}}^{\top}(\boldsymbol{x}, \boldsymbol{x}_\lambda, \boldsymbol{x}_\lambda - \boldsymbol{x}) \phi_\lambda \tag{2.14}$$

Finally, from Eq. (2.2) the shape function in the direct formulation will be defined by

$$\psi_{\mathtt{d}\lambda}(\boldsymbol{x}) = \boldsymbol{H}_{\mathtt{d}}^{\top}(\boldsymbol{x}, \boldsymbol{x}_\lambda, \boldsymbol{x}_\lambda - \boldsymbol{x}) \boldsymbol{M}_{\mathtt{d}}^{-1}(\boldsymbol{x}) \boldsymbol{R}(\boldsymbol{x}) \phi_\lambda \tag{2.15}$$

2.3 Direct and RKPM Shape Function Equivalence

In [9] Liu et al. show that the RKPM formulation satisfies the polynomial reproducing conditions. Here, we prove that this result can be extended to non polynomial reproducing conditions.

For this purpose, firstly, we derive the enriched RKPM shape functions from the reproducing conditions.

The polynomial part can be written as:

$$\sum_\lambda C \phi_\lambda 1 = 1 \quad |\alpha| = 0 \tag{2.16}$$

$$\sum_\lambda C \phi_\lambda (\boldsymbol{x}_\lambda - \boldsymbol{x})^\alpha = 0 \quad |\alpha| \leq m, |\alpha| \neq 0 \tag{2.17}$$

the proof can be found for example in [9].

Using the same procedure, we can also write

$$\sum_{\lambda} C\phi_{\lambda}(\chi(\boldsymbol{x}_{\lambda}) - \chi(\boldsymbol{x}))^{i} = 0 \quad 1 \leq i \leq ne \tag{2.18}$$

According to Eqs. (2.17) and (2.18) the reproducing conditions can be written as

$$\sum_{\lambda} C\phi_{\lambda}\boldsymbol{R}(\boldsymbol{x}, \boldsymbol{x}_{\lambda}, \boldsymbol{x}_{\lambda} - \boldsymbol{x}) = \boldsymbol{R}(0) \tag{2.19}$$

where $\boldsymbol{R}^{\top}(\boldsymbol{x}, \boldsymbol{x}_{\lambda}, \boldsymbol{x}_{\lambda}-\boldsymbol{x}) = \left[P_m(\boldsymbol{x}_{\lambda} - \boldsymbol{x})\ \chi(\boldsymbol{x}_{\lambda}) - \chi(\boldsymbol{x}) \cdots\ (\chi(\boldsymbol{x}_{\lambda}) - \chi(\boldsymbol{x}))^{ne} \right]$ and $\boldsymbol{R}^{\top}(0) = \left[1\ 0 \cdots\ 0 \right]$.

Now we choose for the correction function the same expression than in the direct formulation (2.11). i.e. $\boldsymbol{H}_{\mathtt{r}} = \boldsymbol{H}_{\mathtt{d}}$

$$C_{\mathtt{r}} = \boldsymbol{H}_{\mathtt{r}}^{\top}(\boldsymbol{x}, \boldsymbol{x}_{\lambda}, \boldsymbol{x}_{\lambda} - \boldsymbol{x})\boldsymbol{b}_{\mathtt{r}} \tag{2.20}$$

introducing this expression into Eq. (2.19), it results in:

$$\left(\sum_{\lambda} \boldsymbol{R}(\boldsymbol{x}, \boldsymbol{x}_{\lambda}, \boldsymbol{x}_{\lambda} - \boldsymbol{x})\boldsymbol{H}_{\mathtt{r}}^{\top}(\boldsymbol{x}, \boldsymbol{x}_{\lambda}, \boldsymbol{x}_{\lambda} - \boldsymbol{x})\phi_{\lambda} \right) \boldsymbol{b}_{\mathtt{r}} = \boldsymbol{R}(0) \tag{2.21}$$

or being $\boldsymbol{R} \equiv \boldsymbol{H}_{\mathtt{r}} \equiv \boldsymbol{H}$.

$$\left(\sum_{\lambda} \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{x}_{\lambda}, \boldsymbol{x}_{\lambda} - \boldsymbol{x})\boldsymbol{H}^{\top}(\boldsymbol{x}, \boldsymbol{x}_{\lambda}, \boldsymbol{x}_{\lambda} - \boldsymbol{x})\phi_{\lambda} \right) \boldsymbol{b}_{\mathtt{r}} = \boldsymbol{R}(0) \tag{2.22}$$

whose matrix form results in

$$\boldsymbol{M}_{\mathtt{r}}(\boldsymbol{x})\boldsymbol{b}_{\mathtt{r}}(\boldsymbol{x}) = \boldsymbol{R}(0) \tag{2.23}$$

where $\boldsymbol{M}_{\mathtt{r}}(\boldsymbol{x})$ is the enriched RKPM moment matrix defined by

$$\boldsymbol{M}_{\mathtt{r}}(\boldsymbol{x}) = \sum_{\lambda} \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{x}_{\lambda}, \boldsymbol{x}_{\lambda} - \boldsymbol{x})\boldsymbol{H}^{\top}(\boldsymbol{x}, \boldsymbol{x}_{\lambda}, \boldsymbol{x}_{\lambda} - \boldsymbol{x})\phi_{\lambda} \tag{2.24}$$

Since Eqs. (2.9) and (2.19) are equivalent it directly follows that the vectors $\boldsymbol{b}_{\mathtt{d}}$ and $\boldsymbol{b}_{\mathtt{r}}$ are the same, and given $\boldsymbol{H}_{\mathtt{r}} = \boldsymbol{H}_{\mathtt{d}}$ we can conclude that Direct and the RKPM shape fonctions are the same.

