

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

A Study on Accuracy, Convergence and Stability of Galerkin and Collocation Meshfree Methods

Permalink

<https://escholarship.org/uc/item/2zn4b91b>

Author

Liu, Chang

Publication Date

2015

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

A Study on Accuracy, Convergence and Stability of Galerkin and Collocation
Meshfree Methods

A Thesis submitted in partial satisfaction of the requirements for the degree Master of
Science

in

Structural Engineering

by

Chang Liu

Committee in charge:

Professor J. S. Chen, Chair
Professor Yuri Bazilevs
Professor David J. Benson

2015

The Thesis of Chang Liu is approved and it is acceptable in quality and form for publication on microfilm and electronically

Chair

University of California, San Diego

2015

TABLE OF CONTENTS

Signature Page	iii
TABLE OF CONTENTS.....	iv
LIST OF FIGURES	vii
ABSTRACT OF THE THESIS	xiii
1 Introduction.....	1
1.1. Background.....	1
1.2. Objective	2
1.3. Outline.....	3
2 Literature Review.....	4
2.1. Early Development of Meshfree Methods.....	4
2.2. Galerkin Meshfree Methods	5
2.3. Reproducing Kernel Collocation Method (RKCM)	7
2.4. Radial Basis Collocation Method (RBCM)	8
3 Review of Reproducing Kernel Particle Method (RKPM).....	10
3.1. Reproducing Kernel Approximation.....	10
3.2. Galerkin Formulation.....	13
3.3. Reproducing Kernel Particle Method and Techniques for Imposing Essential Boundary Conditions.....	14
3.3.1. Lagrange Multiplier Method.....	15
3.3.2. Penalty Method	16
3.3.3. Transformation Method	17
3.4. Nodal Integration Techniques for Meshfree Methods.....	21
3.4.1. Direct Nodal Integration (DNI)	21

3.4.2.	Stabilized Conforming Nodal Integration (SCNI).....	23
3.4.3.	Variationally Consistent Integration (VCI)	27
4	Review of Reproducing Kernel Collocation Method (RKCM) and Radial Basis Collocation Method (RBCM).....	30
4.1.	Reproducing Kernel Collocation Method (RKCM)	30
4.2.	Radial Basis Collocation Method (RBCM)	32
4.2.1.	Introduction to RBF	32
4.2.2.	Discretized System of Radial Basis Point Collocation.....	34
5	Comparison of Accuracy and Convergence Among RKCM, RBCM and RKPM.....	35
5.1.	Patch Tests of RKCM in 1-Dimension and 2-Dimensions	35
5.2.	Comparison of RKCM and RBCM in One Dimensional Poisson Problem.....	38
5.2.1.	Effect of Number of Collocation Points on Convergence	38
5.2.2.	Error distribution in 1D model problem	40
5.3.	RKCM and RBCM Comparison in Two Dimensional Poisson Problem.....	44
5.3.1.	Convergence Comparison in 2-D Poisson Problem	44
5.4.	Comparison between Strong Form Collocation Methods (RKCM and RBCM) with Galerkin meshfree Method (RKPM)	47
5.4.1.	Special Collocation Points Arrangements for Enhanced Computational Efficiency.....	52
5.4.2.	Comparison of Errors in RKCM and RBCM using uniform arrangements of collocation points	58
6	Efficiency and Stability of RKCM, RBCM and RKPM.....	64

6.1.	Efficiency of RKCM, RBCM and RKPM	64
6.2.	Conditioning of Linear Systems in RKCM and RBCM	67
6.3.	Solution accuracy using non-uniform discretization	69
7	Conclusion	77
7.1.	Conclusions.....	77
7.2.	Recommendations for future work	80
	Reference	81

LIST OF FIGURES

Figure 5.1	Errors for quadratic patch test with 1-D RKCM	36
Figure 5.2	Errors for cubic patch test with 1-D RKCM	37
Figure 5.3	(a) non-uniform source point and collocation point distribution, (b) Errors for quadratic patch test with 2-D RKCM.....	38
Figure 5.4	Convergence plot of error in L_2 norm of RKCM and RBCM in 1D, varying N_c , legend showing asymptotic rate of convergence	39
Figure 5.5	Convergence plot of error in H^1 norm of RKCM and RBCM in 1D, varying N_c , legend showing asymptotic rate of convergence	39
Figure 5.6	RKCM 1D, error of u , 6 nodes, $N_c = N_s$	42
Figure 5.7	RKCM 1D, error of du/dx , $N_c = N_s$	42
Figure 5.8	RKCM 1D, error of u , 6 nodes $N_c = 2N_s$	42
Figure 5.9	RKCM 1D, error of du/dx , $N_c = 2N_s$	42
Figure 5.10	RKCM 1D, error of u , 6 nodes $N_c = 4N_s$	42
Figure 5.11	RKCM 1D, error of du/dx , $N_c = 4N_s$	42
Figure 5.12	RBCM 1D, error of u , 6 nodes $N_c = N_s$	43
Figure 5.13	RBCM 1D, error of du/dx , $N_c = N_s$	43
Figure 5.14	RBCM 1D, error of u , 6 nodes $N_c = 2N_s$	43
Figure 5.15	RBCM 1D, error of du/dx , $N_c = 2N_s$	43

Figure 5.16	RBCM 1D, error of u , 6 nodes $N_c = 4N_s$	43
Figure 5.17	RBCM 1D, error of du/dx , $N_c = 4N_s$	43
Figure 5.18	Convergence plot of error in L_2 norm of RBCM with $c=1.6$ for 2D Poisson's problem with different N_c to N_s ratios	45
Figure 5.19	Convergence plot of error in H^1 norm of RBCM with $c=1.6$ for 2D Poisson's problem with different N_c to N_s ratios	46
Figure 5.20	Convergence plot of error in L_2 norm of RKCM with quadratic basis for 2D Poisson's problem with different N_c to N_s ratios	46
Figure 5.21	Convergence plot of error in H^1 norm of RKCM with quadratic basis for 2D Poisson's problem with different N_c to N_s ratios	47
Figure 5.22	Convergence plot of error in L_2 norm for 2D Poisson's problem, comparing RKCM and RBCM ($N_c = N_s$) and RKPM DNI, SCNI and VCI	49
Figure 5.23	Convergence plot of error in H^1 norm of 2D Poisson's problem, comparing RKCM and RBCM ($N_c = N_s$) and RKPM with DNI, SCNI and VCI	49
Figure 5.24	Convergence plot of error in L_2 norm for 2D Poisson's problem, comparing RKCM and RBCM with $N_c = N_s$ and $N_c = 2N_s$ and RKPM with DNI, SCNI and VCI	50
Figure 5.25	Convergence plot of error in H^1 norm for 2D Poisson's problem, comparing RKCM and RBCM with $N_c = N_s$ and $N_c = 2N_s$ and RKPM with DNI, SCNI and VCI	50
Figure 5.26	Convergence plot of error in L_2 norm for 2D Poisson's problem, comparing RKCM, RBCM with $N_c = 6N_s$ and RKPM with DNI, SCNI and VCI	51

Figure 5.27	Convergence plot of error in H^1 norm for 2D Poisson's problem, comparing RKCM, RBCM with $N_c = 6N_s$ and RKPM with DNI , SCNI and VCI...	52
Figure 5.28	Special ways of allocating collocation points based on $N_c = N_s$ system, 6x6 source points	54
Figure 5.29	Convergence plot of error in L_2 norm for 2D Poisson's problem, comparing RKCM with special techniques of adding collocation points, compare with RKPM using DNI and SCNI	55
Figure 5.30	Convergence plot of error in H^1 norm for 2D Poisson's problem, comparing RKCM with special techniques of adding collocation points, compare with RKPM with linear basis and iterated using DNI and SCNI.....	55
Figure 5.31	RKCM 2D. Error of u and its gradients for 2D Poisson's problem, 6x6 source points, uniform $N_c = N_s$	56
Figure 5.32	RKCM 2D. Error of u and its gradients for 2D Poisson's problem. 6x6 source points, $N_c = N_s$ plus four corner collocation points.....	56
Figure 5.33	RKCM 2D. Error of u and its gradients for 2D Poisson's problem. 6x6 source points, $N_c = N_s$ plus a ring of collocation points	56
Figure 5.34	RKCM 2D. Error of u and its gradients for 2D Poisson's problem, 18x18 source points, regular $N_c = N_s$	57
Figure 5.35	RKCM 2D. Error of u and its gradients for 2D Poisson's problem. 18x18 source points, $N_c = N_s$ plus four corner collocation points.....	57
Figure 5.36	RKCM 2D. Error of u and its gradients for 2D Poisson's problem. 18x18 source points, $N_c = N_s$ plus a ring of collocation points	57

Figure 5.37	RKCM, error of u and its gradients in domain, 6×6 , $N_c = N_s$	58
Figure 5.38	RBCM, error of u and its gradients in domain, 6×6 , $N_c = N_s$	58
Figure 5.39	RKCM, errors of u and its gradients in the domain, 101×101 source points, $N_c = N_s$	59
Figure 5.40	RBCM, errors of u and its gradients in the domain, 101×101 source points, $N_c = N_s$	59
Figure 5.41	Errors in u , RKCM, quadratic basis, 16×16 source points, $N_c = N_s$	60
Figure 5.42	Errors in du/dx , RBCM, $c=1.6$, 16×16 source points, $N_c = 4N_s$	61
Figure 5.43	Errors in du/dx , RKCM, $n=2$, 16×16 source points, $N_c = N_s$	61
Figure 5.44	Errors in du/dx , RBCM, 16×16 source points, $N_c = N_s$	62
Figure 5.45	Errors in du/dy , RKCM, $n=2$, 16×16 source points, $N_c = N_s$	62
Figure 5.46	Errors in du/dy with RBCM, 16×16 source points, $N_c = 4N_s$	63
Figure 6.1	L_2 error norms and CPU time of RKCM and RBCM under unifrom disrectization	65
Figure 6.2	L_2 error norms and CPU time of collocation methods and RKPM (with linear basis) under unifrom disrectization	65
Figure 6.3	L_2 error norms and CPU time of collocation methods and RKPM (with quadratic basis) under unifrom disrectization	66
Figure 6.4	Condition numbers of linear systems of RBCM with different shape parameters in RBFs	67
Figure 6.6	Effect of shape parameter on error in L_2 norm in RBCM	69

Figure 6.7	Effect of shape parameter on error in H^1 norm in RBCM.....	69
Figure 6.8	Uniform refinement of irregular nodes discretization, collocation points at source points	70
Figure 6.9	Convergence of error in L_2 and H^1 norms RKCM (n=2, non-uniform $N_c = N_s$), RBCM (c=1.6, non-uniform, $N_c = N_s$) and RKPM (non-uniform, quadratic basis, 1-pt Gauss quadrature).....	70
Figure 6.10	Uniform refinement of irregular nodes discretization, uniform collocation points $N_c = 2N_s$ (Cases with 6x6 and 11x11 source points shown).....	71
Figure 6.11	Convergence plot of error in L_2 norm under non-uniform discretization of RKCM (n=2, $N_c = 2N_s$), RBCM (c=1.6, $N_c = 2N_s$) and RKPM (n=2, VCI, DNI, SCNI, GI), comparing results under uniform discretization.....	73
Figure 6.12	Convergence plot of error in H^1 norm under non-uniform discretization of RKCM (n=2, $N_c = 2N_s$), RBCM (c=1.6, $N_c = 2N_s$) and RKPM (n=2, VCI, DNI, SCNI, GI), comparing results under uniform discretization.....	73
Figure 6.13	RKCM, error in domain, 121 non-uniform source points with uniform collocation points $N_c = 2N_s$	75
Figure 6.14	RKCM, error in domain, 121 uniform source points with uniform collocation points $N_c = 2N_s$	75
Figure 6.15	RBCM, error in domain, 121 non-uniform source points with uniform collocation points $N_c = 2N_s$, $c = 1.6$	76
Figure 6.16	RKPM with linear bases and SCNI discretized by 121 non-uniform nodes, error in domain	76

Figure 6.17 RKPM with linear bases and DNI discretized by 121 non-uniform nodes, error in domain.....	76
--	----

ABSTRACT OF THE THESIS

Study on Accuracy, Convergence and Stability of Galerkin and Collocation Meshfree
Methods

by

Chang Liu

Master of Science in Structural Engineering

University of California, San Diego, 2015

Professor J. S. Chen, Chair

In the past two decades substantial advancements of meshfree methods have been made for various engineering and scientific applications. Meshfree methods can be categorized in two branches, methods based on a Galerkin weak form and on a strong form with collocation. For example, the element-free Galerkin (EFG) method and the reproducing kernel particle method (RKPM) are formulated under the Galerkin framework, and the Radial Basis collocation method (RBCM) and reproducing kernel

collocation method (RKCM) are formulated directly on strong form with collocation. While much success has been made in meshfree Galerkin methods and collocation methods, the in-depth comparison of numerical performance of these two classes of methods is scarce. This dissertation investigates the accuracy, convergence, stability, and effectiveness of Galerkin and strong form collocation meshfree methods, with focus on RKPM, RBCM and RKCM. The numerical performance comparison between these methods as well as their advantages and disadvantages are discussed.

In this work, the convergence and accuracy of RKCM, RBCM and RKPM are first studied with varying approximation parameters. Different arrangements of collocation points are used for RKCM and RBCM to investigate how the convergence behaviors are affected by the number and location of collocation points. Meanwhile, RKPM with various quadrature rules for domain integration techniques has also been studied, and the results show that it requires well-formulated domain integration techniques to reach optimal convergence. On the other hand, strong form collocation methods such as RBCM and RKCM can achieve similar accuracy as that in RKPM without encountering complexity involved in advanced quadrature rules. Next, the performance of collocation methods under non-uniform discretization has also been investigated. The results indicate that both RKCM and RBCM solutions are relatively insensitive to the non-uniformity in discretization compared to RKPM with direct nodal integration (DNI) or Gauss integration under the same domain discretization. RKPM, on the other hand, requires the employment of advanced quadrature rules such as the Stabilized conforming nodal integration (SCNI) and the Variational Consistent Integration (VCI) to overcome solution sensitivity to discretization non-uniformity. The comparison of these three methods shows that while RBCM generates best accuracy in coarse discretization, the method is suffered from ill-

conditioning in the model refinement. RKCM and RBCM, on the other hand, are more stable under model refinement but offer less accurate results compared to RBCM in coarse discretization. RKCM appears to be an attractive alternative of RKPM due to its ability in gaining stability and good convergence without the need of well-formulated quadrature rules.

1 Introduction

1.1. Background

Ever since the middle of the 20th century, finite element methods (FEM) have been the most popular numerical methods widely used in almost all fields in science and engineering. However, due to the regularity requirement in construction of approximation based on the mesh, FEM encounters difficulty with mesh distortion and mesh entanglement when dealing large deformation or fracture problems, and adaptive mesh refinement for desired accuracy is also cumbersome, among others. In the past two decades, meshfree methods have emerged as a new class of computational method, which have been used as an enhancement of FEM for better accuracy or as an alternative for solving challenging problems that otherwise cannot be accomplished by FEM.

Unlike FEM, meshfree methods do not rely on a mesh connectivity to construct approximation functions for solving partial differential equations (PDE's), and thus the mesh related numerical difficulties in FEM could be relaxed or resolved. Besides, meshfree methods are effective in domain discretization for arbitrarily distributed scattered points without the need of a mesh generator, constructing approximation for arbitrary and evolving discontinuities in fracture mechanics, developing h-type model refinement, achieving arbitrary order of smoothness (continuity) for higher order PDE's, among others. Reproducing kernel (RK) approximation is one of the earliest methods that have been introduced to construct the meshfree discrete equation under Galerkin weak framework, termed the Reproducing Kernel Particle Method (RKPM). Thereafter, RK approximation was

introduced to directly discretize the strong form using point collocation methods, named as reproducing kernel collocation method (RKCM). RKCM offers several advantages over Galerkin meshfree methods, most noted in its ability to avoid the complexity of domain integration and imposition of essential boundary conditions. The origin of RKCM is the radial basis collocation method (RBCM), which uses radial basis functions as the approximation functions.

In this work, the accuracy, convergence and stability properties of RKPM, RKCM, and RBCM are studied for systematic comparison of the numerical performance in meshfree methods based on strong and weak formulations.

1.2. Objective

The objective of this thesis is to perform a series of numerical studies on the performance of RKPM, RKCM and RBCM, and use the numerical results to make comparison between Galerkin and collocation meshfree methods. One major task in this investigation is to study how the number and location of collocation points affect the performance of RKCM and RBCM, and then to compare the numerical results with RKPM with different nodal integration techniques and quadrature rules. In addition, accuracy of solution of RKCM and RBCM under non-uniform discretization is numerically tested, and the comparison of test results of these strong form collocation methods with Galerkin based RKPM under non-uniform discretization is also made. Finally, the efficiency of these meshfree methods by comparing the CPU time versus error norms under model refinement is also investigated.

1.3. Outline

This dissertation is organized as follows. Chapter 2 provides a literature review of several meshfree methods to be studied, namely, RKPM along with different techniques of domain integration and boundary condition imposition, RBCM and its advancement, and RKCM with proper weighting on the domain and boundary discrete equations. Chapter 3 provides the introduction of RKPM, including its numerical formulations, boundary condition imposition techniques, and the nodal integration techniques such as DNI and SCNI. In Chapter 4, basic equations of two strong form point collocation meshfree methods, RBCM and RKCM, are provided. Numerical analysis of accuracy, convergence, and stability of RKCM, RBCM, and comparison with RKPM using different domain integration techniques are presented in Chapter 5. Further study of efficiency and stability of the three meshfree methods under non-uniform discretization is presented in Chapter 6. The summary and conclusions of this study are given in Chapter 7.

2 Literature Review

In this chapter, the history of meshfree methods development is presented, with focus on three Galerkin and collocation meshfree methods, namely, RKPM, RKCM, and RBCM that are to be studied in this work.

2.1. Early Development of Meshfree Methods

The embryonic meshfree methods can be traced back to the smoothed particle hydrodynamics (SPH) [54], which approximates the solution of conservation laws based on a set of scattered points without a mesh. SPH was first invented to solve problems in astrophysics field, consequently in fluid mechanics, and was later employed in solid mechanics by Libersky et al. [50]. SPH encountered various numerical difficulties: rank instability, tensile instability and lack of partition of unity to guarantee convergence. Several modifications were made to improve the performance of SPH. Johnson and Beissel [42] presented a normalized smoothing function algorithm so that the accuracy for non-uniform strain rates and for free boundaries can be significantly improved. Monaghan [57] showed how tensile instability could be removed by using the so-called artificial stress.

In 1982, Lancaster and Salkauskas [48] introduced the moving least-squares (MLS) method as a new technique for surface fitting. MLS was later introduced to the field of computational mechanics. Nayroles *et al* [58] introduced MLS in the diffuse element method (DEM) for solving PDEs, and was later enhanced to yield the Element Free Galerkin (EFG) method by Belytschko *et al.* [8]. Babuska and Melenk [3] presented the Partition of Unity Method (PUM) as a general approach for the construction of conforming Ansatz spaces with local properties in global conforming approximation.

2.2. Galerkin Meshfree Methods

The Element-Free Galerkin (EFG) method was proposed by the Belytschko *et al.* in [8] as the enhancement of DEM, and is one of the earliest meshfree methods constructed based on Galerkin weak form. In EFG, moving least squares interpolants are used to construct the trial and test functions for the Galerkin approximation. The EFG method relaxed the burden of mesh connectivity, while allowing arbitrary order of continuity in its approximation. The application of EFG is spread to various fields. Belytschko *et al.* [7] utilized EFG in solving fracture and crack growth problems. Lu *et al.* [53] extended EFG to dynamic problems including wave propagation and dynamic fracture. Krysl and Belytschko [46] proposed a technique for modeling for arbitrary three-dimensional dynamically propagating cracks in elastic bodies by EFG with explicit time integration.

In a separate development, Liu and his coworkers [51] proposed the reproducing kernel particle method (RKPM) by adding a correction function to the kernel estimate of SPH to enforce the polynomial reproducibility for completeness. RKPM has been applied to a wide range of problems in science and engineering. In 1996, Chen *et al.* [21] first developed the RKPM formulation for large deformation analysis, and introduced the concept of Lagrangian (material) and Eulerian (spatial) kernel function, and applied the method to large deformation of plasticity and hyperelasticity problems. The earlier applications of RKPM were in the modeling of rubber materials and metal forming analysis by Chen and co-workers ([11], [20]), and the method was later extended to computational geomechanics[70], multi-scale materials modeling [33], computational quantum mechanics [15], computational biomechanics [5], and fragment-impact modeling [33].

An important effort devoted to meshfree Galerkin methods development is the imposition of boundary conditions due to the lack of Kronecker delta property in MLS or RK approximations. Belytschko *et al.* [8] used Lagrange multiplier method to impose the essential boundary condition in EFG, and this method was later slightly modified by Lu *et al* [53] by replacing the Lagrange multiplier by the surface traction. In addition to the additional unknown to be approximated, Lagrange multiplier method may encounter stability problem if the approximation functions are not chosen properly. Transformation method was proposed by Chen *et al* in 1996 [21] and further enhanced by Chen and Wang [23], which is a method that transforms the approximation coefficients from generalized coordinate to nodal coordinate to allow a direct imposition of essential boundary conditions. Two versions of transformation method have been proposed: the full transformation method by Chen *et al.* [21], and the mixed transformation method by Chen and Wang [23]. A boundary singular kernel method has also been introduced for direct imposition of essential boundary conditions [23]. Fernández-Méndez and Huerta [41] used a combination of RK shape function and finite element shape function in the boundary as way of imposition of essential boundary conditions, and Chen *et al.* [18] introduced a primitive function as a correction of RK approximation to recover Kronecker delta properties. Other methods of imposing essential boundary conditions have also been employed from methods traditionally used in FEM, such as penalty method [2] and Nitsche's method [60].

Traditionally the weak form is integrated through Gauss quadrature, however, due to the fact that meshfree shape functions are rational functions, Gauss quadrature is inaccurate. Dolbow and Belytschko [31] investigated that if the background integration cell mismatch with the compact support of the nodes, significant error may

arise. Belytschko [8] used direct nodal integration (DNI) for domain integration, but the method suffers from the spatial instability due to the vanishing first order derivative of approximation functions at the nodes. Beissel and Belytschko [6] introduced a stabilization procedure by adding the square of the residual in the equilibrium equation to the potential energy functional to nodal integrated weak form to resolve the spatial instability issue in DNI. It is shown that by doing so, the close-to-singular modes can be effectively eliminated, but adding the additional term result in losing variational consistency. Chen *et al.* [27] proposed the Stabilized Conforming Nodal Integration (SCNI) to achieve first order Galerkin exactness and to suppress the zero energy modes resulting from NDI. SCNI is constructed based on a strain smoothing stabilization to eliminate spatial instability in nodal integration and shows excellent accuracy over Gauss quadrature and direct nodal integration method. Chen and Wang [66] also introduced constrained RK approximation in conjunction with a sub-domain SCNI to meshfree shell formulation.

2.3. Reproducing Kernel Collocation Method (RKCM)

To resolve the difficulties in domain integration and imposing essential boundary conditions, strong form collocation method based on reproducing kernel approximation has been proposed by Aluru [1]. Hu and Chen *et al.* [37] pointed out the unbalanced errors between the domain and boundary collocation equations and introduced the weighted collocation method to the collocation method based on reproducing kernel approximation and termed it the reproducing kernel collocation method (RKCM) [38]. The estimate of condition number of RKCM by Hu and Chen *et al.* [36] indicated that the stability of RKCM is similar to FEM, and is significantly more stable than collocation methods based on radial basis functions. Later, Hu and

Chen [38] proved that the second order basis functions are required for RKCM to converge. Hu *et al.* [39] further discussed the computational complexity of RKCM.

Improvements have been made to enhance the computational efficiency of RKCM. Chi and Chen *et al.* [29] proposed a gradient RKCM method (G-RKCM) that reduces the order of differentiation to the first order for solving second order PDEs with strong form collocation. They showed that by using G-RKCM, the same number of collocation points and source points can be used to achieve optimal convergence, unlike typical collocation method where much more collocation points than source points are needed to reach this goal.

2.4. Radial Basis Collocation Method (RBCM)

The earlier development of strong form collocation method is credited to the Radial Basis Collocation Method (RBCM). RBCM uses radial basis functions (RBF) as the global approximation functions, and the exponential convergence and infinitely differentiable properties of RBF make it promising in science and engineering applications. Kansa [44] first introduced multiquadrics (MQ) radial basis function to computational fluid dynamics based strong form collocation method. Hu *et al.* [40] introduced radial basis functions (RBF) into the collocation methods to solve elliptic boundary value problems. Hu and Chen *et al.* [37] proposed the weighted RBCM through a weighted least-squares functional to achieve enhanced accuracy and convergence in collocation methods. However, RBCM suffers from ill-conditioning in model refinement due to the global approximation nature of RBF. Efforts have been made to yield a well-conditioned discrete system in RBCM. Schaback and Wendland [69] introduced a new class of RBFs that are positive definite and compactly supported for solving partial differential equations. Chen *et al.* [15]

introduced the reproducing kernel as the localizing function of radial basis function to achieve locality and thus providing a better conditioned discrete system, and the method has been extended to the subdomain collocation method for heterogeneous media [25]. This reproducing kernel enhanced local radial basis collocation method enjoys the benefit of RK approximation and radial basis approximation, and yields a relatively well-conditioned system with higher convergence rate than the regular RK approximation. Wang and Chen *et al.* [68] applied RBCM to fracture mechanics by introducing a domain decomposition technique with appropriate sub-domain interface conditions. Chi and Chen *et al.* [30] performed the dispersion and stability analysis of RBCM in elastodynamics, in which the proper weights for the boundary collocation equations for optimal convergence was derived. This work also showed that the shape parameter strongly influences the dispersion errors and the temporal stability conditions in elastodynamics. Chi and Chen *et al.* [28] further introduced a weighted strong form collocation framework with mixed radial basis approximations for the pressure and displacement fields for incompressible and nearly incompressible linear elasticity. In this work, optimal convergence has been achieved through proper choice of independent source points and collocation points in the pressure and displacement fields in conjunction with proper weights for the boundary collocation equations.

3 Review of Reproducing Kernel Particle Method (RKPM)

In this chapter, we will review the reproducing kernel (RK) approximation and reproducing kernel particle method (RKPM). The essential boundary imposing techniques used for the Galerkin meshfree method and the domain integration techniques summarized in the Lecture Note of Meshfree Methods for Linear and Nonlinear Mechanics by J. S. Chen [19] are also presented.

