

REPRODUCING KERNEL PARTICLE METHODS

WING KAM LIU, SUKKY JUN AND YI FEI ZHANG

Department of Mechanical Engineering, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208, U.S.A.

SUMMARY

A new continuous reproducing kernel interpolation function which explores the attractive features of the flexible time–frequency and space–wave number localization of a window function is developed. This method is motivated by the theory of wavelets and also has the desirable attributes of the recently proposed smooth particle hydrodynamics (SPH) methods, moving least squares methods (MLSM), diffuse element methods (DEM) and element-free Galerkin methods (EFGM). The proposed method maintains the advantages of the free Lagrange or SPH methods; however, because of the addition of a correction function, it gives much more accurate results. Therefore it is called the reproducing kernel particle method (RKPM). In computer implementation RKPM is shown to be more efficient than DEM and EFGM. Moreover, if the window function is C^∞ , the solution and its derivatives are also C^∞ in the entire domain. Theoretical analysis and numerical experiments on the 1D diffusion equation reveal the stability conditions and the effect of the dilation parameter on the unusually high convergence rates of the proposed method. Two-dimensional examples of advection–diffusion equations and compressible Euler equations are also presented together with 2D multiple-scale decompositions.

KEY WORDS multiple scale decomposition; correction function; multi-resolution analysis; reproducing kernel function; wavelet; mesh- (or grid-) free particle methods

1. INTRODUCTION

During the last two decades considerable effort has been devoted to the development of mesh-free or grid-free methods. In most methods the interpolation functions are usually established by enforcing certain continuity requirements around a set of ordered (equally spaced) points. However, owing to deformation, this set of points can become highly disordered and the accuracy deteriorates. In addition, if the interpolation methods, such as finite element and finite difference methods, require a mesh or a grid, the distorted mesh can terminate the calculation owing to mesh entanglement and other problems.

Among the mesh- or grid-free methods are the smooth particle hydrodynamics (SPH) methods, which depend only on a set of disordered points or particles, developed by Lucy¹ and Gingold and Monaghan², among others, the diffuse element methods (DEM) developed by Nayroles *et al.*³ and the element-free Galerkin methods (EFGM) recently proposed by Belytschko *et al.*⁴. The last two methods are based on the moving least squares interpolation functions (MLSM) presented by Lancaster and Salkauskas.⁵ None of these methods requires a finite difference grid or a finite element mesh. Furthermore, if the kernel functions (used in SPH methods) and the weighting functions (used in MLSM, DEM and EFGM) and their derivatives are continuous, the solution and its derivatives are also continuous. The truly mesh-free characteristics, the continuous solution and the continuous derivatives are the key selling points of these methods.

The most attractive feature of SPH methods in a large-deformation analysis is the *free Lagrange* concept. Although SPH methods work well in the absence of boundaries (since the boundary terms are discarded in the formulation; see e.g. Reference 6) and when the number of unknowns (nodes) is large, they are not as accurate as the regular finite element methods; see e.g. Reference 7. From our study of SPH interpolation functions via a simple 1D Galerkin formulation (see Section 7), we found that there is an additional deficiency in the SPH formulation. It is related to the boundary correction term of the reproducing kernel approximation. We shall make an attempt to identify this deficiency (via a *correction function*) and present our views on improving the SPH kernel approximation.

After reviewing the moving least squares interpolation functions and the diffuse element methods, Belytschko *et al.*⁴ pointed out that an assumption made by Nayroles *et al.*,³ namely that the interpolation coefficients are constants, detracts from the accuracy of the method. They showed that by adding more accurate derivatives and enforcing boundary conditions by Lagrange multipliers, the methods could achieve very high rates of convergence. From our experience EFGM is more accurate than the finite element methods and hence the SPH methods, especially for a small set of nodes. One main drawback of EFGM is the computational expense and we found that it is more computationally intensive than the SPH methods.

The objective of this paper is along the same lines of development as SPH, DEM and EFGM: to develop accurate and efficient mesh-free interpolation functions. Since a continuous reproducing kernel can be derived for this proposed method and it is also a Lagrangian particle method, we shall call this development the *Reproducing Kernel Particle Method* (RKPM). The proposed approach is motivated by the theory of wavelets (see e.g. Reference 8), in which a function is represented by a combination of the dilation and translation of a single wavelet, which is a window function. In a wavelet analysis, similarly to the SPH interpolation kernel, the interpolation coefficients are defined in terms of the integral window transform of the window function and the solution itself. We shall borrow three key ideas from wavelet analysis: (i) the integral window transform, (ii) the dilation and translation of a window function and (iii) the continuous and discrete reproducing kernel approximations. It is noted that the window functions used in this paper are *not* wavelets. Good candidates for window functions are the scaling functions used to produce wavelets, since these scaling functions can be constructed to be orthogonal with respect to their translation, and a multiresolution analysis^{8,9} can also be performed. Detailed discussion on multiple-scale wavelet and reproducing kernel methods can be found in Reference 11. In addition, the similarity between the hp-adaptive methods and the multiple scale RKPM is also given in Liu and Chen¹¹. The application of RKPM to large deformation structural dynamics is presented in Liu *et al.*¹⁰

We shall show the similarities among the smooth particle hydrodynamics methods, the diffuse element methods, the element-free Galerkin methods and the reproducing kernel particle methods. We shall also show that SPH and RKPM are indeed developed through a *continuous* reproducing kernel approximation, whereas DEM and EFGM, like finite elements, are developed through a *discrete* reproducing kernel approximation. As a by-product of this development, the concept of dilation of a window function will be used to explain why the accuracy of the diffuse element methods decreases relative to the element-free Galerkin methods.

In the next section some preliminary concepts of integral window transform and SPH interpolation kernel functions are reviewed. In Section 3 the reproducing kernel particle interpolation functions are derived. In Section 4 the effect of the dilation parameter on the reproducing kernel, time–frequency or space–wave number localization, and the stability condition are discussed. In Section 5 some examples of reproducing kernel window functions are presented. The similarities among SPH, DEM, EFGM and RKPM interpolation functions are given in Section 6. Numerical experiments which confirm the theoretical analysis are presented in Section 7, followed by conclusions in Section 8.

2. PRELIMINARIES

2.1. Dilation and translation of a window function

Let \mathbf{x} denote the spatial co-ordinates. If $\Phi(\mathbf{x})$ is a window function located at $\mathbf{x} = \mathbf{0}$ which has a support $B(\mathbf{x})$, then

$$\Phi(\mathbf{x}) \neq 0 \quad \text{in } B(\mathbf{x}) \tag{1a}$$

$$\Phi(\mathbf{x}) = 0 \quad \text{outside } B(\mathbf{x}). \tag{1b}$$

The dilation and translation of $\Phi(\mathbf{x})$, denoted $\Phi_{ab}(\mathbf{x})$ is defined as

$$\Phi_{ab}(\mathbf{x}) = E(a)\Phi\left(\frac{\mathbf{x} - \mathbf{b}}{a}\right), \quad a > 0, \tag{2}$$

is a window function located at $\mathbf{x} = \mathbf{b}$ with a support *scaled* by the dilation parameter a . The function $E(a)$ appearing in (2) scales $\Phi_{ab}(\mathbf{x})$ such that

$$\int_{R_x} \phi_{ab}(\mathbf{x})dR_x = \int_{R_x} \phi(\mathbf{x})dR_x = 1 \tag{3}$$

when the support $B(\mathbf{x})$ is within the spatial region of interest, R_x . It is noted that when \mathbf{b} is close to the boundary of R_x , ∂R_x , the integral of $\Phi_{ab}(\mathbf{x})$ over R_x will be less than one. We believe that this is a drawback of SPH methods as well as all other reproducing kernel methods, including wavelets, which assume the region is unbounded.

2.2. Integral window transform and SPH interpolation kernel functions

The integral window transform of a real function $u(\mathbf{x})$ with a real window function $\Phi_{ab}(\mathbf{x})$ is defined as

$$\langle u, \Phi_{ab} \rangle = \int_{R_x} E(a)\Phi\left(\frac{\mathbf{x} - \mathbf{b}}{a}\right)u(\mathbf{x})dR_x. \tag{4}$$

As a matter of fact, one of the main concepts of the SPH method is to find a suitable smooth reproducing kernel function $\Phi(\mathbf{x})$ that mimics the Dirac delta function. Hence, when a is chosen to approach zero and when $\mathbf{b} = \mathbf{x}$, the reproducing kernel approximation of $u(\mathbf{x})$, denoted $u^h(\mathbf{x})$, is given by

$$u^h(\mathbf{x}) = \langle u, \Phi_{ax} \rangle = \int_{R_x} E(a)\Phi\left(\frac{\mathbf{y} - \mathbf{x}}{a}\right)u(\mathbf{y})dR_y. \tag{5}$$

Discretizing the integral of (5) by NP distinct nodes (points) using a numerical quadrature formula gives

$$u^h(\mathbf{x}) \approx \sum_{J=1}^{NP} E(a)\Phi\left(\frac{\mathbf{x}_J - \mathbf{x}}{a}\right)u(\mathbf{x}_J)\Delta V_J. \tag{6a}$$

Equation (6a) can be written in a more familiar notation, in terms of generalized global shape functions $N_J(\mathbf{x})$, as

$$u^h(\mathbf{x}) = \sum_{J=1}^{NP} N_J(\mathbf{x})u_J, \quad N_J(\mathbf{x}) = E(a)\Phi\left(\frac{\mathbf{x}_J - \mathbf{x}}{a}\right)\Delta V_J, \quad \text{no sum on } J, \tag{6b}$$

where $u_J \equiv u(\mathbf{x}_J)$ and $\Delta V_J \neq 0$ is the J th nodal domain (volume in 3D, area in 2D and length in 1D) associated with quadrature point \mathbf{x}_J . The sum of all ΔV_J gives the total domain V , i.e.

$$\sum_{J=1}^{NP} \Delta V_J = V. \quad (6c)$$

The SPH methods use a similar interpolation formula to that given in (6), but instead of ΔV_J and $E(a)\Phi((\mathbf{x}_J - \mathbf{x})/a)$, ΔM_J and $\rho_J E(a)\Phi((\mathbf{x}_J - \mathbf{x})/a)$ are employed, where ΔM_J and ρ_J are the J th particle mass and density respectively. With this substitution in (6a) or (6b) the SPH approximation can be used in standard Galerkin, collocation or spectral methods, but the particle methods use information from a set of disordered points based on kernel estimation.¹² It is also pointed out by Monaghan¹³ that SPH works quite well for an arbitrarily moving fluid if the number of particles is *large* and in the absence of boundaries; however, there is no systematic way to handle a moving fluid with rigid or moving boundaries in SPH. Furthermore, it is not clear how to generalize SPH to non-uniform mass particles or what is the effect of the dilation parameter 'a' on the accuracy of the solution. Recently, these problems have been addressed by Liu and Chen.¹¹

2.3. Moments

We shall define the following moments for the window function $\Phi(\mathbf{y})$:

$$m_0(\mathbf{x}) = \int_B \Phi(\mathbf{y}) dR_y \quad (\text{zero moment}), \quad (7a)$$

$$m_i(\mathbf{x}) = \int_B y_i \Phi(\mathbf{y}) dR_y \quad (\text{first moment}), \quad (7b)$$

$$m_{ij}(\mathbf{x}) = \int_B y_i y_j \Phi(\mathbf{y}) dR_y \quad \begin{cases} i \neq j & (\text{cross moment}), \\ i = j & (\text{second moment}) \end{cases} \quad (7c)$$

In the above equations the integral is evaluated with respect to the support $B(\mathbf{x})$. Hence the moments are functions of \mathbf{x} , and if $B(\mathbf{x})$ is close to the boundary of the spatial region R_x , $m_0(\mathbf{x})$ is less than one. The subscript indices i and j take values from 1 to NSD , where NSD is the number of space dimensions. If $\Phi(\mathbf{y})$ is symmetric, $m_i(\mathbf{x}) = 0$ in the interior of R_x and $m_i(\mathbf{x}) \neq 0$ when \mathbf{x} is close to the boundary; $m_{ii}(\mathbf{x})$, no sum on i , denotes the standard deviation of $\Phi(\mathbf{y})$ in the x_i -direction; and $m_{ij}(\mathbf{x})$, $i \neq j$, denotes the cross moment. Higher-order moments can be defined similarly. We shall employ these moments to analyse the reproducing kernel particle interpolation functions which are described next.

