An automatic adaptive refinement procedure for the reproducing kernel particle method. Part II : adaptive refinement

Lee, Chi King; Shuai, Y. Y.

2006

Lee, C. K., & Shuai, Y. Y. (2007). An automatic adaptive refinement procedure for the reproducing kernel particle method. Part II: Adaptive refinement. *Computational Mechanics*, 40(3), 415-427.

<https://hdl.handle.net/10356/103289>

<https://doi.org/10.1007/s00466-006-0113-2>

© 2006 Springer. This is the author created version of a work that has been peer reviewed and accepted for publication by *Computational Mechanics*, Springer. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [Article DOI: <http://dx.doi.org/10.1007/s00466-006-0113-2>].

Downloaded on 26 Apr 2023 18:35:33 SGT

An automatic adaptive refinement procedure for the reproducing kernel particle method Part II: Adaptive refinement

*C. K. Lee and Y. Y. Shuai

*School of Civil and Environmental Engineering
Nanyang Technological University
Nanyang Avenue
Singapore 639798*

*email: ccklee@ntu.edu.sg

Pre-printed version of the paper appeared in the journal: Computational Mechanics, Vol. 40, No. 3, pp415-427

Abstract

In Part II of this study, an automatic adaptive refinement procedure using the reproducing kernel particle method (RKPM) for the solution of 2D linear boundary value problems is suggested. Based in the theoretical development and the numerical experiments done in Part I of this study, the Zienkiewicz and Zhu (Z-Z) error estimation scheme is combined with a new stress recovery procedure for the *a posteriori* error estimation of the adaptive refinement procedure. By considering the *a priori* convergence rate of the RKPM and the estimated error norm, an adaptive refinement strategy for the determination of optimal point distribution is proposed. In the suggested adaptive refinement scheme, the local refinement indicators used are computed by considering the partition of unity property of the RKPM shape functions. In addition, a simple but effective variable support size definition scheme is suggested to ensure the robustness of the adaptive RKPM procedure. The performance of the suggested adaptive procedure is tested by using it to solve several benchmark problems. Numerical results indicated that the suggested refinement scheme can lead to the generation of nearly optimal meshes for both smooth and singular problems. The optimal convergence rate of the RKPM is restored and thus the effectivity indices of the Z-Z error estimator are converging to the ideal value of unity as the meshes are refined.

KEYWORDS: *Reproducing Kernel Particle Method; Stress Recovery Procedure; A posteriori error estimation; Adaptive refinement.*

1. Introduction

After nearly three decades' research, the development of an automatic adaptive refinement procedure for the finite element (FE) methods can be considered as a mature area in computational mechanics. This fact is confirmed by the large number of references quoted in the bibliographies and special issues related to adaptive analysis and meshing [1-5] published recently. In fact, many effective adaptive refinement procedures had been implemented in many research programs [6] and even in commercial packages [7]. When comparing with the FE methods, the development of an automatic adaptive refinement procedure for the meshless methods is a relatively new research area and only received attentions in recent years. As a result, fewer papers related to this topic had been published to date [8-14]. One of the possible explanations for such observation is that many meshless methods like the reproducing kernel particle method (RKPM) [15, 16] and the element-free Galerkin method (EFGM) [17, 18] are relatively new numerical methods that only received major attention since the last decade. As a result, a reliable *a posteriori* error estimator for the meshless methods, which is an essential ingredient and pre-requisite for any adaptive refinement procedure, is only available in the past few years [19-25]. However, it should be pointed out that once a reliable *a posteriori* error estimator is available, development and implementation of an automatic adaptive refinement procedure for the meshless methods is an easier task when compared with the case of the FE method. The main reason is that when compared with the FE method, the meshless methods enjoy more flexibility in model generation and no complicated algorithm is needed for the generation of an adaptive finite point meshes even for surfaces [26] and 3D [27] problem domains.

In this paper, an automatic adaptive refinement procedure for the RKPM is suggested. The developed refinement procedure will employ the new stress extraction scheme which is described in Part I of the present study [25] for the construction of an extracted (recovered) stress field. The extracted stress field will then be used with the Zienkiewicz and Zhu (Z-Z) error estimation approach [28] for the *a posteriori* error estimation of the RKPM solution. A new adaptive refinement strategy will be suggested to control the discretization error of the RKPM solution. New point spacing required to reduce the discretization error of the RKPM solution will be determined by making use of the *a priori* convergence rate and the quantitative error distribution information provided by the *a posteriori* error estimator. The refined point mesh is then generated by a simple

point mesh generation scheme [26]. The suggested automatic adaptive refinement procedure will be applied for the solutions of four classic benchmark problems with known exact solutions and its performance and effectiveness will be assessed by comparing the convergence rates obtained with the results of uniform refinement.

Note that when compared with the adaptive refinement strategy developed in reference [24] for the EFGM, two new features are adopted in the current studies. They are (i) a new adaptive refinement strategy using the new concept of error norm contribution associated with nodes and the partition-of-unity property of the RKPM shape functions, and (ii) fully unstructured point meshes are generated when compared with the “semi-unstructured” point meshes generated by the subdivision scheme used in reference [24].

2. *A priori and a posteriori error estimations for the RKPM*

2.1 *RKPM approximation, error measures and a priori error estimation*

In the RKPM, $\hat{\mathbf{u}}$, the approximated solution for the exact displacement $\mathbf{u}$, can be written as

$$\hat{\mathbf{u}}(x, y) = \sum_{i=1}^{NP} N_i(x, y) \hat{\mathbf{u}}_i = \mathbf{N}^T \hat{\mathbf{u}} \quad (1)$$

where $\mathbf{N}$ is the matrix of the RKPM shape functions and $\hat{\mathbf{u}}$ is the vector of nodal displacement values. NP is the number of nodes in the whole problem domain. N_i and $\hat{\mathbf{u}}_i$ are the RKPM shape function and the nodal value of node i respectively. Note that N_i is a compact function that is non-zero only over Ω_i , the subdomain of influence associates with node i with characteristics (support) size h_i .

