A Meshfree method and its error estimate: A literature survey

MATH221 term paper

Xiaohu Liu 15551079
Dept. Civil Engineering

Summary: In this paper, the concept of meshless methods is introduced. In particular, the formulation of Reproducing Kernel Particle Methods(RKPM) is derived and discussed in detail. The error estimate of RKPM is also presented and it shows the convergence of the method.

1.Introduction

In the field of engineering computation, the *Finite Element Method*(FEM) has been the most widely used numerical method for many years. One of the key feature of FEM is the mesh, a topological map that connects the discretization of the continuum. Since mesh is an artifactD, it often does not represent the real physical compatibility of the continuum. For example, in the simulation of manufacturing processes such as modling, the extremely large deformation involved will distort the mesh. As a result, frequent remeshing is needed during the computation. Such remeshing is time-consuming and sometimes can harm the computational accuracy. So it would be better if we discretize the continuum by only a set of particles. This is the motivation of *Meshfree Methods*(meshless methods, particle methods are other common names). Comparing to FEM, the key advantage of meshfree methods is they can easily handle very large deformations, since the connectivity among particles is generated as part of the computation and can change with time. Originated in the late seventies, the meshfree methods experienced fast development during the past ten years. Many different methods are proposed. For the reviewing of meshfree methods, please refer to [1] and [2].

The objective of this survey is to introduce the basic properties of meshfree methods and discuss the results of error estimates of meshfree methods. The organization is as following: Part 2 will discuss some general features of meshfree methods. Comparison with FEM is included. Part 3 will discuss one specific method, Reproducing Kernel Particle Method(RKPM) in detail. Part 4 will discuss the error estimate of RKPM. A short conclusion ends the whole survey.

2. Basic properties of meshfree methods

2.1 Lebesgue space and Sobolev space

For convenience, we list here the definitions of Lebesgue space and Sobolev space. Please refer to [3] for more formal discussion.

DEFINITION 2.1. The Lebesgue space is defined as

$$L^p(\Omega) := \left\{ f : \|f\|_{L^p(\Omega)} < \infty \right\}, 1 \le p \le \infty \tag{2.1}$$

where the L^p norm is defined as:

$$\begin{aligned} \|f\|_{L^p(\Omega)} &:= \left(\int_\Omega \left| f(x)^p \right| \right)^{1/p} \quad \text{for } 1 \le p < \infty \\ \|f\|_{L^\infty(\Omega)} &:= \sup\left\{ \left| f(x) \right| : x \in \Omega \right\} \end{aligned} \tag{2.2}$$

DEFINITION 2.2. The Sobolev space is defined as:

$$W_p^k = \left\{ f \in L^1_{loc}(\Omega) : \|f\|_{W_p^k(\Omega)} < \infty \right\} \tag{2.3}$$

where $L^1_{loc}(\Omega)$ is the set of locally integrable functions defined as:

$$L^1_{loc}(\Omega) = \left\{ f : f \in L^1(K), \forall \text{compact } K \subset \text{interior } \Omega \right\} \tag{2.4}$$

And the associated Sobolev norm is defined by:

$$\|f\|_{W_p^k(\Omega)} := \left(\sum_{|\alpha| \le k} \|D_w^\alpha f\|^p_{L^p_{(\Omega)}} \right)^{1/p} \tag{2.5}$$

Here + is a multi-index. *k* is a non-negative integer.

Particularly, we denote $H^K(\Omega) = W_2^k(\Omega)$.

2.2 Problem setting

Before starting to discuss meshfree methods, we efine the model problem we are solving. Consider a boundary value problem: find $\boldsymbol{u}$ such that:

$$\begin{aligned} \mathbf{L}\boldsymbol{u} &= \boldsymbol{f} && \text{in } \Omega \\ \boldsymbol{u} &= \boldsymbol{u}^0 && \text{on } \Gamma_u \\ \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} &= \boldsymbol{t}^0 && \text{on } \Gamma_t \end{aligned} \tag{2.6}$$

where **L** is a differential operator and $\partial\Omega = \Gamma_t \cup \Gamma_u, \Gamma_t \cap \Gamma_u = \varnothing$.

The variational weak formulation of this problem reads:

$$\text{find } \boldsymbol{u} \in V \quad \text{such that} \quad a(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in V \tag{2.7}$$

where V is a sobolev space. $a(\, , \,)$ is a bilinear form. $(\, , \,)$ is an inner product.

For an example of (2.7), we consider the elasticity problem. In this case, we have K=1, p=2:

$$V = W_2^1(\Omega) = H^1(\Omega)$$

$$a(\boldsymbol{w},\boldsymbol{u}) := \int_\Omega (\nabla \otimes \boldsymbol{w}) : \boldsymbol{C} : (\nabla \otimes \boldsymbol{u}) \, d\Omega \tag{2.8}$$

where $\boldsymbol{C}$ is the elastic moduli tensor.