3. REPRODUCING KERNEL PARTICLE INTERPOLATION FUNCTIONS

The objective is to use the concept of reproducing kernels and the local character of the window function to develop an accurate reproducing kernel function from a suitable smooth window function $\Phi_{ab}(\mathbf{x})$ multiplied by a *correction function* $C(a, \mathbf{x}, \mathbf{b})$. If both $\Phi_{ab}(\mathbf{x})$ and $C(a, \mathbf{x}, \mathbf{b})$ are smooth functions within the spatial region R_x , i.e. the functions and their derivatives are continuous, then we have developed global interpolation functions that do not require a finite element mesh or a finite difference grid. In particular, unlike the SPH methods, the dilation parameter can take a fairly large range of positive values provided that certain stability conditions are met.

Our goal then is to develop reproducing kernel particle interpolation functions which will have the following properties.

1. If $\Phi_{ab}(\mathbf{x})$ is an even function, the correction function $C(a, \mathbf{x}, \mathbf{b})$ should also be one when the support of $\Phi_{ab}(\mathbf{x})$ is not close to the boundary ∂R_x ; it differs from one when $\Phi_{ab}(\mathbf{x})$ is close to the boundary.
2. A truly element- or mesh-free particle method similar to SPH methods but with much better accuracy, especially when the number of particles is *small*.
3. Similarly to MLSM, DEM and EFGM, RKPM provides smoother approximations of the solution as well as its derivatives; however, RKPM is computationally more efficient and a mathematical analysis of RKPM is also available.

Remark. The requirement that the correction function $C(a, \mathbf{x}, \mathbf{b})$ should be constructed to be one when the support of $\Phi_{ab}(\mathbf{x})$ is not close to the boundary yields *efficient computations*, since this correction function is only required when $\Phi_{ab}(\mathbf{x})$ is close to the boundary.

Consider a function $u(\mathbf{x})$ that is to be represented in terms m linearly independent functions $P_i(\mathbf{x})$ and a set of unknowns c_i in the expression

$$u(\mathbf{x}) = \sum_{i=1}^m P_i(\mathbf{x})c_i. \tag{8a}$$

In matrix notation equation (8a) reads

$$u(\mathbf{x}) = \mathbf{P}(\mathbf{x})\mathbf{c}, \tag{8b}$$

where $\mathbf{P}(\mathbf{x}) = \{P_1(\mathbf{x}), P_2(\mathbf{x}), \dots, P_m(\mathbf{x})\}$ is a vector of m linear independent functions and $\mathbf{c} = \{c_1, c_2, \dots, c_m\}$ is the vector of unknown coefficients. A superscript ‘T’ denotes the transpose. In order to define \mathbf{c} in terms of the solution locally around any point \mathbf{x} , we multiply both sides of (8b) by $\mathbf{P}^T(\mathbf{y})$ and perform the integral window transform with respect to a positive window function $\Phi_{ax}(\mathbf{y})$ to yield (\mathbf{x} has been replaced by \mathbf{y} in (8b))

$$\langle \mathbf{P}^T u, \Phi_{ax} \rangle = \langle \mathbf{P}^T \mathbf{P}, \Phi_{ax} \rangle \mathbf{c}, \tag{9a}$$

or the vector of coefficients \mathbf{c} is solved in terms of the solution u , i.e.

$$\mathbf{c} = \mathbf{M}^{-1}(\mathbf{x}) \langle \mathbf{P}^T u, \Phi_{ax} \rangle, \tag{9b}$$

where the $m \times m$ non-singular matrix $\mathbf{M}(\mathbf{x})$ is given by

$$\mathbf{M}(\mathbf{x}) = \langle \mathbf{P}^T \mathbf{P}, \Phi_{ax} \rangle = \int_{R_x} \mathbf{P}^T(\mathbf{y})\mathbf{P}(\mathbf{y})E(a)\Phi\left(\frac{\mathbf{y}-\mathbf{x}}{a}\right)dR_y. \tag{10}$$

It is noted that $\mathbf{M}(\mathbf{x})$ is a continuous function of the translation (\mathbf{x}) of the window function $\Phi_{ax}(\mathbf{y})$. Substitution of (9b) into (8b) gives the approximation of $u(\mathbf{x})$, denoted $u^h(\mathbf{x})$, through a continuous reproducing kernel

$$u^h(\mathbf{x}) = \langle \mathbf{P}(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}^T u, \Phi_{ax} \rangle. \tag{11}$$

Using the definition of the integral window transform, $u^h(\mathbf{x})$ can be shown to be

$$u^h(\mathbf{x}) \int_{R_x} k(a, \mathbf{x}, \mathbf{y})u(\mathbf{y})R_y, \tag{12}$$

where the reproducing kernel, which is a modified window function, is shown to be

$$k(a, \mathbf{x}, \mathbf{y}) = k_{ax}(\mathbf{y}) = C(a, \mathbf{x}, \mathbf{y})E(a)\Phi\left(\frac{\mathbf{y}-\mathbf{x}}{a}\right) \tag{13a}$$

and the *correction function* $C(a, \mathbf{x}, \mathbf{y})$ is given by

$$C(a, \mathbf{x}, \mathbf{y}) = \mathbf{P}(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}^T(\mathbf{y}). \tag{13b}$$

To write (12) in a discrete reproducing kernel particle form, its integral of (12) is discretized by NP distinct points using a numerical quadrature formula to yield the usual approximation formula

$$u^h(\mathbf{x}) = \sum_{J=1}^{NP} N_J(\mathbf{x})u_J, \tag{14a}$$

where the reproducing kernel particle interpolation functions are given by

$$N_J(\mathbf{x}) = C(a, \mathbf{x}, \mathbf{x}_J)E(a)\Phi\left(\frac{\mathbf{x}_J - \mathbf{x}}{a}\right)\Delta V_J. \tag{14b}$$

Comparing (14b) with the SPH interpolation formula (6b), we believe that the *discretized correction function*

$$C(a, \mathbf{x}, \mathbf{y}_J) = \mathbf{P}(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}^T(\mathbf{x}_J) \tag{14c}$$

will improve the accuracy of the interpolation kernel tremendously, partly owing to boundary corrections. Depending on the choice of $\mathbf{P}(\mathbf{x})$, we shall show that the correction function is composed of the moments defined in Section 2.3. In particular, if $\mathbf{P}(\mathbf{x})$ is chosen to be a constant and linear polynomial

$$\mathbf{P}(\mathbf{x}) = \{1, x_1, x_2, x_3\} \tag{15a}$$

it can be shown that the continuous function $C(a, \mathbf{x}, \mathbf{y})$ takes the form

$$C(a, \mathbf{x}, \mathbf{y}) = C_1(a, \mathbf{x}) + C_2(a, \mathbf{x}) \bullet \left(\frac{\mathbf{y} - \mathbf{x}}{a}\right). \tag{15b}$$

In (15b) C_1 and C_2 are continuous scalar and vector functions of \mathbf{x} as well as of the zeroth, first and cross moments. A dot denotes the inner product. It will also be shown that $C_1(a, \mathbf{x}) = 1$ and $C_2(a, \mathbf{x}) = \mathbf{0}$ when the support of $\Phi_{ax}(\mathbf{y})$, $B(\mathbf{x})$, is not close to the boundary of the domain, whereas $C_1(a, \mathbf{x}) \neq 1$ and $C_2(a, \mathbf{x}) \neq \mathbf{0}$ when the window function moves close to the boundary of the domain.

Remark. It can be seen that if $u(\mathbf{x}) = \mathbf{P}(\mathbf{x})$, then

$$\begin{aligned} \mathbf{P}(\mathbf{x}) &= \int_{R_x} \mathbf{P}(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}^T(\mathbf{y})E(a)\Phi\left(\frac{\mathbf{y} - \mathbf{x}}{a}\right)\mathbf{P}(\mathbf{y})dR_y \\ &= \mathbf{P}(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x}) \int_{R_x} \mathbf{P}^T(\mathbf{y})\mathbf{P}(\mathbf{y})E(a)\Phi\left(\frac{\mathbf{y} - \mathbf{x}}{a}\right)dR_y = \mathbf{P}(\mathbf{x}) \end{aligned} \tag{16a}$$

because of the definition of $\mathbf{M}(\mathbf{x})$. Hence the modified window function $k(a, \mathbf{x}, \mathbf{y})$ defined in (13a) satisfies the consistency condition

$$\int_{R_x} k(a, \mathbf{x}, \mathbf{y})y_i^q dR_y = x_i^q \quad \text{for } q = 0, 1, \dots, n, \tag{16b}$$

where n is the highest order used in $\mathbf{P}(\mathbf{x})$. For example, if $\mathbf{P}(\mathbf{x})$ is chosen to be $\{1, x_1, x_2, x_3, x_1^2, \dots\}$, the reproducing kernel will satisfy the usual isoparametric finite element properties. In a discrete approximation, we have

$$\sum_{J=1}^{NP} N_J(\mathbf{x}) = 1, \quad \sum_{J=1}^{NP} N_J(\mathbf{x})\mathbf{x}_J = \mathbf{x}, \quad \text{etc.} \tag{17}$$

4. EFFECT OF THE DILATION PARAMETER ON THE REPRODUCING KERNEL AND STABILITY CONDITION

In this section we restrict our discussion to one dimension. To explore the accuracy of the RKPM interpolation functions, we shall relate the dilation parameter a (sometimes called a scale) to the frequency content (in time) or the wave number/wavelength content (in space). For simplicity, we let x be the time axis and ω the frequency axis. A similar interpretation can be made when x and ω are the space and the wave number respectively.

