L'objectif principal de cette thèse est de contribuer au développement et applications d'une méthode sans maillage qui se base sur le principe des moindres carrés mobiles (MLS). Dans le but de se libérer des problèmes dus à la singularité de la matrice moment dans l'approximation MLS, nous construisons une approximation des moindres carrés mobiles modifiée (MMLS) et une approximation des moindres carrés mobiles régularisée (RMLS) et nous appliquons ces méthodes pour résoudre différentes classes des équations intégrales et intégro-différentielles dans des domaines rectangulaires et non rectangulaires et en dimension deux et trois.

Dans la première partie de cette thèse, nous avons développé une méthode sans maillage qui se base sur le principe des moindres carrés mobiles modifié et nous l'avons appliqué pour résoudre les équations intégrales mixtes du second type en dimension deux et trois. Dans le Chapitre 2, nous appliquons cette nouvelle approche pour la résolution des équations intégrales mixtes linéaires et non linéaires de Fredholm-Volterra, et aussi pour la résolution des équations intégrales de Fredholm-Hammerstein dans des domaines non rectangulaires. Étant donné que les méthodes sans maillage sont très efficaces en raison de leur indépendance de dimension, dans le Chapitre 3, nous étendons cette méthode à la dimension trois pour la résolution numérique des équations intégrales linéaires et non linéaires de Volterra-Fredholm de deuxième type. Une étude d'erreur a été faite et des tests numériques sont présentés pour mettre en évidence le gain en précision des résultats en comparaison avec l'approche MLS classique et pour montrer la fiabilité de notre approche.

La deuxième partie concerne l'étude numérique des équations intégrales stochastiques et des équations intégro-différentielles fractionnaires. Des phénomènes non déterministes apparaissent de plus en plus dans de nombreux domaines et cela peut être naturellement modélisé par des équations intégrales stochastiques qui dépendent généralement du temps et des termes aléatoires. Étant donné que la résolution analytique des équations intégrales stochastiques est très difficile et n'est pas disponible dans la plupart des cas, il est donc indispensable de donner des approximations numériques. Dans le Chapitre 4, une nouvelle adaptation du schéma basé sur l'approximation des moindres carrés mobiles régularisée (RMLS) est utilisée pour résoudre des équations intégrales stochastiques en dimension deux. Cette approche est proposée pour éviter la singularité de la matrice moment qui apparaît dans la méthode MLS classique. Le but du calcul fractionnaire est de généraliser les dérivées d'ordre entier à d'ordre non-entier. L'étude des problèmes fractionnaires est d'actualité, et de plus en plus les recherches se concentrent sur l'étude des équations fractionnaires stochastiques en raison de leur applicabilité pour modéliser des trajectoires aléatoires. Dans le Chapitre 5, nous utilisons l'approximation des moindres carrés mobiles (MLS) pour résoudre certaines classes des équations différentielles fractionnaires stochastiques et pour estimer l'intégrale singulière et stochastique obtenue dans le schéma, nous avons utilisé la formule composite de Gauss Legendre et la somme de Riemann. Des exemples numériques sont traités pour valider notre approche.

Mots clés: Approximation des moindres carrés mobiles modifiée (MMLS), approximation des moindres carrés mobiles régularisée (RMLS), équation intégrale, équation intégro-différentielle, équation intégrale stochastique, équation intégro-différentielle stochastique fractionnaire, estimation d'erreur.

N° d'ordre

Zahra El Majouti

Adaptive moving least squares meshless methods and its applications for solving a class of mixed integral equations and integro-differential equations

Année, 2021/2022

Formation Doctorale: Mathématiques, Informatique et Applications

جامعة الحسن الأول
UNIVERSITÉ HASSAN 1er

كلية العلوم والتقنيات سطات
FACULTÉ DES SCIENCES ET TECHNIQUES SETTAT

Université Hassan 1er

Centre d'Études Doctorales en Sciences et Techniques & Sciences Médicales

Faculté des Sciences et Techniques Settat

THÈSE DE DOCTORAT

Pour l'obtention de grade de Docteur en Sciences et Techniques

Formation Doctorale: Mathématiques, Informatique et Applications

Spécialité: Mathématiques Appliquées

Sous le thème

Adaptive moving least squares meshless methods and its applications for solving a class of mixed integral equations and integro-differential equations

Présentée par :

Zahra El Majouti

Soutenue le: ....27\06\2022...........

A la Faculté des Sciences et Techniques de Settat devant le jury composé de :

Mr. ZAKKARI Mohammed	**Professeur de l'Enseignement Supérieur** Université Hassan 1er, FST Settat	**Président**
Mr. MANIAR Lahcen	**Professeur de l'Enseignement Supérieur** Université Cadi Ayyad, FS semlalia Marrakech	**Rapporteur**
Mr. ALAOUI Mohammed	**Professeur de l'Enseignement Supérieur** Université Hassan 1er, FST Settat	**Rapporteur**
Mr. HANINI Mohamed	**Professeur Habilité** Université Hassan 1er, FST Settat	**Rapporteur**
Mr. FAKHAR Rachid	**Professeur Habilité** Université Sultan Moulay Slimane, FP Khouribga	**Examinateur**
Mr. HAJJAJ Abdelkarim	**Professeur de l'Enseignement Supérieur** Université Hassan 1er, FST Settat	**Co-Encadrant**
Mr. El JID Rachid	**Professeur Habilité** Université Hassan 1er, FST Settat	**Directeur de thèse**

Année Universitaire: 2021/2022

FICHE PRÉSENTATIVE DE LA THÈSE

1. **Nom et Prénom de la doctorante :** El Majouti Zahra
2. **Intitulé du travail:** Adaptive moving least squares meshless methods and its applications for solving a class of mixed integral equations and integro-differential equations.
3. **Encadrant :** El Jid Rachid, Professeur à la Faculté des Sciences et Techniques, Settat, Maroc.
4. **Co-encadrant :** Hajjaj Abdelkarim, Professeur à la Faculté des Sciences et Techniques, Settat, Maroc.
5. **Lieux de réalisation des travaux (laboratoires, institution,...) :** Laboratoire de Mathématiques, Informatique et Sciences de l'ingénieur, Faculté des Sciences et Techniques, Settat, Maroc.
6. **Période de réalisation du travail de thèse :** Décembre 2018 - Mars 2022.

Liste des publications

1. Z. El Majouti, R. El Jid, and A. Hajjaj. *Numerical solution of two-dimensional Fredholm–Hammerstein integral equations on 2D irregular domains by using modified moving least-square method.* International Journal of Computer Mathematics 98(8) (2021): 1574-1593. https://doi.org/10.1080/00207160.2020.1834089

2. Z. El Majouti, R. El Jid and A. Hajjaj. *Solving two-dimensional linear and nonlinear mixed integral equations using moving least squares and modified moving least squares methods.* International Journal of Applied Mathematics. 51(1) (2021).