2.4 MLS Shape Function

In this section we are going to prove that the MLS formulation [3] can be obtained by choosing the following form of the correction function

$$C = \boldsymbol{H}_{\mathtt{m}}^{\top}(\boldsymbol{x}_{\lambda})\boldsymbol{b}_{\mathtt{m}}(\boldsymbol{x}) \tag{2.25}$$

where $\boldsymbol{H}_{\mathtt{m}} = \boldsymbol{R}$. The reproducing conditions (2.9) can be rewritten

$$\sum_{\lambda} \boldsymbol{R}(\boldsymbol{x}_\lambda)\boldsymbol{H}_{\mathtt{m}}^{\top}(\boldsymbol{x}_\lambda)\boldsymbol{b}_{\mathtt{m}}(\boldsymbol{x})\phi_\lambda = \boldsymbol{R}(\boldsymbol{x}) \tag{2.26}$$

or

$$\boldsymbol{M}_{\mathtt{m}}(\boldsymbol{x})\boldsymbol{b}_{\mathtt{m}}(\boldsymbol{x}) = \boldsymbol{R}(\boldsymbol{x}) \tag{2.27}$$

where $\boldsymbol{M}_{\mathtt{m}}(\boldsymbol{x})$ is the MLS form of the moment matrix defined by

$$\boldsymbol{M}_{\mathtt{m}}(\boldsymbol{x}) = \sum_{\lambda} \boldsymbol{H}_{\mathtt{m}}(\boldsymbol{x}_\lambda)\boldsymbol{H}_{\mathtt{m}}^{\top}(\boldsymbol{x}_\lambda)\phi_\lambda \tag{2.28}$$

To derive the standard MLS shape function we consider Eq. (2.4) from which we obtain

$$u(\boldsymbol{x}) = \sum_{\lambda} \boldsymbol{H}_{\mathtt{m}}^{\top}(\boldsymbol{x}_\lambda)\boldsymbol{M}_{\mathtt{m}}^{-1}(\boldsymbol{x})\boldsymbol{R}(\boldsymbol{x})\phi_\lambda u(\boldsymbol{x}_\lambda) \tag{2.29}$$

Since $\boldsymbol{H}_{\mathtt{m}} = \boldsymbol{R}$, the moment matrix is symmetric, and Eq. (2.29) results in

$$u(\boldsymbol{x}) = \boldsymbol{H}_{\mathtt{m}}^{\top}(\boldsymbol{x})\boldsymbol{M}_{\mathtt{m}}^{-1}(\boldsymbol{x}) \sum_{\lambda} \boldsymbol{H}_{\mathtt{m}}(\boldsymbol{x}_\lambda)\phi_\lambda u(\boldsymbol{x}_\lambda) \tag{2.30}$$

or

$$u(\boldsymbol{x}) = \boldsymbol{H}_{\mathtt{m}}^{\top}(\boldsymbol{x})\boldsymbol{a}_{\mathtt{m}} \tag{2.31}$$

where $\boldsymbol{a}_{\mathtt{m}}$ is computed from

$$\boldsymbol{M}_{\mathtt{m}}(\boldsymbol{x})\boldsymbol{a}_{\mathtt{m}} = \sum_{\lambda} \boldsymbol{H}_{\mathtt{m}}(\boldsymbol{x}_\lambda)\phi_\lambda u(\boldsymbol{x}_\lambda) \tag{2.32}$$

Now, we can prove that MLS shape functions coincide with the ones related to the direct procedure.

2.5 Direct and MLS Shape Function Equivalence

To prove the equivalence between both formulations, firstly, we consider the expression of the line i-th component of vector $\boldsymbol{M}_{\mathtt{m}}\boldsymbol{b}_{\mathtt{m}}$.

$$\begin{aligned} [\boldsymbol{M}_{\mathtt{m}}\boldsymbol{b}_{\mathtt{m}}]_i \;=\; & b_{\mathtt{m}p_0} \sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\phi_\lambda \\ & + \sum_{0<|\alpha|\leq m} b_{\mathtt{m}p_\alpha} \sum_{\lambda} R_i(\boldsymbol{x}_\lambda)(\boldsymbol{x}_\lambda)^{\alpha}\phi_\lambda \\ & + \sum_{j=1}^{ne} b_{\mathtt{m}e_j} \sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\chi(\boldsymbol{x}_\lambda)^{j}\phi_\lambda \end{aligned} \tag{2.33}$$

R_i being the i-th component of vector $\boldsymbol{R}$.