3.1. Reproducing Kernel Approximation

The discrete reproducing kernel (RK) approximation of a function $u(\mathbf{x})$, denoted by $u^h(\mathbf{x})$, is expressed as

$$u^h(\mathbf{x}) = \sum_{I=1}^{N_p} \bar{w}_a^{[n]}(\mathbf{x}; \mathbf{x} - \mathbf{x}_I) d_I \quad (1)$$

where d_I are the coefficients associated with nodes $\mathbf{x}_I$, N_p is the number of nodes, the superscript n represents the order of completeness in this approximation, and $\bar{w}_a^{[n]}$ is called the reproducing kernel obtained by correcting a kernel function $\phi_a(\mathbf{x} - \mathbf{x}_I)$ as

$$\bar{w}_a^{[n]}(\mathbf{x}; \mathbf{x} - \mathbf{x}_I) = C^{[n]}(\mathbf{x}; \mathbf{x} - \mathbf{x}_I) \phi_a(\mathbf{x} - \mathbf{x}_I) \quad (2)$$

where $\phi_a(\mathbf{x} - \mathbf{x}_I)$ is a compactly supported kernel function that defines the locality and smoothness of the approximation, and $C^{[n]}(\mathbf{x}; \mathbf{x} - \mathbf{x}_I)$ is the correction function formed by a linear combination of monomial bases

$$C^{[n]}(\mathbf{x}; \mathbf{x} - \mathbf{x}_I) = \sum_{i+j+k=0}^n (x_1 - x_{1I})^i (x_2 - x_{2I})^j (x_3 - x_{3I})^k b_{ijk}(\mathbf{x}) = \mathbf{H}(\mathbf{x} - \mathbf{x}_I) \mathbf{b}(\mathbf{x}) \quad (3)$$

where $\mathbf{H}(\mathbf{x} - \mathbf{x}_I)$ is a vector composed of n -th order complete monomial bases expressed as

$$\mathbf{H}(\mathbf{x} - \mathbf{x}_I) = [1, x_1 - x_{1I}, x_2 - x_{2I}, \dots, (x_3 - x_{3I})^n] \quad (4)$$

The correction function is used to allow the approximation to satisfy the n -th order reproducing conditions

$$\sum_{I=1}^{N_p} \bar{w}_a^{[n]}(\mathbf{x}; \mathbf{x} - \mathbf{x}_I) x_{1I}^i x_{2I}^j x_{3I}^k = x_1^i x_2^j x_3^k \quad \text{for } i + j + k = 0, \dots, n \quad (5)$$

Substituting Equations (3) and (4) into (5), the coefficients $\mathbf{b}(\mathbf{x})$ are solved by

$$\mathbf{M}(\mathbf{x}) \mathbf{b}(\mathbf{x}) = \mathbf{H}(\mathbf{0}) \quad (6)$$

where the moment matrix is

$$\mathbf{M}(\mathbf{x}) = \sum_{I=1}^{N_p} \mathbf{H}(\mathbf{x} - \mathbf{x}_I) \mathbf{H}^T(\mathbf{x} - \mathbf{x}_I) \phi_a(\mathbf{x} - \mathbf{x}_I) \quad (7)$$

and

$$\mathbf{H}(\mathbf{0}) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (8)$$

Note that $\mathbf{M}(\mathbf{x})$ in Equation (6) needs to be invertible, and it requires the support of $\phi_a(\mathbf{x} - \mathbf{x}_I)$ to be greater than a minimum size to cover enough nodes, which is related to the order of basis functions used in the correction function $C^{[n]}(\mathbf{x}; \mathbf{x} - \mathbf{x}_I)$, and the kernel function $\phi_a(\mathbf{x} - \mathbf{x}_I)$ must be a positive function within the support. Besides, the employment of shifted bases in $\mathbf{H}(\mathbf{x} - \mathbf{x}_I)$ can improve the

conditioning of the moment matrix. Substitute the solution $\mathbf{b}(\mathbf{x}) = \mathbf{M}^{-1}\mathbf{H}(\mathbf{0})$ together with Equations (7) and (8) into Equation (5), the RK approximation is rewritten as

$$u^h(\mathbf{x}) = \sum_{l=1}^{N_p} \mathbf{H}^T(\mathbf{0}) \mathbf{M}^{-1}(\mathbf{x}) \mathbf{H}(\mathbf{x} - \mathbf{x}_l) \phi_a(\mathbf{x} - \mathbf{x}_l) d_l \equiv \sum_{l=1}^{N_p} \Psi_l(\mathbf{x}) d_l \quad (9)$$

where $\Psi_l(\mathbf{x})$ are called the RK approximation function

As discussed above, the kernel (weight) function ϕ_a determines the continuity and locality of the RK approximation function in Equation (9), and it needs to be a positive function to guarantee the positive-definiteness of the approximation. It is essential to choose proper kernel function for the solving PDE's. The cubic B-splines and quantic B-splines have been widely used as the kernel function:

1. Cubic B-spline:

$$\phi_a(z) = \begin{cases} \frac{2}{3} - 4z^2 + 4z^3, 0 \leq z \leq \frac{1}{2} & 0 \leq z \leq \frac{1}{2} \\ \frac{4}{3} - 4z + 4z^2 - \frac{3}{4}z^3, \frac{1}{2} \leq z \leq 1 & \frac{1}{2} \leq z \leq 1 \\ 0 & \text{elsewise} \end{cases} \quad (10)$$

2. Quintic B-spline:

$$\phi_a(z) = \begin{cases} \frac{11}{20} - \frac{9z^2}{2} + \frac{81z^4}{4} - \frac{81z^5}{4} & 0 \leq z \leq \frac{1}{3} \\ \frac{17}{40} + \frac{15z}{8} - \frac{63z^2}{4} + \frac{135z^3}{4} - \frac{243z^4}{8} + \frac{81z^5}{8} & \frac{1}{3} \leq z \leq \frac{2}{3} \\ \frac{81}{40} - \frac{81z}{8} + \frac{81z^2}{4} - \frac{81z^3}{4} + \frac{81z^4}{8} - \frac{81z^5}{40} & \frac{2}{3} \leq z \leq 1 \\ 0 & \text{elsewise} \end{cases} \quad (11)$$

where $z = \frac{|\mathbf{x} - \mathbf{x}_l|}{a}$, and in 2D case

$$z = \frac{\sqrt{(x_1 - x_{1l})^2 + (x_2 - x_{2l})^2}}{a} \quad (12)$$

Here a is the normalized compact support size. Note that, in general, the RK shape function does not have Kronecker delta property, that is

$$\Psi_I(x_J) \neq \delta_{IJ} \quad (13)$$

and the nodal coefficient of $\mathbf{x}_I$ is not equal to the approximation value at $\mathbf{x}_I$

$$u^h(\mathbf{x}_I) \neq d_I \quad (14)$$

Therefore special treatment may be required for the imposition of essential boundary conditions while solving a boundary value problem discretized by RK approximation.

3.2. Galerkin Formulation

Given an elastostatics problem, the partial differential equations with boundary conditions are stated as follows

$$\sigma_{ij,j} + b_i = 0 \quad \text{in } \Omega \quad (15)$$

with boundary conditions

$$\begin{aligned} \sigma_{ij} n_j &= h_i & \text{on } \Gamma^{hi} \\ u_i &= g_i & \text{on } \Gamma^{gi} \end{aligned} \quad (16)$$

where Ω is the problem domain with essential boundary Γ^g and natural boundary Γ^h ,

u_i is the component of the displacement, b_i is the body force, C_{ijkl} is the elasticity

tensor, n_i is the surface normal, h_i is the surface traction and g_i is the prescribed

displacement. The corresponding weak form statement is: Find $u_i \in H_g^1(\Omega)$, for

arbitrary $\delta u_i \in H_0^1(\Omega)$, such that

$$\int_{\Omega} \delta u_{i,j} C_{ijkl} u_{k,l} d\Omega = \int_{\Omega} \delta u_i b_i d\Omega + \int_{\Gamma^{hi}} \delta u_i h_i d\Gamma \quad (17)$$

where

$$H_g^1(\Omega) = \left\{ v_i \in H^1(\Omega) : v_i = g_i \text{ on } \Gamma^{gi} \right\} \quad (18)$$

and

$$H_0^1(\Omega) = \left\{ v_i \in H^1(\Omega) : v_i = 0 \text{ on } \Gamma^{gi} \right\} \quad (19)$$

Let the displacement solution u be approximated as u^h , the Galerkin form of the problem is expressed as: Find $u_i^h \in H_g^1(\Omega)$, for arbitrary $\delta u_i^h \in H_0^1(\Omega)$, such that

$$\int_{\Omega} \delta u_{i,j}^h C_{ijkl} u_{k,l}^h d\Omega = \int_{\Omega} \delta u_i^h b_i d\Omega + \int_{\Gamma^h} \delta u_i^h h_i d\Gamma \quad (20)$$

3.3. Reproducing Kernel Particle Method and Techniques for Imposing Essential Boundary Conditions

Let the displacement in Equation (20) be discretized by RK shape functions introduced in Section 3.1, that is

$$u_i^h(\mathbf{x}) = \sum_{I=1}^{N_p} \Psi_I(\mathbf{x}) d_{iI} \equiv \Psi^{iT}(\mathbf{x}) \mathbf{d} \quad (21)$$

where

$$\Psi^1(\mathbf{x}) = \begin{bmatrix} \Psi_1(\mathbf{x}) \\ 0 \\ 0 \\ \Psi_2(\mathbf{x}) \\ 0 \\ 0 \\ \vdots \\ \Psi_{N_p}(\mathbf{x}) \\ 0 \\ 0 \end{bmatrix}, \Psi^2(\mathbf{x}) = \begin{bmatrix} 0 \\ \Psi_1(\mathbf{x}) \\ 0 \\ 0 \\ \Psi_2(\mathbf{x}) \\ 0 \\ \vdots \\ 0 \\ \Psi_{N_p}(\mathbf{x}) \\ 0 \end{bmatrix}, \Psi^3(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \\ \Psi_1(\mathbf{x}) \\ 0 \\ 0 \\ \Psi_2(\mathbf{x}) \\ \vdots \\ 0 \\ 0 \\ \Psi_{N_p}(\mathbf{x}) \end{bmatrix}, \mathbf{d} = \begin{bmatrix} d_{11} \\ d_{21} \\ d_{12} \\ d_{22} \\ d_{32} \\ \vdots \\ d_{1N_p} \\ d_{2N_p} \\ d_{3N_p} \end{bmatrix} \quad (22)$$

The employment of Galerkin weak form (20) using RK approximation is called Reproducing Kernel Particle Method (RKPM). Note that RK approximation functions do not possess interpolating property, and the Galerkin weak form needs to be appropriately modified for applying the essential boundary conditions. In this

section, the Lagrange multipliers method, the penalty method and the transformation method are introduced to impose the essential boundary conditions in the RKPM framework.

3.3.1. Lagrange Multiplier Method

Lagrange multiplier method [8] has been employed for imposition of the essential boundary conditions on Γ^g , where the modified Galerkin weak form states: Find $u_i^h \in H^1(\Omega)$, and $\lambda_i^h \in L^2(\Gamma^{gi})$, for arbitrary $\delta u_i^h \in H^1(\Omega)$, and $\delta \lambda_i^h \in L^2(\Gamma^{gi})$, such that

$$\int_{\Omega} \delta u_{(i,j)}^h C_{ijkl} u_{(k,l)}^h d\Omega + \int_{\Gamma^{gi}} \delta \lambda_i^h (u_i^h - g_i) d\Gamma + \int_{\Gamma^{gi}} \delta u_i^h \lambda_i^h d\Gamma = \int_{\Omega} \delta u_i^h b_i d\Omega + \int_{\Gamma^{hi}} \delta u_i^h h_i d\Gamma \quad (23)$$

where λ_i^h is the Lagrange multiplier function and discretized as

$$\lambda_i^h(\mathbf{x}) = \sum_{l=1}^{N_b} \bar{N}_l(\mathbf{x}) \Lambda_{il}, \quad \mathbf{x} \in \Gamma^g \quad (24)$$

Here Λ_{il} are coefficients of the Lagrange multiplier approximation on the essential boundary Γ^g , and N_b is the number of points where the multipliers are defined. Note that the shape function $\bar{N}_l(\mathbf{x})$ could be different from the standard RK approximation function as long as it satisfies LBB condition.

Substituting Equations (21) and (24) into the modified Galerkin weak form, Equation (23), the following discrete matrix equation is obtained:

$$\begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \Lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{q} \end{bmatrix} \quad (25)$$

where $\mathbf{d}$ and Λ are the vector forms of d_{il} and Λ_{il} , respectively. The components in Equation (25) are

$$\mathbf{K}_{IJ} = \int_{\Omega} \mathbf{B}_I^T \mathbf{D} \mathbf{B}_J d\Omega \quad (26)$$

$$\mathbf{G}_{IK} = \int_{\Gamma^g} \Psi_I \bar{N}_K d\Gamma \quad (27)$$

$$\mathbf{f}_I = \int_{\Gamma^h} \Psi_I t d\Gamma + \int_{\Omega} \Psi_I \mathbf{b} d\Omega \quad (28)$$

$$\mathbf{q}_I = \int_{\Gamma^g} \bar{N}_I \mathbf{g} d\Gamma \quad (29)$$

where $\mathbf{D}$ is the elasticity matrix, and $\mathbf{B}$ is

$$\mathbf{B}_I = \begin{bmatrix} \Psi_{I,x} & 0 \\ 0 & \Psi_{I,y} \\ \Psi_{I,y} & \Psi_{I,x} \end{bmatrix} \quad (30)$$

3.3.2. Penalty Method

An alternative way to impose the essential boundary conditions is the penalty method [7]. Consider the same elastostatics problems given in Equations (15) and (16), the associated functional is

$$\Pi_h = \frac{1}{2} \int_{\Omega} u_{(i,j)}^h C_{ijkl} u_{(k,l)}^h d\Omega - \int_{\Omega} u_i^h b_i d\Omega - \int_{\Gamma^h} u_i^h h_i dS + \frac{\beta}{2} \int_{\Gamma^g} (u_i^h - g_i)(u_i^h - g_i) d\Gamma \quad (31)$$

where β is the penalty parameter, which is taken as a large number to impose the constraint $u_i^h - g_i = 0$ on the essential boundary Γ^g . Taking stationary of Π_h , i.e.

$\delta \Pi_h = 0$, and introducing RK approximation, the matrix form is obtained

$$(\mathbf{K} + \beta \mathbf{K}^p) \mathbf{d} = \mathbf{f} + \beta \mathbf{f}^p \quad (32)$$

where $\mathbf{K}$, $\mathbf{d}$ and $\mathbf{f}$ are shown in previous section, the additional terms $\mathbf{K}^p$ and $\mathbf{f}^p$ are presented due to the imposition of essential boundary conditions:

$$\mathbf{K}_{IJ}^p = \int_{\Gamma^g} \Psi_I \mathbf{S} \Psi_J d\Gamma \quad (33)$$

$$\mathbf{f}_I^p = \int_{\Gamma^g} \Psi_I \mathbf{S} \mathbf{g} d\Gamma \quad (34)$$

where

$$\mathbf{S} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \quad \text{and} \quad S_i = \begin{cases} 1 & \text{if } u_i \text{ is prescribed on } \Gamma^g \\ 0 & \text{if } u_i \text{ is not prescribed on } \Gamma^g \end{cases} \quad (35)$$

3.3.3. Transformation Method

Due to the lack of Kronecker delta properties in reproducing kernel approximation functions, the essential boundary conditions require special treatments. An easy way to impose essential boundary conditions for meshfree methods is the transformation method [23]. Recall the essential boundary conditions expressed as

$$u_i(\mathbf{x}_J) = g_i(\mathbf{x}_J), \quad \forall J \in \eta_g \quad (36)$$

where $g_i(\mathbf{x}_J)$ denotes the prescribed displacement, and η_g is the set of node numbers in which the associated nodes are located on the essential boundary Γ^g . We start with the potential functional of the boundary value problem, Eqs. (15) and (16), with essential boundary condition imposed by a Lagrange multiplier to yield

$$\Pi(\mathbf{u}, \lambda) = \int_{\Omega} \frac{1}{2} u_{(i,j)} C_{ijkl} u_{(k,l)} d\Omega - \int_{\Omega} \delta u_i b_i d\Omega - \int_{\Gamma^{hi}} \delta u_i h_i d\Gamma + \sum_{J \in \eta_g} \lambda_{iJ} [u_i(\mathbf{x}_J) - g_i(\mathbf{x}_J)] \quad (37)$$

where λ_{iJ} is the Lagrange multiplier. Take variation of the functional $\Pi(\mathbf{u}, \lambda)$, and discretize the displacement by RK approximation (21), we have

$$\begin{aligned} \delta \Pi(\mathbf{u}^h, \lambda) = & \int_{\Omega} \delta u_{(i,j)}^h C_{ijkl} u_{(k,l)}^h d\Omega - \int_{\Omega} \delta u_i^h b_i d\Omega - \int_{\Gamma^{hi}} \delta u_i^h h_i d\Gamma + \\ & \sum_{J \in \eta_g} \delta \lambda_{iJ} [u_i^h(\mathbf{x}_J) - g_i(\mathbf{x}_J)] + \delta u_i^h(\mathbf{x}_J) \lambda_{iJ} = 0 \end{aligned} \quad (38)$$

For easy illustration of the concept, only consider a single boundary constraint, i.e. $u_i(\mathbf{x}_J) = g_i(\mathbf{x}_J)$, in Eq. (36). Thus Eq. (38) becomes

$$\begin{aligned} \delta \Pi(\mathbf{u}^h, \lambda) = & \int_{\Omega} \delta u_{(i,j)}^h C_{ijkl} u_{(k,l)}^h d\Omega - \int_{\Omega} \delta u_i^h b_i d\Omega - \int_{\Gamma^{hi}} \delta u_i^h h_i d\Gamma + \\ & \delta \lambda_{ij} [u_i^h(\mathbf{x}_j) - g_i(\mathbf{x}_j)] + \delta u_i^h(\mathbf{x}_j) \lambda_{ij} = 0 \end{aligned} \quad (39)$$

Recall the reproducing kernel approximation (21)

$$\hat{u}_{ij} = u_i^h(\mathbf{x}_j) = \sum_{l=1}^{N_p} \Psi_l(\mathbf{x}_j) d_{il} \equiv \mathbf{A}_j^{iT} \mathbf{d} \quad (40)$$

where $\hat{u}_{ij}$ is the value of the approximation function u_i^h evaluated on $\mathbf{x}_j$, and

$$\mathbf{A}_j^i = \mathbf{Y}^i(\mathbf{x}_j) \quad (41)$$

Thus Eq. (40) can be further written as

$$\hat{\mathbf{u}} = \Lambda \mathbf{d} \quad (42)$$

where $\hat{\mathbf{u}}$ is the nodal displacement vector of $\hat{u}_{ij}$ and Λ is the transformation matrix

$$\hat{\mathbf{u}} = \begin{bmatrix} u_1^h(\mathbf{x}_1) \\ u_2^h(\mathbf{x}_1) \\ u_3^h(\mathbf{x}_1) \\ u_1^h(\mathbf{x}_2) \\ \vdots \\ u_1^h(\mathbf{x}_{N_p}) \\ u_2^h(\mathbf{x}_{N_p}) \\ u_3^h(\mathbf{x}_{N_p}) \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \mathbf{A}_1^{1^T} \\ \mathbf{A}_1^{2^T} \\ \mathbf{A}_1^{3^T} \\ \mathbf{A}_2^{2^T} \\ \vdots \\ \mathbf{A}_{N_p}^{1^T} \\ \mathbf{A}_{N_p}^{2^T} \\ \mathbf{A}_{N_p}^{3^T} \end{bmatrix} \quad (43)$$

By using the RK approximation Eqs. (21) and (40), the Galerkin weak form

Eq. (39) can be discretized into the following linear system

$$\begin{bmatrix} \mathbf{K} & \mathbf{A}_j^i \\ \mathbf{A}_j^{iT} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \lambda_{ji} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ g_{ji} \end{bmatrix} \quad (44)$$

where $g_{ji} \equiv g_i(\mathbf{x}_j)$. Since $\mathbf{d} = \Lambda^{-1} \hat{\mathbf{u}}$, the RK approximation Eq. (21) becomes

$$u_i^h(\mathbf{x}) = \mathbf{Y}^{iT}(\mathbf{x}) \mathbf{d} = \mathbf{Y}^{iT}(\mathbf{x}) \Lambda^{-1} \hat{\mathbf{u}} \quad (45)$$

Note that when $\mathbf{x} = \mathbf{x}_j$, from Eqs. (40), (41) and (45) we have

$$\hat{u}_{ij} = u_i^h(\mathbf{x}_j) = \mathbf{Y}^{i^T}(\mathbf{x}_j)\Lambda^{-1}\hat{\mathbf{u}} = \mathbf{A}_j^i\Lambda^{-1}\hat{\mathbf{u}} \quad (46)$$

That implies $\mathbf{A}_j^i\Lambda^{-1} = \delta_j^i$ and the Kronecker delta property is recovered. Multiply the first set of matrix equation in Eq. (44) by Λ^{-T} , the matrix form becomes

$$\begin{bmatrix} \Lambda^{-T}\mathbf{K} & \mathbf{I}_J \\ \mathbf{A}_J^{i^T} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \lambda_{ji} \end{bmatrix} = \begin{bmatrix} \Lambda^{-T}\mathbf{f} \\ g_{ji} \end{bmatrix} \quad (47)$$

where

$$\mathbf{I}_J^i = \Lambda^{-T}\mathbf{A}_J^i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow (3 \times (J-1) + i)^{th} \quad (48)$$

Now we define the following matrices and vectors for further manipulation of Eq. (47)

$$\hat{\mathbf{K}} \equiv \Lambda^{-T}\mathbf{K} = \begin{bmatrix} \hat{\mathbf{k}}_1^T \\ \vdots \\ \hat{\mathbf{k}}_{3N_p}^T \end{bmatrix}, \quad \hat{\mathbf{k}}_1^T = \begin{bmatrix} \hat{K}_{11} & \hat{K}_{12} & \hat{K}_{13} & \cdots & \hat{K}_{3N_p} \end{bmatrix} \quad (49)$$

$$\hat{\mathbf{f}} = \Lambda^{-T}\mathbf{f}$$

Then Eq. (47) can be rewritten in the following form

$$\begin{bmatrix} \hat{\mathbf{K}} & \mathbf{I}_J \\ \mathbf{A}_J^{i^T} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \lambda_{ji} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}} \\ g_{ji} \end{bmatrix} \quad (50)$$

Exchanging the $(3 \times (J-1) + i)^{th}$ row and the last row of Eq. (50) leads to the following equation

$$\begin{bmatrix} \hat{\mathbf{k}}_1^T & 0 \\ \vdots & \vdots \\ \hat{\mathbf{k}}_{M-1}^T & 0 \\ \boxed{\mathbf{A}_J^{i^T}} & \boxed{0} \\ \hat{\mathbf{k}}_{M+1}^T & 0 \\ \vdots & \vdots \\ \hat{\mathbf{k}}_{3N_p}^T & 0 \\ \boxed{\hat{\mathbf{k}}_M^T} & \boxed{1} \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \lambda_{Ji} \end{bmatrix} = \begin{bmatrix} \hat{f}_1 \\ \vdots \\ \hat{f}_{M-1} \\ \boxed{g_{Ji}} \\ \hat{f}_{M+1} \\ \vdots \\ \hat{f}_{3N_p} \\ \boxed{\hat{f}_M} \end{bmatrix} \leftarrow M = (3 \times (J-1) + i)^{th} \quad (51)$$

In this manner, the displacement coefficient vector and the Lagrange multiplier can be solved separately by

$$\begin{aligned} \tilde{\mathbf{K}}\mathbf{d} &= \tilde{\mathbf{f}} \\ \lambda_{Ji} &= \hat{f}_M - \hat{\mathbf{k}}_M^T \mathbf{d} \end{aligned} \quad (52)$$

where

$$\tilde{\mathbf{K}} = L_J^i(\hat{\mathbf{K}}) = \begin{bmatrix} \hat{\mathbf{k}}_1^T \\ \vdots \\ \hat{\mathbf{k}}_{M-1}^T \\ \boxed{\mathbf{A}_J^{i^T}} \\ \hat{\mathbf{k}}_{M+1}^T \\ \vdots \\ \hat{\mathbf{k}}_{2 \times NP}^T \end{bmatrix} \leftarrow M = (3 \times (J-1) + i) \quad (53)$$

$$\tilde{\mathbf{f}} = l_J^i(\hat{\mathbf{f}}) = \begin{bmatrix} \hat{f}_1 \\ \vdots \\ \hat{f}_{M-1} \\ \boxed{g_{Ji}} \\ \hat{f}_{M+1} \\ \vdots \\ \hat{f}_{3N_p} \end{bmatrix} \quad (54)$$

Here we define a matrix operator L_J^i that replaces the $(3 \times (J-1) + i)^{th}$ row of

$\hat{\mathbf{K}}$ by $\mathbf{A}_J^{iT}$, and a vector operator l_J^i that replaces the $(3 \times (J-1) + i)^{th}$ component of $\hat{\mathbf{f}}$ by g_{Ji} . When dealing with multiple essential boundary constraints, the resulting stiffness matrix and the force vector are obtained through repeating procedures

$$\tilde{\mathbf{K}} = \mathbf{L}_{J_{Nc}}^{i_{Nc}} \cdots \mathbf{L}_{J_2}^{i_2} \mathbf{L}_{J_1}^{i_1}(\hat{\mathbf{K}}), \quad \tilde{\mathbf{f}} = l_{J_{Nc}}^{i_{Nc}} \cdots l_{J_2}^{i_2} l_{J_1}^{i_1}(\hat{\mathbf{f}}) \quad (55)$$

The above approach is called full transformation method. The mixed transformation method has been introduced to enhance the computational efficiency, in which the nodes are separated into essential boundary and non-essential boundary groups, and the coordinate transformation is only applied to the essential boundary group, see [23] for details.

3.4. Nodal Integration Techniques for Meshfree Methods

Gauss quadrature is usually used to evaluate the domain integrals in the Galerkin weak formulation in Equation (20). In the meshfree method, however, Gauss quadrature becomes much costly since the meshfree approximation functions (e.g. RK function) are rational functions, and this requires higher order of Gauss quadrature to guarantee the integration accuracy. In addition, integration errors are caused by the fact that the integration cells in meshfree method are usually constructed independently of the approximation function, thus the compact support of nodes may not align with the integration domain (Dolbow, Belytschko [31]). Therefore, nodal integration techniques were proposed to resolve these issues. Two types of nodal integrations are discussed: Direct Nodal Integration (DNI) (Belytschko *et al.* [8]) and Stabilized Conforming Nodal Integration (SCNI) (Chen *et al.* [27]).

3.4.1. Direct Nodal Integration (DNI)

Numerical integration rules used for approximating a continuous integral over a domain Ω can be generally expressed in a summation form

$$\int_{\Omega} f(\xi) d\xi = \sum_{l=1}^{N_{gs}} f(\xi_l) w_l \quad (56)$$

where $f(\xi)$ is the continuous integrand, N_{gs} is the number of quadrature points, ξ_l is the coordinate of the point l , and w_l is the corresponding weight. Different integration approaches vary only with regard to the locations of the evaluation point and the associated weights. The idea of the direct nodal integration (DNI) technique used in meshfree method is to integrate only at the nodes

$$\int_{\Omega} f(\mathbf{x}) d\Omega = \sum_{l=1}^{N_p} f(\mathbf{x}_l) V_l \quad (57)$$

where $f(\mathbf{x})$ can be taken as any function in the Galerkin form needs to be integrated. In this manner, the quadrature weights V_l represent nodal areas (volumes) associated with the nodes, which can be obtained from a Voronoi diagram constructed from the domain nodes.