4.1. Frequency band

Following Chui,⁸ we define the centre x^* and radius $\Delta\phi$ of a window function $\Phi(x)$ by

$$x^* = \frac{1}{\|\Phi\|_2^2} \int_{-\infty}^{\infty} x|\Phi(x)|^2 dx, \tag{18a}$$

$$\Delta\phi = \frac{1}{\|\Phi\|_2} \left(\int_{-\infty}^{\infty} (x - x^*)^2 |\Phi(x)|^2 dx \right)^{1/2}. \tag{18b}$$

The width of the window function $\Phi(x)$ is defined as $2\Delta\phi$. The norm of $\Phi(x)$ is defined as

$$\|\Phi\|_2 = \langle \Phi, \Phi \rangle^{1/2}. \tag{19a}$$

For our choice of the scaling parameter, the norm of $\Phi_{ab}(x)$ is related to $\Phi(x)$ by

$$\|\Phi_{ab}\|_2^2 = a^{-1} \|\Phi\|_2^2. \tag{19b}$$

Suppose that $\Phi(x)$ is any function such that both $\Phi(x)$ and its Fourier transform $\hat{\Phi}(\omega)$ are window functions. Following the definitions (18), the centre frequency ω^* and the radius $\Delta\hat{\phi}$ of $\hat{\Phi}(\omega)$ are given by

$$\omega^* = \frac{1}{\|\hat{\Phi}\|_2^2} \int_{-\infty}^{\infty} \omega |\hat{\Phi}(\omega)|^2 d\omega, \tag{20a}$$

$$\Delta\hat{\phi} = \frac{1}{\|\hat{\Phi}\|_2} \left(\int_{-\infty}^{\infty} (\omega - \omega^*)^2 |\hat{\Phi}(\omega)|^2 d\omega \right)^{1/2}. \tag{20b}$$

4.2. Time–frequency band of the parent reproducing kernel function

The centre and radius in Section 4.1 are defined over an unbounded domain so that x^* and $\Delta\phi$ are invariant with respect to the translation (x). Since the kernel function is a function of y and is defined within R_x only, we redefine a parent kernel function as

$$k(y) = C(y)a^{-1}\Phi(y), \tag{21}$$

where $C(y)$ is equivalent to $C(a, x, y)$ of (13b) and the arguments a and x are dropped here because they are implicit parameters and only y is a variable. Let us denote the centre and radius of $k(y)$ as \bar{x} and $\Delta\bar{k}$ respectively:

$$\bar{x} = \frac{1}{\|k\|_2^2} \int_{R_x} y |C(y)a^{-1}\Phi(y)|^2 dy, \tag{22a}$$

$$\Delta\bar{k} = \frac{1}{\|k\|_2} \left(\int_{R_x} (y - \bar{x})^2 |C(y)a^{-1}\Phi(y)|^2 dy \right)^{1/2} \tag{22b}$$

If the Fourier transform of $k(y)$ is denoted by $\hat{k}(\omega)$, the centre ($\hat{\omega}$) and radius ($\Delta\hat{k}$) of $\hat{k}(\omega)$ in the frequency space are given by

$$\hat{\omega} = \frac{1}{\|\hat{k}\|_2^2} \int_{-\infty}^{\infty} \omega |\hat{k}(\omega)|^2 d\omega, \tag{23a}$$

$$\Delta\hat{k} = \frac{1}{\|\hat{k}\|_2} \left(\int_{-\infty}^{\infty} (\omega - \hat{\omega})^2 |\hat{k}(\omega)|^2 d\omega \right) \tag{23b}$$

From the parent kernel function $k(y)$ the two-parameter reproducing kernel function $k_{ax}(y) = k(a, x, y)$ can be generated by translation (x) and dilation (a):

$$k_{ax}(y) = C\left(\frac{y-x}{a}\right) a^{-1} \Phi\left(\frac{y-x}{a}\right). \tag{24}$$

The relationship between the norms of $k(y)$ and $k_{ax}(y)$ is obtained by

$$\|k_{ax}\|_2 = \left(\int_R x \left| C\left(\frac{y-x}{a}\right) a^{-1} \Phi\left(\frac{y-x}{a}\right) \right|^2 dy \right)^{1/2} \tag{25a}$$

$$= a^{1/2} \left(\int_{R_x} |C(z) a^{-1} \Phi(z)|^2 dz \right)^{1/2} \tag{25b}$$

$$= a^{1/2} \|k\|^2. \tag{25c}$$

4.3. Time–frequency or space–wave number localization

Because of the linear translation of the reproducing kernel $k_{ax}(y)$, the integral window transform of the response u with $k_{ax}(y)$,

$$\langle u, k_{ax} \rangle = \int_{R_x} C\left(\frac{y-x}{a}\right) a^{-1} \Phi\left(\frac{y-x}{a}\right) u(y) dy, \tag{26a}$$

localizes the response with a *time window* or a *space window*

$$[x + a\bar{x} - a\Delta\bar{k}; x + a\bar{x} + a\Delta\bar{k}]. \tag{26b}$$

Furthermore, since the domain is bounded, x^* and Δk have been replaced by \bar{x} and $\Delta\bar{k}$ such that

$$\bar{x} = x^* = \text{constant}, \quad \Delta\bar{k} = \Delta k = \text{constant} \quad \text{if } B(x) \text{ in } R_x, \tag{26c}$$

$$\bar{x} = x^* = \text{variable}, \quad \Delta k = \Delta k = \text{variable} \quad \text{if } B(x) \text{ is close to } R_x. \tag{26d}$$

In (26d) x^* and Δk have to be computed according to (22). The centre and radius of the reproducing kernel function $k_{ax}(y)$ are shown to be

$$\text{centre} = \frac{1}{\|k_{ax}\|_2^2} \int_{R_x} y \left| C\left(\frac{y-x}{a}\right) a^{-1} \Phi\left(\frac{y-x}{a}\right) \right|^2 dy = \frac{1}{\|k\|_2^2} \int_{R_x} (x + az) |C(z) a^{-1} \Phi(z)|^2 dz = x + a\bar{x}, \tag{27a}$$

$$\begin{aligned} \text{radius} &= \frac{1}{\|k_{ax}\|_2} \left(\int_{R_x} [y - (x + a\bar{x})]^2 \left| C\left(\frac{y-x}{a}\right) a^{-1} \Phi\left(\frac{y-x}{a}\right) \right|^2 dy \right. \\ &= \frac{a}{\|k\|_2} \left(\int_{R_x} (z - \bar{x})^2 |C(z) a^{-1} \Phi(z)|^2 dz \right)^{1/2} = a\Delta\bar{k}. \end{aligned} \tag{27b}$$

The window width is defined as $2a\Delta\bar{k}$. Therefore the integral reproducing kernel window transform can be interpreted as a *time* or *space localization* of the response.⁸ Hence the integral window transform (26a) gives local information on u with the time or space window in (26b).

Since the Fourier transform of the reproducing kernel $k_{ax}(y)$, denoted by $\bar{k}_{ax}(\omega)$, is also a window function, $\hat{k}_{ax}(\omega)$ is given by

$$\bar{k}_{ax}(\omega) = \int_{R_x} e^{-i\omega y} C\left(\frac{y-x}{a}\right) a^{-1} \Phi\left(\frac{y-x}{a}\right) dy = ae^{-i\omega x} \int_{R_x} e^{-i(a\omega)z} (C(z)a^{-1} \Phi(z)) dz = ae^{-i\omega x} \hat{k}(a\omega). \tag{28}$$

To study u in the frequency domain, we shall employ the Parseval identity,⁹ in which the inner product of any two continuous functions is related to the inner product of their Fourier transforms, denoted \hat{u} and \hat{g} respectively, through

$$\langle u, g \rangle = \frac{1}{2\pi} \langle \hat{u}, \hat{g} \rangle. \tag{29}$$

Therefore, if we let $g(y) = k_{ax}(y)$, the integral reproducing kernel transform (12) becomes

$$u^h(x) = \langle u, k_{ax} \rangle = \frac{a}{2\pi} \int_{-\infty}^{\infty} \hat{u}(\omega) e^{i\omega x} \hat{k}(a\omega) d\omega \tag{30}$$

Equation (30) reveals that the integral reproducing kernel window transform also gives local information on $\hat{u}(\omega)$ with a *frequency-window*

$$[\hat{\omega}/a - \Delta\hat{k}/a; \hat{\omega}/a + \Delta\hat{k}/a] \tag{31}$$

and a bandwidth equal to $2\Delta\hat{k}/a$. It is noted that the ratio of the centre frequency and the bandwidth is equal to $\hat{\omega}/2\Delta\hat{k}$, which is independent of the scaling parameter when the support of the window function is within the domain R_x . We wish to construct $C(a, x, y)$ so that the ratio is fairly constant when x is close to the boundary. Employing the definition of the centre frequency, (23a), and the linear transform $\omega' = a\omega$, the centre frequency of $\hat{k}(a\omega)$ is shown to be

$$\text{centre} = \frac{1}{\|\hat{k}_{a\omega}\|_2} \int_{-\infty}^{\infty} \omega |\hat{k}(a\omega)|^2 d\omega = \frac{1}{a} \frac{1}{\|\hat{k}(\omega')\|_2} \int_{-\infty}^{\infty} \omega' |\hat{k}(\omega')|^2 d\omega' = \frac{\hat{\omega}}{2}. \tag{32a}$$

Similarly, using (23b) and the linear transformation $\omega' = a\omega$, the radius of $\hat{k}(a\omega)$ can be shown to be

$$\text{radius} = \frac{1}{\|\hat{k}_{a\omega}\|_2} \left[\int_{-\infty}^{\infty} \left(\omega - \frac{\hat{\omega}}{a} \right)^2 |\hat{k}(a\omega)|^2 d\omega \right]^{1/2} = \frac{1}{a} \frac{1}{\|\hat{k}(\omega')\|_2} \left(\int_{-\infty}^{\infty} (\omega' - \hat{\omega})^2 |\hat{k}(\omega')|^2 d\omega' \right)^{1/2} = \frac{\Delta\hat{k}}{2}, \tag{32b}$$

where

$$\|\hat{k}(a\omega)\|_2 = a^{-1/2} \|\hat{k}(\omega')\|_2 \tag{32c}$$

is the relationship between the norms.

From the above analysis it is interesting to point out that the integral reproducing kernel window transforms in the time or space domain (equation (12)) and in the frequency or wave number domain (equation (30)) study the response u in a rectangular *time-frequency* or *space-wave number* window given by

$$[x + a\bar{x} - a\Delta\bar{k}; x + a\bar{x} + a\Delta\bar{k}] \times [\hat{\omega}/a - \Delta\hat{k}/a; \hat{\omega}/a + \Delta\hat{k}/a]. \tag{33}$$

The above window narrows to pick up the high-frequency or high-wave-number phenomena of u and widens to study the low-frequency or low-wave-number response. This suggests that we employ a flexible time–frequency or space–wave number reproducing kernel window to define adaptive refinement of the local response of u around any point x .¹¹

4.4. Stability analysis

If $\Phi(y)$ is symmetric, $\hat{\omega}$ is zero. Therefore the frequency window is always located at $\omega = 0$ and its frequency band becomes

$$[-\Delta\hat{k}/a; \Delta\hat{k}/a]. \quad (34)$$

The smaller a , the larger is the frequency band; the larger a , the smaller is the frequency band. This implies that the number of sampling points within $B(x)$ must satisfy the relation

$$\Delta x \leq CON \times \pi/(\Delta\hat{k}/2a) = CON \times 2\pi a/\Delta\hat{k} \quad (35)$$

to avoid aliasing.¹⁴ Δx is called the sampling rate. The constant CON will depend on the so-called frame bounds⁹

$$A \leq \frac{2\pi}{\Delta x} |\hat{k}(a\omega)|^2 \leq B. \quad (36)$$

The positive frame bounds coefficients A and B can be estimated numerically. If A/B is close to one, CON will be close to one. Equation (35) is referred to as the stability condition of the reproducing kernel window function methods. In practice CON is chosen much less than one.¹⁵

Remark. With the help of the correction function, we are able to get an explicit expression for the stability condition. See Section 5 and Liu *et al.*¹⁰

It is well known that even if the sampling rate satisfies (35) and (36), there is no good choice for a high-frequency-band window function that can provide accurate frequency and time resolutions of u simultaneously. Therefore for small a an intelligent selection of the frequency band is necessary to be effective. One possible way to employ a larger sampling rate in high-frequency or high-wave-number analysis is the multiple-scale method proposed by Liu *et al.*¹⁵ In this approach the response is divided into multiple frequency bands via a shifting theorem. If $\Phi_{ab}(x)$ is a wavelet, then the centre frequency $\hat{\omega} > 0$. By breaking a up into different scales, we can perform a multiresolution analysis.⁸ An energy ratio method has been developed by Liu and Chen¹¹ to determine the optimal dilation parameter a .

In this paper we restrict ourselves to the analysis of a single frequency band, where $\Phi_{ab}(x)$ is chosen to be the scaling function so that $\hat{\omega} = 0$. Consequently, depending on the choice of a and the scaling function $\Phi(x)$, the integral reproducing kernel window transform $\langle u, k_{ax} \rangle$ is a measure of the amount of change in u at the location $x + a\bar{x}$ with zoom-in (smaller a) and zoom-out (larger a) capability.