With the approximation of $\mathbf{u}$ given by Eqn. 1, the pointwise $\mathbf{e}_\sigma$ and normed $\|\mathbf{e}\|$ measures of the error of the RKPM solution are given by

$$\mathbf{e}_\sigma = \sigma - \hat{\sigma} \quad \text{and} \quad \|\mathbf{e}\| = \left\{ \int_{\Omega} (\mathbf{e}_\sigma)^T \mathbf{D}^{-1} (\mathbf{e}_\sigma) d\Omega \right\}^{1/2} \quad (2)$$

In Eqn. 2, σ and $\hat{\sigma}$ are, respectively, the exact and the RKPM stress fields defined over the problem domain Ω . For the analysis of standard elliptic boundary value problems, the *a priori* error estimation for the convergence of the RKPM solution can be cast in the form

$$\|\mathbf{e}\| \leq Ch^{\min(p, \lambda)} \quad (3)$$

where C is a constant independent of h and p , the typical spacing of node (support size) in the point mesh and the order of exactness of the RKPM shape function respectively. $\lambda < 1$ is the strength of singularity of the exact solution [29, 30].

For 2D problems, the relationship between h and N_{DOF} , the number of degrees of freedom of the point mesh, is given by

$$h \propto (N_{\text{DOF}})^{-1/2} \quad (4)$$

Hence, Eqn. 3 can be rewritten as

$$\|\mathbf{e}\| \leq C'(N_{\text{DOF}})^{-\min(p, \lambda)/2} \quad (5a)$$

In Eqn. 5, C' is another constant independent of N_{DOF} and p . The main objective of an adaptive refinement procedure is to search for an optimal point distribution such that the effect of the singularity will be eliminated. If such optimal distribution is achieved, the convergence rate of the RKPM solution will be increased to [28, 29]

$$\|\mathbf{e}\| \leq C'(N_{\text{DOF}})^{-p/2} \quad (5b)$$

In general, the absolute value of the error norm $\|\mathbf{e}\|$ will vary from problem to problem. Hence, during an adaptive refinement, it is more convenient to use the *relative error* of the RKPM solution, η_{RKPM} , defined as

$$\eta_{\text{RKPM}} = \frac{\|\mathbf{e}\|}{\|\mathbf{u}\|} \quad \|\mathbf{u}\| = \left\{ \int_{\Omega} \boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma} d\Omega \right\}^{1/2} \quad (6)$$

In Eqn. 6, $\mathbf{D}$ is the material matrix and $\|\mathbf{u}\|$ is the exact energy norm of the problem.

2.2 Stress recovery and a posteriori error estimation

In this study, the *a posteriori* estimated error norm of the RKPM solution $\|\bar{\mathbf{e}}\|$ is obtained by using the Z-Z approach [28] by replacing the exact stress field in Eqn. 2 with a recovered (extracted) stress field $\tilde{\boldsymbol{\sigma}}$ such that

$$\bar{\mathbf{e}}_{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}, \quad \|\bar{\mathbf{e}}\| = \left\{ \int_{\Omega} (\bar{\mathbf{e}}_{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} (\bar{\mathbf{e}}_{\boldsymbol{\sigma}}) d\Omega \right\}^{1/2} = \left\{ \int_{\Omega} (\tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} (\tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}) d\Omega \right\}^{1/2} \quad (7)$$

The recovered stress field $\tilde{\sigma}$ is generated by a two-step approach. Firstly, an extraction approach [24] is applied to recover the stress tensor at all the nodes of the point mesh in a way such that

$$\tilde{\sigma}_i(x_i, y_i) = \Psi(x_i, y_i, r_{R,i}) \hat{\sigma} + \Phi(x_i, y_i, r_{R,i}) \quad \text{for } i=1, \dots, NP \quad (8)$$

In Eqn. 8, $\tilde{\sigma}_i(x_i, y_i)$ is the extracted stress tensor at node i with coordinates (x_i, y_i) . $\Psi(x_i, y_i, r_{R,i})$ and $\Phi(x_i, y_i, r_{R,i})$ are, respectively, a boundary extraction operator and a load term operator [25]. $r_{R,i}$ is the size of the recovery subdomain for node i which is a function of h_i , the size of the domain of influence associates with node i .

Secondly, the recovered stress field $\tilde{\sigma}$ is constructed by re-using the shape functions of the RKPM such that

$$\tilde{\sigma}(x, y) = \sum_{i=1}^{NP} N_i \tilde{\sigma}_i \quad (9)$$

In case that if the exact solution of the stress field is known, it is possible to compute the error norm of the recovered stress $\|\tilde{e}\|$, the relative error of the recovered stress η_{Rv} and the effectivity index of the *a posteriori* error estimator θ such that

$$\tilde{e}_\sigma = \sigma - \tilde{\sigma} \quad \|\tilde{e}\| = \left\{ \int_{\Omega} (\tilde{e}_\sigma)^T \mathbf{D}^{-1} (\tilde{e}_\sigma) d\Omega \right\}^{1/2} \quad (10a)$$

$$\eta_{Rv} = \frac{\|\tilde{e}\|}{\|\mathbf{u}\|} \quad (10b)$$

$$\theta = \frac{\|\tilde{e}\|}{\|e\|} \quad (11)$$

It can be shown that [31] that θ will asymptotically converge to unity provides that the convergence rate of the recovered stress field $\tilde{\sigma}$ is higher than that of $\hat{\sigma}$ as N_{DOF} approaches infinity. That is, if

$$\lim_{N_{DOF} \rightarrow \infty} \frac{\|\tilde{e}\|}{\|e\|} = 0 \quad \Rightarrow \quad \lim_{N_{DOF} \rightarrow \infty} \theta = 1 \quad (12)$$

It should be noted that in Part I of this study [25], it was shown that the recovered stress field obtained by using Eqns. 8 and 9 is indeed converging at a rate higher than the RKPM stress field.

3. The adaptive refinement strategy used and point mesh generation

3.1 Variable support size determination for an adaptive RKPM mesh

In an adaptive refinement analysis, unlike the case of uniform refinement in which the spacings for all the nodes are roughly the same, the spacings of nodes in an adaptive point mesh could vary greatly in the problem domain. Hence, a constant support size for all the nodes in the mesh will not be able to reflect the local node spacing of an adaptive mesh. If support sizes for all the nodes are set equal to the smallest node spacing in the mesh, it is possible that the problem domain may not be covered completely. Furthermore, the shape functions for some nodes located at where the node spacing is large may fail to be constructed. However, setting the support size based on the maximum node spacing will increase the computational cost and reduce the accuracy of the RKPM solution [32]. As a result, during the adaptive refinement analysis, a variable support size scheme to define the domains of influence for nodes should be adopted. In the current implementation, the support size for nodes in the mesh will be determined by imposing the follow two essential conditions:

(1) Minimum support size condition

For all the nodes in the problem domain, their domains of influence must contain at least six nodes (Fig. 1).