When we solve (2.7) numerically, we want its finite dimensional approximation. The approximate form of (2.7) reads:

$$\text{find } \boldsymbol{u}^h \in S \quad \text{such that} \quad a(\boldsymbol{u}^h, \boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in S \tag{2.9}$$

where $S \subset V$ is a finite dimensional subspace of V and $\boldsymbol{u}^h$ is the approximation of **u**.

2.3 Discretization

Let/s first recall the discretization method used in FEM: the continuum is discretized to many elements. The discrete approximation is always defined inside an element. For the example shown in fig 2.1, the approximation of the function at point $\boldsymbol{x}$ is defined in the shaded element. Moreover, the approximation will be expressed as a linear combination of the function value at the nodes of the element(*P1, P2, P3, P4* in fig 2.1).

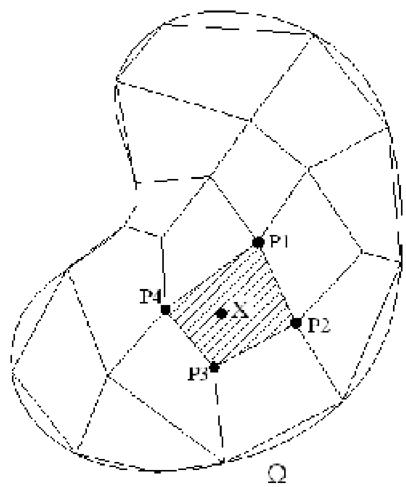

Fig 2.1 FEM discretization

As mentioned before, meshfree methods use a set of particles to discretize the continuum. We then construct the discrete approximation of some unknown functions(e.g. displacement $\boldsymbol{u}$ in elasticity) based on those particles. And for meshfree methods, for each node we define a support. A support is the influential area[3] of that particle and is usually a circle or a rectangle. The supports can be different at nodes. Fig 2.2 shows a particle distribution with uniform circular supports. And to construct approximation at a spatial point $\boldsymbol{x}$, we only use the particles whose support includes $\boldsymbol{x}$. In fig 2.2, the point $\boldsymbol{x}$ is covered by 3 supports(highlighted), so we use 3 particles to construct the approximation for $\boldsymbol{x}$.

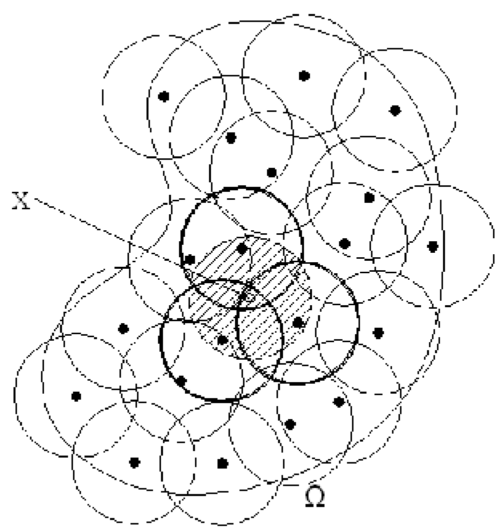

Fig 2.2 Meshfree discretization

Based on this idea, we can define a local domain for ***x***(the shaded circle in fig 2.2), which includes those 3 particles. In a general non-uniform particle distribution, the local domain of ***x*** may be irregular. We can view this local domain as the counterpart of the FEM element. But as we mentioned earlier, this support-based connectivity is far less strict than the mesh-based one and can change with time.

If we write formally what we discussed above, we will have following form of meshfree discretization of an unknown function ***u***:

$$\boldsymbol{u}^h\left(\boldsymbol{x}\right)=\sum_{j=1}^{\mathrm{Ns}}\varphi_j^M\left(\boldsymbol{x}\right)\boldsymbol{u}_j \tag{2.10}$$

where *Ns* is the number of points inside the local domain and $\boldsymbol{u}_j$ is the function value at those points. $\varphi_j\left(\boldsymbol{x}\right)$, usually called [*shape functions*[3] are some known functions.

We have the similar expression for FEM:

$$\boldsymbol{u}^h\left(\boldsymbol{x}\right)=\sum_{j=1}^{\mathrm{Ne}}\varphi_j^F\left(\boldsymbol{x}\right)\boldsymbol{u}_j \tag{2.11}$$

where *Ne* is the number of points of each element.

While the discrete forms look same, there are differences between those two. In FEM, we always use simple polynomials(e.g. Lagragian polynomials) as the shape functions. But as we will see later, the meshfree shape functions will have more complicated expressions. Also, one key property of FEM shape functions is the so-called *Kronecker delta property*:

$$\varphi_i^F\left(\boldsymbol{x}_j\right)=\delta_{ij} \tag{2.12}$$

where δ_{ij} is the *kronecker delta*:

$$\delta_{ij}=\begin{cases}1, & i=j\\ 0, & i\neq j\end{cases} \tag{2.13}$$

This will immediately gives us:

$$\boldsymbol{u}^h\left(\boldsymbol{x}_i\right)=\boldsymbol{u}\left(\boldsymbol{x}_i\right) \tag{2.14}$$

In meshfree case, this property no longer holds due to complicated shape functions. That means:

$$\varphi_i^M\left(\boldsymbol{x}_j\right)\neq\delta_{ij}\Rightarrow u^h\left(\boldsymbol{x}_i\right)\neq u\left(\boldsymbol{x}_i\right) \tag{2.15}$$

So the meshfree approximation is not an interpolation in the strict sense. The result is that in the meshfree formulation, we can not impose Dirichlet type boundary conditions directly. Special procedures are needed(e.g. [4]).