3. Z. El Majouti, R. El Jid and A. Hajjaj. *Numerical solution for three-dimensional nonlinear mixed Volterra–Fredholm integral equations via modified moving least-square method.* International Journal of Computer Mathematics (2021):1-19.
https://doi.org/10.1080/00207160.2021.2014053

4. Z. El Majouti, R. El Jid and A. Hajjaj. *A meshless method for solving two-dimensional stochastic Fredholm integral equations using regularized moving least squares approximation.*Applied Numerical Mathematics, (submitted paper).

5. Z. El Majouti, R. El Jid and E. Taghizadeh. *Moving least squares scheme to approximate the solution of fractional stochastic integro-differential equations.*Applied Mathematics and Computation, (submitted paper).

Communications orales

- Z. El Majouti, R. El Jid, A. Hajjaj, Numerical solution of two-dimensional stochastic Fredholm integral equations on hypercube domains via meshless method based on moving least squares. *International Conference on Mathematics and Data Science (ICMDS) held on 28-30 October 2021, Khouribga..*

- Z. El Majouti, R. El Jid, A. Hajjaj, Numerical solution of fractional integro-differential equations via meshless collocation method based on MLS approximation.
Yarmouk mathematics conference (DEAMN). 18-20 September 2021, Jordan.

- Z. El Majouti, R. El Jid and A. Hajjaj, A meshfree method for solving three-dimensional nonlinear mixed integral equations on the hypercube domains, *8ème edition de la journée Doctorant du CEDoc STSM.* 08 juillet 2021, Settat.

- Z. El Majouti, R. El Jid, A. Hajjaj, Numerical solution of two-dimentional Fredholm-Hammerstein integral equation on 2D irregular domain by using modified moving least square method. *Journée des mathématiques et applications*, 10 Juin 2021, Settat.

- Z. El Majouti, R. El Jid, A. Hajjaj, Application of meshless method for solving Fredholm-Hammerstein integral equation on 2D irregular domains, *International Conference on numerical analysis and optimization days (Jano'13).* 22-24 February 2021, Khouribga.

- Z. El Majouti, R. El Jid, A. Hajjaj, A comparison between moving least squares(MLS) and modified moving least squares (MMLS) methods for solving integral equations, *Séminaire des mathématiques de Ahmed Intissar*, 14 Janvier 2020, Settat.

- Z. El Majouti, R. El Jid, A. Hajjaj, The discrete collocation method for solving two-dimensionnal non linear integral equations on non-rectangular domains, *6éme Congrée Internationale pour la societé marocain des mathématiques aplliquées (SM2A)*, 07-09 Novembre 2019, Beni Mellal.

- Z. El Majouti, R. El Jid, A. Hajjaj, A meshless method based on the moving least squares (MLS) approximation for the numerical solution of two-dimensional linear integral equations of the second kind on non-rectangular domain, *1éme Conférence Internationale en mathématiques appliques et Informatiques (ICRAMCS)*, 29-30 Mars 2019, Faculty of Sciences Ben M'sik, Casablanca.

- Z. El Majouti, R. El Jid, A. Hajjaj, Numerical solution of Volterra-Fredholm integral equation by moving least square method, *3ème édition de la Conférence Internationale Journées Scientifiques en Sciences Appliquées (JSSA'19)*, 15-16 Février 2019, Larache.

- Z. El Majouti, R. El Jid, A. Hajjaj, Approximation of Volterra-Fredholm integral equation by moving least square method, *Journée Nationale de Mathématiques à la mémoire du Professeur Ahmed Intessar*, 24 Novembre 2018, Settat.

- Z. El Majouti, R. El Jid, A. Hajjaj, Moving least squares and Gauss-Legendre for solving the integral equations of of the second kind, *International conference: Numerical Analysis and Optimization Days (JANO'12)*, 18-20 Octobre 2018, ENSAH, Al Hoceima.

Communications affichées

- Z. El Majouti, R. El Jid, A. Hajjaj, A meshless based method for solution of integral equations, *7ème edition de la journée doctorant du CEDoc STSM.* 02 Mai 2019, Settat.

Remerciement

Je tiens tout d'abord à remercier ALLAH le tout puissant de m'avoir donné la santé et la patience pour accomplir cette thèse.
J'adresse de chaleureux remerciements à mon directeur de thèse le Pr. Rachid El jid, il m'a fait découvrir le monde de la recherche et transmis leur passion dans ce domaine. Je lui adresse ma gratitude pour sa gentillesse, sa bonne volonté, sa disponibilité, ainsi que ses orientations et ses guidances avisés et son soutient indéfectible durant la préparation de cette thèse, dès le début de sa confiance à mon égard et à mon travail, il m'a donné une énergie et une inspiration de surmonter toutes les difficultés.
J'adresse mes remerciements les plus sincères à mon co-encadrant le Pr. Abdelkarim Hajjaj pour ses conseils avisés, sa sympathie et de m'avoir encouragé tout au long de cette aventure.
Je remercie très chaleureusement le Pr. Mohamed Hanini d'avoir accepté de rapporter ma thèse, de son intérêt pour mon travail, ainsi que pour ses remarques positives et encourageantes et aussi tous ses conseils sur le plan scientifique et administratif. Je tiens aussi à remercier le Pr. Mohammed Alaoui et le Pr. Lahcen Maniar qui m'ont fait l'honneur d'accepter de rapporter ma thèse. Je leurs suis très reconnaissante d'avoir bien voulu me consacrer leur temps et leur expertise.
J'adresse mes sincères remerciements au Pr. Mohamed Zakari, qui m'a fait l'honneur d'accepter de présider ce jury. J'associe à ces remerciements aussi le Pr. Rachid Fakhar pour avoir accepté d'examiner ce travail.
Un grand merci au Pr. Elham Taghizadeh pour son implication dans cette thèse, et d'avoir partagé son expérience avec moi au cours de longues discussions, son soutien été précieux.
Les années de thèse ne se résument pas à un unique travail de recherche, je tiens à mentionner que j'ai aussi pris beaucoup de plaisir à effectuer ma mission d'enseignement au sein du Département de Mathématiques et Informatique de la FST Settat et voudrais à ce titre remercier les professeurs pour qui j'ai effectué des TD et TP, le Pr. Rachid El Jid et le Pr. Fatima Aqel. Ce fut un véritable plaisir de travailler avec eux et j'en garderai d'excellents souvenirs. Je les remercie énormément pour cette expérience.
Je tiens à remercier tous mes collègues doctorants du Laboratoire MISI et IR2M pour leur aide, services rendus et leur bonne humeur, plus particulièrement, Hiba allah et Amal, pour nos vives discussions et les bons moments que nous avons partagé ensemble!
Cette thèse n'aurait pas du tout été possible sans le soutien inconditionnel et constant de mes fidèles amies Roukia et Khadija, à qui j'adresse un remerciement tout spécifique.
Enfin un remerciement particulier à mes parents, mes frères et mes sœurs, pour leur soutien sans lesquels je n'en serais pas là aujourd'hui.