4.5. Limitation of very-low-frequency/wave-number analysis

The disadvantage of a single-band reproducing kernel analysis is that the method will break down for very-low-frequency/wave-number analysis, such as for large dilation parameter a . This can be seen by interpreting the reproducing kernel (12) as a continuous time or space convolution

$$u^h(x) = \int_{R_x} \left(C(a, x, y) a^{-1} \Phi\left(\frac{y-x}{a}\right) \right) u(y) dy, \quad (37a)$$

where the reproducing kernel is identified as

$$k(a, y) = C(a, y) a^{-1} \Phi\left(\frac{y}{a}\right). \quad (37b)$$

Applying the Fourier transform to both sides of (37a) and employing the Fubini theorem⁹ on the right-hand side yields

$$\hat{u}^h(\omega) = \hat{k}(a, \omega)\hat{u}(\omega). \quad (38)$$

In order for $\hat{u}^h(\omega)$ to approach $\hat{u}(\omega)$, $\hat{k}(a, \omega)$ must be constructed such that $\hat{k}(a, \omega) = 1$ for all ω ; however, this will violate the Riemann–Lebesgue lemma that $\hat{k}(a, \omega) = 0$ when ω approaches $+\infty$. An approximation of the convolution identity constructs $\hat{k}(a, \omega)$ so that $\hat{k}(a, \omega) \approx 1$ as a approaches zero. It is well known that the family of Gaussian functions

$$a^{-1}\Phi\left(\frac{x}{a}\right) = \frac{1}{a\sqrt{\pi}}e^{-x^2/a^2}, \quad a > 0, \quad (39)$$

will satisfy the above conditions provided that $C(a, x)$ is constructed so that it is equal to one when Φ_{ax} is in the interior of the domain and is fairly constant when Φ_{ax} moves close to the boundary. With this construction we have

$$k(a, x) \approx a^{-1}\Phi\left(\frac{x}{a}\right) = \frac{1}{a\sqrt{\pi}}e^{-x^2/a^2} \quad (40a)$$

and the Fourier transform of $k(a, x)$ is

$$\hat{k}(a, \omega) = e^{-a^2\omega^2/4}. \quad (40b)$$

With the Gaussian function it is noted that a is identified as the standard deviation and $\hat{k}(a, \omega)$ approaches one as a approaches zero. However, when a is large, (36) will break down, since $\hat{k}(a, \omega) \neq 1$ unless the frequency content of $u(x)$ is close to zero, which is very restrictive.

From the above argument the dilation parameter should be chosen within a banded range, say $a_{\min} < a < a_{\max}$. The maximum a_{\max} will be chosen so that (40b) is close to one. The minimum a_{\min} can be a small number provided that the stability condition (35) is met. From our numerical experiments we found that a can take a large value. We believe that the correction function $C(a, x, y)$ indeed improves the stability as well as the accuracy of the interpolation function. Furthermore, owing to the presence of $C(a, x, y)$ which contains the term $x - y$, the reproducing kernel particle interpolation function as given in (12b) will further increase the order of the *shape function* by one so that the convergence rate will also be increased by one when a is large! This will be discussed further in the next section.

We shall employ the Gaussian function and the cubic spline, which is a good approximation to the Gaussian function, as the window function $\Phi_{ax}(y)$ in the subsequent development.

4.6. *hp-like adaptivity*

It is noted that the choice of an optimal dilation parameter of a given window function is analogous to choosing the optimal order of the polynomial resulting from the multiple scale RKPM. It is shown in the numerical examples (Section 7), by changing the size of a , the refinement or dilation parameter of a single window function, the rate of convergence of the L_2 norm (and H_1 norm) of a smooth Laplacian solution varies from 2 to 16 (and 1 to 15) for a Gaussian window, and 2 to 5 (and 1 to 4) for a cubic spline window. This shows that the multiple scale RKPM has a similar feature of the adaptive p -finite element methods. However, the high convergence rate, or equivalently, the high value of p , can be achieved simply by increasing the size of the refinement parameter of a single window function. This avoids the awkward implementation of the traditional p -finite elements, and there is no apprehension of the compatibility (or continuity) along the element boundaries of the *hp*-finite element mesh, since RKPM requires only a set of nodes and the global shape functions can be C^∞ .

As in p -finite element methods, exponential convergence of smooth solutions can be achieved for this class of multiple scale RKPM. However, if there is a discontinuity in the solution, such as a shock,

high convergence behaviour is lost. h -like adaptive reproducing kernel particle methods, comparable to the h -finite element method, can be developed by inserting nodes in the high gradient region and at the same time, narrowing the size of the window function (i.e. a smaller a) to pick up the fine-scale structure of the response. This zooming in process together with the additional of nodes will result in an hp -like adaptive refinement algorithm. Based on the multiple scale decomposition and the highest wavelet scale response, a convergence parameter or an error estimation indicator as presented in Liu and Chen¹⁰ can be employed to locate the adaptivity regions.

5. EXAMPLES OF REPRODUCING KERNEL WINDOW FUNCTIONS

We let the independent functions be

$$\mathbf{P}(\mathbf{x}) = \{1, x\} \quad 1\text{D}, \tag{41a}$$

$$\mathbf{P}(\mathbf{x}) = \{1, x_1, x_2\} \quad 2\text{D}, \tag{41b}$$

$$\mathbf{P}(\mathbf{x}) = \{1, x_1, x_2, x_3\} \quad 3\text{D}. \tag{41c}$$

By substituting the above vector of independent polynomials into (10) and (13b), the correction function $C(a, \mathbf{x}, \mathbf{y})$ can be separated into two terms,

$$C(a, \mathbf{x}, \mathbf{y}) = C_1(a, \mathbf{x}) + C_2(a, \mathbf{x}) \bullet \left(\frac{\mathbf{x} - \mathbf{y}}{a}\right), \tag{42}$$

where C_1 and $C_2 = \{C_{21}, \dots, C_{2\text{NSD}}\}$ are a scalar and a vector respectively which can be defined in terms of the zeroth, first and cross moments. It is noted that all moments are in general a function of the location of the window function, \mathbf{x} , though they are constants if $B(\mathbf{x})$ is not close to the boundary. The expressions for C_1 and C_2 and the stability condition in 1D, 2D and 3D are as follows.

One dimension:

$$C_1(a, x) = \frac{m_{11}}{m_0 m_{11} - m_1^2}, \tag{43a}$$

$$C_2(a, x) = \frac{m_1}{m_0 m_{11} - m_1^2}. \tag{43b}$$

In a numerical approximation Δx has to be chosen such that

$$m_0 m_{11} - m_1^2 > 0. \tag{43c}$$

Two dimensions:

$$C_1(a, \mathbf{x}) = \frac{m_{11} m_{22} - m_{12}^2}{m_0(m_{11} m_{22} - m_{12}^2) - (m_1^2 m_{22} - 2m_1 m_2 m_{12} + m_2^2 m_{11})}, \tag{44a}$$

$$C_{21}(a, \mathbf{x}) = \frac{m_1 m_{22} - m_2 m_{12}}{m_0(m_{11} m_{22} - m_{12}^2) - (m_1^2 m_{22} - 2m_1 m_2 m_{12} + m_2^2 m_{11})}, \tag{44b}$$

$$C_{22}(a, \mathbf{x}) = \frac{m_2 m_{11} - m_1 m_{12}}{m_0(m_{11} m_{22} - m_{12}^2) - (m_1^2 m_{22} - 2m_1 m_2 m_{12} + m_2^2 m_{11})}. \tag{44c}$$

In a numerical approximation Δx_1 and Δx_2 have to be chosen such that

$$m_0(m_{11} m_{22} - m_{12}^2) - (m_1^2 m_{22} - 2m_1 m_2 m_{12} + m_2^2 m_{11}) > 0. \tag{44d}$$

Three dimensions:

$$C_1(a, \mathbf{x}) = \{2m_{12}m_{23}m_{31} + m_{11}m_{22}m_{33} - m_{11}m_{23}^2 - m_{22}m_{31}^2 - m_{33}m_{12}^2\}/D, \quad (45a)$$

$$C_{21}(a, \mathbf{x}) = \{m_1(m_{22}m_{33} - m_{23}^2) + m_2(m_{23}m_{31} - m_{12}m_{33}) + m_3(m_{12}m_{23} - m_{31}m_{22})\}/D, \quad (45b)$$

$$C_{22}(a, \mathbf{x}) = \{m_2(m_{33}m_{11} - m_{31}^2) + m_3(m_{31}m_{12} - m_{23}m_{11}) + m_1(m_{23}m_{31} - m_{12}m_{33})\}/D, \quad (45c)$$

$$C_{23}(a, \mathbf{x}) = \{m_3(m_{11}m_{22} - m_{12}^2) + m_1(m_{12}m_{23} - m_{31}m_{22}) + m_2(m_{31}m_{12} - m_{23}m_{11})\}/D, \quad (45d)$$

where D , the Jacobian of the 4×4 matrix $\mathbf{M}(\mathbf{x})$ is given by

$$\begin{aligned} D = & m_0(2m_{12}m_{23}m_{31} + m_{11}m_{22}m_{33} - m_{11}m_{23}^2 - m_{22}m_{31}^2 - m_{33}m_{12}^2) \\ & + m_1^2(m_{23}^2 - m_{22}m_{33}) + m_2^2(m_{31}^2 - m_{33}m_{11}) - m_3^2(m_{12}^2 - m_{11}m_{22}) \\ & + 2m_1m_2(m_{33}m_{12} - m_{23}m_{31}) + 2m_2m_3(m_{11}m_{23} - m_{31}m_{12}) \\ & + 2m_3m_1(m_{22}m_{31} - m_{12}m_{23}). \end{aligned} \quad (45e)$$

In a numerical approximation Δx_1 , Δx_2 and Δx_3 have to be chosen such that

$$D > 0. \quad (45f)$$

Because of the special properties of the moments, $C_1(a, \mathbf{x}) = 1$ and $\mathbf{C}_2(a, \mathbf{x}) = \mathbf{0}$ when the window function is within R_x , whereas $C_1(a, \mathbf{x}) \neq 1$ and $\mathbf{C}_2(a, \mathbf{x}) \neq \mathbf{0}$ when the window function is close to the boundary. It is particularly important to point out that if a is large, the support $B(\mathbf{x})$ is large. Then the linear term appearing in (42), $\mathbf{C}_2(a, \mathbf{x}) \bullet [(\mathbf{x} - \mathbf{y})/a]$, plays an important part in the accuracy and convergence rate of the method.

From (12) and (13) the reproducing kernel function and the approximation $u^h(\mathbf{x})$ are given by

$$k(a, \mathbf{x}, \mathbf{y}) = C(a, \mathbf{x}, \mathbf{y})\mathbf{E}(a)\Phi\left(\frac{\mathbf{y} - \mathbf{x}}{a}\right), \quad (46a)$$

$$u^h(\mathbf{x}) = \int_{R_x} k(a, \mathbf{x}, \mathbf{y})u(\mathbf{y})dR_y. \quad (46b)$$

Therefore, if $\Phi(r)$ ($r = \|\mathbf{x} - \mathbf{y}\|_2$) is the cubic spline function, the order of polynomials of $\Phi(r)$ is equal to three. In a Galerkin formulation the convergence rates of the solution (L2 norm) and its first derivatives (H1 norm) (see Section 7 for definitions) are expected to be four and three respectively. However, when a is larger, observing from the discretized equations (46a) and (42) (with \mathbf{y} replaced by \mathbf{x}_j), the order of the polynomial of the reproducing kernel $k(a, \mathbf{x}, \mathbf{y})$ can be increased by one, so that the L2 and H1 convergence rates can be as high as five and four respectively. This unusual phenomenon is observed in our 1D numerical experiments. The convergence rates are much higher when $\Phi(r)$ is replaced by a Gaussian function.