3. Reproducing kernel particle method(RKPM)

In this section we consider a specific meshfree method proposed by Liu et al[5]. This method starts with the trivial identity:

$$\boldsymbol{u}(\boldsymbol{x}) = \int_{\Omega} \delta(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{u}(\boldsymbol{y})d\boldsymbol{y} \tag{3.1}$$

Where $\delta(\boldsymbol{x})$ is the Dirac delta function.

We want to use a finite-valued function to approximate $\delta(\boldsymbol{x})$:

$$\delta(\boldsymbol{x}-\boldsymbol{y}) \approx \varepsilon^{-d}\phi_{\varepsilon}(\boldsymbol{x}-\boldsymbol{y}) = \varepsilon^{-d}\phi\left(\frac{\boldsymbol{x}-\boldsymbol{y}}{\varepsilon}\right) \tag{3.2}$$

where ε is a tiny number and ϕ is a continuous function called window function. ϕ has the following properties:

$$\begin{cases} \phi(\boldsymbol{x}) > 0 & for \quad |\boldsymbol{x}| < 1 \\ \phi(\boldsymbol{x}) = 0 & for \quad |\boldsymbol{x}| \ge 1 \\ \int \phi(\boldsymbol{x}) d\boldsymbol{x} = 1 \end{cases} \tag{3.3}$$

There are many choices for the window function, e.g. cubic spline function(1-D case):

$$\phi(z) = \begin{cases} \frac{2}{3} - 4|z|^2 + 4|z|^3, & 0 \le |z| \le \frac{1}{2} \\ \frac{4}{3} - 4|z| + 4|z|^2 - \frac{4}{3}|z|^3, & \frac{1}{2} \le |z| \le 1 \\ 0, & |z| \ge 1 \end{cases} \tag{3.4}$$

The higher dimensional window functions can be construct by either replace $|\ |$ above with distance in higher dimension or by product:

$$\phi(\boldsymbol{z}) = \prod \phi(z_i) \tag{3.5}$$

Fig. 3.1 is the plot of (3.4) and the corresponding ϕ_{ε} with ε =0.3.

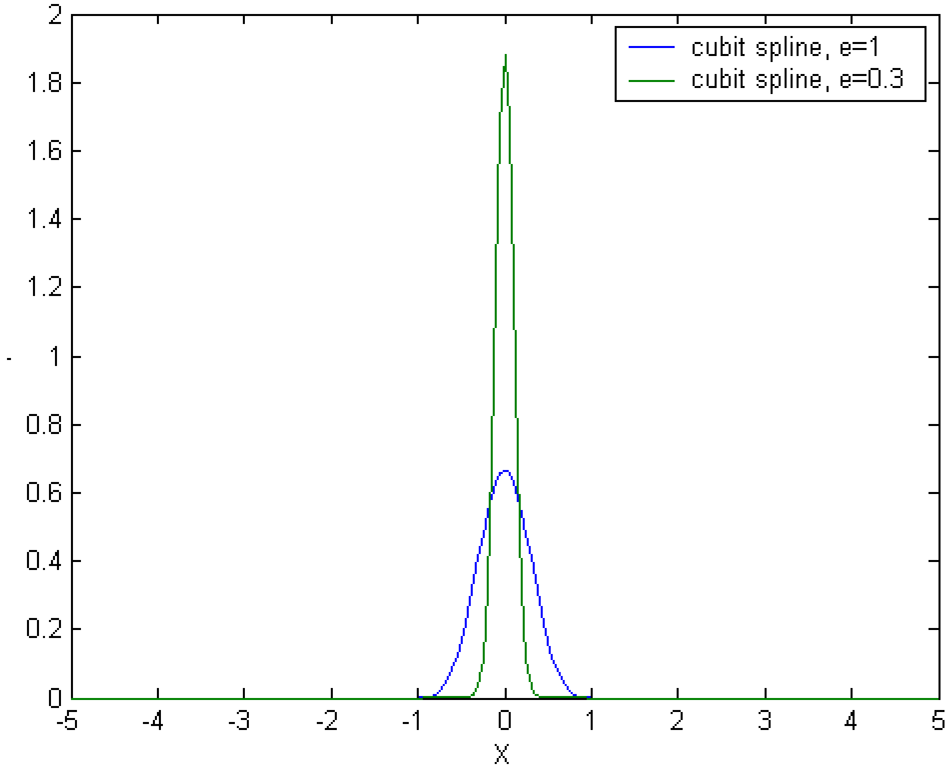

Fig 3.1 The cubit spline window function

Substitute (3.2) back into (3.1), we have:

$$\boldsymbol{u}_{\varepsilon}(\boldsymbol{x}) = \int_{\Omega} \varepsilon^{-d} \phi_{\varepsilon}(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{u}(\boldsymbol{y})d\boldsymbol{y}, \quad \boldsymbol{x} \in \Omega \tag{3.6}$$