Abstract

Based on the meshless method, this thesis aims to ameliorate the concept of moving least square (MLS) method and develop the approximations for cases when the MLS quadratic base functions fail because of a non invertible moment matrix. These schemes are applied for solving a class of integral equations and integro-differential equations to prove the theoretical findings.

The first part consists of implementing the modified moving least square method (MMLS) on two- and three-dimensional mixed integral equations. In Chapter 2, the modified MLS method is proposed to approximate the solution of two-dimensional linear and nonlinear Fredholm-Volterra integral equations. Since the MMLS method does not depend on the geometry of the domain, we employed some irregular domains and we solve two-dimensional Fredholm-Hammerstein integral equations of the second kind. Since the meshless methods perform very efficiently because of their independence of dimension, in Chapter 3, we expand the three-dimensional modified moving least square method for the numerical solution of three-dimensional linear and nonlinear Volterra-Fredholm integral equations of the second kind. The numerical experiments of the MMLS and classical MLS techniques are presented to show the difference between both methods.

The second part concerns the numerical study of stochastic integral equations and fractional integro-differential equations. Non deterministic phenomena appear increasingly in many fields and they can be naturally modeled by stochastic integral equations which are usually dependent on time and a random factor. Since solving analytically stochastic integral equations is very difficult and not available in some cases, it is essential to give their numerical solutions. In Chapter 4, a new adaptive scheme based on the regularized MLS approximation is employed to solve stochastic integral equations. This approach is proposed for handling a singular moment matrix in the context of MLS method. Recently, more and more researchers focus on stochastic fractional equations due to their applicability to model the memory and randomness of many noise systems problems. In Chapter 5, we develop moving least square approximation to solve stochastic fractional integro-differential equations. To establish the scheme; we apply the composite Gauss-Legendre integration rule to compute the singular-fractional integral giving in the scheme and the Riemann sum for estimating Itô integral.

Keywords: Regularized moving least squares approximation, modified moving least squares approximation, integral equations, integro-differential equations, stochastic integral equations, fractional stochastic integro-differential equations, error estimates.

Résumé

L'objectif principal de cette thèse est de contribuer au développement et applications d'une méthode sans maillage qui se base sur le principe des moindres carrés mobiles (MLS). Dans le but de se libérer des problèmes dus à la singularité de la matrice moment dans l'approximation MLS, nous construisons une approximation des moindres carrés mobiles modifiée (MMLS) et une approximation des moindres carrés mobiles régularisée (RMLS) et nous appliquons ces méthodes pour résoudre différentes classes des équations intégrales et intégro-différentielles dans des domaines rectangulaires et non rectangulaires et en dimension deux et trois.

Dans la première partie de cette thèse, nous avons développé une méthode sans maillage qui se base sur le principe des moindres carrés mobiles modifié et nous l'avons appliqué pour résoudre les équations intégrales mixtes du second type en dimension deux et trois. Dans le Chapitre 2, nous appliquons cette nouvelle approche pour la résolution des équations intégrales mixtes linéaires et non linéaires de Fredholm-Volterra, et aussi pour la résolution des équations intégrales de Fredholm-Hammerstein dans des domaines non rectangulaires. Étant donné que les méthodes sans maillage sont très efficaces en raison de leur indépendance de dimension, dans le Chapitre 3, nous étendons cette méthode à la dimension trois pour la résolution numérique des équations intégrales linéaires et non linéaires de Volterra-Fredholm de deuxième type. Une étude d'erreur a été faite et des tests numériques sont présentés pour mettre en évidence le gain en précision des résultats en comparaison avec l'approche MLS classique et pour montrer la fiabilité de notre approche.

La deuxième partie concerne l'étude numérique des équations intégrales stochastiques et des équations intégro-différentielles fractionnaires. Des phénomènes non déterministes apparaissent de plus en plus dans de nombreux domaines et cela peut être naturellement modélisé par des équations intégrales stochastiques qui dépendent généralement du temps et des termes aléatoires. Étant donné que la résolution analytique des équations intégrales stochastiques est très difficile et n'est pas disponible dans la plupart des cas, il est donc indispensable de donner des approximations numériques. Dans le Chapitre 4, une nouvelle adaptation du schéma basé sur l'approximation des moindres carrés mobiles régularisée (RMLS) est utilisée pour résoudre des équations intégrales stochastiques en dimension deux. Cette approche est proposée pour éviter la singularité de la matrice moment qui apparaît dans la méthode MLS classique. Le but du calcul fractionnaire est de généraliser les dérivées d'ordre entier à d'ordre non-entier. L'étude des problèmes fractionnaires est d'actualité, et de plus en plus les recherches se concentrent sur l'étude des équations fractionnaires stochastiques en raison de leur applicabilité pour modéliser des trajectoires aléatoires. Dans le Chapitre 5, nous utilisons l'approximation des moindres carrés mobiles (MLS) pour résoudre certaines classes des équations différentielles fractionnaires stochastiques et pour estimer l'intégrale singulière et stochastique obtenue dans le schéma, nous avons utilisé la formule composite de Gauss Legendre et la somme de Riemann. Des exemples numériques sont traités pour valider notre approche.

Mots clés: Approximation des moindres carrés mobiles modifiée (MMLS), approximation des moindres carrés mobiles régularisée (RMLS), équation intégrale, équation intégro-différentielle, équation intégrale stochastique, équation intégro-différentielle stochastique fractionnaire, estimation d'erreur.

Contents

List of Figures **9**

List of Tables **11**

Abbreviation **14**

Nomenclature **15**

General introduction **16**

1 Preliminaries **19**

1.1 Theoretical foundations of meshless methods 19

1.2 Review on integral equations 33

1.3 Numerical integration 39

1.4 Basics for stochastic and fractional calculus 41

I Modified moving least squares approach for a class of integral equations on 2D and 3D domains **46**

2 The numerical solution of linear and nonlinear Fredholm-Volterra integral equations on 2D regular and irregular domains **47**

2.1 Introduction 47

2.2 Two-dimensional classical MLS approximation 49

2.3 Two-dimensional modified MLS approximation 51

2.4 Numerical scheme 52

2.5 Error estimates 56

2.6 Numerical results 59

2.7 Conclusion 76

3 An implementation of modified moving least square for solving multidimensional mixed integral equations in 3D hypercube domains **77**

3.1 Introduction 77

3.2 Three-dimensional MLS approximation 78

3.3 Three-dimensional modified MLS approximation 79

3.4 Numerical scheme 81

3.5 Error estimates 83

3.6 Numerical results 85

3.7 Conclusion 91

II A meshless methods based on moving least squares and regularised moving least squares approaches for solving stochastic integral equations and fractional stochastic differential equations 92

4 Regularized moving least squares scheme for two-dimensional stochastic Fredholm integral equations 93

4.1 Introduction 93
4.2 Regularized moving least-square approximations 94
4.3 Numerical scheme 96
4.4 Error estimates 97
4.5 Numerical results 99
4.6 Conclusion 104

5 On the numerical solution of fractional stochastic integro-differential equations of fractional order using moving least square approximation 105