(2) Minimum cover condition

The union of the domains of influence of all the nodes in the mesh must completely cover the whole problem domain.

The minimum support size condition is a necessary (but not a sufficient) condition for the generation of RKPM shape function when *linear* RKPM (i.e. $p=1$ in Eqn. 5) is used in the analysis. If the RKPM shape function is required to reproduce exactly a linear function, the correction function [15, 16] must be at least of second (quadratic) order. Hence, in 2D analysis, the support domain for any node must at least contain six nodes (Fig. 1).

The minimum cover condition simply means that the problem domain must be completely covered by the union of the domains of influence of all the nodes. It is a necessary condition for the RKPM shape functions to fulfil the partition of unity condition. That is

$$\Omega = \bigcup_{i=1}^{NP} N_i \Omega_i \quad (13)$$

where Ω_i is the domain of influence associates with node i .

Note that the minimum size condition can only define the lower bound of the support size of nodes in the mesh. Fulfilment of this condition obviously does not imply that the minimum cover condition is satisfied automatically. In addition, it should be mentioned that the minimum support size condition is not sufficient to ensure the existence and uniqueness of the RKPM shape functions and some geometric conditions about the distribution of nodes (e.g. a sufficient number of node in the domain of influence must not be collinear) are needed. In general, it is always preferable to perform some stability checks during the computation of the RKPM shape functions [16] and use the minimum support size whenever is possible as it will reduce the computational cost and increase the accuracy of the RKPM solution. In this study, the support sizes of the nodes in the point mesh will be determined by the following steps.

- (i) Scan over all the nodes within the point mesh, compute h_i , the minimum support sizes for node i , $i=1, \dots, NP$ such that they will satisfy the minimum support size condition. In addition, $d_{\min}$, the minimum distance between any two nodes of the mesh will be computed.
- (ii) A virtual grid which contains square cells with size equal to $d_{\min}$ will be generated to cover the entire problem domain. For a typical grid cell i , the centre of the cell is denoted as O_i .
- (iii) Scan over all the cells that overlap with the problem domain. For a typical cell i (Fig. 2), locate the six nearest nodes that are visible to O_i . The distances between these six nodes with O_i , d_k , $k=1, \dots, 6$ are computed.
- (iv) Check whether the original minimum support sizes of these six nodes satisfy the condition

$$h_k \geq d_k + \sqrt{1/2}d_{\min} \quad \text{for } k = 1, \dots, 6 \quad (14)$$

For the k th node, the minimum support size h_k will be kept if Eqn. 14 is satisfied.

Otherwise, the value of h_k will be increased to $h_k \leftarrow d_k + \sqrt{1/2}d_{\min}$.

The above procedure will ensure that (i) any point of the problem domain will be inside the domain of influence associates to at least one node of the mesh and (ii) the RKPM shape function can be constructed at any point inside the problem domain. Note that the second condition is essential not only for the RKPM analysis but also for the evaluations of the recovered stress field (Eqn. 9) and the estimated error norms (Eqns. 7 and 10) when the RKPM shape functions are required to be evaluated at the integration points of

the problem domain. Details of the proof for these two conditions could be found in the Appendix.

3.2 Adaptive refinement strategy for the RKPM

3.2.1 Acceptable RKPM solution

In any adaptive refinement procedure, the ultimate objective is to achieve an RKPM solution with relative error, η_{RKPM} , less than a user prescribed target value, η_{tar}

$$\eta_{\text{RKPM}} \leq \eta_{\text{tar}} \quad (15)$$

Since during the adaptive refinement analysis, the exact energy norm of the problem (Eqn. 6) is not known in advance, it is approximated by $\|\hat{\mathbf{u}}\|$, the energy norm of the RKPM solution (c.f. Eqn. 6)

$$\|\mathbf{u}\| \approx \|\hat{\mathbf{u}}\| = \left\{ \int_{\Omega} \hat{\boldsymbol{\sigma}}^T \mathbf{D}^{-1} \hat{\boldsymbol{\sigma}} d\Omega \right\}^{1/2} \quad (16)$$

The estimated relative error of the RKPM solution $\bar{\eta}$ can be computed as

$$\eta_{\text{RKPM}} \approx \bar{\eta} = \frac{\|\bar{\mathbf{e}}\|}{\|\hat{\mathbf{u}}\|} \quad (17)$$

The adaptive refinement RKPM procedure will be terminated if the following accuracy condition is reached (c.f. Eqn. 15)

$$\bar{\eta} \leq \eta_{\text{tar}} \quad \text{or} \quad \|\bar{\mathbf{e}}\| \leq \eta_{\text{tar}} \|\hat{\mathbf{u}}\| \quad (18)$$

3.2.2 Global and local refinement indicators and new node spacing calculation

From the experience obtained from adaptive FE analyses [28,31,33-37] and EFGM analyses [8,9,14,23,24], an adaptive refinement procedure will be optimal only when both the global and local errors of the solution are controlled properly and distributed equally among all the degrees of freedom of the numerical model. Since the recovered stress field could be evaluated at any point inside Ω , the notion of estimated error density, $\bar{e}_d$ can be introduced as

$$\bar{e}_d = \gamma^2 (\bar{\mathbf{e}}_{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} (\bar{\mathbf{e}}_{\boldsymbol{\sigma}}) = \gamma^2 (\tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} (\tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}) \quad (19)$$

where γ is a compensation factor >1 employed to improve the adaptive efficiency and accuracy of the error indicators. A value of $\gamma=1.2$ is used in this study.