$\boldsymbol{u}_{\varepsilon}$ is called the *mollification* of ***u***. Define:

$$\Omega_{\varepsilon} = \left\{ x \in \Omega : dist(x, \partial\Omega) > \varepsilon \right\}, \varepsilon > 0$$

Then we have the following theorem([3], appendix):

<u>THEOREM 3.1</u>. If ***u*** is locally integrable and assume $\boldsymbol{u} \in C^k(\Omega)$, then

1. $\boldsymbol{u}_{\varepsilon} \in C^m \; if \; \phi \in C^m$

2. $D^{\alpha}\boldsymbol{u}_{\varepsilon} \to D^{\alpha}\boldsymbol{u}, \quad \forall |\alpha| \le k$

This theorem justifies the convergence of the approximation (3.6) and shows the continuity of $\boldsymbol{u}_{\varepsilon}$ is determined by the window function.

One basic requirement of the approximation is the consistency condition: if the weak form contains derivatives of order *k*, then the approximate solution must be able to exactly represent solutions with constant *k*-th order derivatives and below. That means our approximation must be able to exactly represents polynomials up to *k*-th order. Let $\boldsymbol{u} = 1, (\boldsymbol{x}-\boldsymbol{y}), (\boldsymbol{x}-\boldsymbol{y})^2 \cdots$ in (3.6), we then have a set of equations:

$$\begin{aligned} &\int_{\Omega} \varepsilon^{-d} \phi_{\varepsilon}(\boldsymbol{x}-\boldsymbol{y}) d\boldsymbol{y} = 1 \\ &\int_{\Omega} \varepsilon^{-d} \phi_{\varepsilon}(\boldsymbol{x}-\boldsymbol{y}) \cdot (\boldsymbol{x}-\boldsymbol{y})^m d\boldsymbol{y} = 0, |m| < k \end{aligned} \tag{3.7}$$

Since ϕ_ε is the approximation of Dirac delta function, (3.7) holds.

If we use a set of particles to discretize the domain Ω, we can write the discrete form of (3.6) as:

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{i=1}^{n} \varepsilon^{-d} \phi_{\varepsilon}(\boldsymbol{x}-\boldsymbol{x}_i) \boldsymbol{u}(\boldsymbol{x}_i) w_i \tag{3.8}$$

where w_i are the integral weights.

Unfortunately, the discrete form of (3.7) does not hold, i.e.

$$\begin{aligned} &\sum_i \varepsilon^{-d} \phi_{\varepsilon}(\boldsymbol{x}-\boldsymbol{x}_i) w_i \neq 1 \\ &\sum_i \varepsilon^{-d} \phi_{\varepsilon}(\boldsymbol{x}-\boldsymbol{x}_i) \cdot (\boldsymbol{x}-\boldsymbol{x}_i)^m w_i \neq 0, |m| < k \end{aligned} \tag{3.9}$$

So we need some adjustments for out approximation. Liu et al[5] added a correction function term to (3.6):

$$\boldsymbol{u}_{\varepsilon}(\boldsymbol{x}) = \int_{\Omega} \varepsilon^{-d} \phi_{\varepsilon}(\boldsymbol{x}-\boldsymbol{y}) C(\boldsymbol{x}, \boldsymbol{x}-\boldsymbol{y}) \boldsymbol{u}(\boldsymbol{y}) d\boldsymbol{y} \tag{3.10}$$

The correction function *C* is given by:

$$C(\boldsymbol{x}, \boldsymbol{x}-\boldsymbol{y}) = \boldsymbol{P}(\boldsymbol{x}-\boldsymbol{y}) \boldsymbol{b}(\boldsymbol{x}) \tag{3.11}$$

where $\boldsymbol{b}(\boldsymbol{x}) = [b_0(\boldsymbol{x}), b_1(\boldsymbol{x}), \cdots b_p(\boldsymbol{x})]^T$ is an unknown vector and vector function $\boldsymbol{P}$ has the form:

$$\boldsymbol{P}(\boldsymbol{x}) = [P_0(\boldsymbol{x}), P_1(\boldsymbol{x}), \cdots P_p(\boldsymbol{x})] \tag{3.12}$$

Where the length *p* equals the dimension of the *k*-th order polynomial space:

$$p = \begin{pmatrix} k+d \\ d \end{pmatrix}, \ d \text{ is the spatial dimension.} \tag{3.13}$$