In the same manner, the corresponding component related to vector $\boldsymbol{M}_{\mathtt{d}}\boldsymbol{b}_{\mathtt{d}}$ results in:

$$
\begin{aligned}
[\boldsymbol{M}_{\mathtt{d}}\boldsymbol{b}_{\mathtt{d}}]_i = \; & b_{\mathtt{d}p_0}\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\phi_\lambda \\
& + \sum_{0<|\alpha|\le m} b_{\mathtt{d}p_\alpha}\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)(\boldsymbol{x}_\lambda-\boldsymbol{x})^\alpha\phi_\lambda \\
& + \sum_{j=1}^{ne} b_{\mathtt{d}e_j}\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\big(\chi_j(\boldsymbol{x}_\lambda)-\chi_j(\boldsymbol{x})\big)^j\phi_\lambda
\end{aligned}
\tag{2.34}
$$

Using the binomial theorem, we can write

$$
\begin{aligned}
&\sum_{j=1}^{ne} b_{\mathtt{d}e_j}\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\big(\chi(\boldsymbol{x}_\lambda)-\chi(\boldsymbol{x})\big)^j\phi_\lambda \\
&\qquad = \sum_{j=0}^{ne}\tilde{b}_{\mathtt{d}e_j}(b_{\mathtt{d}e_1},\cdots,b_{\mathtt{d}e_{ne}},\chi(\boldsymbol{x}))\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\chi(\boldsymbol{x}_\lambda)^j\phi_\lambda
\end{aligned}
\tag{2.35}
$$

and

$$
\begin{aligned}
&\sum_{|\alpha|\le m} b_{\mathtt{d}p_\alpha}\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)(\boldsymbol{x}_\lambda-\boldsymbol{x})^\alpha\phi_\lambda \\
&\qquad = \sum_{|\alpha|\le m}\tilde{b}_{\mathtt{d}p_\alpha}(b_{\mathtt{d}p_{|\alpha|=0}},\cdots,b_{\mathtt{d}p_{|\alpha|=m}},\boldsymbol{x})\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\boldsymbol{x}_\lambda^\alpha\phi_\lambda
\end{aligned}
\tag{2.36}
$$

Using Eqs. (2.35) and (2.36), Eq. (2.34) results in

$$
\begin{aligned}
[\boldsymbol{M}_{\mathtt{d}}\boldsymbol{b}_{\mathtt{d}}]_i = \; & \tilde{b}_{\mathtt{d}p_0}\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\phi_\lambda \\
& + \sum_{0<|\alpha|\le m}\tilde{b}_{\mathtt{d}p_\alpha}\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)(\boldsymbol{x}_\lambda)^\alpha\phi_\lambda \\
& + \tilde{b}_{\mathtt{d}e_0}\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\phi_\lambda \\
& + \sum_{j=1}^{ne}\tilde{b}_{\mathtt{d}e_j}\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\chi(\boldsymbol{x}_\lambda)^j\phi_\lambda
\end{aligned}
\tag{2.37}
$$

Subtracting Eqs. (2.37) and (2.33) leads to:

$$
\begin{aligned}
[\boldsymbol{M}_{\mathtt{m}}\boldsymbol{b}_{\mathtt{m}}]_i - [\boldsymbol{M}_{\mathtt{d}}\boldsymbol{b}_{\mathtt{d}}]_i = \; & \Big(b_{\mathtt{m}p_0}-\tilde{b}_{\mathtt{d}p_0}-\tilde{b}_{\mathtt{d}e_0}\Big)\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\phi_\lambda \\
& + \sum_{0<|\alpha|\le m}\Big(b_{\mathtt{m}p_\alpha}-\tilde{b}_{\mathtt{d}p_\alpha}\Big)\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)(\boldsymbol{x}_\lambda)^\alpha\phi_\lambda \\
& + \sum_{j=1}^{ne}\Big(b_{\mathtt{m}e_j}-\tilde{b}_{\mathtt{d}e_j}\Big)\sum_{\lambda} R_i(\boldsymbol{x}_\lambda)\chi(\boldsymbol{x}_\lambda)^j\phi_\lambda
\end{aligned}
\tag{2.38}
$$

As Eqs. (2.27) and (2.13) imply $\boldsymbol{M}_{\mathtt{m}}\boldsymbol{b}_{\mathtt{m}} - \boldsymbol{M}_{\mathtt{d}}\boldsymbol{b}_{\mathtt{d}} = 0$, then Eq. (2.38) reduces to

$$\boldsymbol{M}_{\mathtt{m}} \begin{bmatrix} b_{\mathtt{m}p_0} - \tilde{b}_{\mathtt{d}p_0} - \tilde{b}_{\mathtt{d}e_0} \\ \boldsymbol{b}^{\star}_{\mathtt{m}p} - \tilde{\boldsymbol{b}^{\star}}_{\mathtt{d}p} \\ \boldsymbol{b}^{\star}_{\mathtt{m}e} - \tilde{\boldsymbol{b}^{\star}}_{\mathtt{d}e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.39}$$

As $\boldsymbol{M}_{\mathtt{m}}$ is non singular, we can conclude

$$\begin{bmatrix} b_{\mathtt{m}p_0} - \tilde{b}_{\mathtt{d}p_0} - \tilde{b}_{\mathtt{d}e_0} \\ \boldsymbol{b}^{\star}_{\mathtt{m}p} - \tilde{\boldsymbol{b}^{\star}}_{\mathtt{d}p} \\ \boldsymbol{b}^{\star}_{\mathtt{m}e} - \tilde{\boldsymbol{b}^{\star}}_{\mathtt{d}e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.40}$$