Another interesting observation can also be made from (42). In order to increase the convergence rate by one order, it is suggested to underintegrate the first moment so that it is close to zero, then the term $\mathbf{C}_2(a, \mathbf{x}) \bullet [(\mathbf{x} - \mathbf{y})/a]$ can act as a stabilization term to the SPH methods. At the same time this stabilization term can also improve the accuracy as well as the convergence rate. One way to achieve this is to use the trapezoidal rule to integrate the matrix $\mathbf{M}(\mathbf{x})$ (i.e. the moments) at each \mathbf{x} .

For higher-order polynomials, as well as other independent functions, $\mathbf{P}(\mathbf{x})$ can also be similarly investigated for this class of RKPM interpolation functions; however, from our numerical experiments we found that the use of linear polynomials is accurate enough for most purposes. We also found that linear polynomials give numerically more stable calculations than higher-order polynomials.

6. SIMILARITIES AMONG SPH, DEM, EFGM AND RKPM INTERPOLATION FUNCTIONS

For an illustration of the comparison among the various interpolation functions, only one-dimensional linear polynomials $\mathbf{P}(\mathbf{x}) = \{1, x\}$ are implemented. The window function $\Phi(x)$ is chosen to be a positive even function. If we use the trapezoidal rule to discretize the reproducing kernel (12), the shape function $N_J^R(x)$ of RKPM is ($a > 0$)

$$N_J^R(x) = \left[C_1(x) + C_2(x) \left(\frac{x - x_J}{a} \right) \right] a^{-1} \Phi \left(\frac{x - x_J}{a} \right) \Delta M_J, \quad (47a)$$

$$C_1(x) = m_{11} / (m_0 m_{11} - m_1^2), \quad (47b)$$

$$C_2(x) = m_1 / (m_0 m_{11} - m_1^2), \quad (47c)$$

where ΔM_J is the J th particle mass. If $m_0 = 1$, $m_{11} \neq 0$ and $m_1 = 0$, the SPH interpolation shape function can be obtained. As a matter of fact, the shape functions of RKPM and SPH are equivalent in the interior, but there is a big difference when they are close to a boundary. Hence the SPH methods are not accurate when boundaries are present.

Direct differentiation of (47a) gives

$$N_{J,x}^R(x) = [C_1'(x) + C_2'(x)(x - x_J) + C_2(x)] a^{-1} \Phi(x - x_J) \Delta M_J \\ + [C_1(x) + C_2(x)(x - x_J)] a^{-1} \Phi'(x - x_J) \Delta M_J, \quad (48a)$$

$$C_1' = \frac{m_{11}'}{m_0 m_{11} - m_1^2} - \frac{m_{11}(m_0' m_{11} + m_0 m_{11}' - 2m_1 m_1')}{(m_0 m_{11} - m_1^2)^2}, \quad (48b)$$

$$C_2' = \frac{m_1'}{m_0 m_{11} - m_1^2} - \frac{m_1(m_0' m_{11} + m_0 m_{11}' - 2m_1 m_1')}{(m_0 m_{11} - m_1^2)^2}, \quad (48c)$$

In the above expressions, a prime denotes differentiation with respect to the argument.

Without going into details, the MLSM, DEM and EFGM interpolation functions can be written as⁴

$$\bar{N}_J(x) = \sum_{j=1}^2 P_j(x) [\mathbf{A}^{-1}(x) \mathbf{B}(x)]_{jJ} \quad (49a)$$

when constant and linear polynomials $P_j(x)$ are employed. The matrices $\mathbf{P}(x)$, $\mathbf{A}(x)$ and $\mathbf{B}(x)$ are defined by

$$\mathbf{P}^T(x) = \{P_1, P_2\}, \quad P_1(x) = 1, \quad P_2(x) = x, \quad (49b)$$

$$\mathbf{A}(x) = \sum_{I=1}^n a^{-1} \Phi \left(\frac{x - x_I}{a} \right) \mathbf{P}(x_I) \mathbf{P}^T(x_I), \quad (49c)$$

$$\mathbf{B}(x) = \left[a^{-1} \Phi \left(\frac{x - x_1}{a} \right) \mathbf{P}(x_1), \dots, a^{-1} \Phi \left(\frac{x - x_n}{a} \right) \mathbf{P}(x_n) \right], \quad (49d)$$

where n is the number of points in the neighbourhood of x for which the weight function $a^{-1} \Phi[(x - x_I)/a] \neq 0$, and x_I are the nodal co-ordinates of u_I . The derivatives of $\bar{N}_J(x)$ of EFGM given by Belytschko *et al.*⁴

$$\bar{N}_{J,x}^E(x) = \sum_{j=1}^2 \{P_{j,x} [\mathbf{A}^{-1} \mathbf{B}]_{jJ} + P_j [\mathbf{A}_{,x}^{-1} \mathbf{B} + \mathbf{A}^{-1} \mathbf{B}_{,x}]_{jJ}\} \quad (50a)$$

and the derivative of the matrix \mathbf{A}^{-1} is given by

$$\mathbf{A}_{,x}^{-1} = -\mathbf{A}^{-1}\mathbf{A}_{,x}\mathbf{A}^{-1}. \tag{50b}$$

The derivation of $\bar{N}_{J,x}(x)$ of DEM developed by Nayroles *et al.*³ assumes \mathbf{A} and \mathbf{B} constant, so that

$$\bar{N}_{J,x}^D(x) = \sum_{j=1}^2 P_{j,x}[\mathbf{A}^{-1}\mathbf{B}]_{jJ}. \tag{51}$$

It is noted that no particle mass ΔM_J or nodal length Δx_J is included in (49)–(51). Furthermore, the matrices \mathbf{A} and \mathbf{B} need to be computed at each quadrature point x .

Although it is not very apparent, an interesting result arises under the following three conditions.

1. The nodal co-ordinates x_j are equally spaced.
2. The trapezoidal rule is used to numerically integrate $\mathbf{M}(x)$.
3. The integration weights Δx_j , ΔM_J and ρ_j in (47a) and (48a) are all set equal to one.

\bar{N}_J^R and $\bar{N}_{J,x}^R$ in (47a) and (48a) can be shown to be equivalent to N_J and $\bar{N}_{J,x}^E$ in (49a) and (50a) respectively.

With the above assumptions, $\bar{N}_{J,x}^D$ can be similarly defined from the RKPM interpolation functions (with $\Delta M_J = 1$):

$$N_{J,x}^D(x) = -\frac{m_1}{a(m_0m_{11} - m_1^2)} a^{-1}\Phi\left(\frac{x-x_J}{a}\right)\Delta M_J - \frac{m_0}{a(m_0m_{11} - m_1^2)} \left(\frac{x-x_J}{a}\right) a^{-1}\Phi\left(\frac{x-x_J}{a}\right)\Delta M_J. \tag{52}$$

Comparing (48a) with (52), depending on $\Phi(x)$, the derivative $\bar{N}_{J,x}^D$ might not be an accurate approximation to $\bar{N}_{J,x}^R$, especially for large a -values. For example, let us use the cubic spline functions

$$\Phi(r) = \frac{2}{3} - 4r^2/\Delta x^2 + 4r^3/\Delta x^3 \quad \text{for } 0 \leq r/\Delta x \leq \frac{1}{2}, \tag{53a}$$

$$\Phi(r) = \frac{4}{3} - 4r/\Delta x + 4r^2/\Delta x^2 - 4r^3/3\Delta x^3 \quad \text{for } \frac{1}{2} \leq r/\Delta x \leq 1, \tag{53b}$$

$$r = \|\mathbf{y} - \mathbf{x}\|_2 \tag{53c}$$

as the window function and set the dilation parameter such that $a = 2^{j+0.5}$ for $j \geq 0$. The parameter $j = 0$ is adjusted so that Δx is right at the stability limit (see equations (35) and (36)). This corresponds to the smallest time bandwidth or the largest frequency bandwidth of the window function. Let us consider a set of 21 equally spaced nodes with $\Delta x = 0.3$ representing the domain $0 \leq x \leq 6.0$. If we use a trapezoidal rule to discretize the domain, then $\Delta x_1 = \Delta x_{21} = 0.15$ and $\Delta x_2 = \Delta x_3 = \dots = \Delta x_{20} = 0.3$. With this discretization and $\rho = 1$ the shape function given in (47a) and its derivatives using (48) (exact) and (52) (approximate) are depicted in Figures 1 and 2 respectively for nodes 1, 10 and 21. As can be seen for $j = 0$ (right at the stability limit), the shape function and its derivatives look similar to those for the usual linear finite elements; however, the derivatives of RKPM shape functions are continuous and try to reproduce the finite element discontinuous derivatives. For $j = 1.0$ and 2.0 there is not much difference between the two derivatives, especially for node 10 where the support is within the domain. There is a large difference

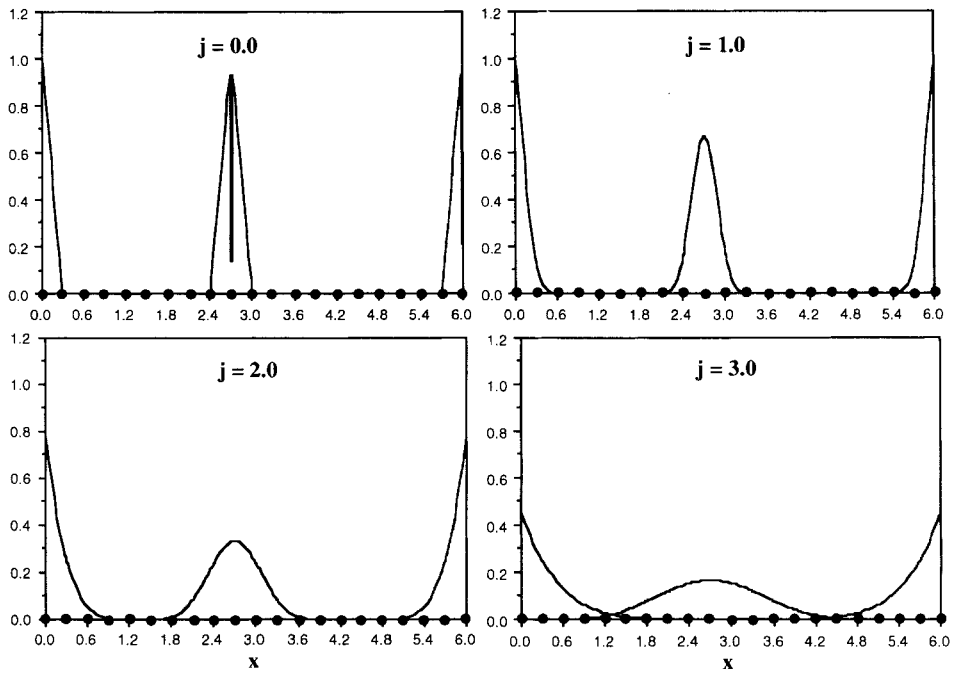


Figure 1. Plots of interpolation functions at node 1 ($x = 0.0$), node 10 ($x = 2.7$) and node 21 ($x = 6.0$)