The $P_i(\boldsymbol{x})$ in (3.12) are chosen to be monomial functions. In 1-D case, we have:

$$\boldsymbol{P}(x) = [1, x, \cdots x^p] \tag{3.14}$$

Then the consistency condition for the modified continuous approximation becomes:

$$\tilde{\boldsymbol{M}}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{P}^T(0) \tag{3.15}$$

where

$$\boldsymbol{P}^T(0) = [1, 0, \cdots 0]^T \tag{3.16}$$

$$\tilde{\boldsymbol{M}}(\boldsymbol{x}) = \int_\Omega \varepsilon^{-d}\phi_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{P}^T(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{y})d\boldsymbol{y} \tag{3.17}$$

<u>PROPOSITION 3.2.</u> $\tilde{\boldsymbol{M}}$ is positive definite.

Proof. Consider the quadratic form:

$$\begin{aligned} Q(\boldsymbol{x},\boldsymbol{a}) = \boldsymbol{a}^T\tilde{\boldsymbol{M}}\boldsymbol{a} &= \int_\Omega \varepsilon^{-d}\phi_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left[\boldsymbol{a}^T\boldsymbol{P}^T(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{a}\right]d\boldsymbol{y} \\ &= \int_\Omega \varepsilon^{-d}\phi_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left[\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{a}\right]^2 d\boldsymbol{y} \geq 0 \end{aligned} \tag{3.18}$$

In the case Q=0, since we have $\phi_\varepsilon > 0$, that means:

$$\sum_{|\alpha|\leq k} a_\alpha \boldsymbol{z}^\alpha = 0, \quad \forall \boldsymbol{z} \tag{3.19}$$

Then a=0 follows. So Q>0, $\tilde{\boldsymbol{M}}$ is positive definite. It is called *moment matrix.*

So (3.15) has a unique solution ***b(x)***.

Now we consider the discrete form of (3.10):

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{i=1}^n r_i^{-d}\phi_{ri}(\boldsymbol{x}-\boldsymbol{x}_i)C(\boldsymbol{x},\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{u}(\boldsymbol{x}_i)w_i \tag{3.19}$$

Here n is the number of particles in the domain. We replace ε with r_i by allowing different support sizes at different particles. Substitute (3.11) into (3.19):

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{i=1}^n r_i^{-d}\phi_{ri}(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{P}(\boldsymbol{x},\boldsymbol{x}-\boldsymbol{x}_i)w_i\boldsymbol{b}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x}_i) \tag{3.20}$$

by absorbing r_i^{-d} and w_i into unknown vector $\boldsymbol{b}$, we have:

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{i=1}^n \phi_{ri}(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{P}(\boldsymbol{x},\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{b}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x}_i) \tag{3.21}$$

The consistency condition now becomes:

$$\boldsymbol{M}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{P}^T(0) \tag{3.22}$$

where

$$\boldsymbol{M}(\boldsymbol{x}) = \sum_{i=1}^{n} \phi_{ri}(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{P}^T(\boldsymbol{x},\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{P}(\boldsymbol{x},\boldsymbol{x}-\boldsymbol{x}_i) \tag{3.23}$$

$\boldsymbol{M}$ is the *discrete moment matrix*. Recall the expression of $\boldsymbol{P}$ (3.12), we can see $\boldsymbol{M}$ is the sum of some rank-one matrices. So for $\boldsymbol{M}$ to be nonsingular, we need at least p nonzero terms in the summation of (3.23), where p is defined by (3.13). In term of particle distribution, this means we need at least p particles whose ϕ does not equal zero at $\boldsymbol{x}$. In other words, any point $\boldsymbol{x}$ must be in the support of at least p particles. To explore the sufficient condition, we look at the quadratic form of $\boldsymbol{M}$:

$$\begin{aligned} Q(\boldsymbol{x},\boldsymbol{a}) = \boldsymbol{a}^T\boldsymbol{M}\boldsymbol{a} &= \sum_{i=1}^{n} \phi_{ri}(\boldsymbol{x}-\boldsymbol{x}_i)\left[\boldsymbol{a}^T\boldsymbol{P}^T(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{a}\right] \\ &= \sum_{i=1}^{n} \phi_{ri}(\boldsymbol{x}-\boldsymbol{x}_i)\left[\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_i)\cdot\boldsymbol{a}\right]^2 \geq 0 \end{aligned} \tag{3.24}$$

and:

$$Q(\boldsymbol{x},\boldsymbol{a}) = 0 \;\; \Rightarrow \;\; \sum_{i=1}^{n}\left[\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_i)\cdot\boldsymbol{a}\right] = 0 \tag{3.25}$$

For (3.25) to have only zero solution, we have:

$$T = \begin{bmatrix} \boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_1) \\ \boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_p) \end{bmatrix} \text{ is nonsingular.} \tag{3.26}$$

(3.26) is the sufficient condition for (3.22) to have unique solution.