Using a similar reasoning, it is easy to prove that

$$\begin{aligned} \boldsymbol{H}^{\top}_{\mathtt{m}}\boldsymbol{b}_{\mathtt{m}} - \boldsymbol{H}^{\top}_{\mathtt{d}}\boldsymbol{b}_{\mathtt{d}} = & \left(b_{\mathtt{m}p_0} - \tilde{b}_{\mathtt{d}p_0} - \tilde{b}_{\mathtt{d}e_0}\right) \\ & + \sum_{0<|\alpha|\leq m} \left(b_{\mathtt{m}p_\alpha} - \tilde{b}_{\mathtt{d}p_\alpha}\right) (\boldsymbol{x}_\lambda)^\alpha \\ & + \sum_{j=1}^{ne} \left(b_{\mathtt{m}e_j} - \tilde{b}_{\mathtt{d}e_j}\right) \chi(\boldsymbol{x}_\lambda)^j \end{aligned} \tag{2.41}$$

that implies, taking into account Eq. (2.40), that $\boldsymbol{H}^{\top}_{\mathtt{m}}\boldsymbol{b}_{\mathtt{m}} - \boldsymbol{H}^{\top}_{\mathtt{d}}\boldsymbol{b}_{\mathtt{d}} = 0$, which proves that both shape functions are the same.

Concluding Remarks

It is well known that Direct, MLS and RKPM shape functions are the same when the reproducing conditions are polynomial (see [8]). Here we have extended this result to general reproducing conditions.

3 Definition of the Continuous Shape Functions with Discontinuous Derivatives

Let Ω be the domain where the problem is defined and Γ_d a point, curve or surface (in 1D, 2D and 3D respectively) where the normal derivative of the problem solution becomes discontinuous. We assume that this discontinuity curve splits the domain in two subdomains Ω_0 and Ω_1 (see Fig. 1)

$$\begin{aligned} \Omega_0 \cup \Omega_1 \cup \Gamma_d = \Omega, \\ \Omega_0 \cap \Omega_1 = \emptyset, \end{aligned} \tag{3.42}$$

The enrichment function $\chi(\boldsymbol{x})$ which will be introduced in the reproduction vector $\boldsymbol{R}(\boldsymbol{x})$ must satisfy the discontinuity conditions. In that follows $\tilde{\boldsymbol{x}}$ represents a point located on the interface, u^0 (resp. u^1) is the function u defined

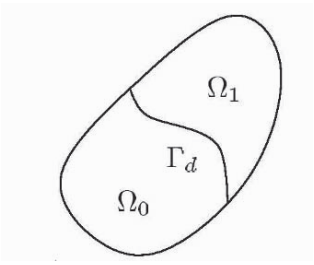


Figure 1. Problem domain containing an interface with a discontinuous normal derivative.

in any point in Ω_0 (resp. Ω_1). Thus, the transmission conditions related to a continuous approximation across the interface Γ_d with discontinuous normal derivative, result in:

$$u^0(\tilde{\boldsymbol{x}}) = u^1(\tilde{\boldsymbol{x}}) \tag{3.43}$$

$$u^0{}_{,i}(\tilde{\boldsymbol{x}}^-)n_i - u^1{}_{,i}(\tilde{\boldsymbol{x}}^+)n_i \neq 0 \tag{3.44}$$

It is also possible to add a discontinuity condition on the second derivative

$$u^0{}_{,ii}(\tilde{\boldsymbol{x}}^-) \neq u^1{}_{,ii}(\tilde{\boldsymbol{x}}^+) \tag{3.45}$$

To locate the interface Γ_d we make use of a "level-set" function $\Theta(\boldsymbol{x})$ defined as the signed distance from $\boldsymbol{x}$ to the interface Γ_d.

Thus, for satisfying the transmission conditions (3.43)-(3.44) the enrichment function $\chi(\boldsymbol{x})$ is assumed to be

$$\chi(\boldsymbol{x}) = H_0(\Theta(\boldsymbol{x}))\Theta(\boldsymbol{x}), \tag{3.46}$$

where $H_0(\boldsymbol{x})$ represents the usual Heaviside function

$$\begin{cases} H_0(\Theta(\boldsymbol{x})) = 1 \text{ if } \Theta(\boldsymbol{x}) \geq 0 \\ H_0(\Theta(\boldsymbol{x})) = 0 \text{ if } \Theta(\boldsymbol{x}) < 0 \end{cases}. \tag{3.47}$$

We are going to verify that this definition of $\chi(\boldsymbol{x})$ represents accurately those transmission conditions. According to the definition of $\chi(\boldsymbol{x})$, given by Eq. (3.46), our approximation reproduce in the domain Ω_0 the function u^0 defined by

$$u^0(\boldsymbol{x}) = \sum_\alpha a_\alpha \boldsymbol{x}^\alpha \tag{3.48}$$

and in the domain Ω_1

$$\begin{aligned} u^1(\boldsymbol{x}) =& \sum_\alpha a_\alpha \boldsymbol{x}^\alpha + e_1\Theta(\boldsymbol{x}) \\ =& u^0(\boldsymbol{x}) + e_1\Theta(\boldsymbol{x}) \end{aligned} \tag{3.49}$$

Since $\Theta(\tilde{\boldsymbol{x}}) = 0$, Eq. (3.49) implies that $u^1(\tilde{\boldsymbol{x}}) = u^0(\tilde{\boldsymbol{x}})$. Thus, the first transmission condition is satisfied.