DNI has the benefits ([10]) including that there is no need to create integration cells, the data structure is as simple as that of a particle method due to all discretized variables are defined only at the nodes, and the cost of evaluation of the integrals of the Galerkin weak form is much less than for complete Gaussian quadrature. Nevertheless, using DNI may cause seriously numerical instability due to the rank deficiency (Chen *et al.* [27]) and poor accuracy and low convergence in non-uniform discretization (Chen *et al.* [14]). This issue can be resolved through the stabilized nodal integration techniques, which will be discussed in the following section.

3.4.2. Stabilized Conforming Nodal Integration (SCNI)

Stabilized conforming nodal integration (SCNI) technique by Chen *et al.* [27] was proposed to stabilize the nodal integration while preserving the first order consistency in the meshfree Galerkin approximation. In SCNI approach, a strain smoothing stabilization for nodal integration and an integration constraint (IC) for Galerkin approximation are introduced, in order to obtain stable and optimally convergent Galerkin solution.

The convergence of Galerkin method for partial differential equations is determined by the approximation for the unknowns and the numerical integration of the weak form. Linearly consistent RK approximations can be achieved if the correction function, Eq. (3), contains complete linear basis functions. But the employment of RK approximation function with linear consistency in the Galerkin formulation, Eq. (20), does not guarantee linear exactness in the solution obtained by the Galerkin method. It is because the compact support for RK approximation functions, $\text{supp}(\Psi_I)$, is generally not conforming to the integration cells. Chen *et al.* [27] proposed the additional requirements associated with the domain integration of the Galerkin method, which are called the Integration Constraints (IC)

$$\sum_{L=1}^{\text{NIT}} \mathbf{B}_I(\mathbf{x}_L) \omega_L = \mathbf{0} \quad \text{for all interior nodes } \{I : \text{supp}(\Psi_I) \cap \Gamma = \emptyset\} \quad (58)$$

and

$$\sum_{L=1}^{\text{NIT}} \mathbf{B}_I(\mathbf{x}_L) \omega_L = \sum_{J=1}^{\text{NITh}} \mathbf{N}_I(\mathbf{x}_J) s_J \quad \text{for boundary nodes } \{I : \text{supp}(\Psi_I) \cap \Gamma^h \neq \emptyset\} \quad (59)$$

where NIT, $\mathbf{x}_L$ and ω_L are the number of integration points, coordinates and corresponding weights of the quadrature scheme in the domain; NITh, $\mathbf{x}_J$ and s_J are the number of integration points, coordinates and corresponding weights of the

quadrature scheme on the natural boundary; $\mathbf{B}_l$ is the standard gradient matrix shown in Eq. (30) and $\mathbf{N}_l$ is expressed

$$\mathbf{N}_l = \begin{bmatrix} \Psi_{ln_1} & 0 \\ 0 & \Psi_{ln_2} \\ \Psi_{ln_2} & \Psi_{ln_1} \end{bmatrix} \quad (60)$$

As long as the integration constraints, i.e. Eqs. (58) and (59) are satisfied, the solution obtained from Galerkin method can guarantee the linear exactness. If the whole problem boundary is pure displacement boundary, only Eq. (58) need to be satisfied. However, if DNI technique is applied on the domain integration of the Galerkin form, the stability issue exists. Thus a strain smoothing method is used as a stabilization of the zero energy modes.

For meeting the IC, the problem domain Ω is discretized by conforming cells. In this approach, Voronoi diagram is adopted and Ω_L is denoted as the Voronoi cell associated with each node $\mathbf{x}_L$. Note that Ω_L can be different from the compact support $supp(\Psi_L)$ of node L . The modified strain smoothing at a node L is

$$\tilde{\epsilon}_{ij}^h(\mathbf{x}_L) = \int_{\Omega} \epsilon_{ij}^h(\mathbf{x}) \Phi(\mathbf{x} - \mathbf{x}_L) d\Omega \quad (61)$$

where ϵ_{ij}^h is the strain obtained from displacement, $\epsilon_{ij}^h = \frac{1}{2}(u_{i,j}^h + u_{j,i}^h)$, and $\Phi(\mathbf{x} - \mathbf{x}_L)$

is a weight function, which is defined as follows for simplicity

$$\Phi(\mathbf{x} - \mathbf{x}_L) = \begin{cases} \frac{1}{A_L} & x \in \Omega_L \\ 0 & x \notin \Omega_L \end{cases} \quad (62)$$

where A_L is the measure of Ω_L . Therefore, by applying the divergence theorem, the domain integration, Eq. (61), can be transformed to boundary integration, the smoothed strain becomes

$$\bar{\epsilon}_{ij}^h(\mathbf{x}_L) = \frac{1}{2A_L} \int_{\Omega_L} \left(\frac{\partial u_i^h}{\partial x_j} + \frac{\partial u_j^h}{\partial x_i} \right) d\Omega = \frac{1}{2A_L} \int_{\Gamma_L} (u_i^h n_j + u_j^h n_i) d\Gamma \quad (63)$$

where Γ_L is the boundary of the Voronoi cell related with node L . Now introduce RK approximation functions into Eq. (63)

$$\bar{\epsilon}_{ij}^h(\mathbf{x}_L) = \sum_{I \in G_L} \tilde{\mathbf{B}}_I(\mathbf{x}_L) d_I \quad (64)$$

where G_L is the set of nodes within the compact support of the RK approximation function of node L . Here we take two dimensions case for example

$$\tilde{\epsilon}^{h^T} = \begin{bmatrix} \tilde{\epsilon}_{11}^h & \tilde{\epsilon}_{22}^h & 2\tilde{\epsilon}_{12}^h \end{bmatrix} \quad (65)$$

$$\mathbf{d}_I^T = \begin{bmatrix} d_{1I} & d_{2I} \end{bmatrix} \quad (66)$$

$$\tilde{\mathbf{B}}_I(\mathbf{x}_L) = \begin{bmatrix} \tilde{b}_{I1}(\mathbf{x}_L) & 0 \\ 0 & \tilde{b}_{I2}(\mathbf{x}_L) \\ \tilde{b}_{I1}(\mathbf{x}_L) & \tilde{b}_{I1}(\mathbf{x}_L) \end{bmatrix} \quad (67)$$

where

$$\tilde{b}_i(x_L) = \frac{1}{A_L} \int_{\Gamma_L} \Psi_I(\mathbf{x}) n_i(\mathbf{x}) d\Gamma \quad (68)$$

The above gradients are called the smoothed gradients. It can be showed the smoothed gradients defined above satisfies the integration constraints:

$$\sum_L \tilde{\mathbf{B}}_I(\mathbf{x}_L) A_L = \begin{bmatrix} \sum_L \tilde{b}_{I1}(\mathbf{x}_L) A_L & 0 \\ 0 & \sum_L \tilde{b}_{I2}(\mathbf{x}_L) A_L \\ \sum_L \tilde{b}_{I2}(\mathbf{x}_L) A_L & \sum_L \tilde{b}_{I1}(\mathbf{x}_L) A_L \end{bmatrix} = \mathbf{0} \quad (69)$$

and consequently,

$$\sum_L \tilde{b}_i(\mathbf{x}_L) A_L = 0 \quad (70)$$

where

$$\sum_L \tilde{b}_i(\mathbf{x}_L) A_L = \sum_L \int_{\Gamma_L} \Psi_I(\mathbf{x}) n_i(\mathbf{x}) d\Gamma \quad (71)$$

Consider the mixed variational principle in order to illustrate the strain smoothing formulation into the Galerkin approximation

$$\delta U(\mathbf{u}, \tilde{\epsilon}) = \int_{\Omega} \delta \tilde{\epsilon}_{ij} C_{ijkl} \tilde{\epsilon}_{kl} d\Omega - \int_{\Omega} \delta u_i b_i d\Omega - \int_{\Gamma^h} du_i h_i d\Gamma = 0 \quad (72)$$

The smoothed strain $\tilde{\epsilon}^h$ approximated by Eq. (64) is introduced. Nodal integration of (72) requires evaluation of matrix $\tilde{\mathbf{B}}$ at node $\mathbf{x}_L$ is employed. Note that to perform the boundary integration of the smoothed gradient, first introduce

$$\Gamma_L = \bigcup_{M=1}^{N_s} \Gamma_L^M \quad (73)$$

where Γ_L^M represent the boundary segments of Γ_L of the nodal domain and N_s is the total number of segments. A two-point trapezoidal rule for each segment of the nodal boundary is employed

$$\tilde{b}_i(\mathbf{x}_L) = \int_{\bigcup_{M=1}^{N_s} \Gamma_L^M} \Psi_I(\mathbf{x}) n_i(\mathbf{x}) d\Gamma = \sum_{M=1}^{N_s} \left[\Psi_I(\mathbf{x}_L^M) n_{iL}^M \frac{l_L^M}{2} + \Psi_I(\mathbf{x}_L^{M+1}) n_{iL}^M \frac{l_L^M}{2} \right] \quad (74)$$

where $\mathbf{x}_L^M$ and $\mathbf{x}_L^{M+1}$ are the two end points of boundary segment Γ_L^M , l_L^M is the length of segment Γ_L^M , and $\mathbf{n}_L^M$ denotes the outward surface normal of Γ_L^M . Finally, the smoothed gradient is introduced into the weak form to yield the following discrete equation

$$\mathbf{Kd} = \mathbf{f} \quad (75)$$

where

$$\mathbf{K}_{IJ} = \sum_{L=1}^{N_p} \tilde{\mathbf{B}}_I(\mathbf{x}_L) \mathbf{C} \tilde{\mathbf{B}}_J(\mathbf{x}_L) A_L \quad (76)$$

$$\mathbf{f}_I = \sum_{L=1}^{N_p} \Psi_I(\mathbf{x}_L) \mathbf{b}(x_L) A_L + \sum_{L=1}^{N_h} \Psi_I(\mathbf{x}_L) \mathbf{h}(\mathbf{x}_L) s_L \quad (77)$$

where N_h is the number of points on the natural boundary, s_L are associated weights with the boundary point.

3.4.3. Variationally Consistent Integration (VCI)

Chen and Hillman *et al.* [14] proposed an arbitrary order variationally consistent integration (VCI) as an extension of SCNI to achieve higher order Galerkin exactness, which is stated as higher order consistency as a more general term in this work. VCI is formulated based on the integration constraints derived from the discrete version of the weak form, which can be used as the correction of many common domain integration techniques to reach optimal convergence.

To illustrate the integration constraints, consider the model problem

$$\begin{aligned} Lu + s &= 0 \quad \text{in } \Omega \\ Bu &= h \quad \text{on } \Gamma^h \\ u &= g \quad \text{on } \Gamma^g \end{aligned} \quad (78)$$

where L is the differential operator of PDE, and B is the differential operator of natural boundary conditions. For Poisson's problem, the differential operators in (78) are

$$\begin{aligned} L &= \nabla^2 \\ B &= \nabla \end{aligned} \quad (79)$$

Define the following bilinear form

$$a(v, u)_\Omega = (\nabla v, \nabla u)_\Omega \quad (80)$$

A generalization of the linear integration constraints in arbitrary order of Galerkin consistency according to Chen and Hillman *et al.* [14] is

$$a\langle \hat{\Psi}_I, x^\alpha \rangle_\Omega = -\langle \hat{\Psi}_I, \nabla^2 x^\alpha \rangle_\Omega + \langle \hat{\Psi}_I, \nabla x^\alpha \cdot \mathbf{n} \rangle_\Gamma \quad \forall I, |\alpha| = 0, 1, \dots, n \quad (81)$$

where $a\langle\cdot,\cdot\rangle_\Omega$ is the quadrature form of $a(\cdot,\cdot)_\Omega$, and $\langle\cdot,\cdot\rangle_\Omega$ is the quadrature form of $(\cdot,\cdot)_\Omega$.

To satisfy n -th order integration constraints in (81), n -th order completeness in the trial functions is needed, and its gradient is expressed as:

$$\nabla u^h = \sum_{I=1}^{N_p} \nabla \Psi_I d_I \quad (82)$$

where the RK approximation function Ψ_I is n -th order complete. The gradient of the test function is expressed in the following form according to Chen and Hillman *et al.* [14]:

$$\tilde{\nabla} v^h = \sum_{I=1}^{N_p} (\nabla \Psi_I + \sum_{|\beta| \leq n} \xi_{\beta_I} \nabla \hat{\Psi}_I^\beta) v_I \quad (83)$$

The additional bases $\{\hat{\Psi}_I^\beta\}_{|\beta| \leq n}$ and Ψ_I need to be linearly independent. Substituting (83) into (81) yields a linear system to solve for the coefficients $\{\xi_{\beta_I}\}_{|\beta| \leq n}$ for the chosen quadrature rule in (81):

$$\begin{aligned} \sum_{|\beta|=1}^n A_{\alpha\beta I} \xi_{\beta I} &= r_{\alpha I} \quad \forall I, \quad |\alpha| = 0, 1, \dots, n \\ A_{\alpha\beta I} &= a \langle \hat{\Psi}_I^\beta, \mathbf{x}^\alpha \rangle_\Omega + \langle \hat{\Psi}_I^\beta, L\mathbf{x}^\alpha \rangle_\Omega - \langle \hat{\Psi}_I^\beta, B\mathbf{x}^\alpha \rangle_{\partial\Omega} \\ r_{\alpha I} &= - \left(a \langle \Psi_I, \mathbf{x}^\alpha \rangle_\Omega + \langle \Psi_I, L\mathbf{x}^\alpha \rangle_\Omega - \langle \Psi_I, B\mathbf{x}^\alpha \rangle_{\partial\Omega} \right) \end{aligned} \quad (84)$$

The final discrete equation of Galerkin approximation is obtained by substituting the resulting assumed gradient for trial functions in (83) and the n -th order complete approximation for the test function into the Galerkin weak form integrated with the quadrature rule used in (81). In VCI, the direct gradient and the assumed gradient is employed into a Petrov Galerkin framework where the trial function and the test function are different, see Chen and Hillman *et al.* [14] for details.

Remark: It is possible to choose basis functions in Eq. (83) to yield a decoupled linear system in Eq. (84). An example given in Chen *et al.* [14] for linear consistency is use of the following corrected gradient

$$\tilde{\nabla} v^h = \sum_{I=1}^{N_p} \left(\nabla \Psi_I + R_I \begin{Bmatrix} \xi_{1I} \\ \xi_{2I} \end{Bmatrix} + S_I \begin{Bmatrix} \xi_{3I} \\ \xi_{4I} \end{Bmatrix} + T_I \begin{Bmatrix} \xi_{5I} \\ \xi_{6I} \end{Bmatrix} \right) v_I \quad (85)$$

where

$$\begin{aligned} S_I &= R_I \cdot (x - x_I) \\ T_I &= R_I \cdot (y - y_I) \end{aligned} \quad (86)$$

The constraints are uncoupled into two sets of equations

$$\begin{aligned} A_I \xi_I^1 &= r_I^1 \\ A_I \xi_I^2 &= r_I^2 \end{aligned} \quad (87)$$

where

$$\mathbf{A}_I = \langle \mathbf{P}(\mathbf{x} - \mathbf{x}_I) \mathbf{P}^T (\mathbf{x} - \mathbf{x}_I) R_I \rangle_{\Omega} \quad (88)$$

$$\mathbf{P}(x) = \begin{bmatrix} 1 & x & y \end{bmatrix}^T \quad (89)$$

The coefficients can be solved hereby. More details can be referred to [14]

4 Review of Reproducing Kernel Collocation Method (RKCM) and Radial Basis Collocation Method (RBCM)

In this chapter, two meshfree strong form collocation methods, the reproducing kernel collocation method (RKCM) and the radial basis collocation method (RBCM) will be reviewed. The key idea of point collocation methods is to enforce zero residuals of partial differential equations (PDEs) and boundary conditions at each collocation point. Different types of strong form collocation formulations can be constructed using different types of approximation functions.

4.1. Reproducing Kernel Collocation Method (RKCM)

Compared to the Galerkin meshfree formulation, RKCM has the following unique properties:

- (1) Once shape functions and their derivatives are calculated at source points, the construction of matrix system is very straightforward.
- (2) Boundary conditions can be easily imposed, while RKPM requires special techniques to impose essential boundary conditions.
- (3) Employment of quadratic or higher order bases is required for convergence in RKCM, see Hu and Chen [38]. Higher order derivatives of RK shape functions are needed in construction of RKCM.
- (4) Since point collocation approach is constructed on a strong form, the issue of domain integration is not encountered.

(5) The local compact support feature of RK approximation yields a well-conditioned discrete system compared to globally approximated methods like radial basis collocation method (RBCM), which will be introduced in next section.

To illustrate point collocation approach, consider a model boundary value problem:

$$\begin{aligned} \mathbf{L}\mathbf{u} &= \mathbf{f} \quad \text{in } \Omega \\ \mathbf{B}^h\mathbf{u} &= \mathbf{h} \quad \text{on } \Gamma^h \\ \mathbf{B}^g\mathbf{u} &= \mathbf{g} \quad \text{on } \Gamma^g \end{aligned} \quad (90)$$

where $\mathbf{L}$ is the differential operator in the domain Ω ; $\mathbf{B}^h$ and $\mathbf{B}^g$ are differential operators for natural and essential boundary conditions, respectively, $\mathbf{f}$ represents the body force; $\mathbf{h}$ is the traction on Neumann boundary, and $\mathbf{g}$ is the prescribed value of $\mathbf{u}$ on the essential boundary.

Denote N_p the number of collocation points in domain, N_d the number of collocation points on the essential boundary, and N_n the number of collocation points on the natural boundary. The total number of collocation points $N_s = N_p + N_d + N_n$.

In the point collocation approach, residuals are enforced to be zero at collocation points:

$$\begin{aligned} \mathbf{L}u^h(\mathbf{x}_i) &= f(\mathbf{x}_i) \quad i = 1, 2, \dots, N_p \\ \mathbf{B}^h u^h(\mathbf{x}_i) &= h(\mathbf{x}_i) \quad i = 1, 2, \dots, N_n \\ \mathbf{B}^g u^h(\mathbf{x}_i) &= g(\mathbf{x}_i) \quad i = 1, 2, \dots, N_d \end{aligned} \quad (91)$$

In matrix form, the discretized system can be written as

$$\begin{bmatrix} \mathbf{A}_1 \\ \sqrt{\alpha^h} \mathbf{A}_2 \\ \sqrt{\alpha^g} \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ d_l \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \sqrt{\alpha^h} \mathbf{b}_2 \\ \sqrt{\alpha^g} \mathbf{b}_3 \end{bmatrix} \quad (92)$$

where $\sqrt{\alpha^h}$ and $\sqrt{\alpha^g}$ are weights taken on the natural and essential boundaries, respectively. For Poisson's problems, the optimal weights are taken as follows in order to get a balance in errors resulting from domain and boundary collocation equations (Hu, Chen, Hu [37])

$$\sqrt{\alpha^h} \approx O(1), \quad \sqrt{\alpha^g} = O(N_s) \quad (93)$$

The matrix components of Eq. (92) are

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{L}\Psi_1(\xi_1) & \mathbf{L}\Psi_2(\xi_1) & \dots & \mathbf{L}\Psi_I(\xi_1) \\ \mathbf{L}\Psi_1(\xi_2) & \mathbf{L}\Psi_2(\xi_2) & \dots & \mathbf{L}\Psi_I(\xi_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{L}\Psi_1(\xi_J) & \mathbf{L}\Psi_2(\xi_J) & \dots & \mathbf{L}\Psi_I(\xi_J) \end{bmatrix} \quad \mathbf{b}_1 = \begin{Bmatrix} f(\xi_1) \\ f(\xi_2) \\ \vdots \\ f(\xi_J) \end{Bmatrix} \quad J = 1, 2, \dots, N_p \quad (94)$$

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{B}^h\Psi_1(\xi_1) & \mathbf{B}^h\Psi_2(\xi_1) & \dots & \mathbf{B}^h\Psi_I(\xi_1) \\ \mathbf{B}^h\Psi_1(\xi_2) & \mathbf{B}^h\Psi_2(\xi_2) & \dots & \mathbf{B}^h\Psi_I(\xi_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}^h\Psi_1(\xi_J) & \mathbf{B}^h\Psi_2(\xi_J) & \dots & \mathbf{B}^h\Psi_I(\xi_J) \end{bmatrix} \quad \mathbf{b}_2 = \begin{Bmatrix} h(\xi_1) \\ h(\xi_2) \\ \vdots \\ h(\xi_J) \end{Bmatrix} \quad J = 1, 2, \dots, N_n \quad (95)$$

$$\mathbf{A}_3 = \begin{bmatrix} \mathbf{B}^g\Psi_1(\xi_1) & \mathbf{B}^g\Psi_2(\xi_1) & \dots & \mathbf{B}^g\Psi_I(\xi_1) \\ \mathbf{B}^g\Psi_1(\xi_2) & \mathbf{B}^g\Psi_2(\xi_2) & \dots & \mathbf{B}^g\Psi_I(\xi_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}^g\Psi_1(\xi_J) & \mathbf{B}^g\Psi_2(\xi_J) & \dots & \mathbf{B}^g\Psi_I(\xi_J) \end{bmatrix} \quad \mathbf{b}_3 = \begin{Bmatrix} g(\xi_1) \\ g(\xi_2) \\ \vdots \\ g(\xi_J) \end{Bmatrix} \quad J = 1, 2, \dots, N_d \quad (96)$$

Second order derivatives of RK approximation function need to be taken for the 2nd order PDEs:

$$\Psi_{I,xx}(\mathbf{x}) = C_{,xx}(\mathbf{x} - \mathbf{x}_I)\phi_a(\mathbf{x} - \mathbf{x}_I) + 2C_{,x}(\mathbf{x} - \mathbf{x}_I)\phi_{a,x}(\mathbf{x} - \mathbf{x}_I) + C(\mathbf{x} - \mathbf{x}_I)\phi_{a,xx}(\mathbf{x} - \mathbf{x}_I) \quad (97)$$

$$C_{,xx}(\mathbf{x}; \mathbf{x} - \mathbf{x}_I) = \mathbf{H}^T(0)\mathbf{M}_{,xx}^{-1}(\mathbf{x})\mathbf{H}(\mathbf{x} - \mathbf{x}_I) + 2\mathbf{H}^T(0)\mathbf{M}_{,x}^{-1}(\mathbf{x})\mathbf{H}_{,x}(\mathbf{x} - \mathbf{x}_I) + \mathbf{H}^T(0)\mathbf{M}^{-1}(\mathbf{x})\mathbf{H}_{,xx}(\mathbf{x} - \mathbf{x}_I) \quad (98)$$

4.2. Radial Basis Collocation Method (RBCM)

4.2.1. Introduction to RBF

Another class of meshfree collocation method is the radial basis collocation method (RBCM). Unlike the localized compact support used in RKCM, RBCM uses radial basis functions (RBF) as a global approximation. The commonly used RBFs are:

Multiquadrics (MQ): $g_I(\mathbf{x}) = (r_I^2 + c^2)^{n-3/2}$ (99)

$$\text{Gaussian: } g_I(x) = \begin{cases} \exp(-\frac{r_I^2}{c^2}) \\ (r_I^2 + c^2)^{n-3/2} \exp(-\frac{r_I^2}{a^2}) \end{cases} \quad (100)$$

In 2D case, $\mathbf{x} = (x_1, x_2)$, $\mathbf{x}_I = (x_{1I}, x_{2I})$, $r_I = ((x_1 - x_{1I})^2 + (x_2 - x_{2I})^2)^{1/2}$, $\mathbf{x}_I$ is the source point of RBF and constant c is the *shape parameter* of RBF, For multiquadrics RBF, if $n = 1$, it is called reciprocal MQ RBF; and if $n = 2$, it is called linear RBF.

A smooth function $u(\mathbf{x})$ can be approximated by RBF:

$$u^h(\mathbf{x}) = \sum_{I=1}^{N_s} g_I(\mathbf{x}) d_I \quad (101)$$

where d_I is the coefficient.

RBFs are infinitely differentiable, which allows easy imposition of radial basis approximation to strong form collocation:

$$\frac{d^n u^h(\mathbf{x})}{dx^n} = \sum_{I=1}^{N_s} \frac{d^n g_I(\mathbf{x})}{dx^n} d_I \quad (102)$$

Madych [56] proved that there exists an exponential convergence rate of RBF:

$$|u(\mathbf{x}) - u^h(\mathbf{x})| \approx O(\eta^{c/h}) \quad (103)$$

where $0 < \eta < 1$ is a real number, h is the radial distance

$$h := h(\Omega, \mathbf{S}) = \sup_{\mathbf{x} \in \Omega} \min_{\mathbf{x}_I \in \mathbf{S}} ((x_1 - x_{1I})^2 + (x_2 - x_{2I})^2)^{1/2} \quad (104)$$

where $\mathbf{S}$ is a set containing all source points.

4.2.2. Discretized System of Radial Basis Point Collocation

To illustrate the discretized form of RBCM, consider the same model problem in(105). Similar to RKCM, the discretized form of RBCM can be expressed as

$$\begin{bmatrix} \mathbf{A}_1 \\ \sqrt{\alpha^h} \mathbf{A}_2 \\ \sqrt{\alpha^g} \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ d_l \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \sqrt{\alpha^h} \mathbf{b}_2 \\ \sqrt{\alpha^g} \mathbf{b}_3 \end{bmatrix} \quad (106)$$

The weights in Eq. (106) are the same as those taken in Eq. (93). The components of the matrix in can be expressed as follows

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{L}g_1(\xi_1) & \mathbf{L}g_2(\xi_1) & \dots & \mathbf{L}g_l(\xi_1) \\ \mathbf{L}g_1(\xi_2) & \mathbf{L}g_2(\xi_2) & \dots & \mathbf{L}g_l(\xi_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{L}g_1(\xi_J) & \mathbf{L}g_2(\xi_J) & \dots & \mathbf{L}g_l(\xi_J) \end{bmatrix} \quad \mathbf{b}_1 = \begin{Bmatrix} f(\xi_1) \\ f(\xi_2) \\ \vdots \\ f(\xi_J) \end{Bmatrix} \quad J = 1, 2, \dots, N_p \quad (107)$$

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{B}^h g_1(\xi_1) & \mathbf{B}^h g_2(\xi_1) & \dots & \mathbf{B}^h g_l(\xi_1) \\ \mathbf{B}^h g_1(\xi_2) & \mathbf{B}^h g_2(\xi_2) & \dots & \mathbf{B}^h g_l(\xi_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}^h g_1(\xi_J) & \mathbf{B}^h g_2(\xi_J) & \dots & \mathbf{B}^h g_l(\xi_J) \end{bmatrix} \quad \mathbf{b}_2 = \begin{Bmatrix} h(\xi_1) \\ h(\xi_2) \\ \vdots \\ h(\xi_J) \end{Bmatrix} \quad J = 1, 2, \dots, N_n \quad (108)$$

$$\mathbf{A}_3 = \begin{bmatrix} \mathbf{B}^g g_1(\xi_1) & \mathbf{B}^g g_2(\xi_1) & \dots & \mathbf{B}^g g_l(\xi_1) \\ \mathbf{B}^g g_1(\xi_2) & \mathbf{B}^g g_2(\xi_2) & \dots & \mathbf{B}^g g_l(\xi_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}^g g_1(\xi_J) & \mathbf{B}^g g_2(\xi_J) & \dots & \mathbf{B}^g g_l(\xi_J) \end{bmatrix} \quad \mathbf{b}_3 = \begin{Bmatrix} g(\xi_1) \\ g(\xi_2) \\ \vdots \\ g(\xi_J) \end{Bmatrix} \quad J = 1, 2, \dots, N_d \quad (109)$$

5 Comparison of Accuracy and Convergence Among RKCM, RBCM and RKPM

In this chapter, RKCM, RBCM and RKPM are used to solve 1-D and 2-D model problems. Various domain and quadrature integration techniques are employed in RKPM, while point collocation approaches with different number and allocation of collocation points are considered in KRCM and RBCM. Solution accuracy and convergence of these three methods are compared. Unless specified, quadratic bases with a normalized compact support size of $a = 3.01$ is used in RKCM, linear bases with normalized support of 2.01 is employed for RKCM, and linear MQ RBF in Eq. (99) is utilized in RBCM with a shape parameter $c = 1.6$. The use of quadratic bases in RKCM is due to its minimum polynomial order requirement as proved by Hu and Chen et al. [38].