Now re-write (3.21) in the form of (2.9):

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{i=1}^{n} \varphi_i(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x}_i) \tag{3.27}$$

where the shape functions has the form:

$$\varphi_i(\boldsymbol{x}) = \phi_{ri}(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{P}(\boldsymbol{x},\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{b}(\boldsymbol{x}) \tag{3.28}$$

In practical programming, at each spatial point $\boldsymbol{x}$ we first solve (3.22) to get the $\boldsymbol{b}$ vector. Then we plug $\boldsymbol{b}$ into (3.28) to get the shape function value at point $\boldsymbol{x}$.

It can be easily shown that:

$$\varphi_i\left(\boldsymbol{x}\right)\in C^m \text{ if } \phi\left(\boldsymbol{x}\right)\in C^m \tag{3.29}$$

So it is easy to construct arbitrarily smooth shape functions. This will make it easy when solving higher order PDEs.

It can also be shown[6] that $\{\varphi_i\}_{i=1}^n$ are independent. So our approximation (3.27) belongs to the space with $\{\varphi_i\}_{i=1}^n$ as basis.

We end this section with several shape function plots. Fig 3.2 shows a set of 1-D RKPM shape functions. Fig 3.3 and fig 3.4 show a 2-D shape function and its derivative along *x*-direction.

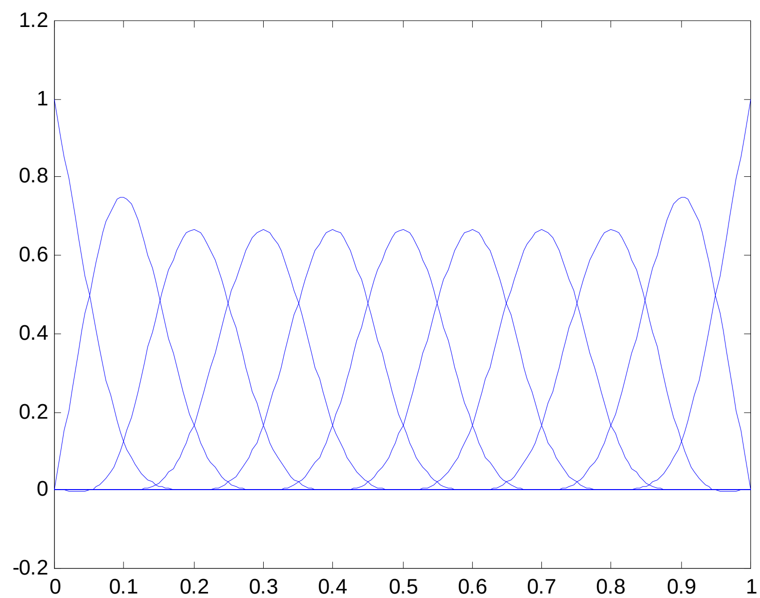

Fig. 3.2 1D RKP shape functions

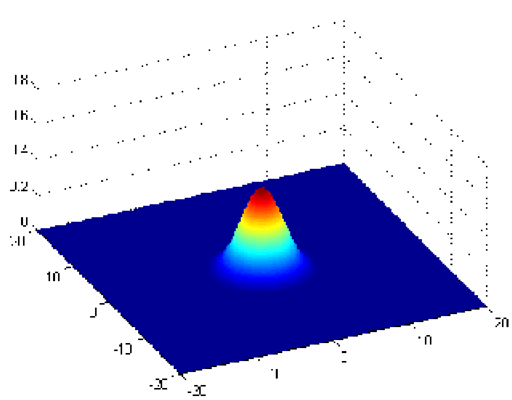

Fig. 3.2 2D RKP shape function

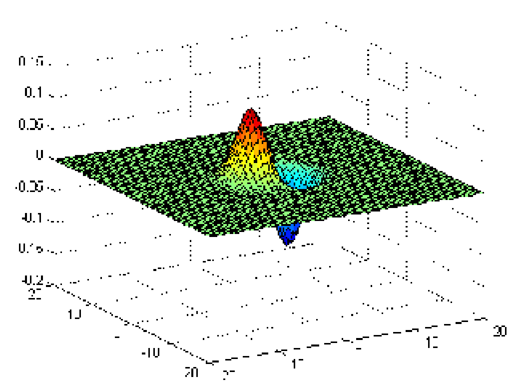

Fig. 3.4. *x*-derivative of 2D shape function

4. Error estimates of RKPM

We first try to get the interpolation error estimate, i.e. try to get a bound for $\left\| u(x) - u^h(x) \right\|$.