Using Eqn. 19, the global error in energy norm for the whole problem domain can be re-defined as (c.f. Eqn. 7)

$$\|\bar{\mathbf{e}}\| = \gamma \left\{ \int_{\Omega} \bar{\mathbf{e}}_d d\Omega \right\}^{1/2} \quad (20)$$

Hence, using Eqn. 18, a *global refinement indicator*, ξ_g , can be defined as

$$\xi_g = \frac{\|\bar{\mathbf{e}}\|}{\eta_{\text{tar}} \|\hat{\mathbf{u}}\|} \quad (21)$$

Obviously, a value of $\xi_g \leq 1$ implies that the global target accuracy criterion is satisfied while $\xi_g > 1$ indicates further refinement is needed. Note that the global error indicator can only indicate whether further refinement is needed or not. It does not indicate where and how to refine the point mesh. Thus, in order to construct an optimal point mesh that will be able to reduce the effect of singularity, a *local error refinement* indicator is required. Ideally, in an optimal mesh [38], the error contribution should be equally distributed among all the degrees of freedom of the model. In case of finite element analysis, it implies that the error norm should be equally distributed among all the elements of the mesh. In the meshless methods, since the domains of influence of nodes in the point mesh will overlap each another, the notion of error norm contribution from individual node of the mesh should be different from that of adaptive FE analysis. Recall that the RKPM shape functions and the domains of influence of nodes will satisfy the partition of unity condition (Eqn. 13), the global error norm of the RKPM solution (Eqn. 2) can be re-written in the form

$$\|\mathbf{e}\| = \left\{ \int_{\Omega} (\mathbf{e}_{\sigma})^T \mathbf{D}^{-1} (\mathbf{e}_{\sigma}) d\Omega \right\}^{1/2} = \left\{ \int_{\Omega} \sum_{i=1}^{\text{NP}} N_i ((\mathbf{e}_{\sigma})^T \mathbf{D}^{-1} (\mathbf{e}_{\sigma})) d\Omega \right\}^{1/2} \quad (22a)$$

By changing the order of integration of Eqn. 22a, one has

$$\|\mathbf{e}\| = \left\{ \sum_{i=1}^{\text{NP}} \left(\int_{\Omega_i} N_i ((\mathbf{e}_{\sigma})^T \mathbf{D}^{-1} (\mathbf{e}_{\sigma})) d\Omega \right) \right\}^{1/2} = \left\{ \sum_{i=1}^{\text{NP}} (\|\mathbf{e}\|_i)^2 \right\}^{1/2} \quad (22b)$$

where $\|\mathbf{e}\|_i$ is the local error norm contribution *associates* with node i such that

$$\|\mathbf{e}\|_i = \left\{ \int_{\Omega_i} N_i(\mathbf{e}_\sigma)^T \mathbf{D}^{-1}(\mathbf{e}_\sigma) d\Omega \right\}^{1/2}, \quad i = 1, \dots, NP \quad (23)$$

The equal error norm contribution condition implies that in an optimal point mesh, for any two nodes i and j in the problem domain, the following conditions should be valid

$$\|\mathbf{e}\|_i = \|\mathbf{e}\|_j \quad \text{for } i, j = 1, \dots, NP \quad (24a)$$

In Eqn. 24a, $\|\mathbf{e}\|_i$ and $\|\mathbf{e}\|_j$ are, respectively, the error norm contributions associate with the nodes i and j . In case that the above equal distribution of error norm condition is achieved, one can write

$$\|\mathbf{e}\|^2 = \sum_{i=1}^{NP} (\|\mathbf{e}\|_i)^2 = NP(\|\mathbf{e}\|_i)^2 \Rightarrow \|\mathbf{e}\|_i = \frac{\|\mathbf{e}\|}{\sqrt{NP}} \quad \text{for } i = 1, \dots, NP \quad (24b)$$

During the adaptive refinement process, the estimated error norm contribution associates with node i of the point mesh can be defined in a similar manner (c.f. Eqn. 23)

$$\|\bar{\mathbf{e}}\|_i = \gamma \left\{ \int_{\Omega_i} N_i(\tilde{\mathbf{e}}_\sigma)^T \mathbf{D}^{-1}(\tilde{\mathbf{e}}_\sigma) d\Omega \right\}^{1/2} = \gamma \left\{ \int_{\Omega_i} N_i \bar{\mathbf{e}}_d d\Omega \right\}^{1/2} \quad (25a)$$

It is hoped that during the adaptive refinement, an equilibration of local estimated error norm contribution can be achieved. That is (c.f. Eqn. 24)

$$\|\bar{\mathbf{e}}\|_i = \|\bar{\mathbf{e}}\|_j \quad \text{and} \quad \|\bar{\mathbf{e}}\|_i = \frac{\|\bar{\mathbf{e}}\|}{\sqrt{NP}} \quad \text{for } i, j = 1, \dots, NP \quad (25b)$$

With the above results, the local error indicator for node i , ξ_i , can be defined as

$$\xi_i = \frac{\|\bar{\mathbf{e}}\|_i}{(\|\bar{\mathbf{e}}\|/\sqrt{NP})} \quad (26a)$$

Using Eqn. 18, Eqn. 26a can be more conveniently written in the form

$$\xi_i = \frac{\|\bar{\mathbf{e}}\|_i}{\|\mathbf{e}\|_{\text{allow}}} = \frac{\sqrt{NP} \|\bar{\mathbf{e}}\|_i}{\eta_{\text{tar}} \|\hat{\mathbf{u}}\|} \quad (26b)$$

where in Eqn. 26b, the term $\|\mathbf{e}\|_{\text{allow}}$ is the allowable error norm contribution and is given by

$$\|\mathbf{e}\|_{\text{allow}} = \frac{\eta_{\text{tar}} \|\hat{\mathbf{u}}\|}{\sqrt{\text{NP}}} \quad (26c)$$

Obviously, a value of $\xi_i > 1$ means that the point mesh requires local refinement at node i . It should be stressed that from Eqn. 23, $\|\mathbf{e}\|_i$, the error norm contribution for node i is **not** equal to the error norm defined **over** Ω_i , the domain of influence associates with node i . That is

$$\|\mathbf{e}\|_i \neq \|\mathbf{e}\|_{\Omega_i} = \left(\int_{\Omega_i} (\mathbf{e}_{\sigma})^T \mathbf{D}^{-1} (\mathbf{e}_{\sigma}) d\Omega \right)^{1/2} \quad (27a)$$

In fact, since the domains of influence of nodes of the point mesh will overlap each another, the following inequality will be held (c.f. Eqn. 24b)

$$\|\mathbf{e}\|^2 \leq \sum_{i=1}^{\text{NP}} \left(\|\mathbf{e}\|_{\Omega_i} \right)^2 \quad (27b)$$

However, the *a priori* convergence rate of Eqn. 3 is also applicable to $\|\mathbf{e}\|_i$. That is

$$\|\mathbf{e}\|_i \leq C(h_i)^{\beta_i} \quad (28)$$

where h_i is the local node spacing (support size, Section 3.1) defined for node i . β_i is the local convergence rate for node i and is given by

$$\beta_i = \begin{cases} p, & \text{if } \Omega_i \text{ contains no singular point} \\ \lambda, & \text{if } \Omega_i \text{ contains a singular point with strength} = \lambda \end{cases} \quad (29)$$

With the definitions of the global (ξ_g) and the local (ξ_i) error indicators, the new node spacing of the point mesh can be computed by the following steps.