Now, we evaluate the gradient of u^1 in the neighborhood of Γ_d from Eq. (3.49)

$$u^1_{,i}(\tilde{\boldsymbol{x}}^+)n_i = u^0_{,i}(\tilde{\boldsymbol{x}}^-)n_i + e_1\Theta_{,i}(\tilde{\boldsymbol{x}}^+)n_i \tag{3.50}$$

Since $\Theta_{,i}(\tilde{\boldsymbol{x}}^+)n_i \neq 0$ we verify that the second transmission condition (3.44) is also verified.

When $ne = 1$, that is, when the enrichment consists of a single function, the third condition (3.45) is verified if $\Theta_{,ii}(\tilde{\boldsymbol{x}}^+) \neq 0$. To avoid this difficulty one could consider $ne = 2$. In this case it is easy to prove that the last condition is verified for all Θ.

Thus, the vector $\boldsymbol{H}$ must be defined in the MLS framework as:

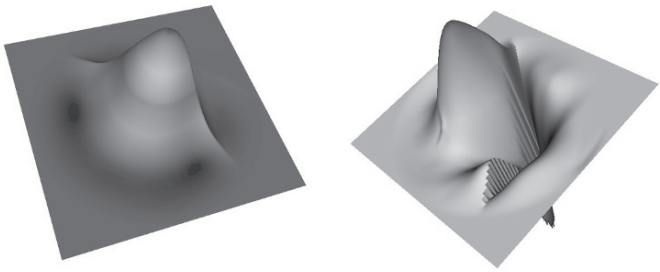
$$\boldsymbol{H}^\top_{\mathtt{m}}(\boldsymbol{x}) = \left[\, P_m(\boldsymbol{x})\ \chi(\boldsymbol{x})\ \chi^2(\boldsymbol{x}) \,\right] \tag{3.51}$$

and in the RKPA context as:

$$\boldsymbol{H}^\top_{\mathtt{r}}(\boldsymbol{x}, \boldsymbol{x}_\lambda, \boldsymbol{x}_\lambda - \boldsymbol{x}) = \left[\, P_m(\boldsymbol{x}_\lambda - \boldsymbol{x})\ \chi(\boldsymbol{x}_\lambda) - \chi(\boldsymbol{x})\ (\chi(\boldsymbol{x}_\lambda) - \chi(\boldsymbol{x}))^2 \,\right] \tag{3.52}$$

with $\chi(\boldsymbol{x}) = H_0(\Theta(\boldsymbol{x}))\Theta(\boldsymbol{x})$.

Figure (2(a)) (resp (2(b))) shows $\psi_\lambda(\boldsymbol{x})$ (resp. $\psi_\lambda(\boldsymbol{x})_{,x}$) evaluated for a node close to a circular interface.



(a) $\psi_\lambda(\boldsymbol{x})$. (b) $\psi_\lambda(\boldsymbol{x})_{,x}$.

Figure 2. Enriched shape function and its derivative evaluated for a node close to a circular interface

4 Enrichment Function Expressed by a Polynomial

Consider the evaluation of the enriched shape function in the direct formulation $\psi_\lambda = (\boldsymbol{H}_p^\top \boldsymbol{b}_p + \boldsymbol{H}_e^\top \boldsymbol{b}_e)\phi_\lambda$. The matrix form of the reproducing conditions is

$$\begin{bmatrix} \boldsymbol{M}_{pp} & \boldsymbol{M}_{pe} \\ \boldsymbol{M}_{ep} & \boldsymbol{M}_{ee} \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_p \\ \boldsymbol{b}_e \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_p \\ \boldsymbol{R}_e \end{bmatrix} \tag{4.53}$$

where the index p refers to the polynomial part and e to the non polynomial one. For the sake of simplicity, from now on, we omit the index referring the direct formulation.

Now consider a shape function computed only with the polynomial part $\hat{\psi}_\lambda = \boldsymbol{H}_p^\top \hat{\boldsymbol{b}}_p \phi_\lambda$. We have

$$\boldsymbol{M}_{pp} \hat{\boldsymbol{b}}_p = \boldsymbol{R}_p \tag{4.54}$$

From Eqs. (4.53) and (4.54) we can write

$$\boldsymbol{M}_{pp} \left(\hat{\boldsymbol{b}}_p - \boldsymbol{b}_p \right) = \boldsymbol{M}_{ep} \boldsymbol{b}_e \tag{4.55}$$