5.1 Introduction 105
5.2 Numerical scheme 106
5.3 Error estimates 107
5.4 Numerical results 110
5.5 Conclusion 118

Conclusion and perspectives 119

Annex 122

7.1 Preliminary functional analysis 122
7.2 Implementation of MLS shape functions 126

Bibliography 130

List of Figures

1.1 Data fitting using least square method . . . 24
1.2 Disretization using meshless methods: nodes, domains of influence (circular shape) 27
1.3 Different domains of influence . . . 27
1.4 Different cases with a linear basis for a random distribution of nodes . . . 27
1.5 Comparison between weight (a) and shape (b) functions . . . 29
1.6 Comparison between weight (a) and shape function of the central node . . . 29
1.7 Shape function first derivatives of the central node: (a) $\phi_{,x}$ and (b) $\phi_{,y}$ . . . 30
1.8 Comparison between shape function and shape function first derivatives of the central node: (a) ϕ and (b) ϕ_x . . . 31
1.9 Discretized Brownian path $B(t)$ . . . 42

2.1 Approximation error :(a) Linear basis(MLS), (b) Quadratic basis(MLS), (c) Quadratic basis(MMLS) of Example 1 . . . 61
2.2 Approximation error :(a) Linear basis(MLS), (b) Quadratic basis(MLS), (c) Quadratic basis(MMLS) of Example 3 . . . 63
2.3 Approximation error: (a) Linear basis(MLS), (b) Quadratic basis(MLS), (c) Quadratic basis(MMLS) of Example 4 . . . 64
2.4 The consideration domain D for Example 5 . . . 65
2.5 Node distribution for Example 5 . . . 66
2.6 The consideration domain D for Example 6 . . . 66
2.7 Node distribution for Example 6 . . . 68
2.8 The consideration domain for Example 7 . . . 69
2.9 Comparison between MLS and MMLS methods for Example 7 . . . 70
2.10 Node distribution for Example 7 . . . 70
2.11 The consideration domain D for Example 8 . . . 71
2.12 Nodes distribution for Example 8 . . . 71
2.13 A comparison between weight functions for Example 8 . . . 72
2.14 The consideration domain for Example 9 . . . 73
2.15 Comparison between MLS and MMLS methods for Example 9 . . . 73
2.16 Node distribution for Example 9 . . . 74
2.17 The consideration domain D for Example 10 . . . 75
2.18 Node distribution for Example 10 . . . 76

3.1 $3 \times 3 \times 3$ uniformly distributed nodes on the cub domain and the red sphere is the support of the central node . . . 81
3.2 A comparison between MMLS and RBF(IQ) methods for Example 11 . . . 86
3.3 The consideration domain D for Examples 11,12,13,14,15 . . . 87
3.4 Regular nodes distribution for Examples 11,12,13,14,15 . . . 87
3.5 A comparison between MLS and MMLS methods for Example 12 . . . 88
3.6 A comparison between MLS and MMLS methods for Example 13 . . . 89

4.1 RMLS and MMLS shape functions of the central node . . . 95

4.2 MLS and RMLS circular support of the central node 96
4.3 Absolute error $e(x, y)$ of MMLS and RMLS methods ($n = 36$) for Example 16 . . 100
4.4 Contour plots of maximum error for MMLS and RMLS methods for Example 16 100
4.5 Absolute error $e(x, y)$ of MMLS and RMLS methods ($n = 36$) for Example 17 . 101
4.6 Contour plots of maximum error for MMLS and RMLS methods for Example 17 102
4.7 Absolute error $e(x, y)$ of MLS and RMLS methods ($n = 47$) for Example 18 . . 103
4.8 Contour plots for MMLS and RMLS methods for Example 18 103

5.1 Error distributions using different basis and $\sigma = 0$ for Example 19 113
5.2 The approximate solution using different values of σ and α for Example 19 . . . 114
5.4 The approximate solution for different values of α and $n = 10$ for Example 20 . 114
5.3 Contour plots of maximum error (nodes, the distance between two consecutive nodes) for $m = \{1, 2\}$ and $\sigma = \alpha = 0$ for Example 19 115
5.5 Contour plots of maximum error (nodes, the distance between two consecutive nodes) for $m = \{1, 2, 3\}$ for Example 21 117
5.6 Error distributions using different basis and $\sigma = 0$ for different values of m and n for Example 21 118
5.7 Comparison between the numerical solution and the exact solution for $\sigma = 0$ and $n = 15$ for Example 21. 118
7.1 Node distribution 126
7.2 MLS circular support 126
7.3 Node distribution 127
7.4 MLS circular support 128

List of Tables

1.1 Global and local weak form-based meshless methods 20
1.2 Strong form-based meshless methods. 21
1.3 Gauss–Legendre nodes and coefficients 40

2.1 Maximum errors using different values of n for Example 1 60
2.2 Maximum errors using different values of degree basis m for Example 2 62
2.3 Maximum errors using different values of n for Example 3 62
2.4 Maximum errors using different values of n for Example 4 64
2.5 Maximum errors using different values of ν and $n = 5 \times 5$ for Example 4 64
2.6 Numerical results for Example 5 with $m_1 = 2$ 65
2.7 Numerical results for Example 6 with different values of n and $m_1 = 2$ 67
2.8 Numerical results for Example 7 with different values of n and $m_1 = 2$ 69
2.9 Maximum error and rate of convergence for Example 8 using MMLS method and $m_1 = 2$ 72
2.10 Numerical results for Example 9 with different values of n and $m_1 = 2$ 73
2.11 Numerical results for Example 10 with different values of n and with $m_1 = 2$ . . 75
2.12 Numerical results for Example 10 using different values of ν and $n = 39$ 75

3.1 Numerical results for Example 11 with $m_1 = 2$ 86
3.2 Maximum error and rate of convergence for Example 11 with different values of n 86
3.3 Numerical results for Example 12 with different values of n and m_1 88
3.4 Maximum error and rate of convergence for Example 12 with different values of n and $m_1 = 2$ 88
3.5 Numerical results for Example 12 using MMLS method for different values of ν and $n = 125$ 88
3.6 Numerical results for Example 13 with different values of n and $m_1 = 2$ 89
3.7 Maximum error and rate of convergence for Example 13 with different values of n and $m_1 = 2$ 89
3.8 Numerical results for Example 14 with $m_1 = 2$ 90
3.9 Maximum error and rate of convergence for Example 14 using different values of n and $m_1 = 2$ 90
3.10 Maximum error for Example 14 using MMLS method with different values of m_1 and $n = 27$ 90
3.11 Numerical results for Example 15 with $m_1 = 2$ and $n = 27$ 91
3.12 Maximum error and rate of convergence for Example 15 with different values of n and $m_1 = 2$ 91

4.1 Maximum error for Example 16 using RMLS and MMLS methods 99
4.2 Maximum error and CPU time for Example 17 using RMLS and MLS methods 101
4.3 Maximum error and CPU time for Example 18 using RMLS and MLS methods 102