- (i) Retrieve the existing node spacing information, h_i^{old} from the existing mesh.
- (ii) Compute the global error indicator, ξ_g and the local error indicators ξ_i , $i=1, \dots, \text{NP}$.
- (iii) Check for convergence condition

The adaptive refinement procedure will be stopped if it is found that the condition $\xi_g \leq 1$ is true. Otherwise, the new node spacing will be computed by using step (iv).

(iv) Define the new node spacing, h_i^{new} .

By using Eqn. 28, if h_i^{new} is the required new node spacing for node i , one has

$$\|\mathbf{e}\|_i \approx C(h_i^{\text{old}})^{\beta_i} \quad \text{and} \quad \|\mathbf{e}\|_{\text{allow}} \approx C(h_i^{\text{new}})^{\beta_i} \quad (30a)$$

Hence, by using Eqn. 26b, the relationship between h_i^{old} and h_i^{new} can be expressed as

$$\xi_i = \frac{\|\mathbf{e}\|_i}{\|\mathbf{e}\|_{\text{allow}}} = \left(\frac{h_i^{\text{old}}}{h_i^{\text{new}}} \right)^{\beta_i} \quad (30b)$$

Therefore, the new node spacing for node i , h_i^{new} can be computed as

$$h_i^{\text{new}} = \frac{h_i^{\text{old}}}{(\xi_i)^{-1/\beta_i}} \quad (30c)$$

3.3 *New point mesh generation and determination of h_i*

The new point mesh spacing information given by Eqn. 30c will be defined at the nodal point of the current mesh. It can be directly converted to a set of metric specification [39] for the generation of the new adaptive finite point mesh. In this study, the surface point mesh generator used is the one developed by Lee [26] that employs on the boundary offsetting technique to generate nodal points and the metric specification to control the node spacing of the mesh. After the new point mesh is generated, the support sizes for the nodal points of the mesh will be determined by using the procedure described in Section 3.1 so that the minimum support size and minimum cover conditions will be satisfied.

4. **Numerical Examples**

4.1 *Scope of study*

In this section, the four benchmark problems with known exact solutions which are used in Part I of the present study [25] will be re-used to investigate the performance of the proposed adaptive RKPM refinement procedure. Detailed descriptions of the benchmark problems can be found in reference [25]. A summary of the key properties of these problems is listed in Table 1.

In order to validate the effectiveness of the adaptive refinement procedure, both uniform and adaptive refinement will be carried out. The point meshes employed during the uniform refinements are the same as those used in Part I of the current study. In all examples, a coarse uniform mesh will be used as the initial mesh for the adaptive analysis. The target relative errors (η_{tar} , Eqn. 15) for the four benchmark problems are

listed in Table 1. The adaptive procedure will be continued until Eqn. 15 is satisfied or when three refinements (four analyses) are carried out. For all the uniform and adaptive analyses done, the following results will be computed:

- (1) The exact relative error of the RKPM solution, η_{RKPM} (Eqn. 6).
- (2) The relative error of the recovered solution, η_{RV} (Eqn. 10b).
- (3) The estimated relative error, $\bar{\eta}$ (Eqn. 17) computed with and without using the compensation factor γ (Eqns. 19 and 20).
- (4) The convergence rate of the RKPM solution, R_{RKPM} , defined as

$$R_{\text{RKPM}} = \frac{\log(\eta_{\text{RKPM},i} / \eta_{\text{RKPM},i+1})}{\log(N_{\text{DOF},i+1} / N_{\text{DOF},i})} \quad (31)$$

where $\eta_{\text{RKPM},i}$ and $N_{\text{DOF},i}$ are the relative error and the number of degrees of freedom of the i th refinement analysis respectively.

- (5) The convergence rate of the recovered stress, R_{RV} , defined as

$$R_{\text{RV}} = \frac{\log(\eta_{\text{RV},i} / \eta_{\text{RV},i+1})}{\log(N_{\text{DOF},i+1} / N_{\text{DOF},i})} \quad (32)$$

with $\eta_{\text{RV},i}$ is the relative error for the recovered solution obtained from the i th refinement analysis.

- (6) The effectivity index of the *a posteriori* error estimator, θ (Eqn. 11) computed with and without using the compensation factor γ (Eqns. 19 and 20).

In all the analyses, linear RKPM ($p=1$, Eqn. 5) will be applied for the solution of the problem. Hence, it is expected that the optimal convergence rate (with respect to N_{DOF}) of the RKPM solution should equal to $p/2=0.5$. During the RKPM analysis, the penalty parameter α (Eqn. 8 of reference [25]) used will be set equal to $\alpha=5 \times 10^6 E$ where E is the Young's modulus of the problem. For the stress recovery procedure, based on the numerical results obtained in Part I of the present study, the radius of subdomain for stress extraction r_R (Eqn. 46 of reference [25]) is set equal to $2h$ where h is the support size of the node under consideration.

4.2 Numerical results

4.2.1 Example 1: Timoshenko beam (convex domain, smooth solution)

For this example, all the results obtained are summarized in Table 2 while the adaptive point meshes generated are shown in Fig. 3. The rates of convergence for both uniform and adaptive refinements are plotted in Fig. 4. From Table 2 and Fig. 4, it can be seen that the optimal convergence rate was achieved by both uniform and adaptive refinements with adaptive refinement slightly outperformed the uniform refinement. In fact, it can be seen from Fig. 3 that even the exact solution for this problem is smooth, spacings of nodes were reduced near the support end of the beam in which high stress gradients are located. Regarding the effectivity indices obtained, from Table 2, it can be seen that regardless whether the compensation factor γ were used or not, their values were converging to unity as the meshes were refined. This implies that both the use of the compensation factor and the adaptive strategy contributed to the improvements of the effectivity indices. Furthermore, the recovered stress fields are more accurate and converged at a higher rate than the RKPM solution.