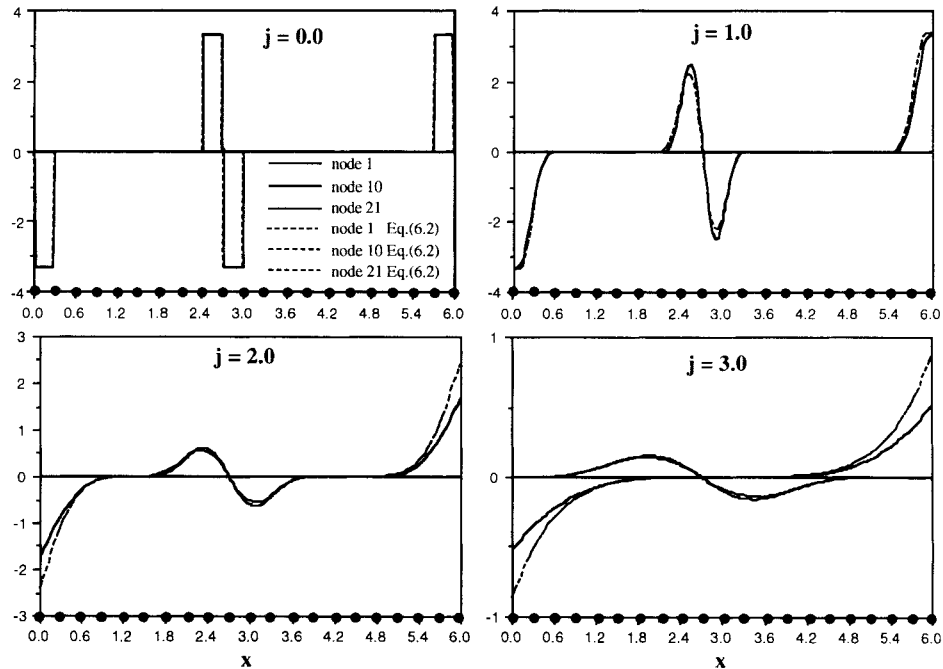


Figure 2. Derivatives of interpolation functions using exact and approximate formulae at node 1 ($x = 0$), node 10 ($x = 2.7$) and node 21 ($x = 6.0$)

between the two derivatives when the shape functions are located at the boundaries ($x = 0.0$ and 6.0). This difference will produce inaccurate derivatives in (52) and the solution deteriorates.

If we choose $\Phi(r)$ as a Gaussian function such that

$$a^{-1}\Phi\left(\frac{x-y}{a}\right) = \frac{1}{a\sqrt{\pi}}e^{-(x-y)^2/a^2}, \quad a > 0, \tag{54a}$$

$$a = 2^{j+0.5}\sigma, \quad j \geq 0, \quad \text{and} \quad \frac{\Delta x}{\sigma\sqrt{\pi}} = \frac{1}{\sqrt{2}}. \tag{54b}$$

With σ as the standard deviation, the *exact* derivative given in (48) and the *approximate* derivative given in (52) become respectively

$$N_{J,x}^R(x) = \left[C_1'(x) + C_2'(x)\left(\frac{x-x_J}{a}\right) + \frac{C_2(x)}{a} \right] a^{-1}\Phi\left(\frac{x-x_J}{a}\right)\Delta M_J - \frac{2}{a} \left[C_1(x) + C_2(x)\left(\frac{x-x_J}{a}\right) \right] \left(\frac{x-x_J}{a}\right) a^{-1}\Phi\left(\frac{x-x_J}{a}\right)\Delta M_J, \tag{55a}$$

$$N_{J,x}^D(x) = -\frac{m_0}{a(m_0m_{11} - m_1^2)}\left(\frac{x-x_J}{a}\right) a^{-1}\Phi\left(\frac{x-x_J}{a}\right)\Delta M_J - \left\{ \frac{m_1}{a(m_0m_{11} - m_1^2)} \right\} a^{-1}\Phi\left(\frac{x-x_J}{a}\right)\Delta M_J. \tag{55b}$$

When the derivatives of the shape function are not evaluated close to the boundaries ($m_1 = 0$), the two derivatives are represented with the *same* function $a^{-1}\Phi((x - x_J)/a)$ but different coefficients. When the derivatives are evaluated close to the boundaries, the two derivatives are again represented by the same Gaussian function and from numerical experiments the quadratic term $[(x - x_J)/a]^2$ appearing in (55a) does not play an important role. Therefore (55b) is a good approximation to (55a). This is further elaborated in the next section.

7. NUMERICAL EXPERIMENTS

7.1. Convergence study

We employ a one-dimensional Laplacian-type equation

$$u_{,xx} + 2s^2\text{sech}^2[s(x-3)]\tanh[s(x-3)] = 0 \tag{56a}$$

with essential boundary conditions

$$u(0) = -\tanh(3s), \tag{56b}$$

$$u(6) = \tanh(3s). \tag{56c}$$

The parameter s controls the degree of localization of the gradient of u ($u_{,x}$). As s increases, $u_{,x}$ has an increasing gradient. The exact solution is

$$u(x) = \tanh[s(x-3)]. \tag{56d}$$

In a numerical approximation we employ a Galerkin formulation of (56a) and the boundary conditions are enforced via the standard Lagrange multiplier approach. For a detailed description of this problem and the implementation of the boundary conditions see Reference 4.

We shall utilize the cubic spline and Gaussian function described in Section 6 as window functions. For simplicity, linear polynomials $\mathbf{P}(x) = \{1, x\}$ and $s = 10$ are used throughout. Five different discretizations with $\Delta x = 0.3$ (21 nodes), 0.15 (41 nodes), 0.075 (81 nodes), 0.0375 (161 nodes) and

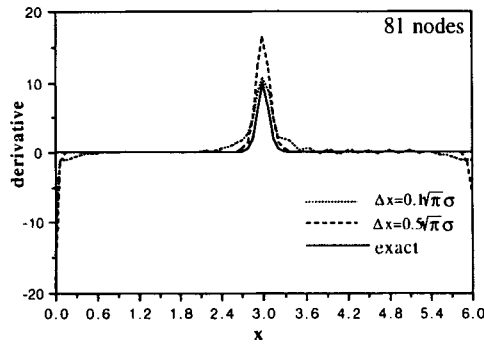


Figure 3. Comparison of SPH solutions with exact solution

0-025 (241 nodes) are solved. It is noted that, similarly to SPH, DEM and EFGM, RKPM does not require an element or element connectivities. The standard L2 and H1 error norms are defined as

$$(L2norm)^2 = \int_0^6 (u^{exact} - u^h)^2 dx, \tag{57a}$$

$$(H1norm)^2 = \int_0^6 (u_x^{exact} - u_{,x}^h)^2 dx, \tag{57b}$$

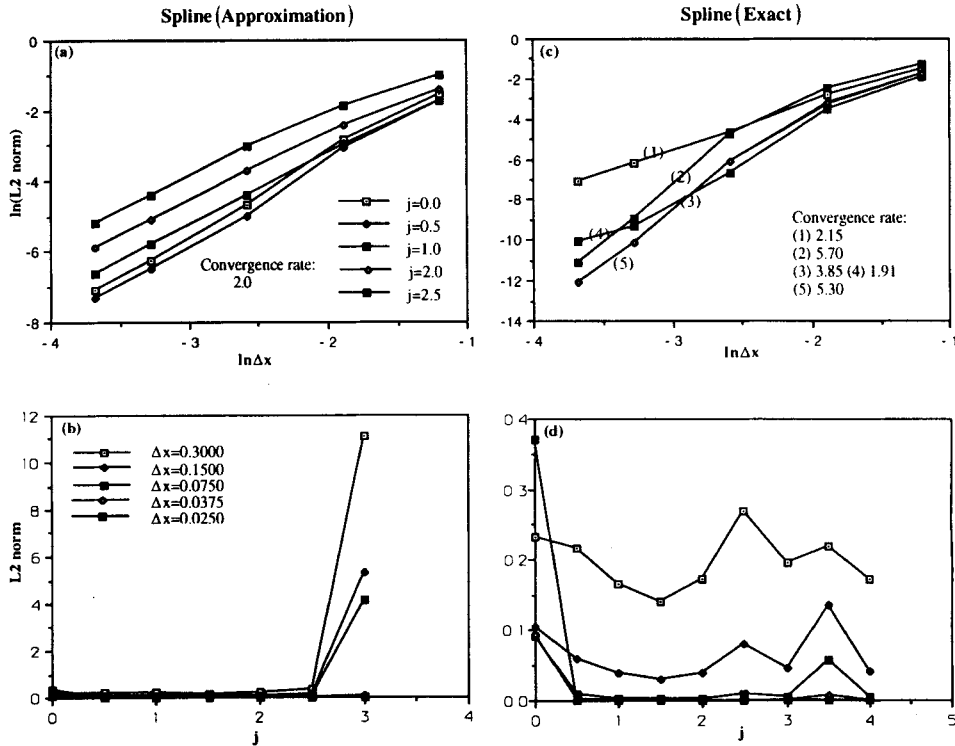


Figure 4. L2 norm and convergence plots using approximate and exact formulae for the derivatives of the interpolation function using a cubic spline window function

The rates of convergence are defined as the slopes p and q appearing in the $\ln(\text{error})$ versus $\ln\Delta x$ equations

$$\ln(\text{L2 norm}) = \ln G_1 + p \ln \Delta x, \tag{58a}$$

$$\ln(\text{H2 norm}) = \ln G_2 + q \ln \Delta x. \tag{58b}$$

As the constants G_1 and G_2 become smaller, the method becomes more accurate. Also, the higher p and q , the faster is the rate of convergence. To obtain the convergence plots, 10 and 12 Gauss quadrature points are used to integrate the matrices and errors respectively. Nevertheless, only three to five Gauss quadrature points are sufficient to integrate the matrices accurately. The trapezoidal rule is used to integrate all the moments.

The exact solution of u_x and the SPH Galerkin approximations using a Gaussian window, 81 nodes ($\Delta x = 0.075$) and $\sigma = \Delta x/(0.1\sqrt{\pi})$ and $\Delta x/(0.5\sqrt{\pi})$ are depicted in Figure 3. It is seen that the SPH solution depends very much on the dilation parameter (or the standard deviation). As a matter of fact, for a given Δx , the larger σ (the larger the window), the better is the gradient at the centre. However, both choices of σ give bad approximations of the gradient of u at the boundaries. This confirms that SPH interpolation functions do not work well with boundaries.

The L2 and H1 norm plots for the spline window function are shown in Figures 4 and 5 respectively. As can be seen in Figures 4(b) and 4(d) and Figures 5(b) and 5(d), (48) works better than (52) when a is large (i.e. j is large). Similar conclusions can also be drawn from Figures 4(a) and 4(c) and Figures

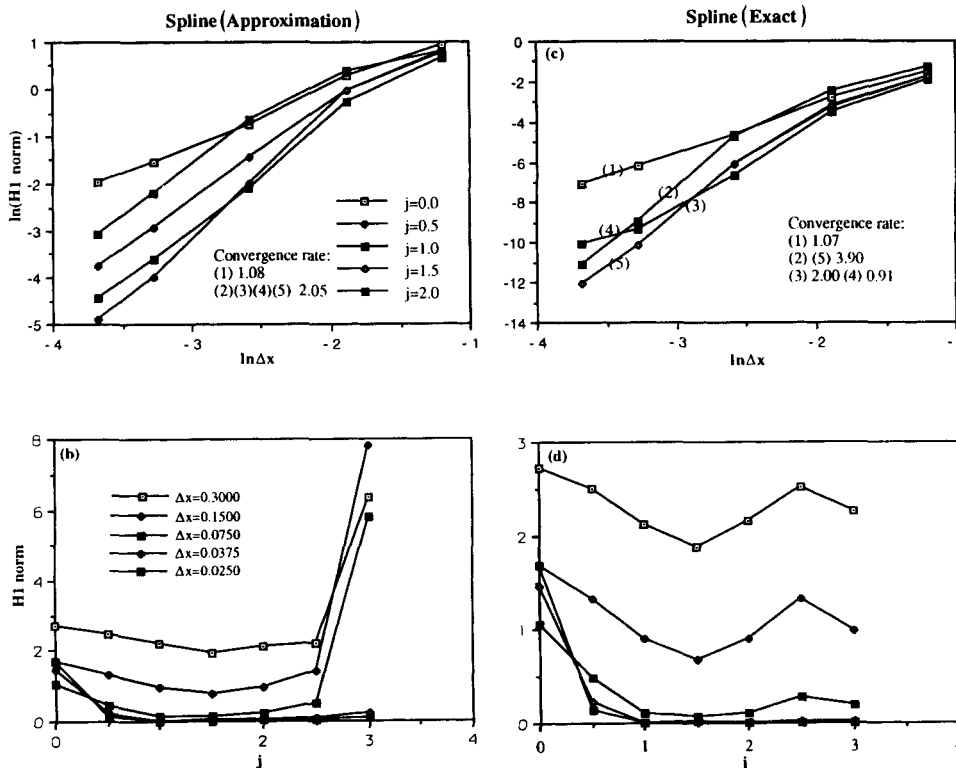


Figure 5. H1 norm and convergence plots using approximate and exact formulae for the derivatives of the interpolation function using a cubic spline window function

5(a) and 5(c). One interesting observation is that the L2 and H1 convergence rates (see Figures 4(c) and 5(c)) can be as high as 5.7 and 3.9 respectively. This confirms our analysis in Section 6.