5.1 Errors using different values of n and $\sigma = 0$ for Example 19 111

5.2 Numerical results using different values of x, α, $\sigma = 1$ and $n = 4$ for Example 19 112
5.3 Errors, rates and CPU times using different values of m and n for Example 19 . 112
5.4 Errors using different values of x, α, σ and $n = 4$ for Example 20 113
5.5 Numerical results using different values of x, α, $\sigma = 1$ and $n = 4$ for Example 20 113
5.6 Errors using regular and irregular points and $m = 3$ for Example 21 115
5.7 Errors using different values of n and m for Example 21 116

Abbreviation

AO-IIMLS Adaptive Orthogonal Improved Interpolating Moving Least-Square. 49

BEM Boundary Element Method. 16

DEM Diffuse Element Method. 16

EFGM Element Free Galerkin Method. 16, 19

FDM Finite Difference Method. 16

FEM Finite Element Method. 16

GRKCM Gradient Reproducing Kernel Collocation Method. 21

HCM Hermite Collocation Method. 21

LBIE Local Boundary Integral Equation. 20

MIP Point Interpolation method. 21

MLPG Meshless Local Petrov Galerkin. 20

MLS Moving Least Squares. 17

MLSCM Moving least Squares Collocation Method. 21

MLSW Meshless Local Strong-Weak. 21

MMLS Modified Moving Least Squares. 17

MMs Meshless methods. 17

MPS Moving Particle Semi-implicit. 21

MWS Meshless Weak Strong. 19

NEM Natural Element Method. 20

PIM Point Interpolation Method. 17

PUFEM Partition of Unity Finite Element Method. 20

RBCM Reproducing Kernel. 21

RBF Radial Basis Function. 20

RKCM Reproducing Kernel Collocation Method. 21

RKPM Reproducing Kernel Particle Method. 21

RMLS Regularized Moving Least Squares. 18

SPH Smoothed Particle Hydrodynamics. 16

WLS Weighted Least Squares. 21

Nomenclature

Matrices

Φ	MLS shape functions matrix
A	MLS moment matrix
B	Matrix term in MLS
M	MMLS moment matrix
R	RMLS moment matrix
W	MLS weight functions matrix

Operators

D^{α}	Caputo fractional derivative of order α
J^{α}	Riemann-Liouville fractional derivative operator of order α
P_n	MLS collocation operator

Scalars

θ	Normalized distance
$B(t)$	Brownian motion
D	Problem domain
D_e	Local domain
R	Size of the support domain
x	x-axis coordinate
x_i	Position of a node
y	y-axis coordinate
z	z-axis coordinate
ω_p	The quadrature weights
ε_e	RMLS additional coefficients
M	composite Gauss-Legendre uniform subdivisions number
m	Polynomial basis degree
v_e	MMLS additional coefficients
h_X	The distance between two consecutive nodes
m_1	Number of points quadrature
n	Number of nodes in the local support

Vectors

a	Vector of unknown coefficients
ϕ	Shape functions vector
$\\|e\\|_{\infty}$	Maximum errors
e(x)	Absolute error
ω	Weight function vector
K	Kernel function vector
p	Polynomial basis vector
u^{ext}	Exact solution vector
u^h	Approximation vector

General introduction

Most physical phenomena in nature can be described by a set of integral equations or partial differential equations (PDEs) such as the reformulation of radiative heat transfer problems [52], finding the propagation of acoustical and elastical waves [11], the exploration of electrostatic and low frequency electromagnetic [82], the investigation of hydrodynamic interaction among elements of a polymer chain [9], the distribution of surface water waves by a vertical barrier with a gap [19]. However, only limited cases with simple geometry and boundary conditions have analytical solutions and due to the mathematical complexities, the majority cannot provide analytical solutions. Therefore, numerical methods become indispensable tools to provide approximated solutions for a wide range of problems.

The main idea of numerical simulation is to transform a complex practical problem into a simple discrete form of mathematical description, recreate and solve the problem on a computer, and finally reveal the phenomena virtually according to the requirements of the analysts. There are many numerical methods to solve these equations such as finite difference method (FDM), finite element method (FEM), boundary element method (BEM), and meshless methods. The finite element method has been used with great success in many fields and widespread applications. However, it has its shortcomings and limitation. The reliance of the method on a mesh leads to complications for certain classes of problems. Consider the modeling of large deformation processes; considerable loss in accuracy arises when the elements in the mesh become extremely skewed or compressed. The use of a mesh in modeling these problems creates difficulties in the treatment of discontinuities that do not align with element edges. Problems with distorted mesh always require remeshing which is costly and not trivial in three dimensions.

To overcome these problems meshfree or element-free method has been proposed and achieved remarkable progress in recent years. The Smoothed Particle Hydrodynamics (SPH) proposed by Gingold and Monaghan [43] and Lucy et al. [72] in 1977 is considered to be one of the earliest meshfree methods in the literature. In the 1990s, a new class of meshfree methods emerged based on the Galerkin method. Among which, the first one called diffuse element method (DEM) was proposed by Nayroles et al. [98]. Thereafter in the framework of DEM, Belytschko made some improvements and proposed the Element Free Galerkin Method [14] (EFGM). Then in the following few decades, a variety of new meshless methods have sprung up. These methods are widely implemented in solid and fluid mechanics. However, only a few meshless methods are applicable for solving integral equations and ordinary or partial differential equations. The meshless methods consist of two main steps: the approximation of unknown functions and the discretization of the PDE. The latter step has two main categories: Galerkin-based technique and the collocation approach. Since background mesh and integration are required in Galerkin-based meshless method, it leads to expensive computational cost. The collocation-based meshless method is sometimes efficient since no integration is needed, but it is difficult to solve large-scale problems due to ill-conditioned matrices.

The MLS approximation was introduced firstly in the late of 1960s by Shepard [107] as a means of generating smooth surface interpolating between specified points, then Lancaster and Salkauskas [76] present MLS approximation as another approach to construct shape functions in meshless methods. MLS method is one of the most effective meshless approaches that has implemented a significant role in numerical analysis field, and it does not require domain elements or background cells. This method allows an easy adaptation of the nodal density, then the distribution of nodes could be chosen regularly or randomly in the consideration domain. Here, we review some of the most recent works for the numerical solution of integral equations and partial differential equations utilizing the meshless methods. The MLS method has been applied for numerical solution of one and two-dimensional integral equations [83, 5], one-dimensional Volterra–Fredholm integral equations [68], one-dimensional Fredholm-Hammerstein integral equations [69], Volterra integral equations with proportional delay [70], and one-dimensional non-linear integro-differential equations [25].