Now we consider a polynomial form of the enrichment functions:

$$\chi^j(\boldsymbol{x}) = \sum_{|\alpha| \leq m} \boldsymbol{b}_{ej} (\boldsymbol{x}_\lambda - \boldsymbol{x})^\alpha (a_j)_\alpha \tag{4.56}$$

where $(a_j)_\alpha$ are the coefficients of the polynomials. Eq. (4.55) can be rewritten as

$$\begin{aligned} \sum_{|\alpha| \leq m} \left(\hat{\boldsymbol{b}}_p - \boldsymbol{b}_p \right)_\alpha \sum_\lambda R_i(\boldsymbol{x}_\lambda)(\boldsymbol{x}_\lambda - \boldsymbol{x})^\alpha \phi_\lambda \\ = \sum_{|\alpha| \leq m} \sum_{j=1}^{ne} \boldsymbol{b}_{ej} (a_j)_\alpha \sum_\lambda R_i(\boldsymbol{x}_\lambda)(\boldsymbol{x}_\lambda - \boldsymbol{x})^\alpha \phi_\lambda \end{aligned} \tag{4.57}$$

which implies

$$\left(\hat{\boldsymbol{b}}_p - \boldsymbol{b}_p \right)_\alpha = \sum_{j=1}^{ne} \boldsymbol{b}_{ej} (a_j)_\alpha \quad |\alpha| \leq m \tag{4.58}$$

Now, we evaluate $\hat{\psi}_\lambda(\boldsymbol{x}) - \psi_\lambda(\boldsymbol{x})$, we obtain

$$\begin{aligned} \hat{\psi}_\lambda(\boldsymbol{x}) - \psi_\lambda(\boldsymbol{x}) &= \boldsymbol{H}_p^\top \hat{\boldsymbol{b}}_p \phi_\lambda - (\boldsymbol{H}_p^\top \boldsymbol{b}_p + \boldsymbol{H}_e^\top \boldsymbol{b}_e) \phi_\lambda \\ &= \phi_\lambda \sum_{|\alpha| \leq m} \left(\left(\hat{\boldsymbol{b}}_p - \boldsymbol{b}_p \right)_\alpha - \sum_{j=1}^{ne} \boldsymbol{b}_{ej} (a_j)_\alpha \right) (\boldsymbol{x}_\lambda - \boldsymbol{x})^\alpha \end{aligned} \tag{4.59}$$

Using (4.58) in the previous equation whe conclude that $\psi_\lambda = \hat{\psi}_\lambda$.

From the equation (4.57) we notice that this result is valid only if the polynomial degree of χ is lower or equal than the degree of the polynomial part of the approximation, and if $(a_j)_\alpha$ do not depend on $\boldsymbol{x}_\lambda$.

In the enrichment just described, the presence of the Heaviside function implies that close to the interface $\psi_\lambda \neq \hat{\psi}_\lambda$ despite that the enrichment has a polynomial representation. ψ_λ will be equal to $\hat{\psi}_\lambda$ when all the nodes $\boldsymbol{x}_\lambda$

whose supports include the point $\boldsymbol{x}$ are located in the same subdomain (Ω_1 or Ω_0).

Thus, one could expect local enrichment of the approximation function in the neighborhood of the interface. However, the distance function previously proposed for enriching the approximation involving discontinuous normal derivatives across a fixed or moving interface, is rarely a polynomial. Thus, the resulting enrichment will result global which implies in the case of moving interfaces recompute at each time step all the shape functions. One possibility to circumvent this difficulty lies in the use of a Taylor expansion of the distance function, which results polynomial, according to:

$$\tilde{\Theta}^j(\boldsymbol{x}_\lambda) = \sum_{|\alpha| \leq m} \frac{1}{\alpha!} \partial^\alpha \Theta^j(\boldsymbol{x})(\boldsymbol{x}_\lambda - \boldsymbol{x})^\alpha \tag{4.60}$$

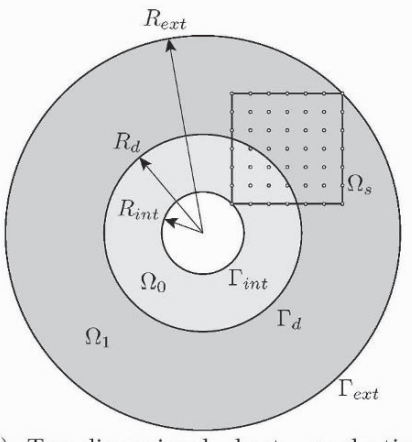
5 Numerical Results

Let's consider the two-dimensional heat conduction problem in a bi-material cylinder depicted on figure (3(a)). We note Ω_0 and Ω_1 the domain of each material, and Γ the interface between both domains. Note that $\Omega_i \cap \Gamma = \emptyset$ and $\Omega = \Omega_0 \cup \Omega_0 \cup \Gamma$.

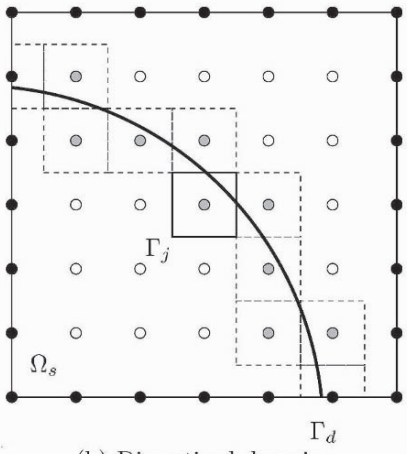
The governing equations are given by:

$$\lambda(\boldsymbol{x}) \left(T(\boldsymbol{x})_{,xx} + T(\boldsymbol{x})_{,yy}\right) = g \quad \text{for} \quad \boldsymbol{x} \in \Omega \tag{5.61}$$

with the conditions on the interface



(a) Two-dimensional heat conduction problem in a bi-material cylinder.