5.1. Patch Tests of RKCM in 1-Dimension and 2-Dimensions

Since RKCM is able to provide n^{th} order completeness by satisfying the reproducing conditions, solutions of n^{th} order polynomial in boundary value problems (BVPs) can be exactly obtained if collocation points are properly introduced. Consider the boundary value problem

$$\begin{aligned} u_{,xx} &= 2, \quad x \in (1, 0) \\ u(0) &= 1, \quad u(1) = 4 \end{aligned} \tag{110}$$

The exact solution of problem (110) is $u = x^2 + 2x + 1$.

In RKCM, quadratic bases are employed to satisfy 2^{nd} order completeness needed in the exact solution, ten non-uniformly source points are distributed, and the

compact support size used is $4.00h$, where h is the average nodal distance of the source points. The error of displacement and its gradient in the domain is shown in Figure 5.1. The L_2 error norm calculated is $3.538\text{E-}14$ and the H^1 error norm is $3.630\text{E-}13$, therefore the 1D patch test is passed.

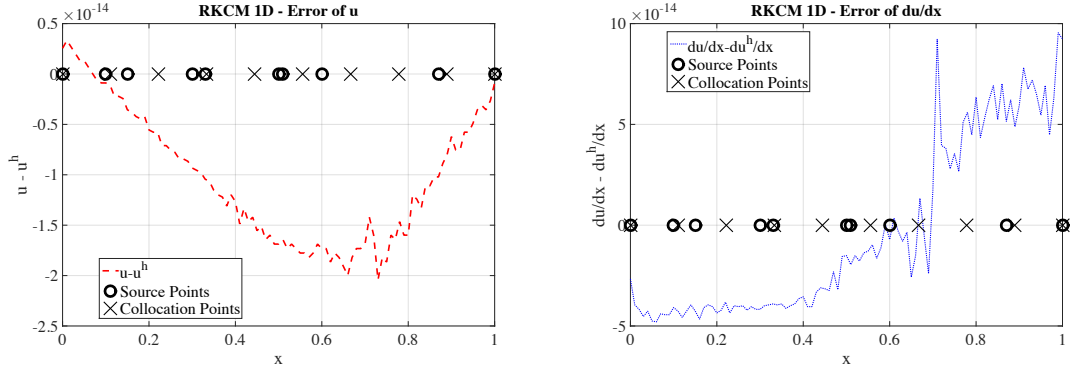


Figure 5.1 Errors for quadratic patch test with 1-D RKCM

Now consider another example to test the cubic basis

$$\begin{aligned} u_{,xx} &= 4x + 4 \quad x \in (0,1) \\ u(0) &= 3 \\ u(1) &= 11 \end{aligned} \tag{111}$$

of which the exact solution is $u = x^3 + 2x^2 + 5x + 3$. The error distributions of the numerical solution and its gradient shown in Figure 5.2 reach the machine error, which indicates that the RKCM with cubic basis can reproduce 3rd order exactness in BVP. The above two numerical examples demonstrate that RKCM possesses the correct consistency in 1D case.

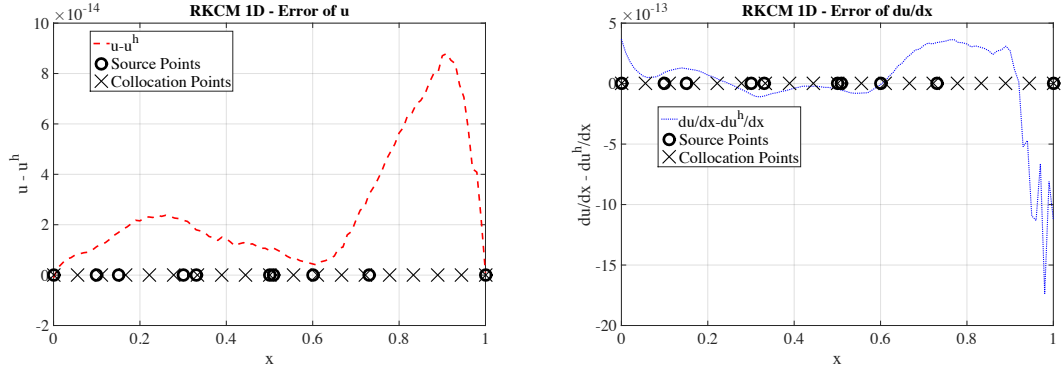


Figure 5.2 Errors for cubic patch test with 1-D RKCM

Now consider the following 2-D boundary value problem

$$u_{,xx} + u_{,yy} = 14 \quad \text{in } \Omega = (0,1) \times (0,1) \quad (112)$$

whose essential boundary conditions are prescribed as

$$\begin{aligned} u(x=0) &= 5y^2 + 4y + 3 & u(x=1) &= 5y^2 + 5y + 8 \\ u(y=0) &= 2x^2 + 3x + 3 & u(y=1) &= 2x^2 + 4x + 12 \end{aligned} \quad (113)$$

The exact solution is $u(x,y) = 2x^2 + 3x + 5y^2 + 4y + xy + 3$. Non-uniform discretization of the domain is introduced as shown in Fig. 5.3(a). The error distributions of the numerical solution and its gradients plotted in Figure 5.3(b) show that the 2-D patch test is also passed. The consistency of RKCM is validated and the convergence will be tested in the following section.

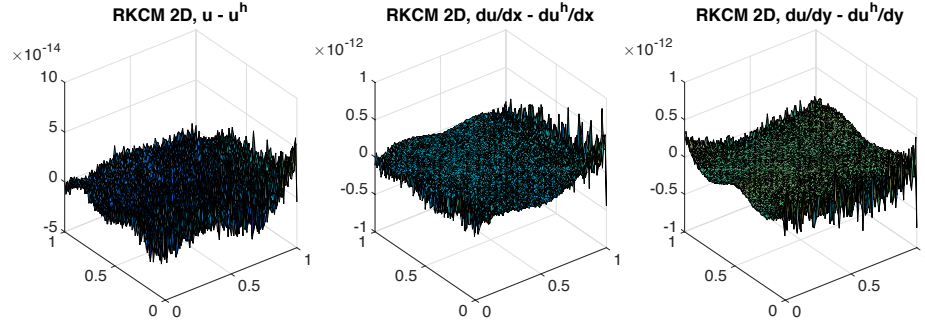


Figure 5.3 (a) non-uniform source point and collocation point distribution, (b) Errors for quadratic patch test with 2-D RKCM

5.2. Comparison of RKCM and RBCM in One Dimensional Poisson Problem

One dimensional boundary value problem is considered in this section

$$u_{,xx} = e^x \quad x \in (0,1) \quad (114)$$

with pure essential boundary conditions

$$u(0) = 1 \text{ and } u(1) = e \quad (115)$$

The exact solution is $u(x) = e^x$.

5.2.1. Effect of Number of Collocation Points on Convergence

First, the effect of number of collocation points on the accuracy and convergence of RKCM and RBCM is investigated by solving the 1D problem in Eq. (114). The domain is discretized using 6, 10, 14 and 18 uniformly distributed source points (N_s), respectively. For each discretization, the number of collocation points N_c is increased proportionally up to eight times of N_s . For both collocation methods, the weight $\sqrt{\alpha^g}$ of the system in Eq. (80) applied on the essential boundary is set as $\sqrt{\alpha^g} = N_x$ according to Hu and Chen [37]. Figure 5.4 and Figure 5.5 show the error of solutions in L_2 and H^1 norms, respectively, obtained from RKCM and RBCM with various number of collocation points.

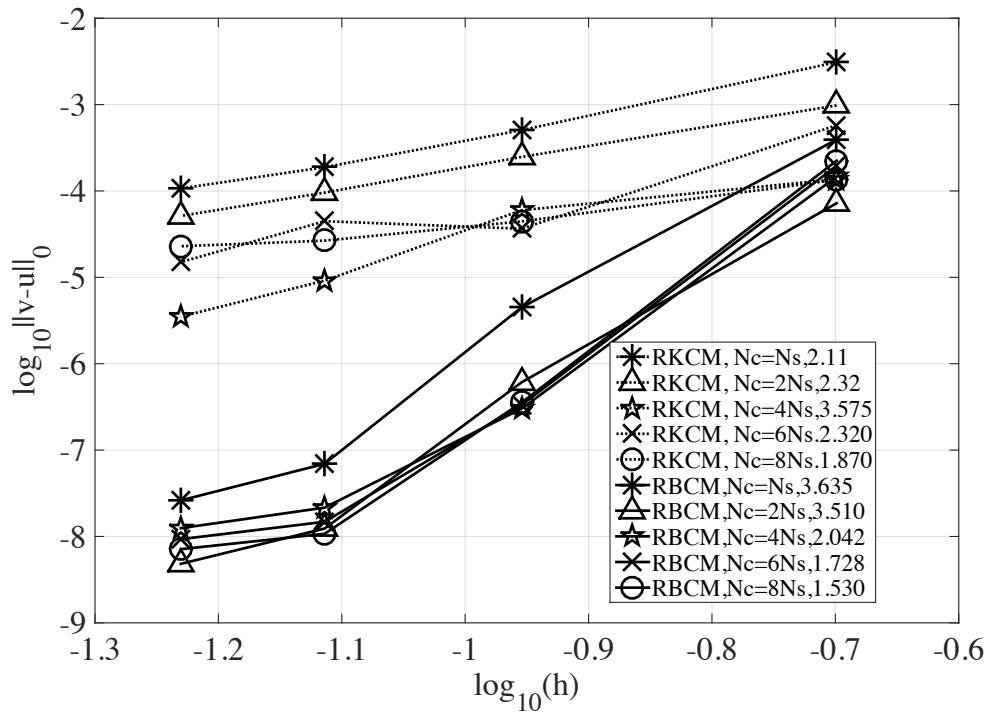


Figure 5.4 Convergence plot of error in L_2 norm of RKCM and RBCM in 1D, varying N_c , legend showing asymptotic rate of convergence

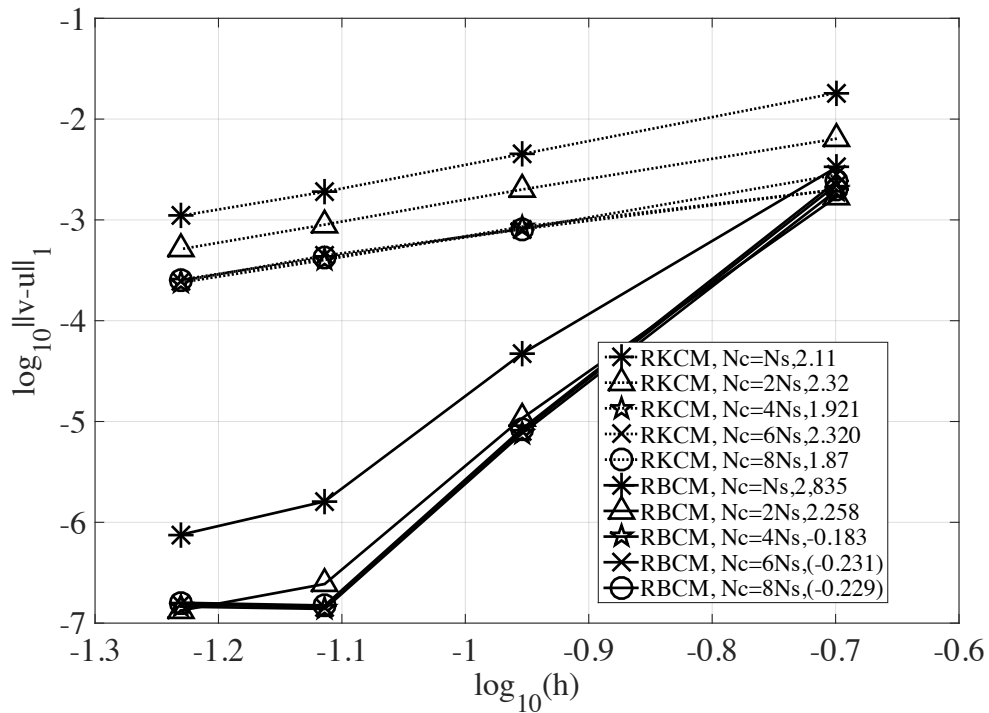


Figure 5.5 Convergence plot of error in H^1 norm of RKCM and RBCM in 1D, varying N_c , legend showing asymptotic rate of convergence

The figures show that RKCM gives the optimal convergence rate in H^1 norm, but not in L_2 norm. Under the same discretization, RBCM show better accuracy than RKCM, and provides nearly exponential convergence in the L_2 and H^1 norms. However, under very fine discretization, the convergence rate of RBCM reduces significantly, which is caused by the ill-conditioning of the linear system. Regarding to RKCM, an over-determined system ($N_c > N_s$) can provide better accuracy, but the result does not necessarily become better with the increase of collocation points. In this example, RKCM shows the best accuracy in both norms when $N_c = 4N_s$. Same observation is found for RBCM, where an over-determined system ($N_c > N_s$) give better accuracy than determined system ($N_c = N_s$), but keep increasing the number of collocation points does not yield better solution accuracy.

5.2.2. Error distribution in 1D model problem

The results of numerical errors in model problem in Eq. (114), $u - u^h$ and $du/dx - du^h/dx$, using RKCM are shown from Figure 5.6 to Figure 5.11. It can be observed that adding more collocation points up to four times the source points makes the solution errors reduced. RKCM gives accurate displacement solution on boundary points in 1D case, while relative large error in the domain. The gradient error, $du/dx - du^h/dx$, is large on boundary when $N_c = N_s$ and can be significantly reduced by adding collocation points.

Figure 5.12 to Figure 5.17 show the errors of u and its gradients using RBCM. In RBCM, increasing N_c does not appear to yield reduction of errors. Under certain situation, increasing N_c may cause increase of error. It is also observed that RKCM solution is in general more accurate than RBCM solution except for the case $N_c = N_s$

in this 1D Poisson problem, which is not the case in 2D problem to be discussed below.

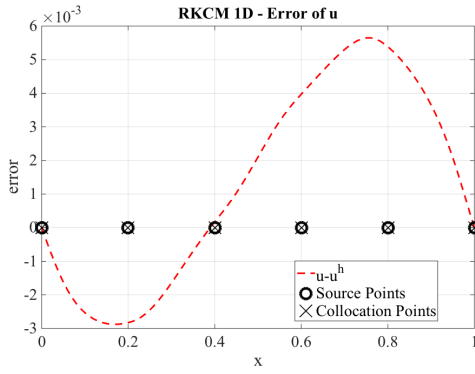


Figure 5.6 RKCM 1D, error of u , 6 nodes, $N_c = N_s$

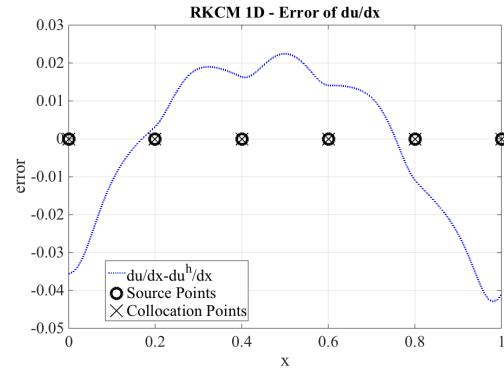


Figure 5.7 RKCM 1D, error of du/dx , $N_c = N_s$

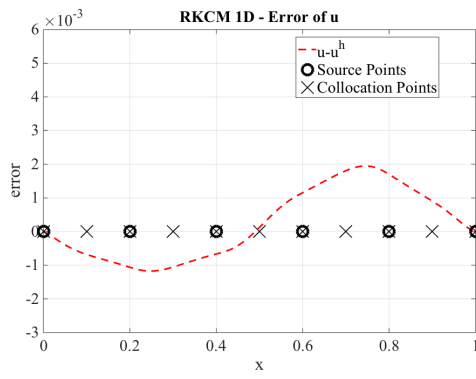


Figure 5.8 RKCM 1D, error of u , 6 nodes $N_c = 2N_s$

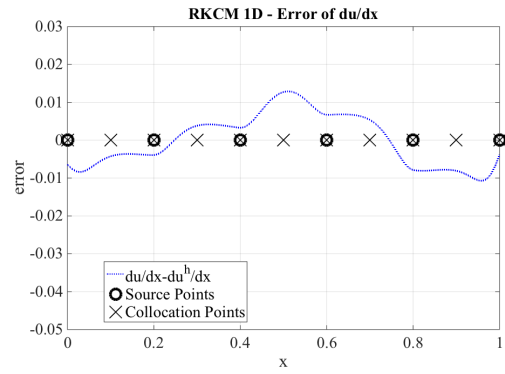


Figure 5.9 RKCM 1D, error of du/dx , $N_c = 2N_s$

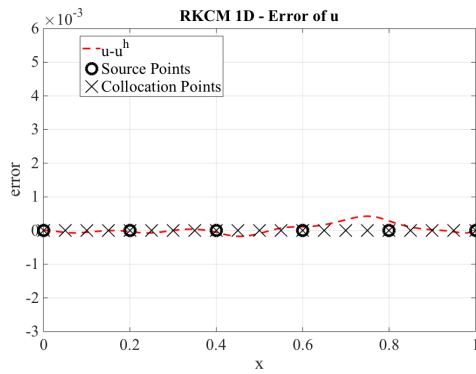


Figure 5.10 RKCM 1D, error of u , 6 nodes $N_c = 4N_s$

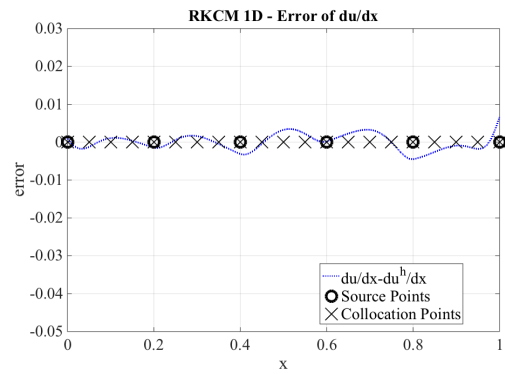


Figure 5.11 RKCM 1D, error of du/dx , $N_c = 4N_s$

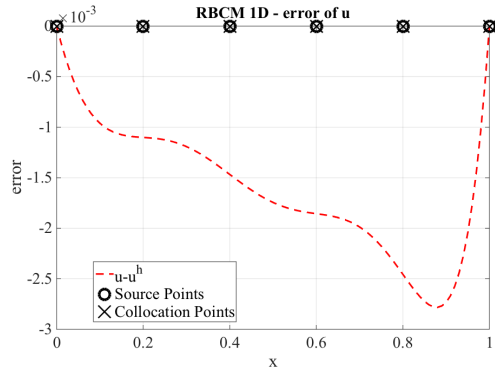


Figure 5.12 RBCM 1D, error of u , 6 nodes $N_c = N_s$

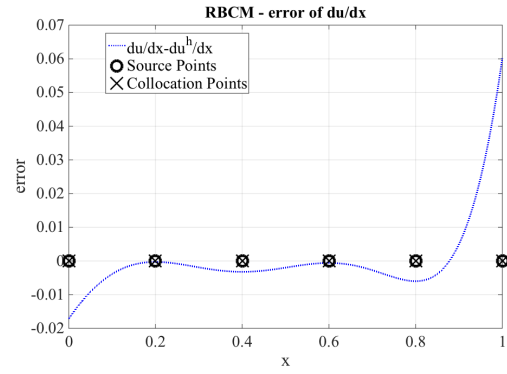


Figure 5.13 RBCM 1D, error of du/dx , $N_c = N_s$

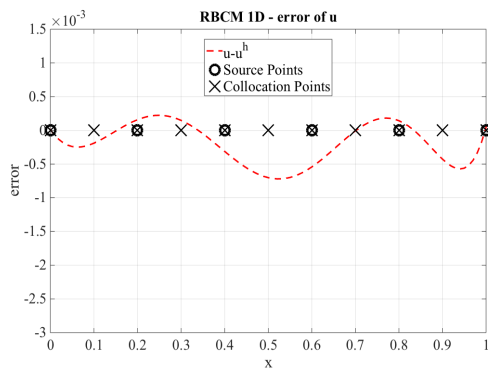


Figure 5.14 RBCM 1D, error of u , 6 nodes $N_c = 2N_s$

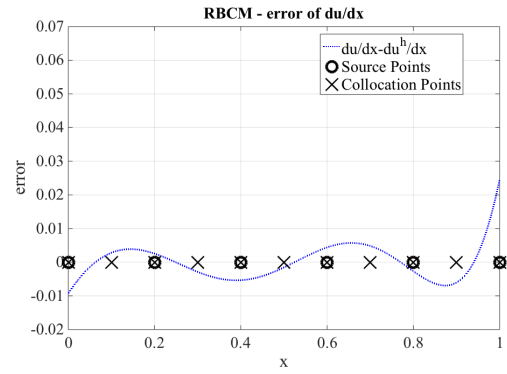


Figure 5.15 RBCM 1D, error of du/dx , $N_c = 2N_s$

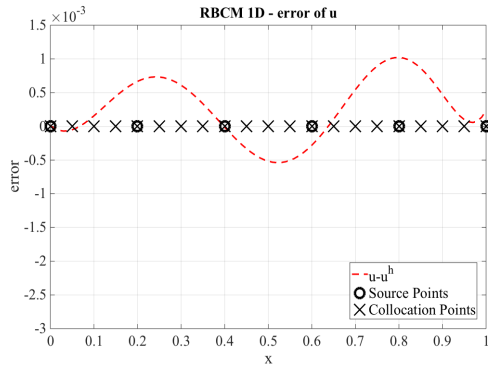


Figure 5.16 RBCM 1D, error of u , 6 nodes $N_c = 4N_s$

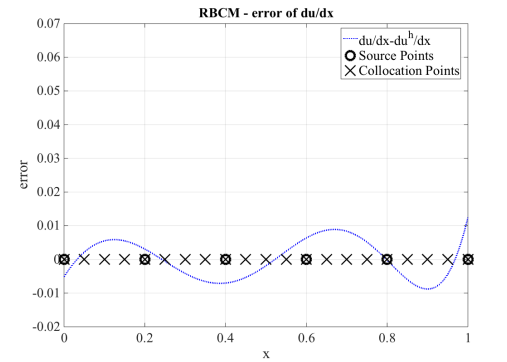


Figure 5.17 RBCM 1D, error of du/dx , $N_c = 4N_s$

5.3. RKCM and RBCM Comparison in Two Dimensional Poisson Problem

Now consider the following two dimensional Poisson problem:

$$u_{,xx} + u_{,yy} = (x^2 + y^2)e^{xy} \quad \text{in } (0,0) \times (1,1) \quad (116)$$

and the pure essential boundary conditions are applied as follows

$$\begin{aligned} u(0,y) &= 1 & u(1,y) &= e^y \\ u(x,0) &= 1 & u(x,1) &= e^x \end{aligned} \quad (117)$$

The exact solution is $u(x,y) = e^{xy}$.

In this two-dimensional analysis, we use N_c and N_s to denote the number of collocation points and source points in each direction, respectively. If $N_c = \alpha N_s$, it means that the number of collocation points is α^2 times of the number of source points in the entire 2D domain.

5.3.1. Convergence Comparison in 2-D Poisson Problem

Convergence studies in L_2 norm and H^1 norm of RBCM with different number of collocation points are presented separately in Figure 5.18 and Figure 5.19. The figures show that in the relatively coarse discretization, the accuracy of RBCM can be improved by moderately adding collocation points. However it becomes more susceptible to an ill-conditioned system as the discretization is refined. The numerical solution shows exponential convergence in the range of coarse discretization, and with significantly slower convergence, as the model is refined.

Same testes are performed for RKCM and the results are shown in Figure 5.20 and Figure 5.21. Compared to the solutions obtained from RBCM, RKCM provides a more stable convergence for all $N_c > N_s$ cases irrespective of the refinement of the source points although poor accuracy is observed when $N_c = N_s$ is used. Optimal

convergence rate is reached even though the convergence rate is not as high as that of RBCM in the range of coarse discretization, in particular a super convergence in L_2 and H^1 norms is observed when $N_c = 8N_s$. Generally, increasing the number of collocation improves the accuracy of RKCM, and $N_c = 2 \sim 4N_s$ is a good choice in balancing accuracy and efficiency.

This example shows that the ratio of the number of collocation points to the number of source points has profound influence on solution convergence and the accuracy in RKCM. For RBCM, the numerical errors caused by ill-conditioned matrix become dominant beyond certain discretization refinement. While RBCM offers better accuracy than RKCM in coarse discretization, the situation could reverse as the simulation model size increases. Overall, RKCM offers more stable convergence properties regardless the dimension of the approximation space.