In MLS method a local evaluation of the approximating function is required, and then, the compact support domain for each data point is usually chosen as a sphere box centered on the point. Each data point has an associated dilatation parameter, which describes the size of its compact support domain. If the degree of the polynomial base function increases, the MLS approximation is implemented in a more validated fashion for complex data distributions. However, it becomes more difficult to ensure the independence of the shape functions on using higher order polynomial base functions. There have been several techniques proposed for handling a singular moment matrix in the context of meshless methods, such as perturbation of nodal positions, coordinate transformation and the matrix triangularization algorithm (MTA) [71]. These techniques have been developed in the context of the point interpolation method (PIM), and therefore assume that sufficient nodes exist in the support domain; they also do not ensure the smoothness and continuity of the approximation. The Tikhonov–Miller regularization [117] is another special technique to deal with the MLS singular moments, this technique can be easily extended to weighted least squares regularization and higher degree polynomial basis functions. In the context of the Tikhonov–Miller regularization, this thesis aims to develop new adaptive schemes based on the regularized MLS approximation and modified MLS approximation for handling a singular moment matrix in the context of meshfree methods based on Moving Least Squares (MLS) approximation. These methods are applied for solving a class of mixed integral equations and integro-différentiel equations in regular and irregular domains.

Apart from this introduction, The first chapter is devoted to the mathematical tools necessary for a better understanding of the problems studied in this thesis. More specifically, in Chapter 1, we review the MMs methods including their approximations and classifications, we give a detail description of MLS approximation that will be used in the sequel. We review also the basic definitions regarding integral operator, integral equations, integro-differential equations and their classifications, numerical integrations, some basic formulations of fractional and stochastic calculus that will be used throughout thesis.

The rest of this thesis is divided into two parts.

In the first part, which is composed of two chapters, we are interested to develop a new adaptive scheme based on modified MLS approximation MMLS for the case when the MLS quadratic base functions fail because of a non invertible moment matrix. To prove the theoretical findings, the MMLS and MLS methods are extended to solve a class of integral equations in 2D and 3D domains. In Chapter 2, we apply the meshless methods based on moving least squares (MLS) and modified moving least squares (MMLS) approximations to approximate the solution of the two-dimensional linear and nonlinear Fredholm-Volterra integral equations. These schemes are applied for solving Fredholm-Hammerstein integral equations on 2D irregular domains. The modification is proposed on the quadratic base functions by imposing additional terms based on the coefficients of the polynomial base functions. This approach prevents the singular moment matrix in the context of MLS based on meshfree methods. The numerical schemes presented are simples and effective to solve mixed integral equations and its algorithm can be easily implemented. The convergence rate and the error bound of the presented method are obtained. Finally, numerical examples are included to show the validity and efficiency of the MMLS method in comparison with the MLS method and presented in various regular and irregular domains. In addition, in Chapter 3, we expand the three-dimensional modified moving least square method for the numerical solution of three-dimensional linear and nonlinear Volterra-Fredholm integral equations of the second kind. This method does not arise the difficulties for higher-dimensional problems because of the simple adaption of the MMLS and it is more flexible for most classes of multi-dimensional integral equations. The numerical experiments of the MMLS and classical MLS techniques are presented to show the difference between both methods for multi-dimensional problems. The convergence analysis is provided and some numerical tests are given to prove the applicability of this technique.

The second part consists of two chapters in which we are interested in stochastic and deterministic integral equations and differential equations. Many time-dependent processes in science have elements of randomness. Modeling such phenomena naturally requires the use of various stochastic integral equations. More recently, the development of numerical methods for numerical solution of these equations has become a field of increasing interest, since analytical solutions of these equations are not usually available. In Chapter 4, a new adaptive meshless scheme based on the regularized moving least squares approximation(RMLS) combined with Itô approximation is employed to solve two-dimensional stochastic integral equations. The computational complexity is presented to measure the usage time of the proposed approach. We also analyzed the convergence and stability of the proposed method by specific theorems. To reveal the accuracy and efficiency of the proposed method some numerical examples are included. Several real-world phenomena can be better described by a mathematical model involving fractional derivatives. This is due to two reasons: first, we can choose any real derivative/integral operator, second, as a fractional-order derivative is a non-local operator, we can model systems with long term memory, and we know that most real world data or experimental data are noisy. In many cases, it is not possible to find the exact solutions of stochastic functional equations. Therefore, the need to obtain the numerical solution of stochastic fractional integro-differential equations has increased significantly. These phenomena can be more satisfactorily modeled by various SFIDEs. Chapter 5 is devoted to approximate solution of SFDEs using MLS method, the problem solving turns into a linear system solution of equations that can be solved easily. To establish the scheme; we apply the composite Gauss-Legendre integration rule to compute the singular fractional integral appearing in the scheme and the Riemann sum for estimating Itô integral. We have also compared different basis in terms of CPU time. The convergence analysis of the proposed method is proved. In the end, several numerical tests are presented and compared with the results obtained by other methods to verify that this method is accurate and efficient.

The manuscript concludes with the results obtained and an outline of possible perspectives that are in line with the continuity of this research work.

Chapter 1

Preliminaries

In this introductory chapter we recall some notions that will be used in the sequel. In Section 1.1, a detailed literature review on meshless methods has been given and it has been shown that strong form-based collocation methods possess some attractive features, especially the MLS method which is described in detail. In Section 1.2, we illustrate different criteria of classification of integral equations, then we mention some theorems to prove existence of solutions of some kind of integral equations. Section 1.3 is devoted to introduce some fundamental results of numerical integration, especially those based on Gauss-Legendre and composite Gauss-Legendre methods. Finally, in Section 1.4, we review some basic formulations of fractional and stochastic calculus.

1.1 Theoretical foundations of meshless methods

Nowadays meshless methods (MMs) have been considered as very promising alternatives to the well known finite element method (FEM) for certain limitations: In finite element analysis, meshing consumes too much time, large deformation, large mesh distortion or element splitting may bring difficulties or even failure in numerical computation.

1.1.1 Classification

A classification of meshless methods is presented to provide a clear picture of the relation between meshless methods in general and the point collocation method. According to whether or not they use integration, the meshless methods can be divided into three categories based on

- Weak-strong form formulation (meshless weak-strong (MWS) form method).
- Strong-form formulation (collocation method).
- Weak-form formulation (element-free Galerkin method (EFGM), and meshless local Petrov-Galerkin method (MLPG)).

1.1.1.1 Weak form meshless methods

A compact integration domain, which is required in weak form for numerical integration, can be constructed globally or locally

- Global weak forms which based on background integration in the whole domain.
- Local weak forms (Local Petrov-Galerkin method) which based on background in a rather small local sub-domain and no background mesh is required.

Meshfree weak-form methods, such as the EFGM. [14], have the following advantages:

- They have very good stability and excellent accuracy.
- The Neumann boundary conditions can be naturally satisfied through the use of the weak form.

Table 1.1: Global and local weak form-based meshless methods

Local weak form		Global weak form	
Method	Basis functions	Method	Basis functions
NEM	Natural neighbour	DEM	MLS/RK
PUFEM	Local polynomials	EFGM	MLS
MLPG	MLS/RK or smooth kernel	PIM	Polynomials
LBIE	MLS	MG-RBF	RBF

In spite of the potential benefits of using weak form meshless methods (as mentioned above), there are also some following drawbacks

- The existence of integration may lead to expansive computational costs, posing considerable complexities [87]. The term "meshless" only indicates that no mesh for the approximation of the field variables.
- Most weak form meshless methods have been criticised for not actually being truly meshless as global or local background meshes have to be created to integrate the governing PDEs.
- The integral process and the introduction of the displacement boundary condition are complicated in practical operation.