(b) Discretized domain.

Figure 3. Domains definition

$$\lambda_0 T_{,i}(\boldsymbol{x}^-)\boldsymbol{n}_i(\boldsymbol{x}) = \lambda_1 T_{,i}(\boldsymbol{x}^+)\boldsymbol{n}_i(\boldsymbol{x}) \quad \text{for} \quad \boldsymbol{x} \in \Gamma \tag{5.62}$$

$$T(\boldsymbol{x}^-) = T(\boldsymbol{x}^+) \quad \text{for} \quad \boldsymbol{x} \in \Gamma \tag{5.63}$$

where $\boldsymbol{n}$ is the unit outwards vector defined on the interface. The boundary conditions are

$$T(\boldsymbol{x}) = T_{int} \quad \text{for} \quad \boldsymbol{x} \in \Gamma_{int} \tag{5.64}$$

$$T(\boldsymbol{x}) = T_{ext} \quad \text{for} \quad \boldsymbol{x} \in \Gamma_{ext} \tag{5.65}$$

The analytical solution of this problem can be found in [2].

5.1 Analysis of the Enriched Approximation

In this numerical test we analyze the representation of the analytical solution of the problem just described. The evaluation is done on the domain Ω_s depicted in figure (3(b)). Table (1) defines the different enriched functional approximations according to ne and $\Theta(\boldsymbol{x})$. In this table $\Theta_c(\boldsymbol{x})$ is the distance function associated with the circular interface which results $\Theta_c(\boldsymbol{x}) = \sqrt{x^2 + y^2} - R_d^2$, and $\tilde{\Theta}_c(\boldsymbol{x})$ is its Taylor expansion evaluated using Eq. (4.60). In all cases $m = 2$ and ϕ_λ is the classical cubic spline function (see [9]). The domain is discretized by a regular grid as shown in figure (3(b)). The support radius of ϕ_λ is $2.6 \times dx$, where dx is the internodal distance along the coordinate axes. The other problem parameters are $R_{int} = 0.1$, $R_{ext} = 1$, $R_d = 0.9$, $\lambda_2 = 100$, $g = 1$, $T_{int} = 0$, $T_{ext} = 100$.

For each simulation scenario (related to a different enriched approximation in table (1)) a convergence test is carried out with a number of nodes varying from 21×21 to 101×101. Convergence rates using the L^2 and the H^1 norms are given in table (2).

We can notice that the different enriched approximations exhibit the same behavior in term of rate of convergence and accuracy. They manifest a significant increase in the order of convergence (of approximatively one order) with respect to the standard (non enriched) RKPA approximation.

Table 1. Definition of the different enriched approximations: $P2$ refers to the second polynomial degree, $D1$ to the enrichment based on the distance function, $D2$ to the enrichment based on the distance function as well as on its square; and finally T refers to the Taylor's expansion of that distance function.

Approximation	ne	$\Theta(\boldsymbol{x})$
P2	0	
P2D1	1	$\Theta_c(\boldsymbol{x})$
P2D1T	1	$\tilde{\Theta}_c(\boldsymbol{x})$
P2D2	2	$\Theta_c(\boldsymbol{x})$
P2D2T	2	$\tilde{\Theta}_c(\boldsymbol{x})$

Table 2. Convergence analysis of the different enriched approximations: slope (coordinate at the origin) of the $log(Error)$ versus $log(1/\Delta x)$ curves

Functional approximation	λ_1	Analysis of the approximation L^2	Analysis of the approximation H^1	Analysis of the discretization L^2	Analysis of the discretization H^1
P2	0.01	-1.47(-0.64)	-0.49(-0.44)	-0.52(-0.86)	-0.41(-0.35)
	1	-1.47(-0.54)	-0.49(-0.29)	-0.41(-0.91)	-0.40(-0.22)
	10	-1.47(-0.79)	-0.49(-0.34)	-0.16(-1.41)	-0.30(-0.44)
P2D1	0.01	-2.51(-1.25)	-1.50(-1.15)	-1.96(-0.66)	-1.45(-0.74)
	1	-2.44(-0.66)	-1.43(-0.49)	-1.83(-0.30)	-1.45(-0.14)
	10	-2.44(-0.79)	-1.44(-0.38)	-1.37(-0.97)	-1.43(-0.17)
P2D1	0.01	-2.51(-1.25)	-1.50(-1.15)	-1.96(-0.66)	-1.45(-0.74)
	1	-2.44(-0.66)	-1.43(-0.49)	-1.83(-0.30)	-1.45(-0.14)
	10	-2.44(-0.79)	-1.44(-0.38)	-1.37(-0.97)	-1.43(-0.17)
P2D1T	0.01	-2.26(-0.58)	-1.52(-0.40)	-1.20(-0.85)	-1.17(-0.38)
	1	-2.43(-0.39)	-1.48(-0.09)	-1.25(-0.65)	-1.29(-0.05)
	10	-2.43(-0.57)	-1.47(-0.08)	-1.12(-1.06)	-1.22(-0.28)
P2D2	0.01	-2.47(-0.71)	-1.48(-0.58)	-2.03(-0.45)	-1.33(-0.64)
	1	-2.47(-0.19)	-1.48(-0.02)	-1.98(-0.03)	-1.38(-0.04)
	10	-2.47(-0.39)	-1.48(-0.02)	-1.44(-0.96)	-1.43(-0.05)
P2D2T	0.01	-2.44(-0.88)	-1.46(-0.72)	-2.06(-0.47)	-1.23(-0.77)
	1	-2.44(-0.36)	-1.46(-0.14)	-2.10(+0.09)	-1.28(-0.16)
	10	-2.44(-0.53)	-1.46(-0.11)	-1.37(-1.04)	-1.42(-0.10)