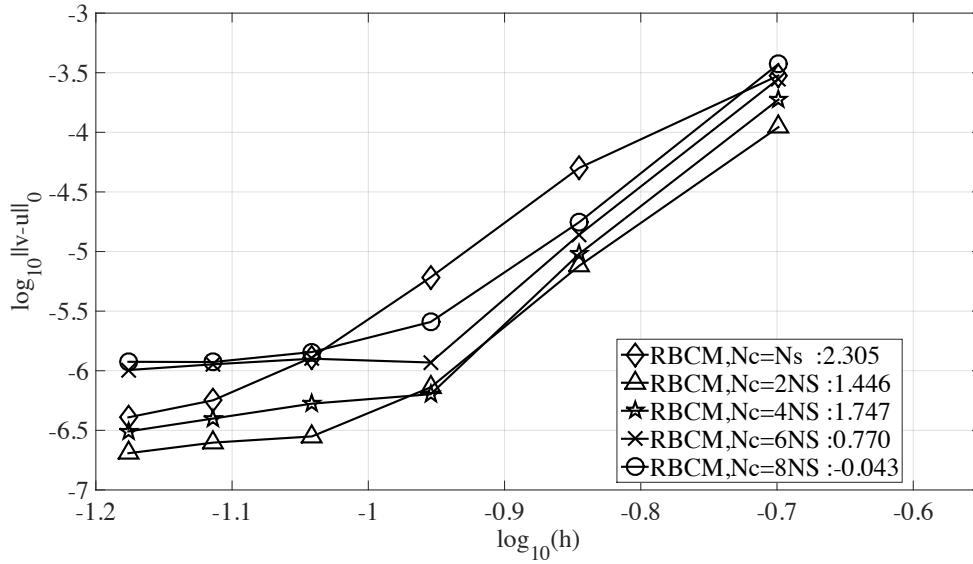


Figure 5.18 Convergence plot of error in L_2 norm of RBCM with $c=1.6$ for 2D Poisson's problem with different N_c to N_s ratios

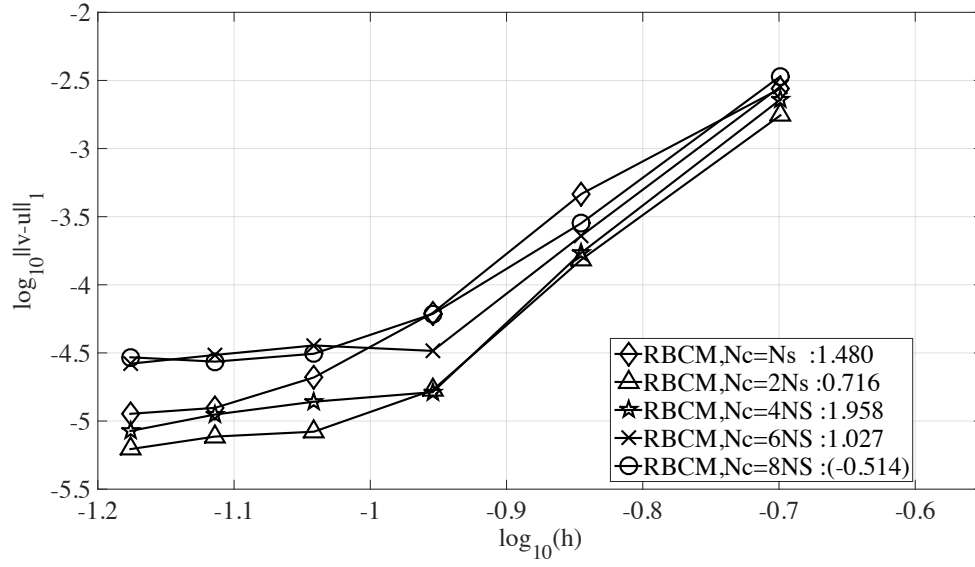


Figure 5.19 Convergence plot of error in H^1 norm of RBCM with $c=1.6$ for 2D Poisson's problem with different N_c to N_s ratios

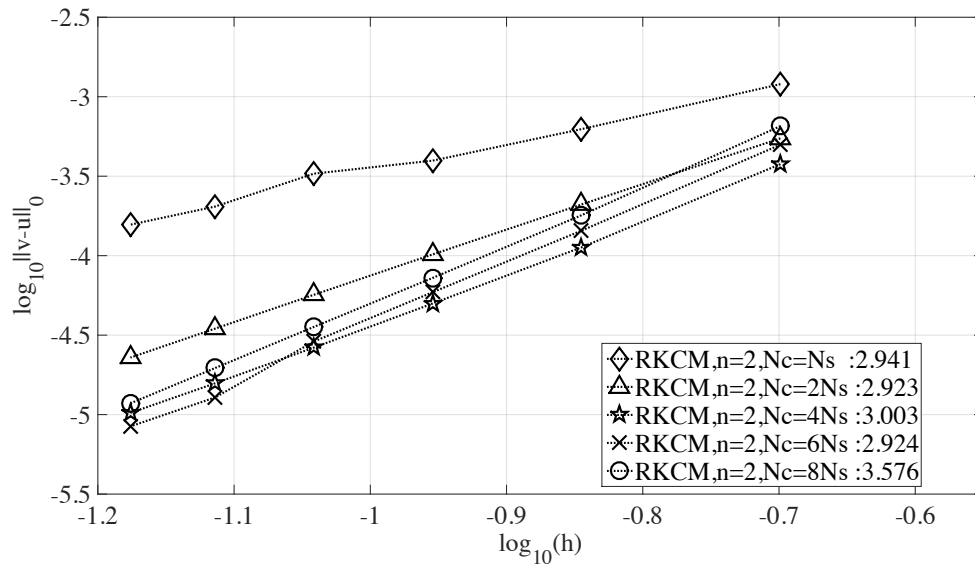


Figure 5.20 Convergence plot of error in L_2 norm of RKCM with quadratic basis for 2D Poisson's problem with different N_c to N_s ratios

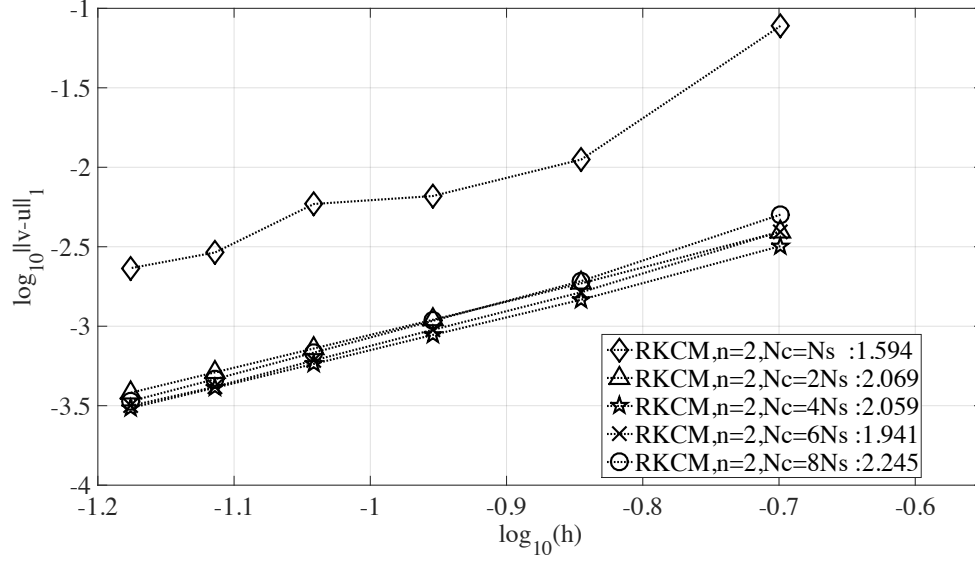


Figure 5.21 Convergence plot of error in H^1 norm of RKCM with quadratic basis for 2D Poisson's problem with different N_c to N_s ratios

5.4. Comparison between Strong Form Collocation Methods (RKCM and RBCM) with Galerkin meshfree Method (RKPM)

In this section, two strong form collocation methods (RKCM and RBCM) are compared with RKPM with different integration techniques by solving the 2-D model problem presented in Section 5.3. Discretization started with $N_c = N_s$ and is proportionally increased to $N_c = 8N_s$ in this study. Linear and quadratic bases are used in RKPM while quadratic bases are used in RKCM due to its regularity requirement. Linear multiquadrics radial basis functions are used in RBCM.

In Figure 5.22 and Figure 5.23, the solutions of RKCM and RBCM collocation methods with $N_c = N_s$ are compared with RKPM using linear bases integrated by DNI, SCNI and quadratic bases integrated by VCI. RKPM gives steady and optimal convergence rate in L_2 norm (1.981 and 1.984, respectively) with both DNI and SCNI, super-convergence and accuracy are obtained with VCI; while

RKCM also achieves optimal convergence and behaves more accurately than RKPM with linear bases when $N_s \geq 10^2$ but less accurate than RKPM with quadratic bases and integrated by VCI. The accuracy of solution obtained by RBCM surpasses all other solutions in this test problem in the discretization with relatively small degrees of freedom. In Figure 5.24 and Figure 5.25, the results obtained from RKCM and RBCM with $N_c = 2N_s$ in each dimension and RKPM with quadratic bases and integrated with VCI 3-point Gauss quadrature are compared. RKCM ($N_c = N_s$) provides less accuracy than RKPM (VC-GI 3-pt) in L_2 norm but very similar accuracy in H^1 norm, within both methods quadratic bases are used. Similar to the examples discussed earlier, increasing the number of collocation points by a factor of 2 gives a significant enhancement in the accuracy of RKCM and RBCM. It is also noted that RKPM with DNI actually work well in problems with uniform discretization as has been reported in the literature (Chen, Hillman, and Ruter, 2013), whereas they show poor accuracy of these approaches in non-uniform discretizations.

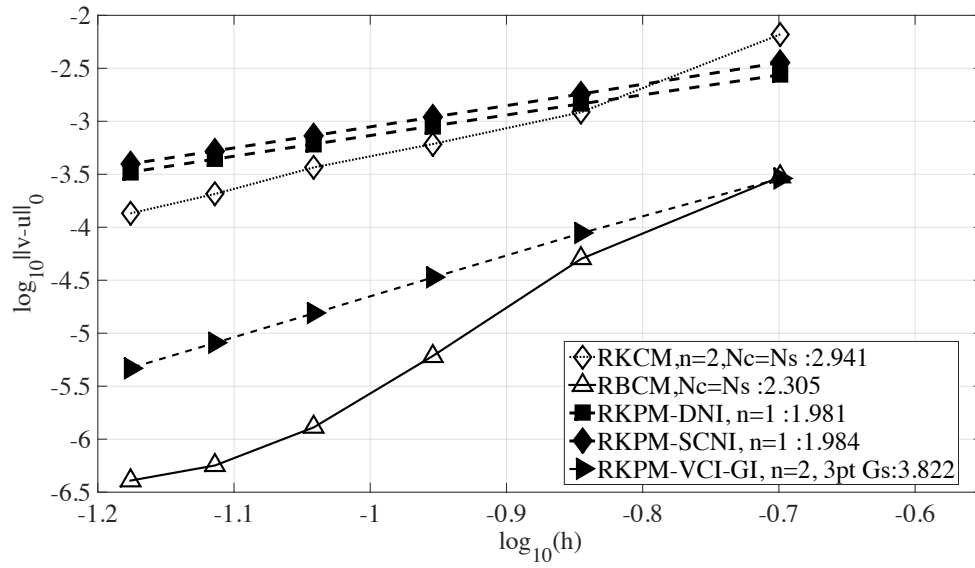


Figure 5.22 Convergence plot of error in L_2 norm for 2D Poisson's problem, comparing RKCM and RBCM ($N_c = N_s$) and RKPM DNI, SCNI and VCI

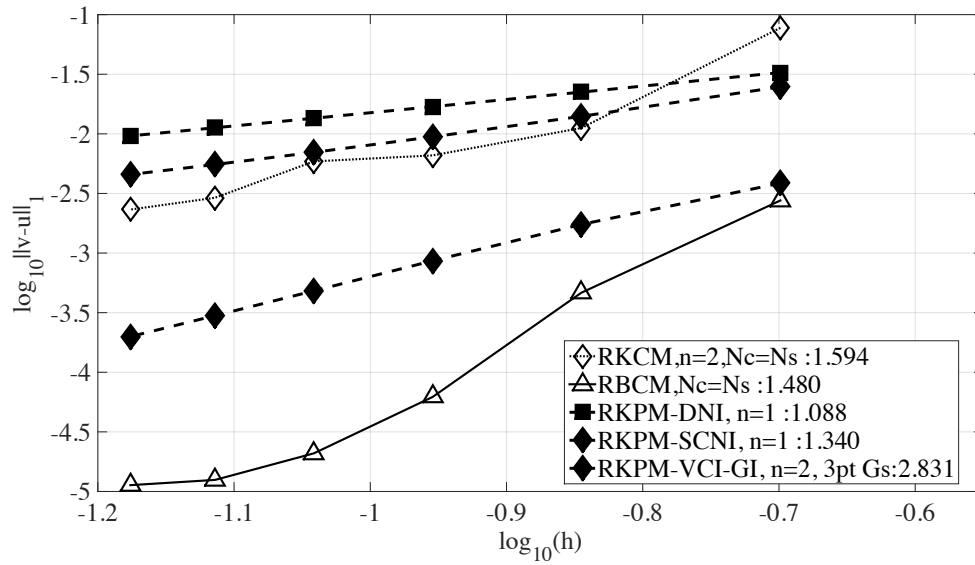


Figure 5.23 Convergence plot of error in H^1 norm of 2D Poisson's problem, comparing RKCM and RBCM ($N_c = N_s$) and RKPM with DNI, SCNI and VCI

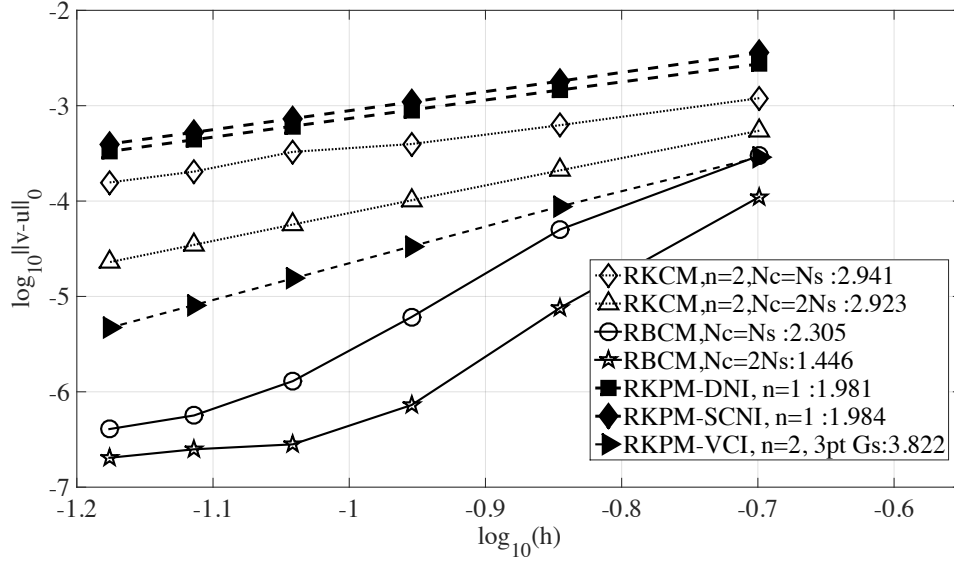


Figure 5.24 Convergence plot of error in L_2 norm for 2D Poisson's problem, comparing RKCM and RBCM with $N_c = N_s$ and $N_c = 2N_s$ and RKPM with DNI, SCNI and VCI

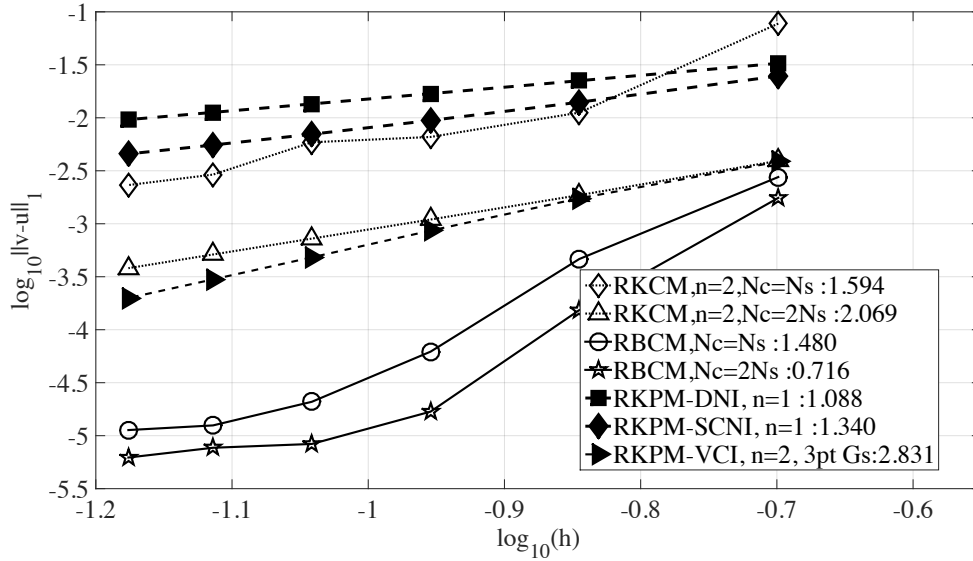


Figure 5.25 Convergence plot of error in H^1 norm for 2D Poisson's problem, comparing RKCM and RBCM with $N_c = N_s$ and $N_c = 2N_s$ and RKPM with DNI, SCNI and VCI

Further increasing the collocation points improves the accuracy of RKCM but not for RBCM as shown in Figure 5.26 and Figure 5.27. The same phenomena were observed in the examples shown in the last section. With $N_c = 6N_s$, RKCM gives

similar accuracy to VCI with quadratic bases in both L_2 norm and H^1 norm. In terms of convergence, RKCM gives stable and optimal convergence while RKPM with VCI offers super-convergence. According to the result, RKCM can be seen as a alternative of VCI since it avoids the complexity of VCI but offers similar accuracy and convergence property.

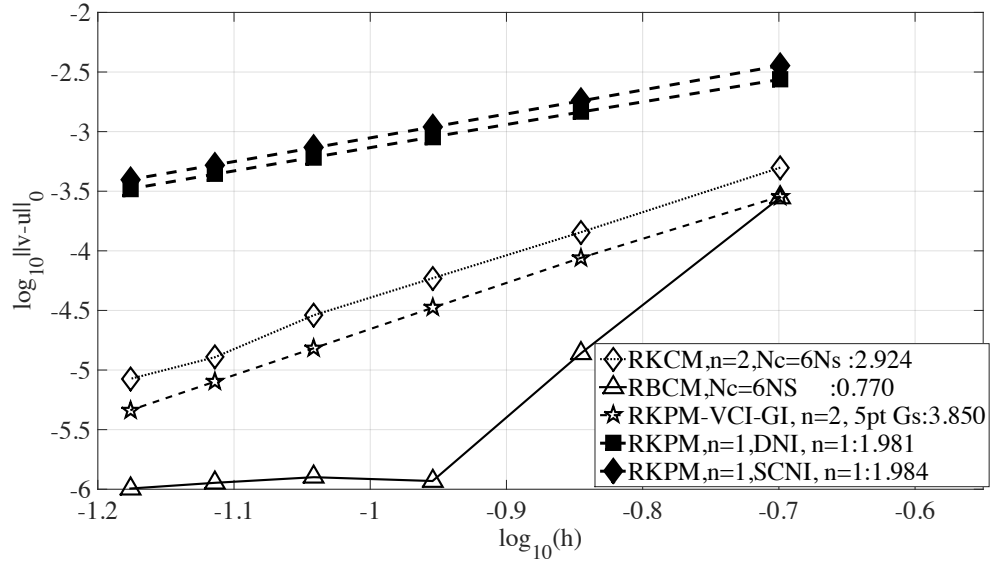


Figure 5.26 Convergence plot of error in L_2 norm for 2D Poisson's problem, comparing RKCM, RBCM with $N_c = 6N_s$ and RKPM with DNI , SCNI and VCI

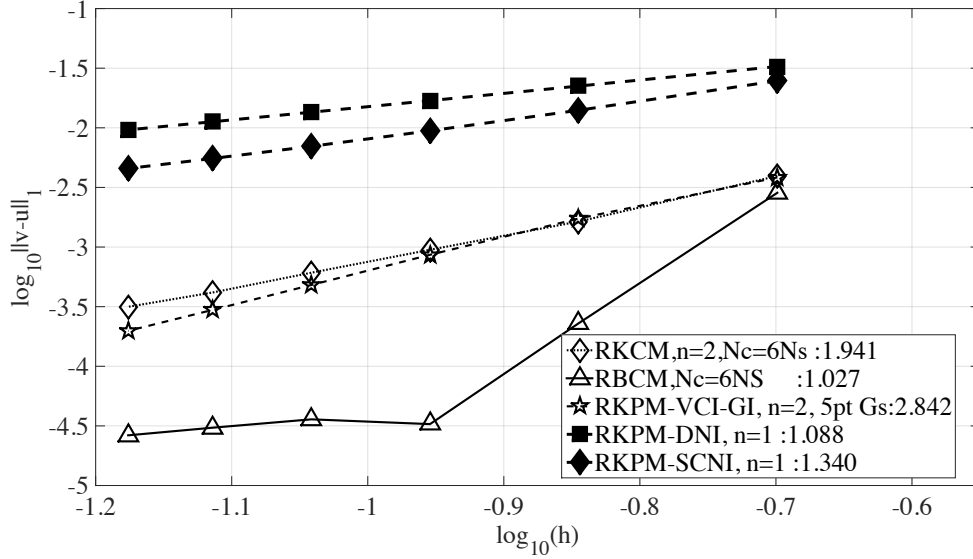


Figure 5.27 Convergence plot of error in H^1 norm for 2D Poisson's problem, comparing RKCM, RBCM with $N_c = 6N_s$ and RKPM with DNI , SCNI and VCI

5.4.1. Special Collocation Points Arrangements for Enhanced Computational Efficiency

The RKCM test results from the previous sections indicate that a continuous increase in the number of collocation points does not necessarily enhance solution accuracy, and an optimum arrangement of collocation points needs further investigation. When the collocation points exactly coincide with the source points, collocation methods yield a determinate system. The solutions in this situation show low accuracy with sub-optimal convergence rate. While uniformly adding collocation points can significantly improve the solution accuracy, it is computationally expensive. Therefore, strategies of adaptively adding collocation points instead of uniformly increasing collocation points are of great interest. Figure 5.28 shows different ways of adding collocation points based on the base case with $N_c = N_s$. The

convergence of error in L_2 and H^1 norms using different strategies of adding collocation points are displayed in Figure 5.29 and Figure 5.30, along with RKPM results using nodal integration.

The results show that by simply adding 4 collocation points in the domain, the solution shows a stable convergence rate in model refinement. Adding 4 collocation points can also achieve faster convergence and better accuracy compared with RKPM integrated with DNI and SCNI with very little additional CPU. By adding a ring of collocation points based on $N_c = N_s$, RKCM can provide accurate and stable results close to $N_c = 2N_s$ case, with optimal convergence rate and lower computational cost. The distribution of errors in u and its gradients obtained from RKCM with these four arrangements of collocation points shown in Figure 5.28 are plotted in Figure 5.31 to Figure 5.36. The results in these figures show that adding 4 corner collocation points or a ring of collocation points near the boundary can significantly suppress the oscillation of errors in the domain.

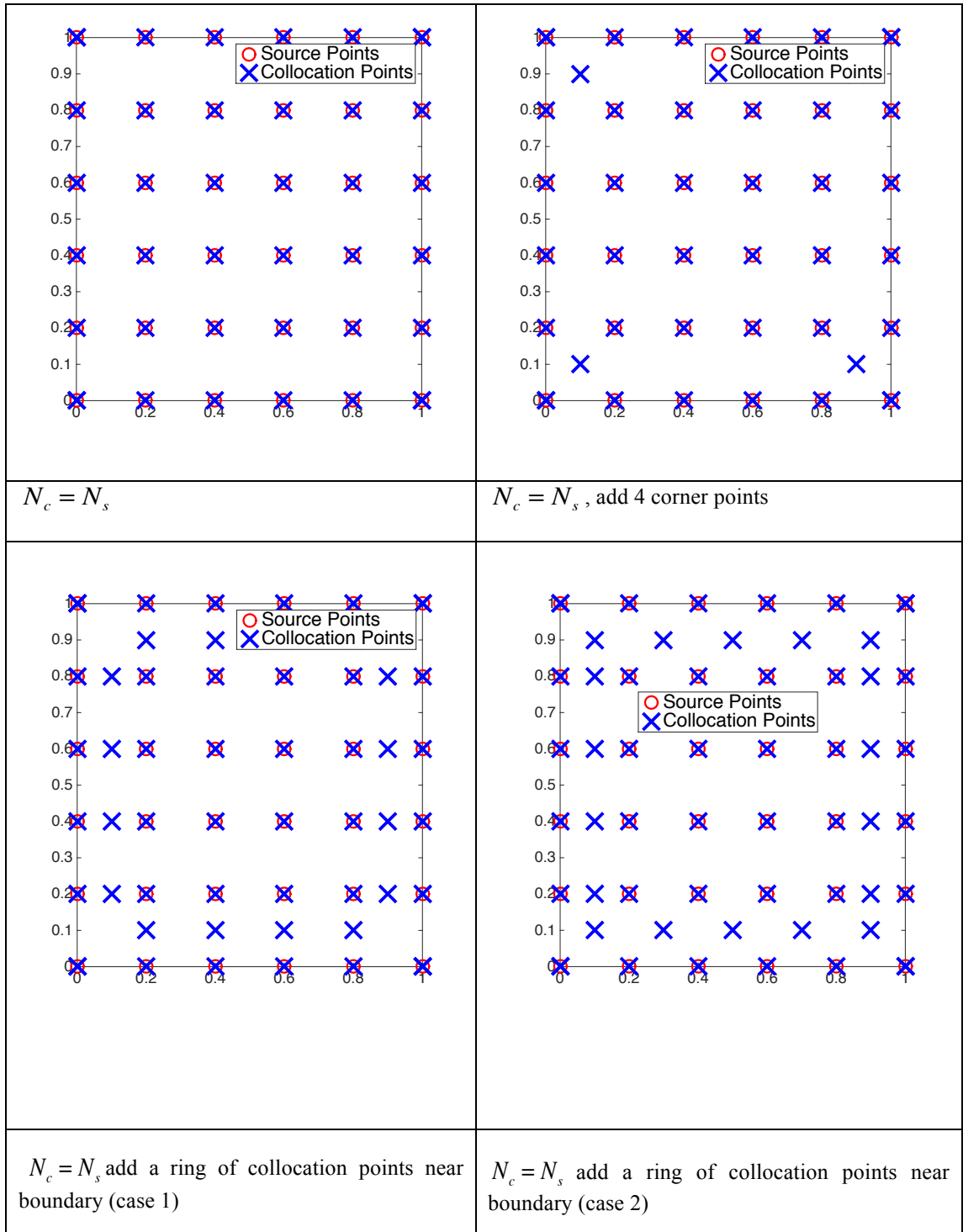


Figure 5.28 Special ways of allocating collocation points based on $N_c = N_s$ system, 6x6 source points

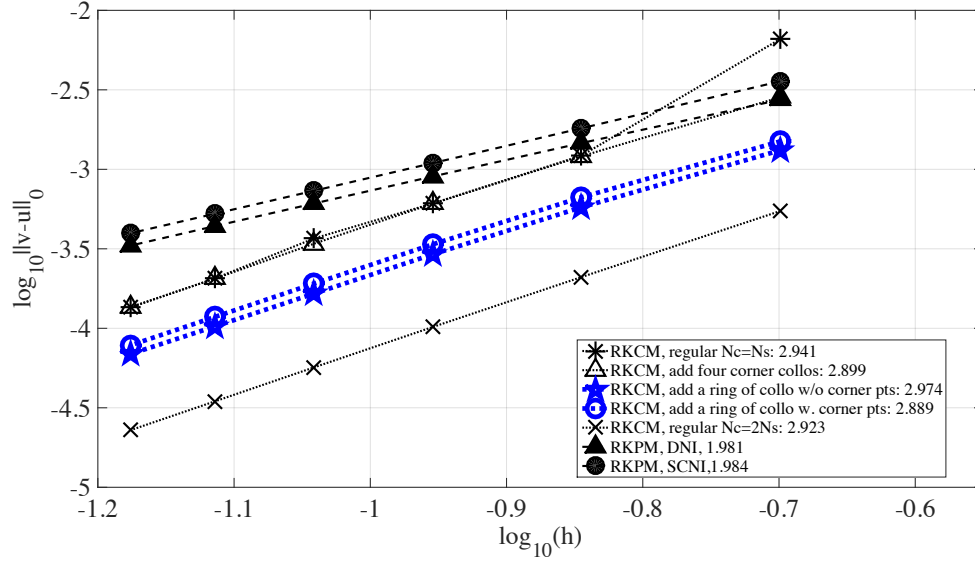


Figure 5.29 Convergence plot of error in L_2 norm for 2D Poisson's problem, comparing RKCM with special techniques of adding collocation points, compare with RKPM using DNI and SCNI