The numerical integration seems to be the most significant issue which reduces the efficiency of weak form-based meshless methods as compared to the standard FEM for instance [106, 27]. An alternative approach, the strong form-based point collocation method, have been investigated to develop more efficient and accurate integration techniques for weak form meshless methods.

1.1.1.2 Strong form meshless methods

The strong form method does not need the numerical integration. Thus the background mesh even locally is not needed for the strong form methods. A typical meshfree strong-form method is the meshfree collocation method [122]. Compared with meshfree weak-form methods, meshfree strong-form methods have the following attractive advantages:

- They are truly meshless methods and background mesh is not required for both field variable approximation and integration.
- They can be applied with a simple algorithm.
- The are very efficient in constructing the final system of equations since no integration is required and shape functions are only evaluated at nodes rather than at integration points.
- They can be discretised arbitrarily in theory using points distribution.

Meshfree strong-form methods have been successfully used in fluid mechanics. However, they are often unstable and less accurate for problems governed by differential equations with Neumann (derivative) boundary conditions.

The meshfree strong-form methods and meshfree weak-form methods have their own advantages and shortcomings, and they are complementary. The next proposed methods conbined the week form with the strong form together in a proper manner to fully take their advantages and avoid their disadvantages.

Table 1.2: Strong form-based meshless methods.

Method	Basis functions
RBCM	Local RBF
MPS	WLS
RKCM	RKPM
MLSCM	MLS
HCM	Enriched MLS
GRKCM	Diffuse RKPM

1.1.1.3 Weak-strong form meshless methods

Liu and Gu [73] proposed a novel meshfree weak-strong (MWS) form method to combine the local weak form and the strong form together to fully take their advantages and avoid their disadvantages. The local weak-form is used for all the nodes that are on or near boundaries with derivative boundary conditions and the strong-form is used for all the other nodes. In the MWS method, the problem domain and its boundary is represented by a set of points or nodes. The strong form or collocation method is used for all the internal nodes and the nodes on the essential (Dirichlet) boundaries. The local weak form (Petrov-Galerkin weak form) is used for nodes on the natural (Neumann) boundaries.

The meshless local Petrov-Galerkin (MLPG) method is a representative meshless method, and is widely applied in computational mechanics [59] and fluid mechanics [75]. But, this method is necessary to execute the boundary integral operation, and it is always difficult to solve irregular domain problems. In order to remove this kind of limitation of the MLPG method, a meshless local strong-weak (MLSW) method [59] was proposed recently by Yang and Zheng. The proposed method uses the MLPG method for domain discretization, adopts the meshless intervention-point (MIP) method for imposing the natural boundary conditions, and employs a collocation method for imposing the essential boundary conditions. Thus, the boundary integral is completely eliminated, and it favours to solve all kinds of irregular domain problem. Theoretically, the MLSW method deduced by coupling algorithm, not only has inherited the advantage of the MLPG method, which is always stable and accurate for numerical solution, but also has attained the superiority of the collocation-type method, which is naturally simple and flexible to cope with the domain of complex structure. Therefore, the method realizes advantageous complementarities of the weak-form method and the strong-form method.

1.1.2 Meshless approximation

The first and most important step in meshless methods is to approximate the field functions and create shape functions of the problem from a cloud of points. In this section, various approximations for meshless methods will be recalled.

1.1.2.1 Kernel particle and reproducing kernel particle approximation

The kernel methods approximate the field function $u^h(x)$ as the convolution of the original function $u(x)$ with a kernel function in a domain ω:

$$u^h(x) = \int_{\Omega} w_h(x-y,h)u(x)d\Omega_y$$

where $u^h(x)$ is the approximation of $u(x)$, $w(x-y,h)$ is a kernel or weight function, and h is a measure of the size of the support. The kernel functions should satisfy the following conditions

- Positivity $w_h(x-y,h) > 0$ in Ω_i, where $\Omega = \cup\Omega_i$.

- Compact support $w_h(x-y,h)=0$ out of Ω_i.
- Partition of unity $\int_{\Omega} w_h(x-y,h)=1$.
- Dirac-like shape $w_h(x-s,h) \to \delta\big(\|x-s\|\big)$ when $h \longrightarrow 0$ where $\delta(s)$ is the Dirac function.
- $w_h(x-s,h)$ is monotonically decreasing.

The kernel approximation was invoked for the first time by Lucy in the smoothed particle hydrodynamics (SPH), which is the oldest meshfree method introduced firstly by Gingold and Monaghan [43] and Lucy [72] in 1977 but is now used in hydraulics [23], structure dynamics [18], solid mechanics [102].

Liu developed reproducing kernel particle method (RKPM)[74] based on SPH in order to increase the order of completeness of the approximation, a correction function $c(x,x-y)$ is introduced into the approximation:

$$u^h(x)=\int_{\Omega} c(x,x-y)w_h(x-y,h)u(x)d\Omega_y,$$

the reproducing equation should exactly reproduce polynomials and can be expressed by a linear combination of polynomial basis functions; α is the dilation parameter of the kernel function $\Phi_\alpha(x-y)$, it gives more accurate results because of the addition of the correction function.

1.1.2.2 Radial basis approximation

Radial basis functions (RBFs) were first developed by Hardy to represent topographical surfaces given sets of sparse scattered measurements [47]. The main feature of the RBF approximation is that the interpolant is a linear combination of translations of a basis function which only depends on the Euclidean distance from its center. This basis function is therefore radially symmetric with respect to its center. That is how its name radial basis function comes about. Given a set of n distinct known data points $\{x_j; u(x_j)\}$, the approximation function can be approximate by RBF as

$$u(x)=\sum_{i=1}^{n} c_j\phi\big(\|x-x_j\|\big), \tag{1.1.1}$$

the coefficient c_j must be determined as unknown, ϕ is called radial basis function, there have been several choices for the radial basis function as

- Gaussian(GA): $\phi(h)=e^{-\alpha h^2}, \quad \alpha>0.$
- Multiquadric (MQ): $\phi(h)=(h^2+\alpha^2)^{\frac{1}{2}}, \quad \alpha>0,$
- Inverse multiquadric (IMQ): $\phi(h)=(h^2+\alpha^2)^{\frac{-1}{2}}, \quad \alpha>0,$

where $h=\|x-x_i\|$ and α is a parameter for controlling the shape of functions. Eq. (1.1.1) can be reformulated as a system of linear equations of the form:

$$AC=U, \tag{1.1.2}$$

where A, C, U are defined respectively

$$A_{i,j}=\phi\big(\|x_i-x_j\|\big), \quad U=[u(x_1),u(x_2),....,u(x_n)]^T, \quad C=[c_1,c_2,....,c_n]^T.$$

If the matrix A is nonsingular the solution of the system is guaranteed. More details about the inversibility of matrix A about the different radial basis function can be found in the nice review references of [16]. Therefore, solving Eq. (1.1.2) can acquire

$$C=A^{-1}U. \tag{1.1.3}$$

RBFs have been applied to solve integral equations [123] and integro-differential equations [44]. Although the training is faster in RBF network but in the limit the basis functions become increasingly flat because of several types of radial basis functions that include a free parameter α, then the linear system to solve becomes highly ill-conditioned, and the expansion coefficients diverge.