Let the support sizes be *quasi-uniform*([7], 4.4), i.e. there exist constants c1, c2 and r such that:

$$c_1 \le \frac{r_i}{r} \le c_2 \quad \forall i \tag{4.1}$$

Liu et al[8] showed the derivative of ***b*** vector is bounded by:

$$D^\alpha \boldsymbol{b}(\boldsymbol{x}) \le c \cdot r^{-|\alpha|} \tag{4.2}$$

Based on this result, we can get the bound of the shape function[6]:

$$\max_{1\le i\le l} \max_{|\beta|=l} \left\| D^\beta \varphi_i \right\|_\infty \le \frac{c}{r^l}, \quad l = 0 \cdots k \tag{4.3}$$

Denote $B_i := B_{ri}(\boldsymbol{x}_i)$ as the ball centered at $\boldsymbol{x}_i$. Now we consider the averaged *k*-th order Taylor polynomial of ***u***([7], 4.1) over *Bi*:

$$Q_j^k \boldsymbol{u}(\boldsymbol{x}) = \int_{B_i} T_y^k \boldsymbol{u}(\boldsymbol{x}) \phi(\boldsymbol{y}) d\boldsymbol{y} \tag{4.4}$$

where ϕ is window function on B_i. $T_y^k \boldsymbol{u}(\boldsymbol{x})$ is the Taylor polynomial at ***y***:

$$T_y^p \boldsymbol{u}(\boldsymbol{x}) = \sum_{|\alpha|<p} \frac{1}{\alpha!} D^\alpha \boldsymbol{u}(\boldsymbol{y}) (\boldsymbol{x} - \boldsymbol{y})^\alpha \tag{4.5}$$

Obviously $Q_j^k \boldsymbol{u}(\boldsymbol{x})$ is an approximation of ***u(x)***. We can define the remainder term:

$$R_j^k \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x}) - Q_j^k \boldsymbol{u}(\boldsymbol{x}) \tag{4.6}$$

We have the following lemma([7], 4.3):

LEMMA 4.1.(Bramble-Hilbert).

Let *B* be a ball in Ω such that Ω is star-shaped with respect to *B* and let $Q^m u$ be the Taylor polynomial of degree *m* of ***u*** averaged over *B* where $\boldsymbol{u} \in W_p^m(\Omega)$ and $p \ge 1$. Then

$$\left\| R^k \boldsymbol{u} \right\|_{W_p^k(\Omega)} \le c \cdot d^{k-p} \left| \boldsymbol{u} \right|_{W_p^m(\Omega)} \; p = 0,1,\cdots k \tag{4.7}$$

where d is the diameter of the domain.

Use this lemma, we have the results for each support *Bi*:

$$\left\| R_j^k \boldsymbol{u}(\boldsymbol{x}) \right\|_{W_l^p(B_j)} \le c \cdot r_j^{k-p} \left| \boldsymbol{u} \right|_{W_p^k(B_j)}, \; p = 0,1,\cdots k \tag{4.8}$$

The difference between ***u(x)*** and its approximation:

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}^h(\boldsymbol{x}) &= Q_j^k \boldsymbol{u}(\boldsymbol{x}) + R_j^k \boldsymbol{u}(\boldsymbol{x}) - \sum_{i=1}^{n} \left(Q_j^k \boldsymbol{u}(\boldsymbol{x}_i) + R_j^k \boldsymbol{u}(\boldsymbol{x}_i) \right) \varphi_i(\boldsymbol{x}) \\ &= Q_j^k \boldsymbol{u}(\boldsymbol{x}) - \sum_{i=1}^{n} Q_j^k \boldsymbol{u}(\boldsymbol{x}_i) \varphi_i(\boldsymbol{x}) + R_j^k \boldsymbol{u}(\boldsymbol{x}) - \sum_{i=1}^{n} R_j^k \boldsymbol{u}(\boldsymbol{x}_i) \varphi_i(\boldsymbol{x}) \end{aligned} \tag{4.9}$$

Note that k is the highest order of derivative appeared in the weak form. We already showed our RKP approximation can exactly reproduce polynomials up to k-th order. And $Q_j^k \boldsymbol{u}(\boldsymbol{x})$ is a k-th order polynomial. So we have:

$$\sum_{i=1}^{n} Q_j^k \boldsymbol{u}(\boldsymbol{x}_i) \varphi_i(\boldsymbol{x}) = Q_j^k \boldsymbol{u}(\boldsymbol{x}) \tag{4.10}$$

Then (4.9) reduces to:

$$\boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}^h(\boldsymbol{x}) = R_j^k \boldsymbol{u}(\boldsymbol{x}) - \sum_{i=1}^{n} R_j^k \boldsymbol{u}(\boldsymbol{x}_i) \varphi_i(\boldsymbol{x}) \tag{4.11}$$

Take norm of both sides of (4.11) and use Cauchy-Schwarz inequality, we have:

$$\left\| \boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}^h(\boldsymbol{x}) \right\|_{W_l^q(B_j)} \le \left\| R_j^k \boldsymbol{u}(\boldsymbol{x}) \right\|_{W_l^q(B_j)} + \left\| R_j^k \boldsymbol{u}(\boldsymbol{x}_i) \right\|_{W_l^q(B_j)} \sum_{i=1}^{n} \left\| \varphi_i(\boldsymbol{x}) \right\|_{W_l^q(B_j)} \tag{4.12}$$