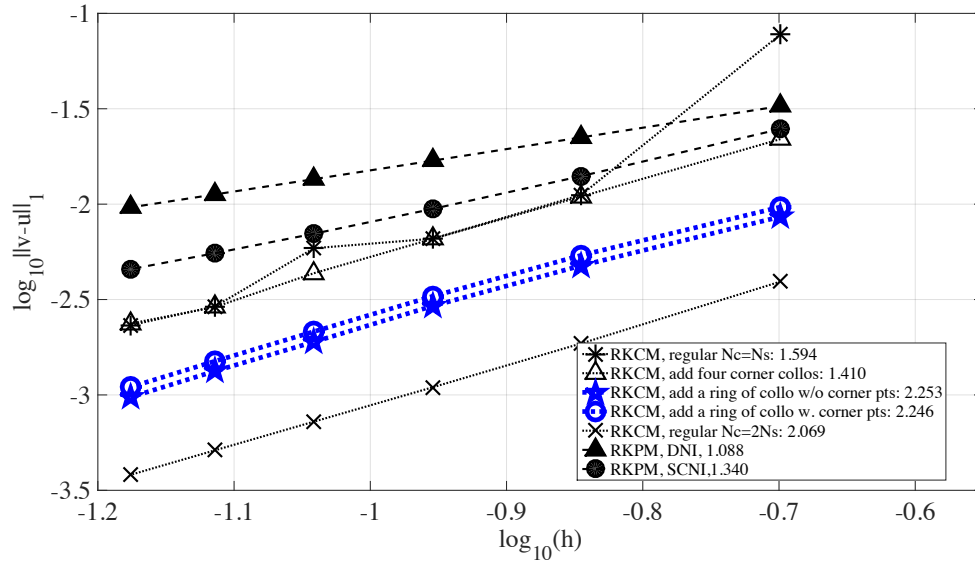


Figure 5.30 Convergence plot of error in H^1 norm for 2D Poisson's problem, comparing RKCM with special techniques of adding collocation points, compare with RKPM with linear basis and iterated using DNI and SCNI

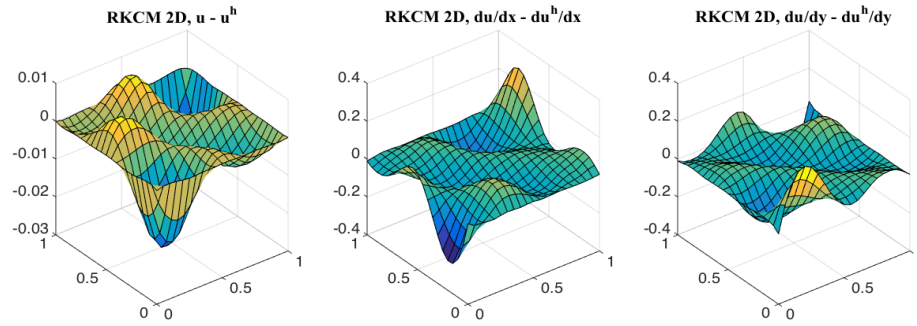


Figure 5.31 RKCM 2D. Error of u and its gradients for 2D Poisson's problem, 6x6 source points, uniform $N_c = N_s$

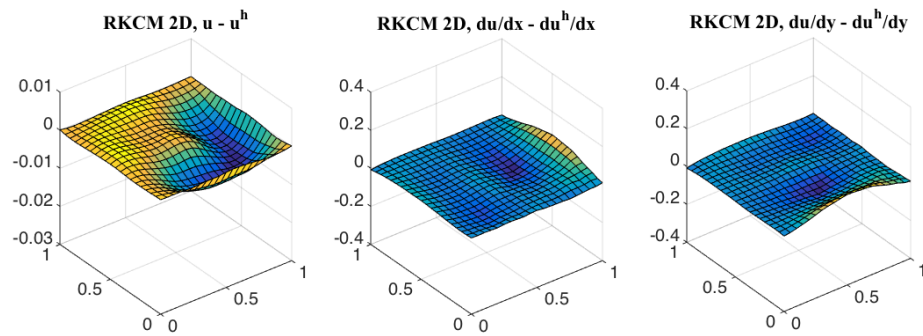


Figure 5.32 RKCM 2D. Error of u and its gradients for 2D Poisson's problem. 6x6 source points, $N_c = N_s$ plus four corner collocation points

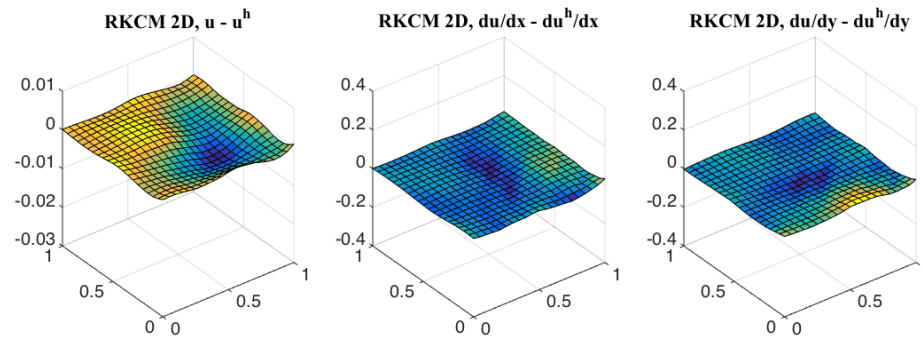


Figure 5.33 RKCM 2D. Error of u and its gradients for 2D Poisson's problem. 6x6 source points, $N_c = N_s$ plus a ring of collocation points

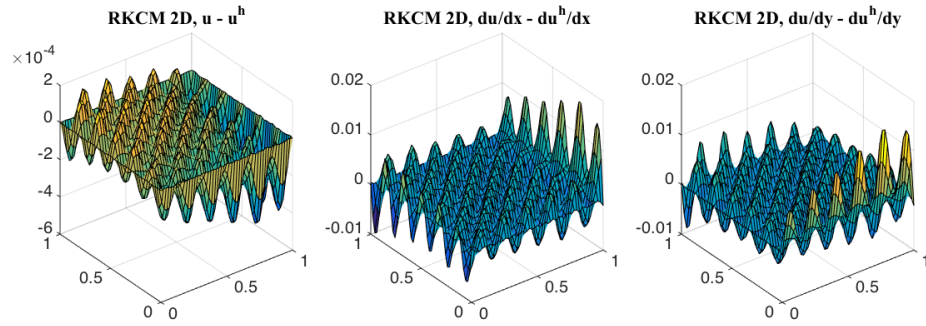


Figure 5.34 RKCM 2D. Error of u and its gradients for 2D Poisson's problem, 18x18 source points, regular $N_c = N_s$

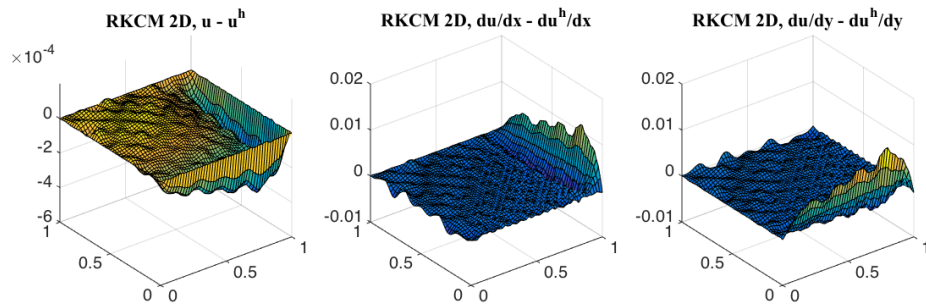


Figure 5.35 RKCM 2D. Error of u and its gradients for 2D Poisson's problem. 18x18 source points, $N_c = N_s$ plus four corner collocation points

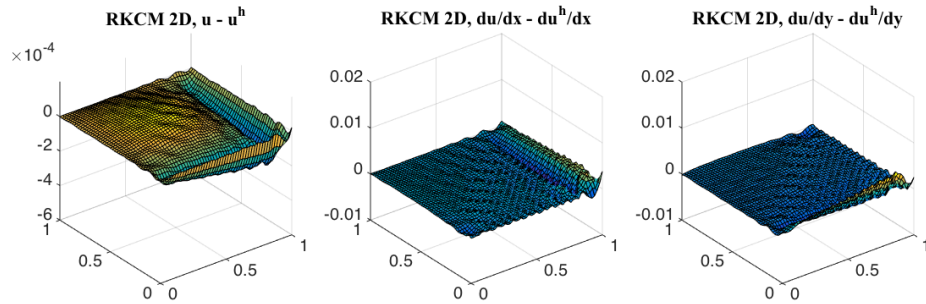


Figure 5.36 RKCM 2D. Error of u and its gradients for 2D Poisson's problem. 18x18 source points, $N_c = N_s$ plus a ring of collocation points

5.4.2. Comparison of Errors in RKCM and RBCM using uniform arrangements of collocation points

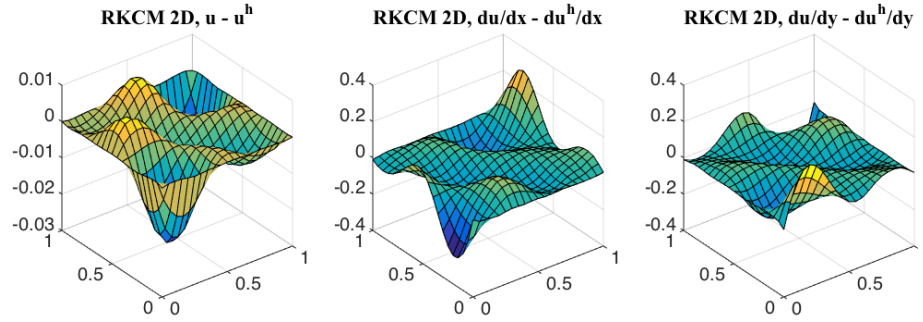


Figure 5.37 RKCM, error of u and its gradients in domain, 6×6 , $N_c = N_s$

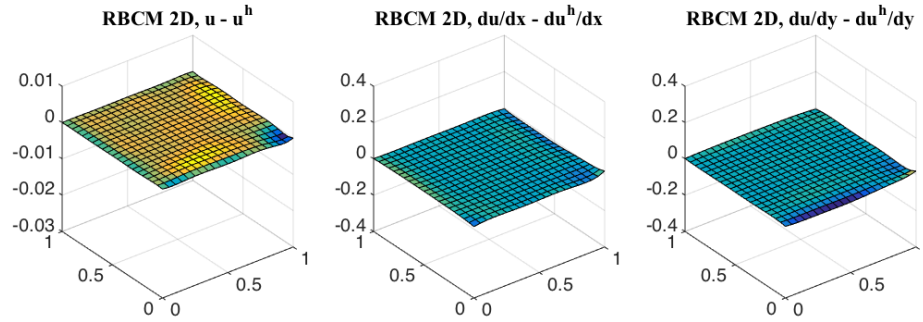


Figure 5.38 RBCM, error of u and its gradients in domain, 6×6 , $N_c = N_s$

Figure 5.37 and Figure 5.38 show the error of u^h and its derivatives throughout the domain in a coarse discretization (6×6 source points, 6×6 collocation points) using RKCM and RBCM, respectively. When the domain is discretized coarsely the errors of u^h and its derivatives are smaller by using RBCM than by RKCM. In RKCM, the errors of u^h are negligible on essential boundaries while error of derivatives of u^h on boundaries is relatively larger than that in the domain using coarse discretization.

Figure 5.39 and Figure 5.40 show the error distribution of u^h and du^h/dx_i throughout the domain with fine discretization (101 by 101 source points and

collocation points same as source points) using RKCM and RBCM, respectively. Under finer domain discretization, RKCM yields better accuracy than RBCM. According to the convergence study for RBCM, the convergence rates of error norms reduced significantly when the domain is discretized beyond certain refinement, while RKCM converges at a stable rate in model refinement. The possible reason for RKCM exceeds RBCM in accuracy is that fine discretization leads to ill-conditioned system in RBCM in fine discretization. Further investigation on the conditioning of linear systems will be discussed in the next chapter.

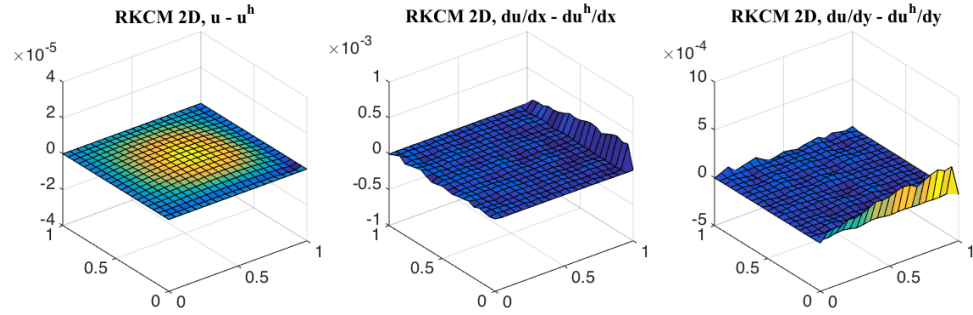


Figure 5.39 RKCM, errors of u and its gradients in the domain, 101×101 source points, $N_c = N_s$

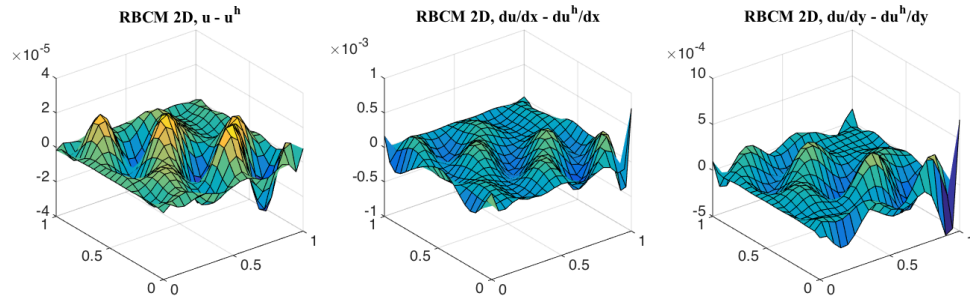


Figure 5.40 RBCM, errors of u and its gradients in the domain, 101×101 source points, $N_c = N_s$

Figure 5.41 to Figure 5.46 are plots of errors along the lines $y=0$, $y=0.25$, $y=0.5$, $y=0.75$ and $y=1.0$ using 16 by 16 source points. For this coarse discretization

case, RBCM gives much smaller error than RKCM as expected. These error plots offer more evidence for the accuracy of RBCM when the linear system is well-conditioned.

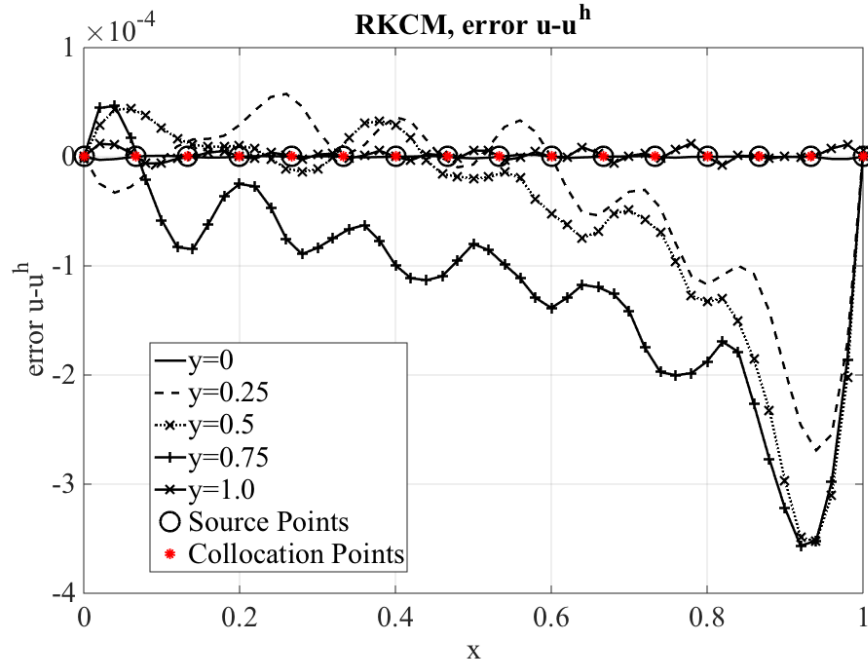


Figure 5.41 Errors in u , RKCM, quadratic basis, 16x16 source points, $N_c = N_s$

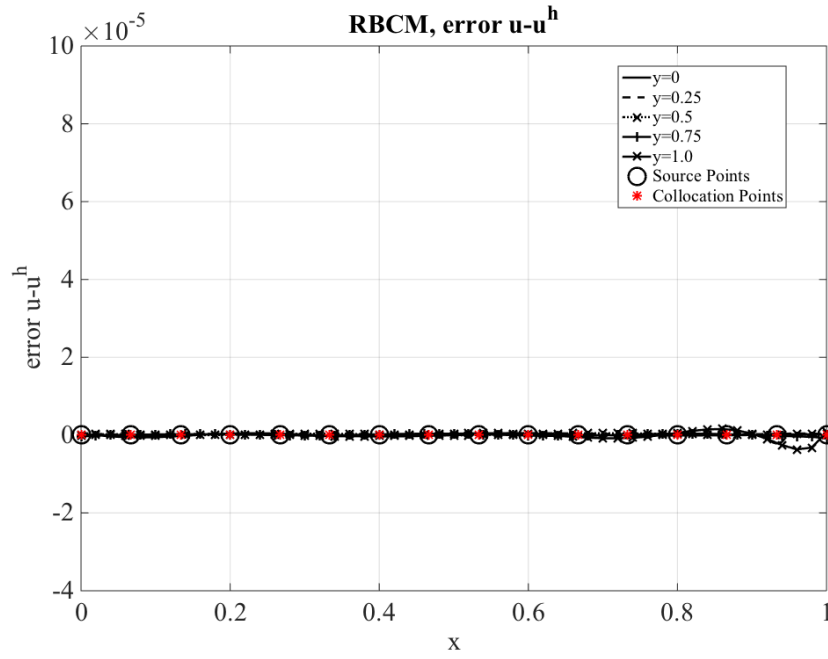


Figure 5.42 Errors in du/dx , RBCM, $c=1.6$, 16×16 source points, $N_c = 4N_s$

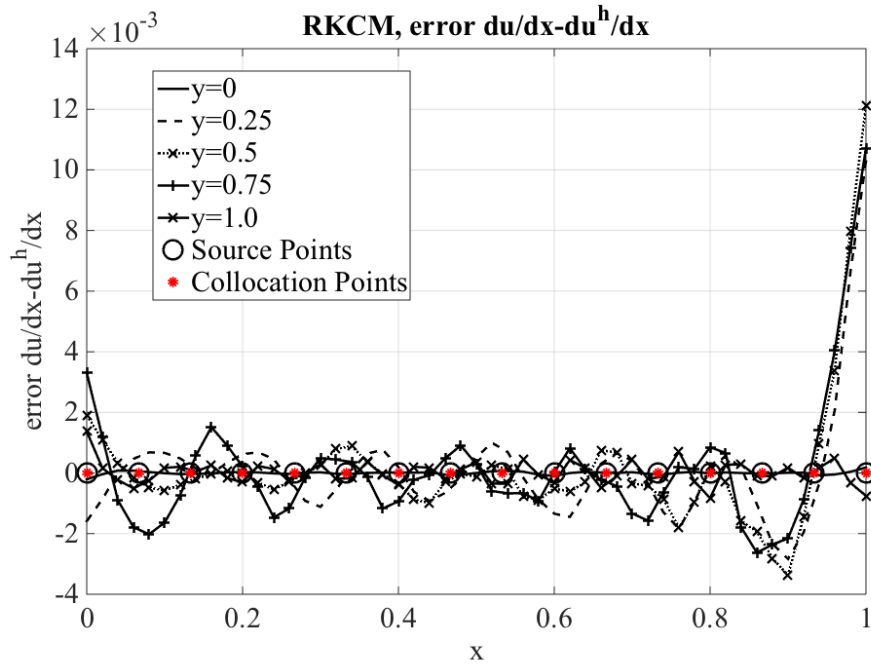


Figure 5.43 Errors in du/dx , RKCM, $n=2$, 16×16 source points, $N_c = N_s$

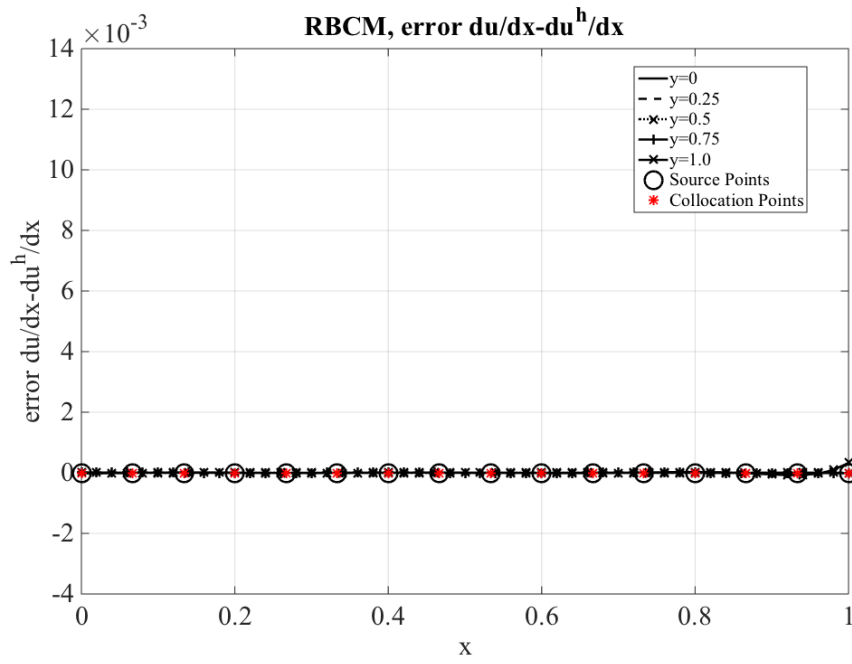


Figure 5.44 Errors in du/dx , RBCM, 16x16 source points, $N_c = N_s$

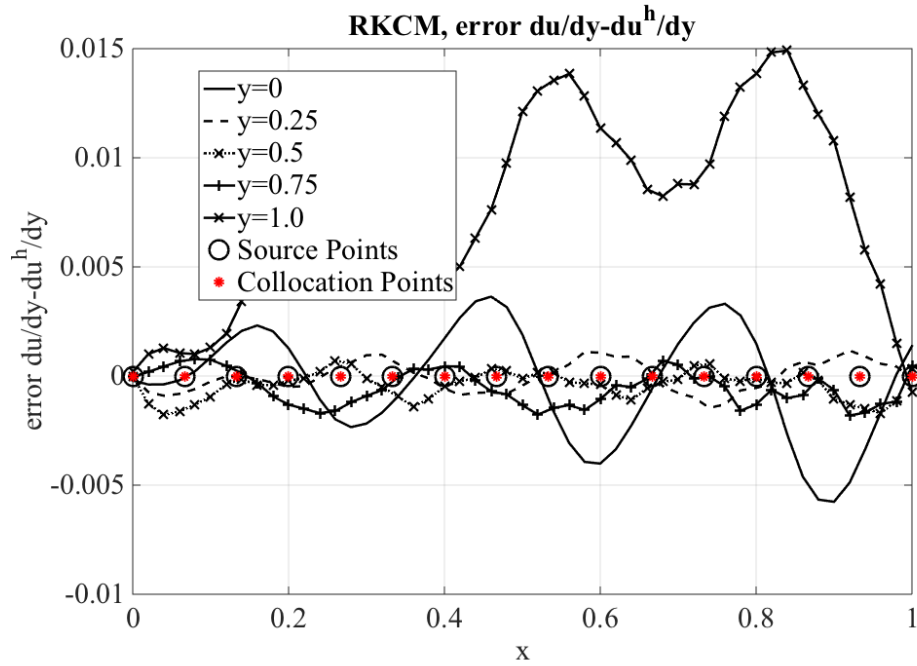


Figure 5.45 Errors in du/dy , RKCM, $n=2$, 16x16 source points, $N_c = N_s$

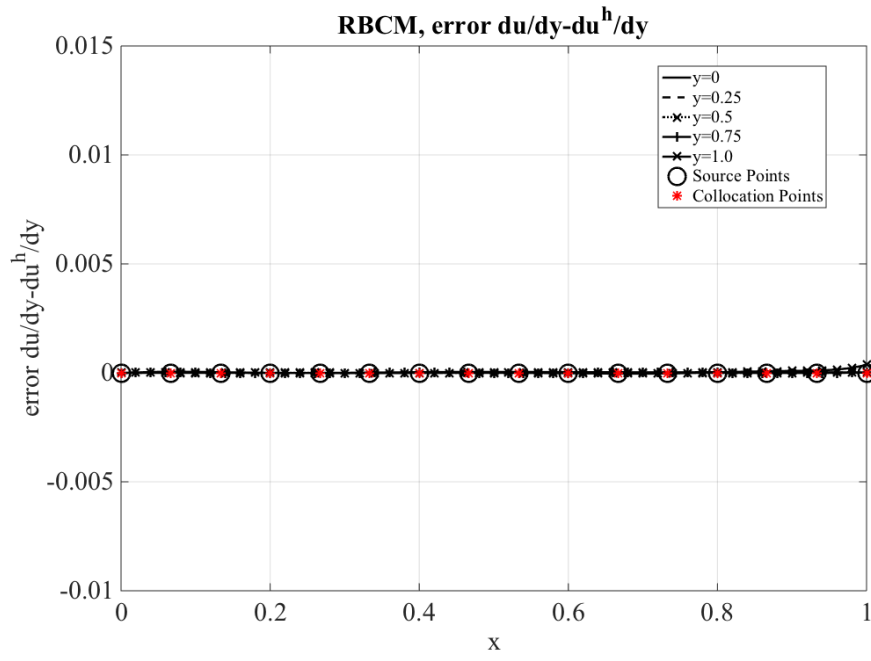


Figure 5.46 Errors in du/dy with RBCM, 16x16 source points, $N_c = 4N_s$

6 Efficiency and Stability of RKCM, RBCM and RKPM

In this chapter, the efficiency and stability of RKCM and RBCM are investigated and compared with RKPM based on weak form. Here the stability refers to the conditioning of the linear system, see Hu and Chen *et al.* [36].