4.2.2 Example 2: infinite plate with a hole (concave domain, smooth solution)

For this example, the results obtained for adaptive refinement are shown in Table 3. The adaptive meshes generated and the convergence rates are shown in Figs. 5 and 6 respectively. From the numerical results obtained, it can also be concluded that adaptive refinement again slightly outperformed the uniform refinement and this can be explained by the fact that denser nodes were generated near the opening of the plate where the stress gradients are high. Note that the adaptive point meshes generated are not symmetric with respect to the line $\theta=\pi/4$ since the exact solution of stress (and hence the boundary tractions) is not symmetric (Eqn. 48 of reference 25). The recovered stress field are more accurate and were converging at a higher rate than the RKPM solution. Similar to the first example, due to the use of the compensation factor and the adaptive strategy, the effectivity indices were converging to the ideal value of unity.

4.2.3 Example 3: Region near the tip of a crack (convex domain, singular solution)

For this problem, the results obtained are listed in Table 4 and the adaptive meshes generated are shown in Fig. 7. The convergence history of the adaptive refinement is plotted in Fig. 8. From Table 4 and Fig. 8, it can be seen that significant improvement in

the rate of convergence was obtained (0.25 for uniform refinement, ≈ 0.40 for adaptive refinement). During adaptive refinement, rapid reduction of node spacing was carried out near the singular point and the effect of singularity was significantly reduced. Even though in this problem a singular point exists inside the problem domain, both the accuracy and convergence rates of the recovered solutions are higher than the RKPM solutions. As a result, convergence of the effectivity indices, **again, with and without using the compensation factor** can be observed. Note that the relatively poor value of the effectivity index for the first refinement mesh may be due to the coarse initial mesh used ($N_{\text{DOF}}=30$). From this example, it can be concluded that the adaptive refinement scheme suggested and the point mesh generation procedure successfully led to the generation of nearly optimal point meshes.

4.2.4 Example 4: L-shape domain (concave domain, singular solution)

In the last example, the results obtained from adaptive refinement are listed in Table 5 while the adaptive point meshes and the convergence history are shown in Fig. 9 and Fig. 10 respectively. Similar to Example 3, significant improvement in the convergence rate was obtained during adaptive refinement (from 0.27 for uniform refinement to 0.4 for adaptive refinement). Extensive reduction in node spacing can be observed near the re-entrance corner and both the accuracy and the convergence rates of the recovered solutions are higher than the RKPM solution. Similar to the case of Example 2, due to the asymmetric property of the exact stress field (Eqn. 50 of reference 25), the adaptive point meshes generated are not symmetric with respect to the x axis. Regarding the performance of the *a posteriori* error estimator, it can be concluded that **regardless whether the compensation factor was used or not, asymptotic convergence trend of the effectivity indices can be observed**. Again, the relatively poor results for the initial and first refinement meshes were due to the coarse initial mesh ($N_{\text{DOF}}=66$) used. In this example, the adaptive refinement scheme and the point mesh generator again largely reduced most of the effects of the singular point.

Regarding the computational complexity of the adaptive refinement procedure, Table 6 shows the ratios of CPU time used by all major processes (RKPM analysis, error estimation and adaptive refinement) of the adaptive refinement procedure. From Table 6, it can be seen that the computational costs for error estimation and adaptive refinement

are modest when compared with the RKPM analysis cost. This can be explained by the fact that while the computational complexity of the RKPM analysis is in general proportional to $(N_{\text{DOF}})^{1.5}$, the computational complexities for both error estimation and adaptive refinement are only proportional to N_{DOF} .

5. Conclusions

In this study, a simple and automatic adaptive refinement procedure for the reproducing kernel particle method (RKPM) is suggested. In this procedure, the Zienkiewicz and Zhu (Z-Z) error estimator is combined with a new stress recovered scheme for the *a posteriori* error estimation of the RKPM solution. A new adaptive refinement strategy based on the partition of unity property of the RKPM shape functions for the definition of the local refinement indicators is suggested. The local refinement indicators are then used for the definition of new node spacings during the adaptive point mesh generation. In addition, a simple but effective variable support size definition scheme has also been suggested for adaptive meshes to ensure that the RKPM shape functions can be constructed in any point inside the problem domain.

Numerical studies were carried out on four benchmark problems and it was found that by using the proposed adaptive refinement procedure, the user prescribed target accuracy can be reached in a few refinement steps. For problems with smooth exact solutions, even though no significant improvement in the convergence rate was obtained, the adaptive scheme could lead to solutions with higher accuracy. For problems with singular solutions, it was found that the proposed refinement process could lead to the generation of nearly optimal point meshes. As a result, when compared with uniform refinement, the effect of singularity was largely reduced and nearly optimal convergence rates were achieved. The numerical tests also re-confirmed that the recovered stress fields generated by the recovery procedure suggested in Part I of the current study [25] converged at a higher rate than the RKPM solutions.

References

1. Mackerle, J., (2001), "Error estimates and adaptive finite element methods: A bibliography (1990-2000)", Engineering Computations, Vol. 18 pp. 802-914.
2. Mackerle, J., (2001), "2D and 3D finite element meshing and remeshing: A bibliography (1990-2001)", Engineering Computations, Vol. 18 pp. 1108-1197.
3. Special Issue on Adaptive Meshing, Part I, Fin. Ele. Anal. Des., 25, No. 1/2, (1997), Guest Editor: S. H. Lo, pp. 1-198
4. Special Issue on Adaptive Meshing, Part II, Fin. Ele. Anal. Des., **25**, No. 3/4, (1997), Guest Editor: S. H. Lo, pp 199-350
5. Lo S. H., Finite element mesh generation and adaptive meshing, Progress in Structural Engineering and Materials, **4**, 381-300 (2002)
6. Talyor R. L. FEAP, A finite element analysis program version 7.5 Released, <http://www.ce.berkeley.edu/~rlt/feap/>
7. LUSAS, MYSTRO user guide, FEA Ltd, UK, 1997.
8. Duarte C. A. M., Oden J. T. (1996), "A hp Adaptive Method Using Clouds", Computer Methods in Applied Mechanics and Engineering, Vol. 139, pp. 237-262.
9. Haussler-Combe U., Korn C. (1998), "An adaptive approach with the Element-Free-Galerkin method", Computer Methods in Applied Mechanics and Engineering, Vol. 162, pp. 203-222.
10. Jun S. and Im. S. (2000), "Multiple-scale meshfree adaptivity for the simulation of adiabatic shear band formation", Computational Mechanics, Vol 25, pp. 257-266
11. Hueatra A. and Fernandez-Mendez S. (2000), "Enrichment and coupling of the finite element and meshless methods", International Journal for Numerical Methods in Engineering, Vol. 48, pp. 1615-1636
12. Lee S. H., Kim H. J. and Jun S. (2000), "Two scale meshfree method for the adaptivity of 3-D stress concentration problems", Computational Mechanics, Vol. 26, pp. 376-387
13. Kang M. S. and Youn. S. K. (2001), "Dof splitting p-adaptive meshless method", Structural Engineering and Mechanics, Vol. 11, pp. 535-546
14. Liu G. R. and Tu Z. H., (2002), "An adaptive procedure based on background cells for meshless methods", Computer Methods in Applied Mechanics and Engineering, Vol. 191, pp. 1923-1943.
15. Liu W. K., Jun S., Li S., Adee J., Belytschko T. (1995), "Reproducing kernel particle methods for structural dynamics, International Journal for Numerical Methods in Engineering, Vol. 38, pp. 1655-1679.