In (4.12), $1 \le q \le \infty$. Now plug (4.3) and (4.8) into the above inequality, we have:

$$\left\| \boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}^h(\boldsymbol{x}) \right\|_{W_l^q(B_j)} \le c \cdot r_j^{k-l} \left| \boldsymbol{u} \right|_{W_k^q(B_j)}, 0 \le l \le k, 1 \le j \le n \tag{4.13}$$

Expand this result to the whole domain, we have the interpolation error estimates:

$$\left\| \boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}^h(\boldsymbol{x}) \right\|_{W_l^q(\Omega)} \le c \cdot r^{k-l} \left| \boldsymbol{u} \right|_{W_k^q(\Omega)}, 0 \le l \le k, q \in [1, \infty] \tag{4.14}$$

Now we can derive the error bound for approximate solutions obtained by RKPM. Now we revisit the BVP defined in the beginning of section 2. Since we did not discuss how to treat Dirichlet boundary condition, we assume for now our problem does not have Dirichlet b.c., i.e. $\Gamma_u = \varnothing$ in (2.6). The weak formulation using RKPM reads:

$$\text{find } \boldsymbol{u}^R \in V_R \quad \text{such that} \quad a\left(\boldsymbol{u}^R, \boldsymbol{v}\right) = \left(\boldsymbol{f}, \boldsymbol{v}\right) \quad \forall \boldsymbol{v} \in V_R \tag{4.15}$$

where $V_R = span\left\{\varphi_i\left(\boldsymbol{x}\right) : 1 \le i \le n\right\}$ is the space generated by RKP shape functions.

By Lax-Milgram theorem([7], 2.7), the problem has a unique solution. Moreover, by C&aD theorem([7], 2.8), we have the following inequality:

$$\left\|\boldsymbol{u} - \boldsymbol{u}^R\right\|_V \le c \cdot \min_{v \in V_R} \left\|\boldsymbol{u} - \boldsymbol{v}\right\|_V \tag{4.16}$$

Since the RKP approximation $\boldsymbol{u}^h$ belongs to V_R, replace $\boldsymbol{v}$ in (4.16) by $\boldsymbol{u}^h$ will not affect the inequality. Then we have:

$$\left\|\boldsymbol{u} - \boldsymbol{u}^R\right\|_V \le c \cdot \left\|\boldsymbol{u} - \boldsymbol{u}^h\right\|_V \tag{3.46}$$

So the error estimates of the approximation is reduced to the error estimate of interpolation. Using the interpolation error bound we just got, we have the error estimate for the RKPM approximate solution[6]:

$$\left\|\boldsymbol{u} - \boldsymbol{u}^R\right\|_{H^n(\Omega)} \le c \cdot r^{k-n} \left|\boldsymbol{u}\right|_{H^k(\Omega)} \tag{3.47}$$

In (3.47), *n* is the level of continuity of ϕ , $n \le k$ and $W_l^k\left(\Omega\right)$ is replaced by $H^k\left(\Omega\right)$.

Han and Meng[6] showed this bound is still valid when considering Dirichlet boundary conditions. So the error of RKPM is in the order of $O\left(r^{p-n}\right)$. This bound has been verified by various numerical results[6],[8]. This error bound shows the approximate result will converge to the real solution as we decrease the size of the support(increase the number of particles in the meantime).

5.Conclusion

In this paper we discuss the mechanism of meshfree methods, in particular RKPM. By discussing the formulation of RKPM and some results of its error estimate, we have shown the validity of RKPM as a numerical method solving for PDEs. As for me, by studying the relative literature, I learned the basic methodology of error estimates. This will help me to justify the ideas when developing new numerical scheme.

References

[1]. T.Belytschko, et al. Meshless methods: An overview and recent developments. *Comput. Methods Appl. Mech*. Engrg. 139(1996): 3-47.

[2]. S.Li and W.K.Liu. Meshfree and particle methods and their applications. *Appl. Mech. Rev.* 55(2002): 1-34.

[3]. L.C.Evans. Partial Differential Equations. Berkeley mathematics lecture notes Vol. 3A. 1993

[4]. J.S Chen, et al. A reproducing kernel method with nodal interpolation property. *Int. J. Numer. Meth. Engrg.* 56(2003): 935-960

[5]. W.K. Liu, et al. Reproducing kernel particle methods, *Int. J. Numer. Meth. Engrg.* 38(1995): 1655-1679.

[6]. W.Han and X.Meng. Error analysis of the reproducing kernel particle method. *Comput. Methods Appl. Mech. Engrg.* 190(2001): 6157-6181

[7]. S.C.Brenner and L.R.Scott. The Mathematical Theory of Finite Element Methods. *Springer, New York.* 1994

[8]. Moving least-square reproducing kernel methods (I): Methodology and convergence. *Comput. Methods. Appl. Mech. Engrg.* 143(1997): 113-154