Similarly, the L2 and H1 convergence rate plots shown in Figures 6 and 7 respectively are produced using the Gaussian window function. The standard deviation and dilation parameter are chosen such that

$$\frac{\Delta x}{\sigma\sqrt{\pi}} = \frac{1}{\sqrt{2}}, \quad a = 2^{j+0.5}\sigma. \tag{58c}$$

As predicted in Section 6, the two formulae for the derivatives give virtually identical results. It is interesting to point out that the L2 and H1 convergence rates can go as high as 15.92 and 14.91 respectively. These numerical results show that a meshless p -like adaptive variable-node multiple scale RKPM can be obtained by a combination of the translation and dilation of a single flexible window function (Liu and Chen¹¹). Finally, in Table I the peak values of u_x are compared with the standard linear finite element method (FEM) and the exact solution. RKPM is able to capture the high resolution of the steep localized gradient using a flexible space-wave number Gaussian window function.

Remark. The two parameters in the scaling function provide the ability to translate and dilate the window function. Translation is required to move the window function around the domain since the window functions themselves have a compact support. The ability to translate replaces the need to define elements. The dilation parameter is used to provide refinement. This dilation parameter also

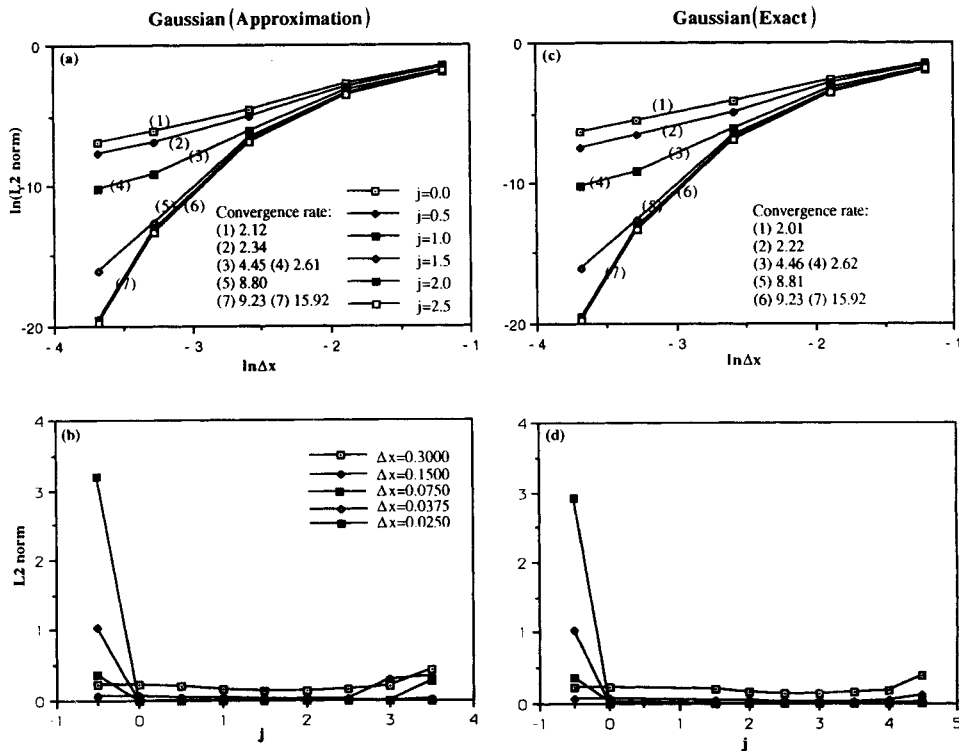


Figure 6. L2 norm and convergence plots using approximate and exact formulae for the derivatives of the interpolation function using a Gaussian window function

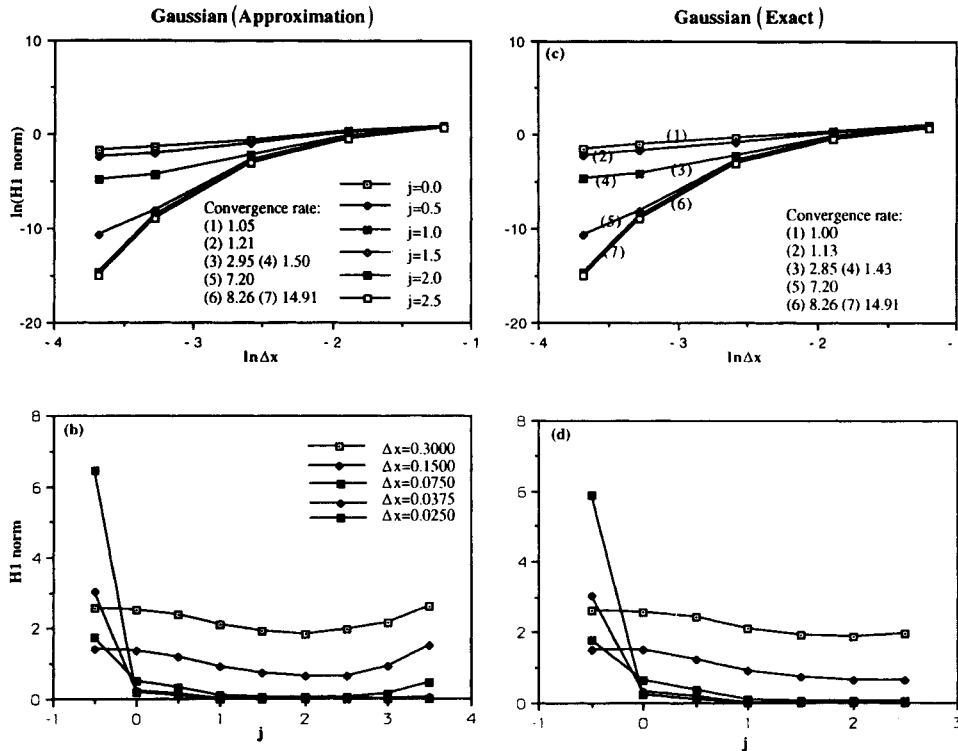


Figure 7. H1 norm and convergence plots using approximate and exact formulae for the derivatives of the interpolation function using a Gaussian window function

controls the convergence rate of the multiple scale RKPM. As shown in the above numerical examples, the rate of convergence of the L_2 norm (and H_1 norm) of a smooth Laplacian solution varies from 2 to 16 (and 1 to 15) for a Gaussian window; and 2 to 5 (and 1 to 4) for a cubic spline window by simply changing the refinement or dilation parameter of a single window function. This combination of translation and dilation produces a meshless p -like adaptive variable-node multiple scale RKPM. The larger the dilation parameter, the smaller the frequency band is in the solution, and the larger the critical time step becomes in dynamic analyses (Liu *et al.*¹⁰). The refinement parameter transformation between the time and frequency domain (or space and wave number) controls the solution space. This introduces the ability to choose the size of the frequency or wave number range in the calculation. The application of this multiple frequency and/or wave number concepts to multi-grid methods will be exploited in a companion paper.

Table I. Comparison of derivative peak values among RKPM, EM and exact solution

Reproducing kernel particle method (Gaussian)								
Δx	$j = 0.5$	$j = 1.0$	$j = 1.5$	$j = 2.0$	$j = 2.5$	$j = 3.0$	FEM	Exact
0.300	3.5100	3.8315	4.6672	5.3217	5.4837	5.1865	3.3168	9.9923
0.15	6.3599	6.7798	7.6968	8.3095	8.5252	8.5419	6.0343	9.9980
0.075	8.7556	9.0684	9.6081	9.8416	9.8864	9.9010	8.4687	9.9995
0.0375	9.6788	9.7992	9.9605	9.9984	9.9995	9.9996	9.5562	9.9998
0.005	9.8580	9.9155	9.9861	9.9998	9.9999	9.9999	9.7967	9.9999

7.2. 2D advection–diffusion equation

The scalar advection–diffusion equation is given as

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \nu \nabla^2 \phi. \quad (59)$$

Advection-dominated flow skewed to the mesh is considered here. The problem statement is given by many authors; see e.g. Reference 16. It has been noted that the streamline upwind/Petrov–Galerkin method requires a discontinuity-capturing term in the weighting function for smoothing the solution near the discontinuity. However, the multiple solution decomposition analysis developed by Liu and Chen¹¹ improves the solution without the discontinuity-capturing term and enables us to determine where the local refinement is needed. In this numerical example a square window with the dilation parameter $a = 1.1892$ is employed. A 3×3 dyadic (power of 2) multiple scale decomposition (a scaling function scale and two wavelet scales) is performed on the computed solution using 21×21 nodes. The reference solution, the total solutions (using 21×21 nodes) and the low scale solution

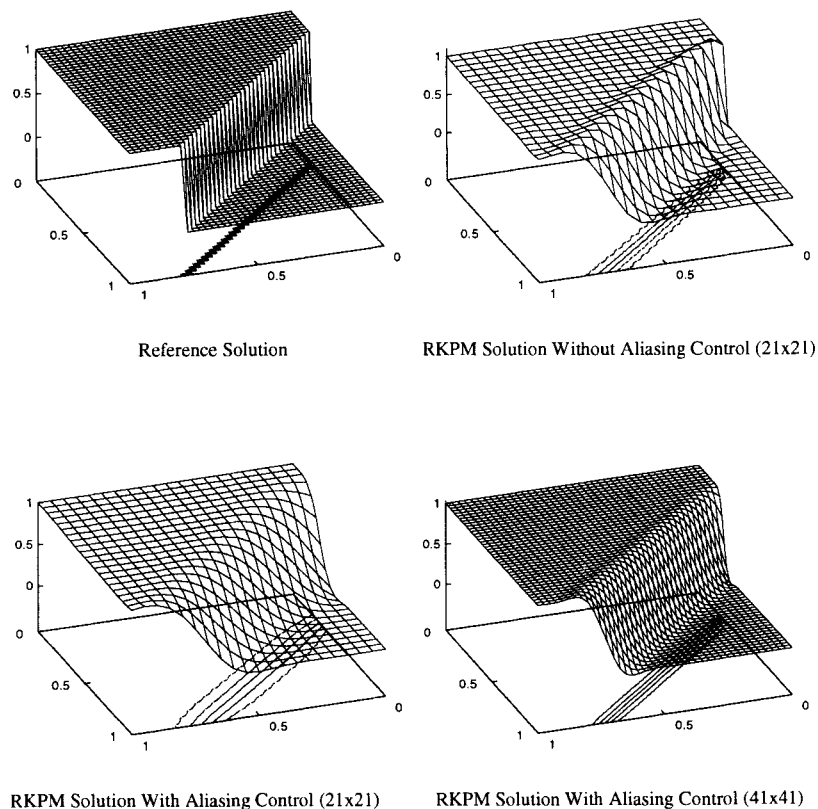


Figure 8. Solutions of 2D advection–diffusion equations

using different refinements are depicted in Figure 8. It is noticed that the low scale solution consists of only the scaling function solution, and the higher wavelet scale solutions, which contain mostly aliasing wave number solutions, are removed. As can be seen, the comparison with the 41×41 nodes solution and the reference solution is fairly good.