5.2 Analysis of the Discretization of a Poisson Problem

The problem previously described is now solved using the mixed point-subdomain collocation technique deeply described in [6]. The exact solution is prescribed on the boundary of Ω_s. We consider three sets of nodes. The first one, named Γ_{bc} contains the nodes located on the domain boundary where the temperature is prescribed. The second set, named Ω_d, consists of the nodes whose associated Voronoi cells intersect the interface (see figure (3(b))). The last one, named Ω_i, contains all the remaining nodes. The discretization is performed as follows:

- For each node $\boldsymbol{x}_j$ in Ω_i we consider the usual point collocation:

$$\lambda(\boldsymbol{x}_j)\left(\sum_{i=1}^{NP}(\psi_i^{[(2,0)]}(\boldsymbol{x}_j)+\psi_i^{[(0,2)]}(\boldsymbol{x}_j))T_i\right)=g(\boldsymbol{x}_j) \tag{5.66}$$

- For each node $\boldsymbol{x}_j$ in Ω_d proceed from a subdomain collocation

$$\int_{\Gamma_j}\lambda(\boldsymbol{x})(T_{,x}n_x+T_{,y}n_y)\,d\Gamma_j=g(\boldsymbol{x}_j)A_j \tag{5.67}$$

  where Γ_j is the boundary of the Voronoi cell associated with the node $\boldsymbol{x}_j$, A_j being its area. The integration on Γ_j is done using 5 gauss points on each edge.

- For each node $\boldsymbol{x}_j$ in Ω_d we enforce the known temperature on the domain boundary

$$\sum_{i=1}^{NP} \psi_i(\boldsymbol{x}_j) T_i = T(\boldsymbol{x}_j) \tag{5.68}$$

A convergence analysis is achieved for each simulation scenario. The results are grouped in table (2).

We can notice again that the different enriched approximations exhibit the same behavior in term of rate of convergence and accuracy. They manifest a significant increase in the order of convergence (of approximatively one order) with respect to the standard (non enriched) RKPA approximation.

6 Conclusion

In this paper we have proposed a new approximation technique within the context of meshless methods able to reproduce functions with discontinuous derivatives. This approach involves some concepts of the reproducing kernel particle method (RKPM), which have been extended in order to reproduce functions with discontinuous derivatives. The accuracy of the proposed technique has been compared with usual RKP approximations (which only reproduces polynomials) evidencing a significant increase in the order of convergence. Moreover, a Taylor's expansion of the distance function used to define the enrichment, allowed restricting the enrichment to the regions where the enrichment is required.

References

1. G. Touzot B. Nayrolles and P. Villon, *Generalizing the finite element method: Diffuse approximation and diffuse elements computational mechanics*, Journal of Computational Mechanics **10** (1992), no. 5, 307–318.
2. R.C. Batra, M. Porfiri, and D. Spinello, *Treatment of material discontinuity in two meshless local petrov-galerkin (MLPG) formulations of axisymmetric transient heat conduction*, Int. J. Numer. Meths. Eng. **61** (2004), 2461–2479.
3. T. Belytschko, Y. Kronggauz, D. Organ, and M. Fleming, *Meshless methods: an overview and recent developments*, Comput. Meths. Appl. Mech. Engrg. **139** (1996), 3–47.
4. M. Fleming, Y.A. Chu, B. Moran, and T. Belytschko, *Enriched element-free galerkin methods for crack tip fields*, Int. J. Numer. Meths. Eng. **40** (1997).
5. R.A. Gingold and J.J. Monaghan, *Smoothed particle hydrodynamics: Theory and application to non-spherical stars*, Monthly Notices Royal Astronomical Society **181** (1977), 375–389.
6. D. W. Kim and Y. Kim, *Point collocation methods using the fast moving least-square reproducing kernel approximation*, Int. J. Numer. Meths. Eng. **56** (2003), 1445–1464.

7. Y. Kronggauz and T. Belytschko, *Efg approximation with discontinuous derivatives*, Int. J. Numer. Meths. Eng. **41** (1998), 1215–1233.
8. S. Li and W.K. Liu, *Meshfree and particle methods and their applications*, Appl. Mech. Rev. **55** (2002), 1–34.
9. W. K. Liu, S. Li, and T. Belytshko, *Moving Least Square Reproducing Kernel Method. (I) Methodology and convergence*, Comput. Meths. Appl. Mech. Engrg. **143** (1997), 113–145.
10. W.K. Liu, S. Jun, S. Li, J. Adee, and T. Belytschko, *Reproducing kernel particle methods for structural dynamics*, Int. J. Numer. Meths. Eng. **38** (1995), 1655–1679.