16. Liu W. K., Sukky J. Zhang Y. F. (1995), "Reproducing kernel particle methods", *International Journal for Numerical Methods in Fluids*, Vol. 20, pp. 1081-1106.
17. Belytschko T., Lu Y. Y., Gu L. (1994). "Element-free Galerkin methods", *International Journal for Numerical Methods in Engineering*, Vol. 37, pp. 229-256.
18. Lu Y. Y., Belytschko T., Gu L. (1994), "A new implementation of the element free Galerkin method", *Computer Methods in Applied Mechanics and Engineering*, Vol. 113, pp. 397-414.
19. Chung H. J., Belytschko T. (1998), "An Error Estimate in the EFG Method", *Computational Mechanics*, Vol. 21, pp. 91-100.
20. Gavete L., Falcon S., and Ruiz A. (2001), "An error indicator for the element free Galerkin method", *Eur. J. Mech. A/Solids* Vol. 20, pp. 327-341
21. Gavete L., Cuesta J. L. and Ruiz A. (2002), "A procedure for approximation of the error in the EFG method", *International Journal for Numerical Methods in Engineering*, Vol. 53, pp. 677-690.
22. Gavete M., Gavete L., Alonso B. and Martin A. J., (2003) "A posteriori error approximation in EFG method", *International Journal for Numerical Methods in Engineering*, Vol. 58, pp. 2239-2263
23. Lee C. K. and Zhou C. E. (2004), "On error estimation and adaptive refinement for element free Galerkin method. Part I: stress recovery and a posteriori error estimation", *Computers and Structures*, Vol. 82, pp. 413-428.
24. Lee C. K. and Zhou C. E. (2004), "On error estimation and adaptive refinement for element free Galerkin method. Part II: adaptive refinement", *Computers and Structures*, Vol. 82, pp. 429-443.
25. Lee C. K. and Shuai Y. Y., "An automatic adaptive refinement procedure for reproducing kernel particle method, Part I: Stress recovery and a posteriori error estimation", Submitted to *Computational Mechanics*.
26. Lee, C. K., (2000), "A new finite point generation scheme using metric specifications", *International Journal for Numerical Methods in Engineering (United States)*, Vol. 48, No. 10, pp. 1423 - 1444.
27. Lohner R. and Onate E., (1998) "An advancing front point generation technique", *Commun. Numer. Meth. Engng*, Vol. 14, pp. 1097-1108
28. Zienkiewicz O. C., Zhu J. Z. (1987), "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis", *International Journal for Numerical Methods in Engineering*, Vol. 24, pp. 337-357.
29. Szabo B. A. and Babuska I. (1991), *Finite Element Analysis*, Wiley & Sons, Inc.

30. Lee C. K., Lie S. T., and Shuai Y. Y., (2005), "On coupling of reproducing kernel particle method and boundary element method", *Computational Mechanics*, Vol. 34, pp 282-297
31. Zienkiewicz O. C., Zhu J. Z., "The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique", *International Journal for Numerical Methods in Engineering*, Vol. 33, pp. 1331-1364.
32. Sukumar N., Moran B., Belytschko T., (1998) "The natural element method in solid mechanics", *International Journal for Numerical Methods in Engineering*, Vol. 43, pp. 839-887.
33. Zienkiewicz O. C., and Zhu J. Z., (1992). "The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity", *International Journal for Numerical Methods in Engineering*, Vol. 33, pp. 1365-1382
34. Lee C. K. and Lo S. H., (1992) "An automatic adaptive refinement finite element procedure for 2D elastostatic analysis", *International Journal for Numerical Methods in Engineering*, Vol. 35, pp. 1967-1989
35. Lee C. K. and Hobbs R. E., (1997), "Automatic adaptive refinement for shell analysis using nine-node assumed strain element", *International Journal for Numerical Methods in Engineering*, Vol. 40, pp. 3601-3638
36. Lee C. K. and Hobbs R.E., (1998) "On solving nearly incompressible 2D problems using an adaptive refinement procedure", *Communications in Numerical Methods in Engineering*, Vol. 41 pp. 409-418
37. Lee C. K. and Hobbs R. E., (1998) "Automatic adaptive refinement for plate bending problems using Reissner-Mindlin plate bending elements", *International Journal for Numerical Methods in Engineering*, Vol. 41, pp. 1-63
38. Onate. E. and Bugeda G. (1993), "A study of mesh optimality criteria in adaptive finite element analysis", *Engineering Computation*, Vol. 10, pp. 307-321
39. Lee C. K., (1999), "Automatic adaptive mesh generation using metric advancing front approach", *Engineering Computation*, Vol. 16. pp. 230-263

Appendix: Determination of support size for nodes in the point mesh

In this appendix, it will be proved that the procedure described in Section 3.1 will

- (1) ensure that any point inside the problem domain Ω is within the domain of influence of at least one node of the mesh, and
- (2) the RKPM shape function can be constructed at any point within Ω .