6.1. Efficiency of RKCM, RBCM and RKPM

Again consider the 2-D Poisson problem in Section 5.3 with same uniform discretization with approximation parameters set as: RKPM with linear bases, RKCM with quadratic bases, RBCM with linear multiquadrics radial basis functions using shape parameter of 1.6. The domain is uniformly discretized to 6x6, 8x8, 10x10, 12x12, 14x14 to 16x16 source points.

The relative performance of the two meshfree collocation methods are shown in Figure 6.1. Under uniform coarse discretizations, RBCM can achieve better accuracy than RKCM with much lower CPU time consumed. From computational effort point of view, RBFs are simple and easy to construct, and RBCM avoids the computational complexity of constructing approximation functions and their derivatives. Further, RKPM results with different orders of Gauss quadrature rule are included for comparison in Figure 6.2 and Figure 6.3.

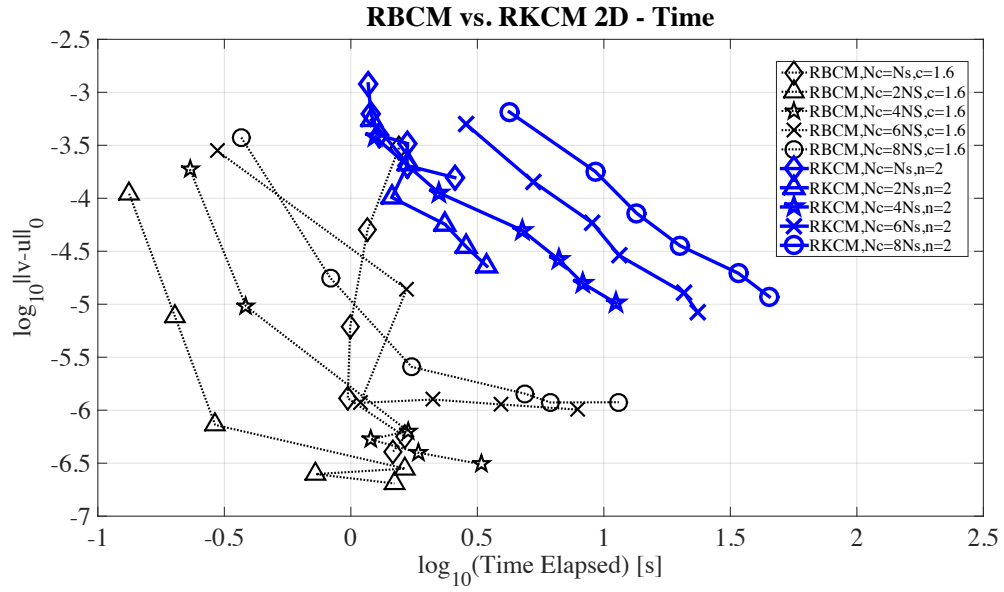


Figure 6.1 L_2 error norms and CPU time of RKCM and RBCM under unifrom discretization

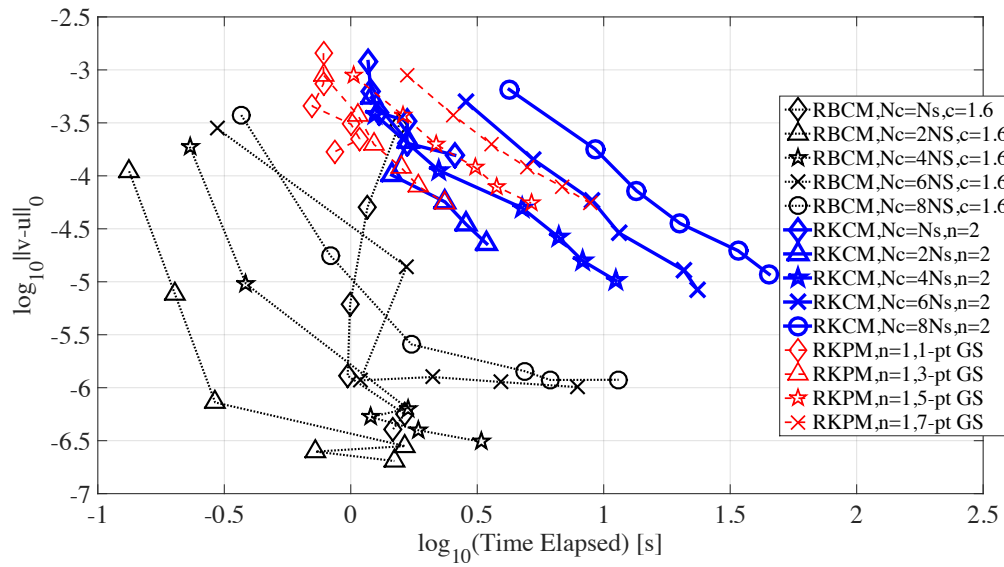


Figure 6.2 L_2 error norms and CPU time of collocation methods and RKPM (with linear basis) under unifrom discretization

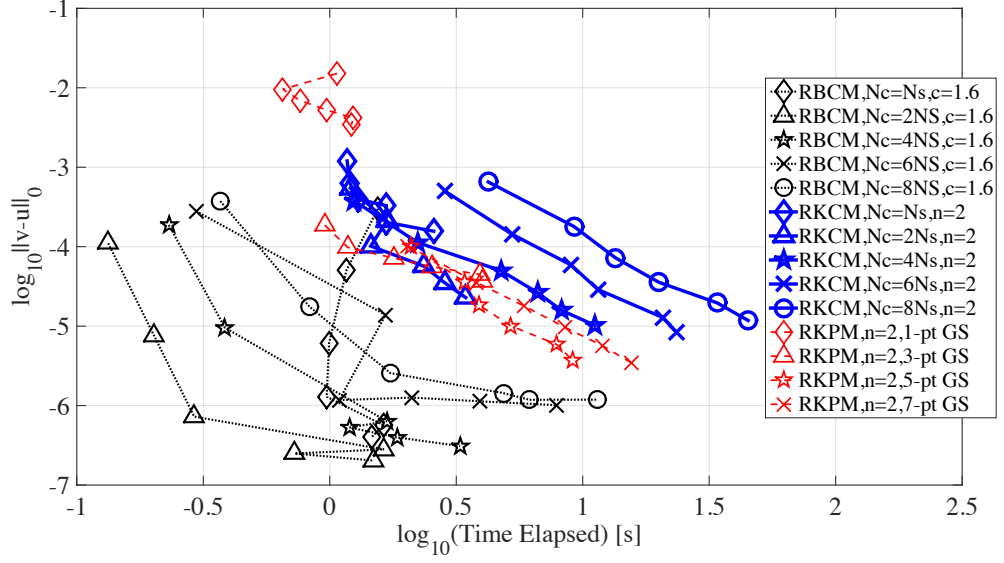


Figure 6.3 L_2 error norms and CPU time of collocation methods and RKPM (with quadratic basis) under unifrom discretization

Hu and Chen [39] have made comparison of operation counts between RKPM and RKCM, the total operation counts include the construction of RK approximation functions and solution for the linear system:

$$\underbrace{O(\mu \cdot N_s^4) + O(\frac{1}{6} N_s^3)}_{RKPM} > \underbrace{O(100s^3 \cdot N_s^2) + O(2N_s^3)}_{RKCM}$$

The coefficient μ is related to the order of Gauss quadrature in weak form, N_s is the number of source points and $s = n + 1$ where n is the order of bases. According to this analytical result, RKPM is more costly in larger systems than RKCM. However as Figure 6.2 and Figure 6.3 indicate, RKPM with both linear and quadratic bases is more efficient than RKCM under uniform discretization if enough integration points are employed. Nevertheless, it is noted that RKPM with Gauss quadrature loses its accuracy under non-uniform discretization, thus its efficiency will be less than RKCM in this case. This will be discussed in later sections.

As the numerical results in Figure 6.3 show, RBCM exhibits outstanding performance on efficiency and accuracy under same discretization compared to RKCM and RKPM for a relatively smaller discrete system. However, due to the fully densified system of RBCM, solving a problem with finer discretization using RBCM can be extremely costly and less accurate than RKCM and RKPM.

6.2. Conditioning of Linear Systems in RKCM and RBCM

Condition numbers of linear systems under different discretization of RBCM with different shape parameters are displayed in log scale in Figure 6.4, and Figure 6.5 provides comparison of condition numbers between RKCM and RBCM. It can be seen that RBCM with smaller shape parameter ($c = 0.5$) yields a better-conditioned system than RBCM with a larger shape parameter does ($c = 1.6$). Results in Figure 6.6 and Figure 6.7 show exponential convergence in RBCM, and the cases with larger parameter factors yield smaller errors.

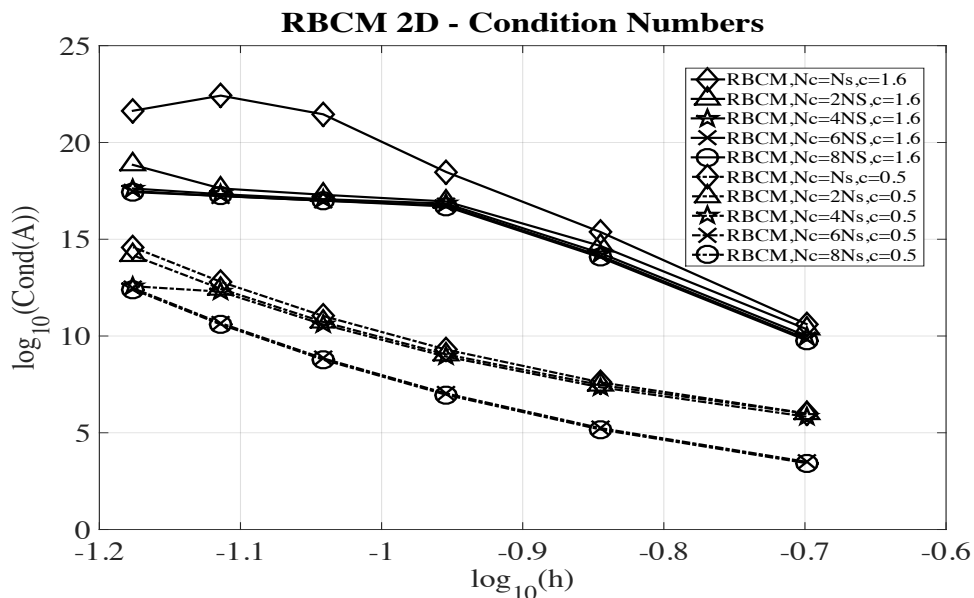


Figure 6.4 Condition numbers of linear systems of RBCM with different shape parameters in RBFs

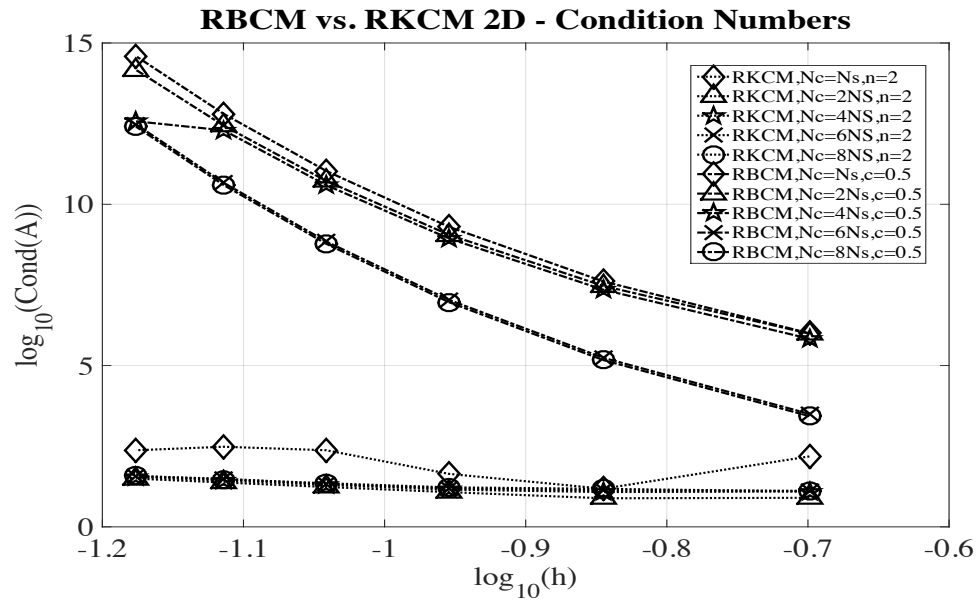


Figure 6.5 Condition numbers of linear systems of RBCM and RKCM

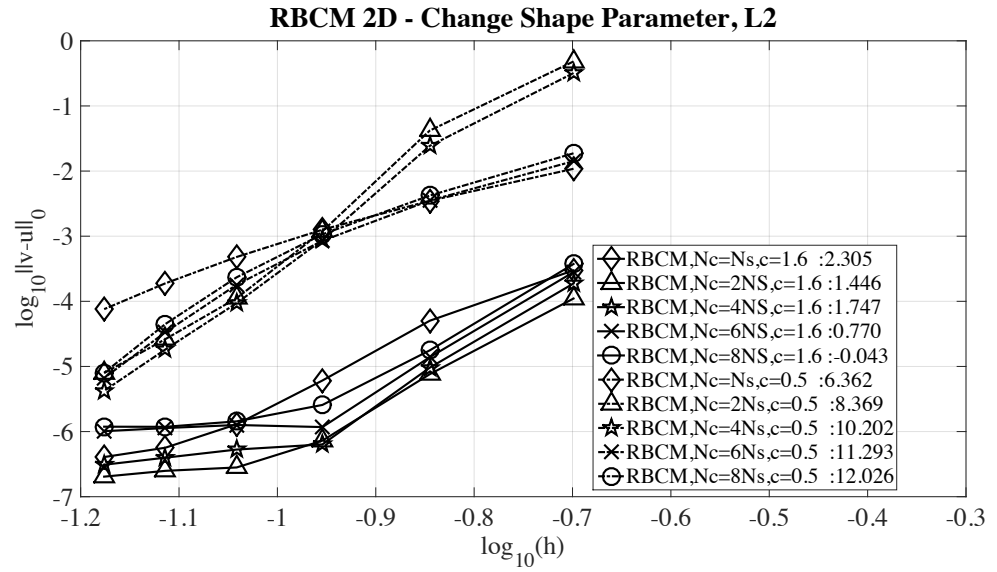


Figure 6.6 Effect of shape parameter on error in L_2 norm in RBCM

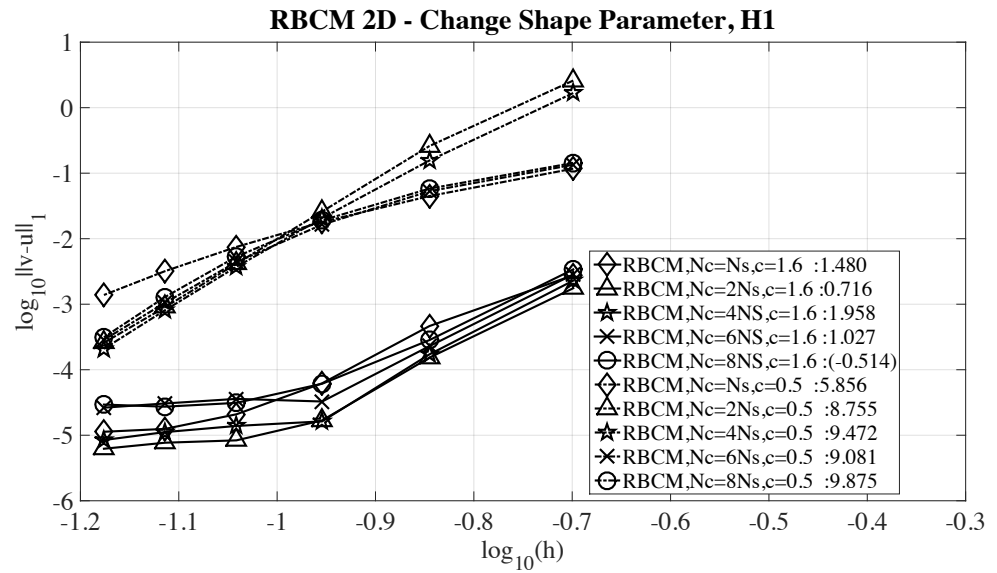


Figure 6.7 Effect of shape parameter on error in H^1 norm in RBCM

6.3. Solution accuracy using non-uniform discretization

In this section, solution accuracy using non-uniform discretization is investigated. The 4 plots in Figure 6.8 are the 4 levels of refinements of source points and collocation points employed. In the test, N_s and N_c varies from 36, 121, 441 to 1681. The collocation points are exact the same as the source points.

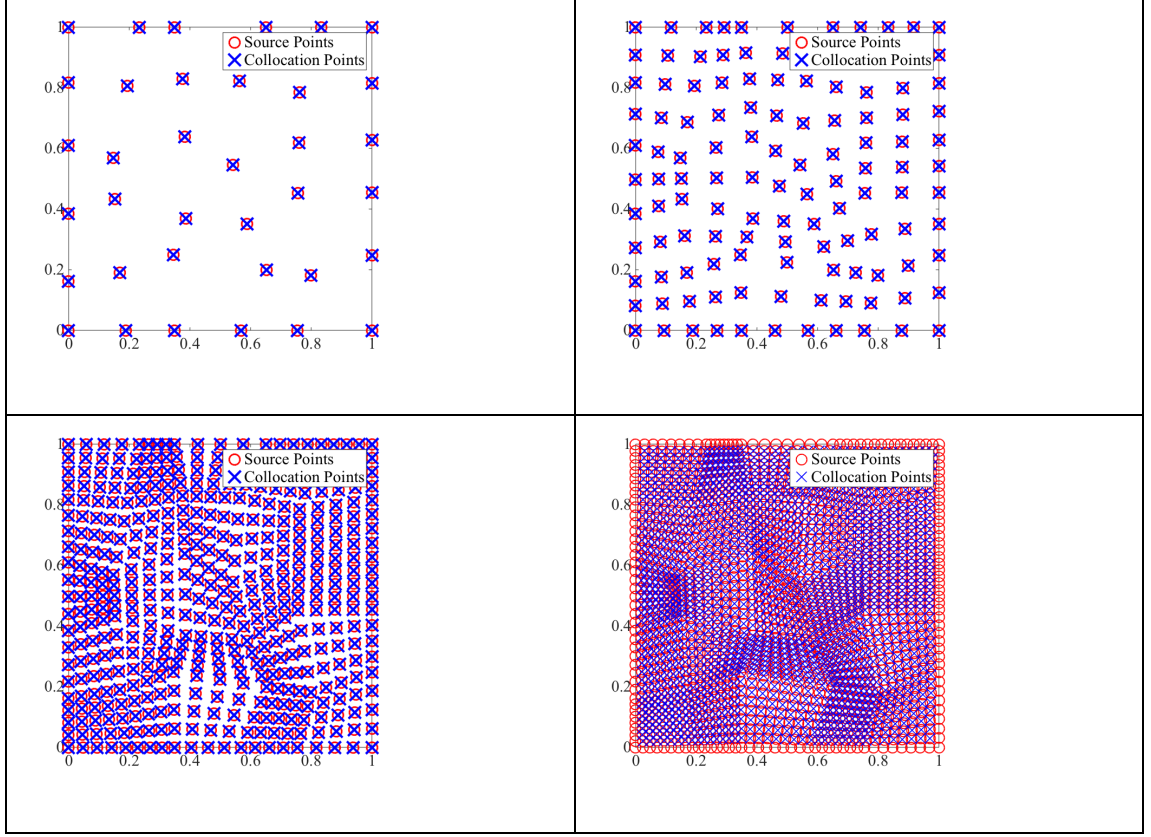


Figure 6.8 Uniform refinement of irregular nodes discretization, collocation points at source points

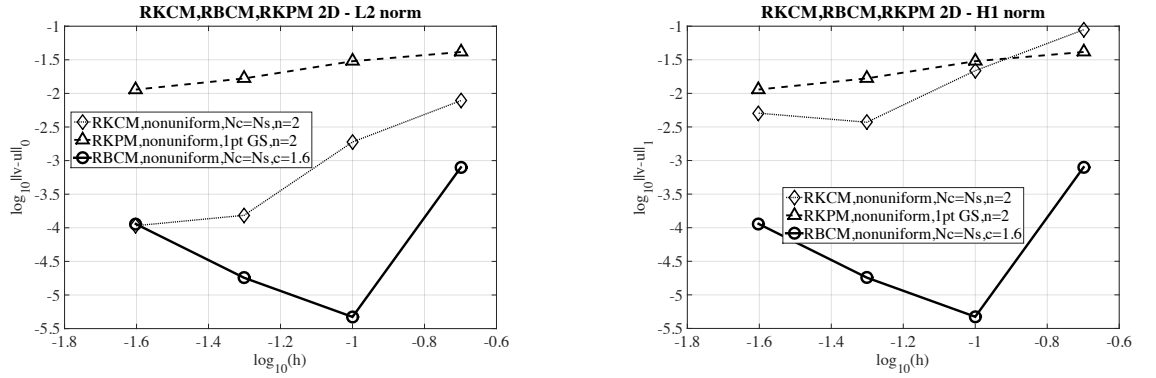


Figure 6.9 Convergence of error in L_2 and H^1 norms RKCM ($n=2$, non-uniform $N_c = N_s$), RBCM ($c=1.6$, non-uniform, $N_c = N_s$) and RKPM (non-uniform, quadratic basis, 1-pt Gauss quadrature)

Results using discretization in Figure 6.8 are shown in Figure 6.9. None of RKCM or RBCM with $N_c = N_s$ or RKPM with 1-pt Gauss integration gives optimal convergence under non-uniform discretization. In general, RKCM and RBCM show

better accuracy than RKPM with 1-pt Gauss quadrature. Meanwhile, RBCM suffers from ill-conditioning in the tests and start to diverge after $N_s > 121$. Overall, RKCM and RBCM are not recommended for using non-uniform source points with collocation points exactly at source points.

Now consider the case in which $N_c = 2N_s$ for RKCM and RBCM, and uniform collocation points are used in the domain discretized by non-uniform source points.

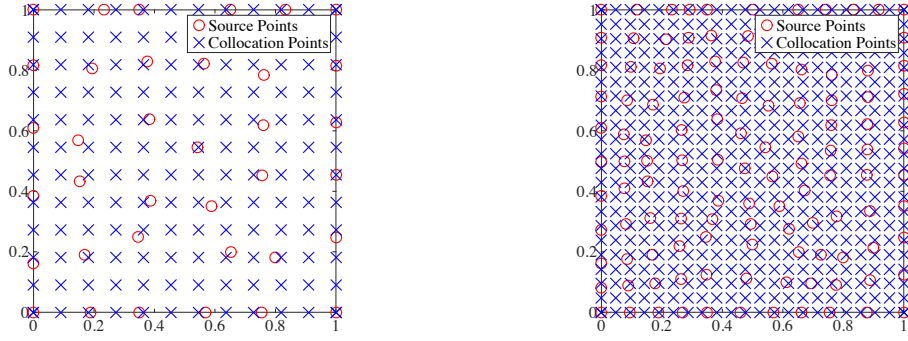


Figure 6.10 Uniform refinement of irregular nodes discretization, uniform collocation points $N_c = 2N_s$ (Cases with 6x6 and 11x11 source points shown)

With the non-uniform discretization shown in Figure 6.10, error norms are shown in Figure 6.11 and Figure 6.12. RKPM does not show convergence with 3x3 Gauss quadrature rule, and its accuracy is much less than that of RBCM and RKCM. Again, RBCM suffers from ill-conditioning when N_s exceeds 121. RKCM with non-uniform source points gives slightly smaller convergence rate than the case with uniform source points. Compared with results of RKPM with DNI, SCNI and VCI. under non-uniform discretizations, it shows that DNI loses convergence in H^1 error norms. SCNI on the other hand, converges optimally. Note that RKCM yields better accuracy than RKPM with SCNI due to the use of higher order bases and

large number of collocation points. With the employment of VCI, quadratic bases can be used in RKPM with optimal rate of convergence. The result of RKCM is less accurate than that of RKPM with VCI in L_2 norm but very close to RKPM with VCI in H^1 norm. Therefore, RKCM can be an alternative of VCI that addresses the issue of achieving optimal convergence and accuracy with higher order bases under non-uniform discretizations, but it avoids the complexity of VCI. Overall, RKCM performs quite well under non-uniform discretization and appears to be insensitive to the non-uniformity in the discretization.

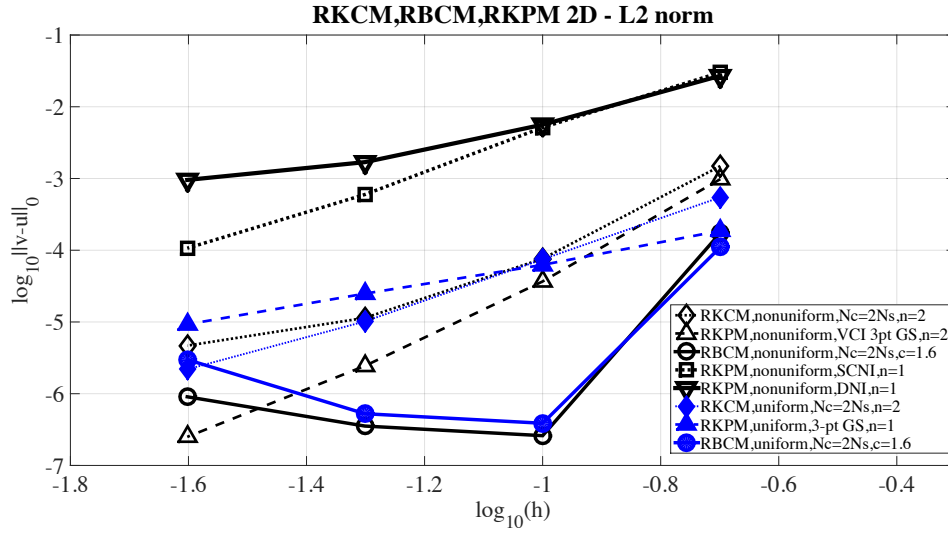


Figure 6.11 Convergence plot of error in L_2 norm under non-uniform discretization of RKCM ($n=2$, $N_c = 2N_s$), RBCM ($c=1.6$, $N_c = 2N_s$) and RKPM ($n=2$, VCI, DNI, SCNI, GI), comparing results under uniform discretization

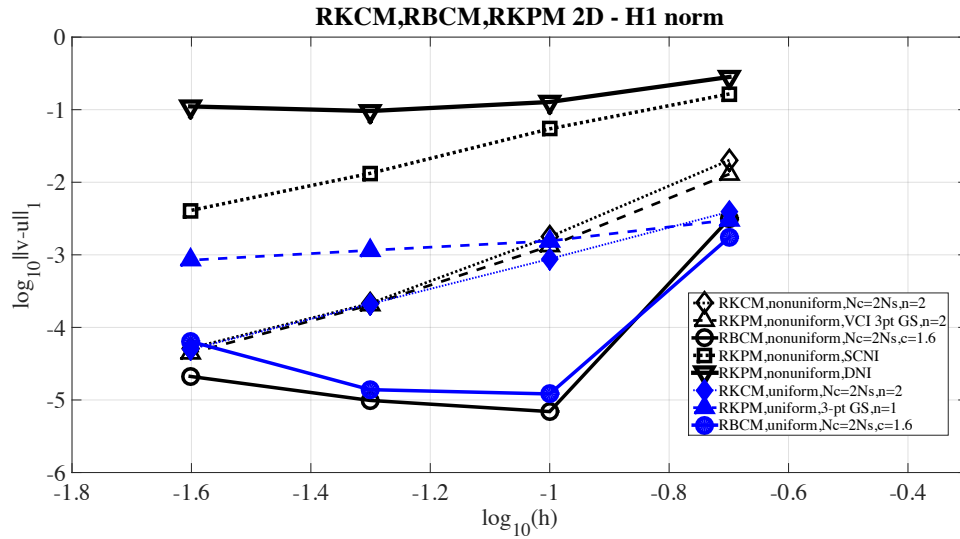


Figure 6.12 Convergence plot of error in H^1 norm under non-uniform discretization of RKCM ($n=2$, $N_c = 2N_s$), RBCM ($c=1.6$, $N_c = 2N_s$) and RKPM ($n=2$, VCI, DNI, SCNI, GI), comparing results under uniform discretization

The errors of solution and its derivatives are shown in Figure 6.13 to Figure 6.17. Figure 6.13 and Figure 6.14 compare RKCM solutions under uniform and non-uniform discretizations with 11×11 source points. RKCM using non-uniform source

points with uniform $N_c = 2N_s$ only slightly loses accuracy compared to the uniform discretization case, and the result does not show large oscillation. Figure 6.15 displays RBCM under the same source points and collocation points arrangements as those used in RKCM. In this coarse discretization case the discrete system presents no ill-conditioning, and RBCM performs very well. Figure 6.16 demonstrate the situation where RKPM can encounter serious instability and inaccuracy in solution when using DNI. This situation can be corrected by SCNI for domain integration as shown in Figure 6.16. On the other hand, RKCM with sufficient collocation points can achieve good accuracy without the complexity of RKPM with SCNI.