7.3. 2D compressible Euler equation

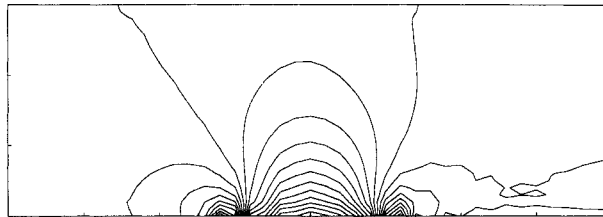
The compressible Euler equations in conservation form are written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial \mathbf{x}_i} = 0, \tag{60}$$

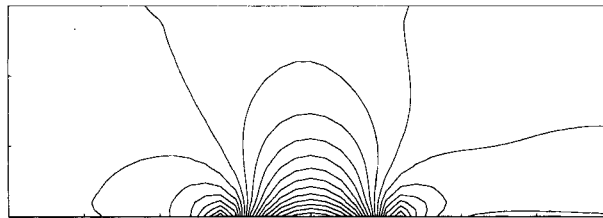
where the 2D conservative variables \mathbf{U} and the Euler flux \mathbf{F}_i are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho \varepsilon \end{bmatrix}, \quad \mathbf{F}_i = \begin{bmatrix} \rho u_i \\ \rho u_1 u_i + p \delta_{i1} \\ \rho u_2 u_i + p \delta_{i2} \\ u_i (\rho \varepsilon + p) \end{bmatrix}. \tag{61}$$

(a) RKPM solution without aliasing control



(b) RKPM solution with aliasing control (2 x 2 decomposition)



(c) High wave number band (wavelet) solution

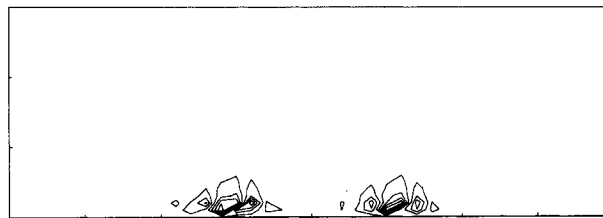


Figure 9. Solutions of 2D compressible Euler equations ($M_0 = 0.5$)

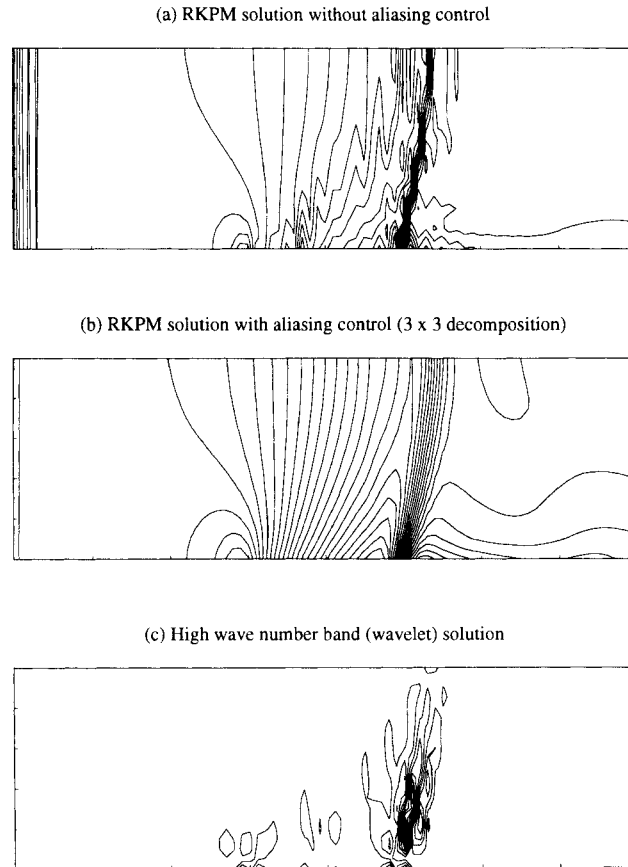
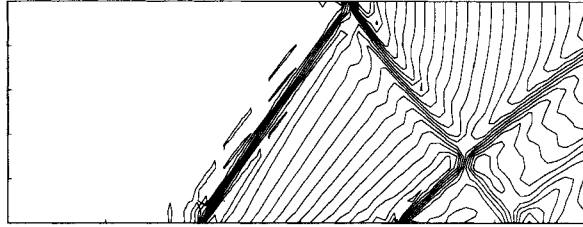


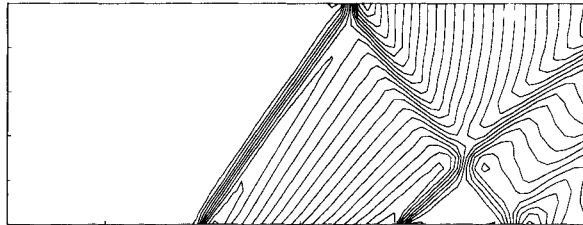
Figure 10. Solutions of 2-D Compressible Euler Equations ($M_0 = 1.4$)

The numerical example is a thin biconvex airfoil approximation problem discussed by Hughes and Tezduyar.¹⁷ Three Mach numbers ($M_0 = 0.5, 0.84$ and 1.4) of inflow conditions are used for subsonic, transonic and supersonic flows, respectively. For the case of $M_0 = 0.5$, 41×31 nodal points are employed and the computational domain is $(-2.0, 2.0) \times (0, 1.5)$. For transonic flow condition ($M = 0.84$), 81×21 nodes and the domain of $(-2.0, 2.0) \times (0.0, 1.0)$ are used while the domain of $(-1.5, 1.5) \times (0, 1.0)$ and 66×21 nodes are employed for supersonic flow ($M = 1.4$). In this example, a square window of $a = 1.23114$ is used. Figures 9, 10 and 11 show the contour plots for the pressure for each inflow Mach number. Similar to the previous example, a 2×2 multiple scale decomposition for subsonic and supersonic flow and a 3×3 for transonic flow have been performed to remove the high wave number band solutions. Those figures depict the numerical solutions without aliasing control, numerical solutions with aliasing control (by taking high and medium wave number band solutions out), and the high wave number band (wavelet) solutions. The detailed computational procedures will be presented in a companion paper.

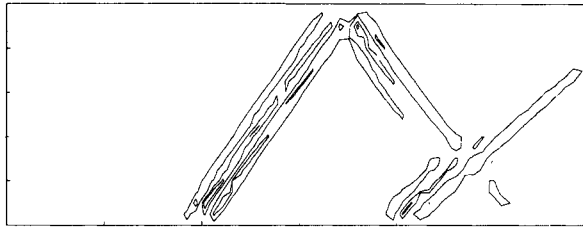
(a) RKPM solution without aliasing control



(b) RKPM solution with aliasing control (2 x 2 decomposition)



(c) High wave number band (wavelet) solution

Figure 11. Solutions of 2-D Compressible Euler Equations ($M_0 = 1.4$)

8. CONCLUSIONS

In this paper, mesh- or grid-free interpolation functions are reviewed and studied. The dilation and translation of a window function, the integral window transform and the SPH interpolation kernel functions are reviewed. Through an understanding of the merits and deficiencies of SPH, MLSM, DEM and EFGM, a new continuous reproducing kernel particle interpolation function is derived in terms of a flexible time–frequency or space–wave number localized window function. Comparing with the SPH interpolation function, a continuous correction function to the SPH methods, which is composed of the various moments of the window function, is identified. This is the key success of the RKPM. It is noted that no artificial viscosity is needed in RKPM as compared to SPH methods.

The effect of the dilation parameter and the stability condition of this new discretized reproducing kernel particle interpolation function are discussed. The convergence rate of this class of RKPM interpolation functions using a Galerkin method is shown to be at least one order higher than that of the window function when the dilation parameter is large. In particular, exponential convergence is observed with a Gaussian window function. This results in the development of a meshless *hp*-like

adaptive variable-node multiple scale RKPM, see Liu and Chen.¹¹ Since the correction function and the window function can be chosen to be smooth, the solution as well as its derivatives are continuous throughout the entire domain of interest, unlike the usual finite element methods. In addition, borrowing the multi-resolution idea of wavelet theories, RKPM can provide a multiple scale decomposition which enhances the physical interpretation of the computed solution. The numerical experiments confirm the theoretical analysis presented in this paper.

ACKNOWLEDGEMENTS

The support of this research by AFSOR grant number F49620-92-J-0505 and ONR grant number N00014-93-I-0292 to Northwestern University is gratefully acknowledged.

REFERENCES

1. L. Lucy, 'A numerical approach to testing the fission hypothesis', *A. J.*, **82**, 1013–1024 (1977).
2. R. A. Gingold and J. J. Monaghan, 'Smoothed particle hydrodynamics: theory and application to non-spherical stars', *Mon. Not. R. Astron. Soc.*, **181**, 275–389 (1977).
3. B. Nayroles, G. Touzot and P. Villon, 'Generalizing the finite element method: diffuse approximation and diffuse elements', *Comput. Mech.*, **10**, 307–318 (1992).
4. T. Belytschko, Y. Y. Lu and L. Gu, 'Element free Galerkin methods', *Int. numer. methods eng.*, **37**, 229–256 (1994).
5. P. Lancaster and K. Salkauskas, 'Surfaces generated by moving least squares methods', *Math. Comput.*, **37**, 141–158 (1981).
6. L. D. Libersky and A. G. Petschek, 'Smooth particle hydrodynamics with strength of materials', *Lecture Notes in Physics*, Vol. 395, *Advances in the Free-Lagrange Method*, Springer, Berlin, 1990, pp. 248–257.
7. G. R. Johnson, E. H. Peterson and R. A. Stryk, 'Incorporation of an SPH option into the EPIC code for a wide range of high velocity impact computations', *Preprint*, 1993.
8. C. K. Chui, *An Introduction to Wavelets*, Academic, New York, 1992.
9. Daubechies, *CBMS/NSF Series in Applied Mathematics*, No. 61, *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.
10. W. K. Liu, S. Jun, S. Li, J. Adee and T. Belytschko, 'Reproducing Kernel Particle Methods for Structural Dynamics', *Int. j. numer. methods. eng.*, **38**, 1655–1680 (1995).
11. W. K. Liu and Y. Chen, 'Wavelet and multiple scale reproducing kernel methods', *Int. j. numer. methods fluids*, (1995).
12. J. J. Monaghan, 'Why particle methods work', *SIAM J. Sci. Stat. Comput.*, **3**, 422–433 (1982).
13. J. J. Monaghan, 'An introduction to SPH', *Comput. Phys. Commun.*, **48**, 89–96 (1988).
14. E. O. Brigham, *The Fast Fourier Transform*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
15. W. K. Liu, Y. Zhang and M. R. Ramirez, 'Multiple scale finite element methods', *Int. j. numer. methods eng.*, **32**, 969–990 (1991).
16. A. N. Brooks and T. J. R. Hughes, 'Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **32**, 199–259 (1982).
17. T. J. R. Hughes and T. E. Tezduyar, 'Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations', *Comput. Methods Appl. Mech. Eng.*, **45**, 217–284 (1984).