In order to complete the proof, first assume that there exists a point $\mathbf{p}$ inside Ω such that it is not covered by the domain of influence of any node in the mesh (Fig. A1). Since the virtual grid cells are constructed in such a way that their sizes are all equal to $d_{\min}$, it can be concluded that the greatest distance between $\mathbf{p}$ and the centre of any grid cell should be at most equal to $\sqrt{1/2}d_{\min}$ (Fig. A1). Now assume that the cell under consideration is labelled as cell i and the node k is one of the six nearest nodes around O_i , the centre of cell i . By using the same notations in Section 3.1, the distance between O_i and node k is denoted as d_k . From Fig. A1, l_k , the distance between $\mathbf{p}$ and node k must satisfy the inequality

$$l_k \leq \sqrt{1/2}d_{\min} + d_k \quad (\text{A1})$$

However, according to Section 3.1 and from Eqn. 14, the size of the domain of influence of node k must satisfy the condition

$$h_k \geq d_k + \sqrt{1/2}d_{\min} \geq l_k \quad (\text{A2})$$

Hence, this will result in contradiction and point $\mathbf{p}$ must be within the domain of influence of node k .

To prove that the RKPM shape function can be constructed at any point $\mathbf{p}$ within Ω , it is necessary to show that $\mathbf{p}$ will be covered by the domains of influence corresponding to at least six distinct nodes in the point mesh. This condition can be proved by recalling that during the construction of h_k , Eqn. 14 will be enforced for the six nearest nodes around O_i . Thus, during the error estimation process, it is always possible to construct the RKPM shape functions at any integration point within the problem domain.

However, it should be mentioned that just the minimum support size condition is not sufficient to exclude all the degenerated geometric arrangements of nodes (e.g. all the nodes are collinear). Hence, in actual implementation, the minimum support size will be used whenever is possible and some stability checks are preformed during the computation of the RKPM shape functions [16].

Fig. 1: The minimum size of domains of influence of nodes: h_i satisfies the minimum size condition while h_j does not

Fig. 2: Determination of support sizes

(a) Initial Mesh, 15 nodes

(b) Mesh 1, 84 nodes

(c) Mesh 2, 217 nodes

(d) Final mesh, 456 nodes

Fig. 3: Adaptive meshes generated from Example 1, Timoshenko beam

Fig. 4: Rate of convergence for Example 1

Fig. 6: Rate of convergence for Example 2

Fig. 5: Adaptive meshes generated form Example 2, Infinite plate with a hole

(a) Initial mesh, 15 nodes

(b) Mesh 1, 149 nodes

(c) Mesh 2, 469 nodes

(d) Final mesh, 1150 nodes

Fig. 7: Adaptive meshes generated form Example 3, Region near the tip of a crack

Fig. 8: Rate of convergence for Example 3

Fig. 10: Rate of convergence for Example 4

(a) Initial mesh, 33 nodes

(b) Mesh 1, 121 nodes

(c) Mesh 2, 379 nodes

(d) Final mesh, 1055 nodes

Fig. 9: Adaptive meshes generated form Example 4, L-shape domain

Example no.	Descriptions	Domain geometry	Reference figures	Presence of singularity	λ	η_{tar} (%)
1	Timoshenko beam	Convex	*Fig. 6	No	---	2.5
2	Infinite plate with a hole	Concave	*Fig. 8	No	---	2.5
3	Region near the tip of a crack	Convex	*Fig. 10	Yes	0.5	5
4	L-shape domain	Concave	*Fig. 12	Yes	0.544	5

* Figure numbers of reference [25] in which the definition of the problems are given

Table 1: Summary of the four benchmark problems used

Mesh	Initial mesh	Mesh 1	Mesh 2	Final mesh
N_{DOF}	30	168	434	912
η_{RKPM} (%)	21.54	6.45	3.41	2.1
η_{Rv} (%)	11.67	2.61	1.16	0.62
$\bar{\eta}$ (%)	16.37 (13.65)	5.55 (4.63)	3.31 (2.76)	2.18 (1.82)
R_{RKPM}		0.70	0.67	0.65
R_{Rv}		0.87	0.85	0.84
θ	0.76 (0.63)	0.86 (0.72)	0.97 (0.81)	1.04 (0.87)

Table 2: Results for Example 1

(For $\bar{\eta}$ and θ , values in brackets are obtained without using the compensation factor γ)

Mesh	Initial mesh	Mesh 1	Mesh 2	Final mesh
N_{DOF}	54	98	224	482
η_{RKPM} (%)	9.41	6.24	3.59	2.22
η_{Rv} (%)	4.97	2.94	1.44	0.76
$\bar{\eta}$ (%)	7.25 (6.04)	5.05 (4.20)	3.27 (2.73)	2.15 (1.79)
R_{RKPM}		0.69	0.67	0.63
R_{Rv}		0.88	0.86	0.83
θ	0.77 (0.64)	0.81 (0.67)	0.91 (0.76)	0.97 (0.81)

Table 3: Results for Example 2

(For $\bar{\eta}$ and θ , values in brackets are obtained without using the compensation factor γ)

Mesh	Initial mesh	Mesh 1	Mesh 2	Final mesh
N_{DOF}	30	298	938	2300
η_{RKPM} (%)	29.7	12.13	7.76	5.52
η_{Rv} (%)	17.97	6.70	4.14	2.89
$\bar{\eta}$ (%)	16.33 (13.60)	8.85 (7.38)	6.05 (5.04)	4.75 (3.96)
R_{RKPM}		0.39	0.39	0.38
R_{Rv}		0.43	0.42	0.40
θ	0.55 (0.49)	0.73 (0.61)	0.78 (0.65)	0.86 (0.72)

Table 4: Results for Example 3

(For $\bar{\eta}$ and θ , values in brackets are obtained without using the compensation factor γ)

Mesh	Initial mesh	Mesh 1	Mesh 2	Final mesh
N_{DOF}	66	242	758	2110
η_{RKPM} (%)	23.5	13.79	8.73	5.86
η_{Rv} (%)	14.12	7.97	4.88	3.17
$\bar{\eta}$ (%)	12.93 (10.78)	9.93 (8.28)	6.90 (5.75)	4.92 (4.10)
R_{RKPM}		0.41	0.40	0.39
R_{Rv}		0.44	0.43	0.42
θ	0.55 (0.46)	0.72 (0.60)	0.79 (0.66)	0.84 (0.70)

Table 5: Results for Example 4

(For $\bar{\eta}$ and θ , values in brackets are obtained without using the compensation factor γ)

Major process	Ratios of CPU time used
RKPM Analysis	90%
Error estimation (include stress recovery)	9%
Adaptive refinement (including point mesh generation)	<1%

Table 6: Ratios of CPU time used in different major processes for adaptive refinement

Fig. A1: Coverage of a point $\mathbf{p}$ inside Ω