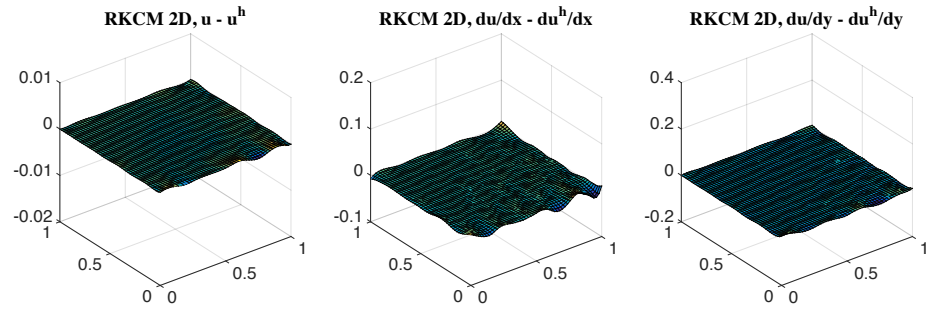


Figure 6.13 RKCM, error in domain, 121 non-uniform source points with uniform collocation points $N_c = 2N_s$

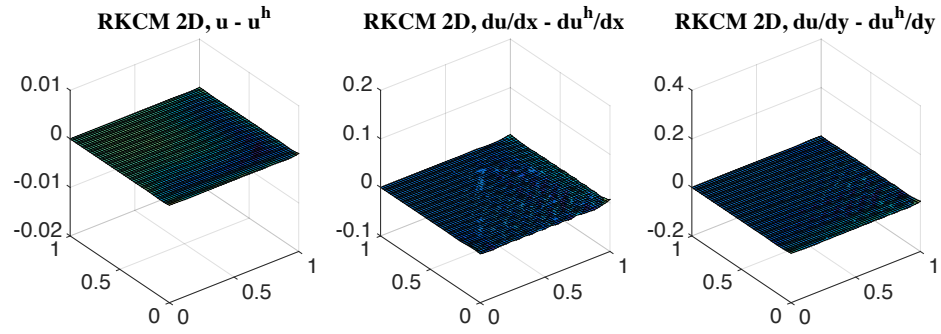


Figure 6.14 RKCM, error in domain, 121 uniform source points with uniform collocation points $N_c = 2N_s$

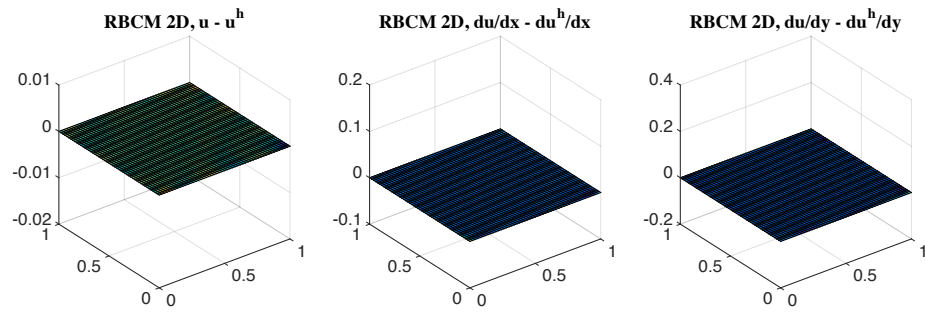


Figure 6.15 RBCM, error in domain, 121 non-uniform source points with uniform collocation points $N_c = 2N_s$, $c = 1.6$

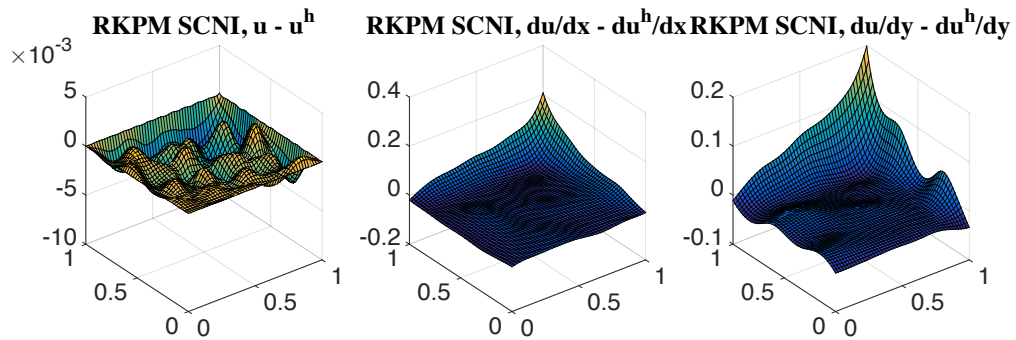


Figure 6.16 RKPM with linear bases and SCNI discretized by 121 non-uniform nodes, error in domain

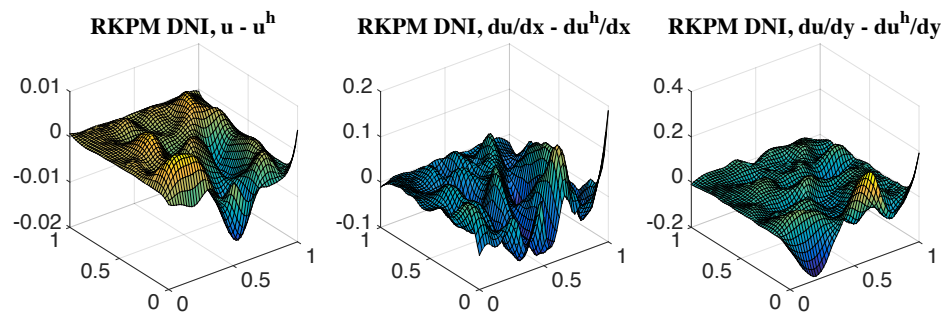


Figure 6.17 RKPM with linear bases and DNI discretized by 121 non-uniform nodes, error in domain

7 Conclusion

7.1. Conclusions

In the presented work, the performance in terms of accuracy, convergence, efficiency and stability of three meshfree methods, RKCM, RBCM and RKPM, are studied and compared. This study focuses on the performance of the strong form collocation meshfree methods using RK and RBF functions, and the Galerkin meshfree method with RK approximation functions. Observations from the numerical results are summarized as follows:

1. How basis functions affect convergence and stability of collocation methods?

RKCM offers optimal algebraic convergence. Another advantage of RKCM is that the compact support feature offers locality in the approximation and gives a well-conditioned linear system compared to RBCM. This method gives accurate result and optimal convergence rate when more collocations are used than the number of source points. Accuracy can be slightly reduced in non-uniform discretization compared to that in the uniform discretization, but the result is still promising. Under non-uniform discretization, RKCM is attractive alternative of RKPM with SCNI or CVI, since it avoids the domain integration complexity.

RBCM generally provides the best accuracy under coarse discretizations, and the solution accuracy is highly related to the choice of shape parameter in the radial basis functions. One apparent drawback of RBCM is that it suffers from ill-conditioning when the discretization is fine or when the shape parameter is relatively large. As a result of such ill-conditioning, convergence behavior of RBCM can be greatly deteriorated in fine discretization. The use of smaller shape parameter can provide a better conditioned linear system, but the accuracy is sacrificed. RBCM offers optimal result when collocation points are exactly located at the source nodes,

and adding more collocation points does not yield noticeable accuracy improvement. Similar performance of RBCM under non-uniform source points is also observed. While RBCM is efficient in computing RBF and its derivatives, the computational implementation is straightforward, and the complexity in the domain integration process is avoided, but the dense matrix in RBCM consumes higher CPU among the three methods when solving large systems.

2. How the complexity in Galerkin method can be simplified in strong form collocation method?

RKPM yields poor accuracy and convergence under non-uniform discretization when using Gauss quadrature or direct nodal integration (DNI) method. Such problems can be resolved by introducing Stabilized Conforming Nodal Integration (SCNI) and variationally consistent integration (VCI). It is noted that only linear bases are used in RKPM-SCNI. For higher order bases, RKPM requires method such as the Variationally Consistent Integration (VCI, Chen *et al.* [14].) to assume optimal convergence. This complexity in the domain integration for RKPM motivated the strong form collocation methods such as RKCM.

RKCM is a strong form version of the reproducing kernel based meshfree methods. Comparing with RKPM, the complexity of imposing essential boundary conditions and domain integration are reduced while maintaining ideal accuracy and convergence. Moreover, higher order bases can be directly employed without the complexity of VCI.

3. How the collocation points can be arranged to optimize the performance of strong form collocation methods?

RKCM offers optimal algebraic convergence when $N_s = 2N_c \sim 4N_c$ in each dimension; keeping adding collocation points beyond this range would not deliver further improvement in accuracy. For RBCM, optimal accuracy are obtained when the range of number of collocation points lies in $N_s = N_c \sim 2N_c$. However, ill-conditioning is the dominant influence on accuracy and convergence in RBCM, rather than the system discretization.

Additional boundary layer of collocation points can significantly improve accuracy and stability of RKCM, with only a little additional CPU more than the basic $N_c = N_s$ system.

4. How non-uniform discretization affects the convergence and stability of meshfree methods?

Non-uniform discretization leads to poor accuracy and convergence in Galerkin meshfree method using DNI and lower order Gauss quadrature unless SCNI is used, but it has only marginal influence in accuracy and convergence in collocation method such as RKCM and RBCM. RKCM and RBCM keep stable and accurate results under non-uniform discretizations without additional treatments.

5. What affects the efficiency of meshfree methods?

Considering computational efficiency, although taking higher order derivatives in the strong form approximation can be computationally costly, the cost in domain integration is reduced. Especially, RKCM can provide higher convergence rate by using quadratic bases, which is harder to achieve by RKPM, since either formidable number of quadrature points are needed or more advanced domain integration method such as VCI is required. In this regard, RKCM is an attractive alternative of RKPM when employing higher order bases.

7.2. Recommendations for future work

The followings are suggested for future work.

1. Extension of RKCM to nonlinear problems.
2. Improvement of computational efficiency in the localized RBF using reproducing kernel as the localizing function proposed by Chen *et al.* 2008 [25].

Reference

- [1] Aluru, N. R., 2000. “A Point Collocation Method Based on Reproducing Kernel Approximations.” *International Journal for Numerical Methods in Engineering* **47**: 1083–1121.
- [2] Atluri, S. N., and T. Zhu. 1998. “A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics.” *Computational Mechanics* **22**: 117–27.
- [3] Babuška, I., and Melenk J. M. 1997. “The Partition of Unity Method.” *International Journal for Numerical Methods in Engineering* **40**: 727–58.
- [4] Babuška, I., Banerjee, U., Osborn J., and Li, Q. 2008. “Quadrature for Meshless Methods.” *International Journal for Numerical Methods in Engineering* **76**: 1434–70.
- [5] Basava R. R., Chen J. S., Zhang Y. 2014 Pixel Based Meshfree Modeling of Skeletal Muscles. Lecture Notes in Computer Science, [Y.J. Zhang and J.M.R.S. Tavares Eds.], Springer, CompIMAGE, LNCS **8641**, pp. 316–327.
- [6] Beissel, Stephen, and Belytschko, T. 1996. “Nodal Integration of the Element-Free Galerkin Method.” *Computer Methods in Applied Mechanics and Engineering* **139**: 49–74.
- [7] Belytschko, T, Gu, L., and Lu, Y. Y. 1994. “Fracture and Crack Growth by Element Free Galerkin Methods.” *Modelling and Simulation in Materials Science and Engineering* **2**: 519-534.
- [8] Belytschko, T., Lu, Y. Y., and Gu, L.. 1994. “Element-Free Galerkin Methods.” *International Journal for Numerical Methods in Engineering* **37** : 229–56.
- [9] Belytschko, T., and Tabbara, M., 1996. “Dynamic Fracture Using Element-Free Galerkin Methods.” *International Journal for Numerical Methods in Engineering* **39** (6): 923–38.
- [10] Belytschko, T., Krongauz Y., Organ, D., and Krysl, P., 1996. “Meshless Methods: An Overview an Drecent Developments.” *Computer Methods in Applied Mechanics and Engineering* **7815** (96): 3–37.
- [11] Chen, J. S., Pan, S., and Wu, C.T. 1997. “Large Deformation Analysis of Rubber Based on a Reproducing Kernel Particle Method.” *Computational Mechanics* **19** (3): 211–27.
- [12] Chen, J. S., and Belytschko, T. 2013. “Meshless and Meshfree Methods.” *Encyclopedia of Applied and Computational Mathematics*.

- [13] Chen, J. S., Chi, S. W., and Hu, H.Y. 2011. "Recent Developments in Stabilized Galerkin and Collocation Meshfree Methods." *Computer Assisted Mechanics and Engineering Sciences*, **18**: 3–21.
- [14] Chen, J. S., Hillman, M., and Ruter, M. 2013. "An Arbitrary Order Variationally Consistent Integration for Galerkin Meshfree Methods." *International Journal for Numerical Methods in Engineering*, **95**: 387–418.
- [15] Chen, J. S., Hu, W., and Hu, H.Y., 2008. "Reproducing Kernel Enhanced Local Radial Basis Collocation Method." *International Journal for Numerical Methods in Engineering*, **75**: 600–627.
- [16] Chen, J. S., Hu, W., Hu, H. Y., 2008 "Reproducing Kernel Enhanced Local Radial Basis Collocation Method," *International Journal for Numerical Methods in Engineering*, **75**, 600-627.
- [17] Chen, J. S. and Mehraeen, S. 2004, "Variationally Consistent Multi-scale Modeling and Homogenization of Stressed Grain Growth", *Computer Methods in Applied mechanics and Engineering*, **193**, 1825-1848.
- [18] Chen, J. S., Han, W., You, Y., Meng, X., 2003 "A Reproducing Kernel Method with Nodal Interpolation Property," *International Journal for Numerical Methods in Engineering*, **56**, pp. 935-960,.
- [19] Chen, J. S., "Lecture Note: Meshfree Methods for Linear and Nonlinear Mechanics," UC San Diego, 2015.
- [20] Chen, J. S., Pan, C, Roque, C., and Wang, H. P., 1998. "A Lagrangian Reproducing Kernel Particle Method for Metal Forming Analysis." *Computational Mechanics* **22**: 289–307.
- [21] Chen, J. S., Pan, C., Wu, C.T., and Liu, W. K.. 1996. "Reproducing Kernel Particle Methods for Large Deformation Analysis of Non-Linear Structures." *Computer Methods in Applied Mechanics and Engineering* **139** (96): 195–227.
- [22] Chen, J. S., and Wang, D.. 2006. "A Constrained Reproducing Kernel Particle Formulation for Shear Deformable Shell in Cartesian Coordinates." *International Journal for Numerical Methods in Engineering* **68** (2): 151–72.
- [23] Chen, J. S., and Wang, H. P.. 2000. "New Boundary Condition Treatments in Meshfree Computation of Contact Problems." *Computer Methods in Applied Mechanics and Engineering* **187**: 441–68.
- [24] Chen, J. S., Wang, H. P., Yoon, S., and You, S., 2000. "Some Recent Improvements in Meshfree Methods for Incompressible Finite Elasticity Boundary Value Problems with Contact." *Computational Mechanics* **25**: 137–156.
- [25] Chen, J. S., Wang, L., Hu, H.Y., and Chi, S. W. 2009. "Subdomain Radial Basis Collocation Method for Heterogeneous Media." *International Journal for Numerical Methods in Engineering*, **83**: 851-876.

- [26] Chen, J. S., Sangpil Yoon, and Cheng Tang Wu. 2002. "Non-Linear Version of Stabilized Conforming Nodal Integration for Galerkin Mesh-Free Methods." *International Journal for Numerical Methods in Engineering* **53**: 2587–2615.
- [27] Chen, J. S., Wu, C. T., and Yoon, S. 2001. "A Stabilized Conforming Nodal Integration for Galerkin Mesh-Free Methods." *International Journal for Numerical Methods in Engineering*, **50**: 435–66.
- [28] Chi, Sheng-Wei, J. S. Chen, and Hu, H.Y. 2013. "A Weighted Collocation on the Strong Form with Mixed Radial Basis Approximations for Incompressible Linear Elasticity." *Computational Mechanics* **53**: 309–24.
- [29] Chi, S. W., Chen, J. S., Hu, H.Y., and Yang, J. 2013. "A Gradient Reproducing Kernel Collocation Method for Boundary Value Problems." *International Journal for Numerical Methods in Engineering*, **93**: 1381–1402.
- [30] Chi, S. W, Chen, J. S., Luo, H., Hu, H.Y., and Wang, L.. 2013. "Dispersion and Stability Properties of Radial Basis Collocation Method for Elastodynamics." *Numerical Methods for Partial Differential Equation* **23** (4): 904–22.
- [31] Dolbow, J., and Belytschko, T.. 1999. "Numerical Integration of the Galerkin Weak Form in Meshfree Methods." *Computational Mechanics* **23**: 219–30.
- [32] Fernandez-Mendez, S., and Huerta A. 2003. "Imposing Essential Boundary Conditions in Mesh-Free Methods," *Computer Methods in Applied Mechanics and Engineering* **193**: 1257-1275
- [33] Guan C., Chi S. W., Chen, J. S., Slawson T. R., Roth, M. J. 2011 "Semi-Lagrangian Reproducing Kernel Particle Method for Fragment-Impact Problems," *International Journal of Impact Engineering*, **38**, 1033-1047.
- [34] Hao, S., Liu, W. K., and Chang, C. T. 2000. "Computer Implementation of Damage Models by Finite Element and Meshfree Methods." *Computer Methods in Applied Mechanics and Engineering* **7825** (99): 401–40.
- [35] Hon, Y. C., and Schaback, R., 2001. "On Unsymmetric Collocation by Radial Basis Functions." *Applied Mathematics and Computation* **119** (2): 177–86.
- [36] Hu, H. Y., Chen, J. S., and Chi, S. W.. 2011. "Perturbation and Stability Analysis of Strong Form Collocation Reproducing Kernel Approximation." *International Journal for Numerical Methods in Engineering*, **88**: 157–79.
- [37] Hu, H. Y., Chen, J. S. and Hu, W. 2006. "Weighted Radial Basis Collocation Method for Boundary Value Problems." *International Journal for Numerical Methods in Engineering*, **69**(13) 2736–57.
- [38] Hu, H. Y., Chen, J. S., Hu, W. 2009. "Error Analysis of Collocation Method Based on Reproducing Kernel Approximation." *Numerical Methods for Partial Differential Equations* **23**: 904–22.

- [39] Hu, H. Y., Lai, C. K., and Chen, J. S. 2009. "A Study on Convergence and Complexity of Reproducing Kernel Collocation Method." *Interaction and Multiscale Mechanics* **2** (3): 295–319.
- [40] Hu, H. Y., and Li, Z. C., 2005. "Radial Basis Collocation Methods for Elliptic Boundary Value Problems." *Computers and Mathematics with Applications*, **50**, 289–320.
- [41] Huerta, A., and Fernández, S. 2000. "Enrichment and Coupling of the finite Element and Meshless Methods." *International Journal for Numerical Methods in Engineering* **48**: 1615–36.
- [42] Johnson, G. R., and Beissel, S. 1996. "Normalized Smoothing Functions for SPH Impact Computations." *International Journal for Numerical Methods in Engineering* **39** (February): 2725–41.
- [43] Kansa, E. J., and Hon, Y. C. 2000. "Circumventing the Ill-Conditioning Problem with Multiquadric Radial Basis Functions: Applications to Elliptic Partial Differential Equations." *Computers and Mathematics with Applications* **39**: 123–37.
- [44] Kansa, E. J. 1990. "Multiquadrics - A Scattered Data Approximation Scheme with Applications to Computational Fluid-dynamics—I Surface Approximations and Partial Derivative Estimates." *Computers & Mathematics with Applications* **19** (8): 127–45.
- [45] Kim, D. W., and Kim, Y. 2003. "Point Collocation Methods Using the Fast Moving Least-Square Reproducing Kernel Approximation." *International Journal for Numerical Methods in Engineering* **56**: 1445–64.
- [46] Krysl, P., and Belytschko, T. 1999. "The Element Free Galerkin Method for Dynamic Propagation of Arbitrary 3-D Cracks." *International Journal for Numerical Methods in Engineering* **44** (6): 767–800.
- [47] Krysl, P., and Belytschko, T.. 1997. "Element-Free Galerkin Method: Convergence of the Continuous and Discontinuous Shape Functions." *Computer Methods in Applied Mechanics and Engineering* **148** (3-4): 257–77.
- [48] Lancaster, P., and Salkauskas, K. 1981. "Surfaces Generated by Moving Least Squares Methods." *Mathematics of Computation* **37** (155): 141–141.
- [49] Li, S., and Liu, W. K.. 2002. "Meshfree and Particle Methods and Their Applications." *Applied Mechanics Reviews* **55** (1): 1-34 .
- [50] Libersky, L. D., Petschek, A. G., Carney, T., Jim R. Hipp, and Firooz a. Allahdadi. 1993. "High Strain Lagrangian Hydrodynamics." *Journal of Computational Physics*, **109**: 67-75.
- [51] Liu, W. K., Jun, S., and Zhang, Y. 1995. "Reproducing Kernel Particle Methods." *International Journal for Numerical Methods in Fluids* **20**: 1081–1106.

- [52] Lu, Y. Y., T. Belytschko, and L. Gu. 1994. "A New Implementation of the Element Free Galerkin Method." *Computer Methods in Applied Mechanics and Engineering* **113** (2): 397–414.
- [53] Lu, Y. Y., Belytschko, T., and Tabbara, M. 1995. "Element-Free Galerkin Method for Wave Propagation and Dynamic Fracture." *Computer Methods in Applied Mechanics and Engineering* **126** (2): 131–53.
- [54] Lucy, L. B. 1977. "A Numerical Approach to the Testing of the Fission Hypothesis." *The Astronomical Journal* **82**: 1013.
- [55] Luo, H., Liu, X., and Huang, X. 2011. "Stability and Dispersion Analysis of Reproducing Kernel Collocation Method for Transient Dynamics." *Applied Mathematics and Mechanics* **32**: 777–88.
- [56] Madych, W. R. 1992. "Miscellaneous Error Bounds for Multiquadric and Related Interpolators." *Computers and Mathematics with Applications Mathematics with Applications* **24** (12): 121–38
- [57] Monaghan, J. 2000. "SPH without a Tensile Instability." *Journal of Computational Physics* **159** (2): 290–311.
- [58] Nayroles, B., Touzot, G. and Villon, P. 1992. "Generalizing the Finite Element Method: Diffuse Approximation and Diffuse Elements." *Computational Mechanics* **10**: 307–18.
- [59] Nguyen, V. P., Rabczuk, T., Bordas, S and Duflot, M. 2008. "Meshless Methods: A Review and Computer Implementation Aspects." *Mathematics and Computers in Simulation* **79**: 763–813.
- [60] Nitsche, J. A., Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, Vol. **36** pp. 9-15, 1970-71.
- [61] Oden, J. T., Diller, K. R., Bajaj, C., Browne, J. C., Hazle J., Babuška, I., *et al.* 2007. "Dynamic Data-Driven Finite Element Models for Laser Treatment of Cancer." *Numerical Methods for Partial Differential Equation* **23** (4): 904–22.
- [62] Oñate, E., Idelsohn, S., Zienkiewicz, O. C. and Taylor, R. L. 1996. "A Finite Point Method in Computational Mechanics. Applications to Convective Transport and Fluid Flow." *International Journal for Numerical Methods in Engineering* **39**: 3839–66.
- [63] Puso, M. A, Chen, J. S., Zywucz, E. and Elmer W. 2011. "Meshfree and Finite Element Nodal Integration Methods." *International Journal for Numerical Methods in Engineering*, **74**: 416-446.
- [64] Schaback, R, and Wendland, H. 1995. "Using Compactly Support Radial Basis Functions to Solve Partial Differential Equations."

- [65] Shaw, A, Roy D., Reid, S.R., and Aleyaasin, M. 2007. "Reproducing Kernel Collocation Method Applied to the Non-Linear Dynamics of Pipe Whip in a Plane." *International Journal of Impact Engineering* **34**: 1637–54.
- [66] Wang, D., and Chen J. S. 2004. "Locking-Free Stabilized Conforming Nodal Integration for Meshfree Mindlin-Reissner Plate Formulation." *Computer Methods in Applied Mechanics and Engineering* **193**: 1065–83.
- [67] Wang, L., Chen J. S., and Hu, H. Y.. 2009. "Radial Basis Collocation Method for Dynamic Analysis of Axially Moving Beams." *Interaction and Multiscale Mechanics* **2** (4): 333–52.
- [68] Wang, L., Chen J. S., Hu H. Y. 2010. "Subdomain Radial Basis Collocation Method for Fracture Mechanics." *International Journal for Numerical Methods in Engineering*, n **83**: 851–76.
- [69] Wendland, H. 1995. "Piecewise Polynomial, Positive Definite and Compactly Supported Radial Functions of Minimal Degree." *Advances in Computational Mathematics* **4**: 389–96.
- [70] Wu, C. T., Chen, J. S. Chi, L., and Huck, F. 2001. "Lagrangian Meshfree Formulation for Analysis of Geotechnical Materials." *Journal of Engineering Mechanics*, **127**: 440–449.
- [71] Zhang, X, Liu, X., Song, K., and Lu, M. 2001. "Least-Squares Collocation Meshless Method." *International Journal for Numerical Methods in Engineering* **51**: 1089–1100.