1.1.3 Moving least-squares approximation

1.1.3.1 Least squares

Considering the well known data fitting problem where we want to find a function $u^h(x)$ fitting the data points (x_i, u_i) with $u_i = u(x_i)$. Assuming that the approximation function $u^h(x)$ is a polynomial of order m:

$$u^h(x) = a_0 + a_1 x + a_2 x^2 + ... + a_m x^m,$$

written in compact form

$$u^h(x) = p^T(x)a,$$

the parameters a can be determined by minimizing the square of difference between u_i and u^h

$$J = \sum_{i=1}^{n} [u^h(x_i) - u_i]^2 = \sum_{i=1}^{n} [p^T(x_i)a - u_i]^2,$$

differentiating with respect to a leads to the equation

$$\sum_{i=1}^{n} p(x_i)p^T(x_i)a = \sum_{i=1}^{n} p(x_i)u_i, \tag{1.1.4}$$

which allows us to solve for unknowns a, then the approximation function $u^h(x)$ is completely defined. Let us present a simple example with data given as

$$(x_1, u_1) = (1, 1.5), \quad (x_2, u_2) = (2, 2), \quad (x_3, u_3) = (3, 5.5), \quad (x_4, u_4) = (3, 6),$$

and

$$p^T(x) = [1 \quad x \quad x^2], \quad a = [a_0 \quad a_1 \quad a_2].$$

Eq.(1.1.4) becomes

$$\sum_{i=1}^{4} \begin{bmatrix} 1 & x_i & x_i^2 \\ x_i & x_i^2 & x_i^3 \\ x_i^2 & x_i^3 & x_i^4 \end{bmatrix} a = \sum_{i=1}^{4} \begin{bmatrix} 1 \\ x_i \\ x_i^2 \end{bmatrix} u_i,$$

$$\begin{bmatrix} 4 & 10 & 30 \\ 10 & 30 & 100 \\ 30 & 100 & 354 \end{bmatrix} a = \begin{bmatrix} 15 \\ 46 \\ 155 \end{bmatrix},$$

The solution is

$$a = \begin{bmatrix} -2 & \frac{2754}{195} & \frac{-155}{39} \end{bmatrix}.$$

Then the approximating is given by

$$u^h = -2 + \frac{2754}{195}x + \frac{-155}{39}x^2.$$

It is clear that this way of approximation equals the role of every data points which often gives inaccurate results if some points are more important than others.

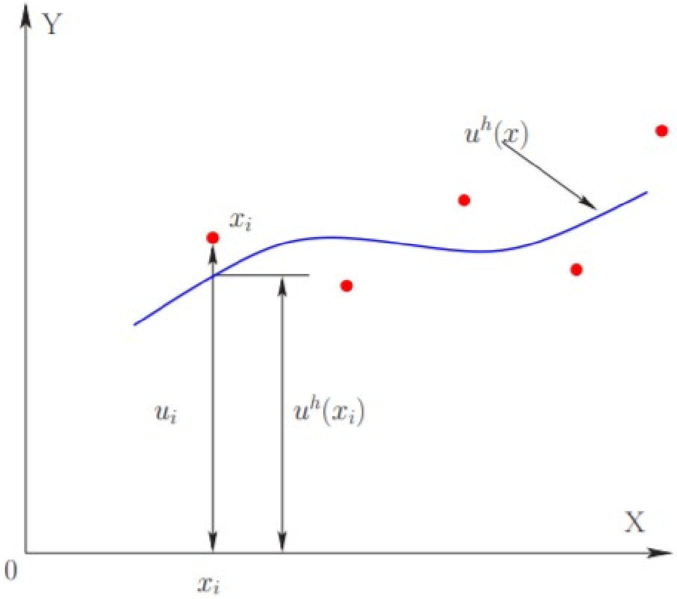


Figure 1.1: Data fitting using least square method

1.1.3.2 Weighted least square data fitting

The standard least square technique solves a large linear system and produces a global solution. MLS allow the fit to change locally depending on where the function is evaluated. To achieve this property, the solution for each point in the problem domain is solved for using weighted least square approximation. In the weighted least square estimation, the unknown a is estimated by minimizing the sum of the squared residual. Unlike least squares, however, each term in the weighted least squares approximation contain an additional weight ω, that determines how much each observation in the data set influence the final parameter estimation. and in order to improve the least square fit, we need modify the method by weighting the data in a way that emphasizes the effect of distance from a chosen point. Then we can write the weighted least square fit

$$J = \sum_{i=1}^{n} w(x_i - x_0)[u^h(x_i) - u_i]^2.$$

1.1.3.3 Moving least square method

In the moving least square method, the procedure is exactly the same as the weighted least square fit but now this procedure is applied for every points of the domain. Consider the data values of the function $u(x)$ at nodal points $X = \{(x_i\}_{i=1}^n$ in domain D. Let $u : D \to \mathbb{R}$ be a continuous real function. Let P_q be the space of polynomials of degree $q \ll n$. The local MLS approximation $u^h(\mathbf{x})$ of $u(\mathbf{x})$ around a point $\bar{x}$, can be given as

$$u_L^h(x, \bar{x}) = p^T(x)a(x), \tag{1.1.5}$$

where p(x) is a complete polynomial of order m

$$p^T(x) = [1, x, x^2, ..., x^m],$$

and $a(x)$ are unknown non constant coefficients to be determined

$$a(x) = [a_0, a_1, ..., a_m].$$

The unknown coefficients are obtained by minimizing the difference between the local approximation at that point and the nodal parameters u_i as

$$J(x) = \sum_{i=1}^{n} \omega_i(x - x_i)(p^T(x_i)a(x) - u_i)^2 = [P.a(x) - u]^T.W.[P.a(x) - u], \tag{1.1.6}$$

where $\omega_i(x - x_i)$ is the weight function associated with node i with $w_i(x - x_i) > 0$, n is the number of nodes in the neighborhood of x where the weight function $w(x - x_i) \neq 0$. and u_i are the fictitious nodal values, but not the nodal values of the unknown trial function $u^h(x)$, i.e. $u^h(x_i) \neq u_i$. The minimum of J in Eq. (1.1.6) with respect to $a(x)$ can be determined by setting the derivative of J with respect to $a(x)$ equal to zero, then we have

$$\begin{aligned}
&\sum_{i=1}^{n} w(x - x_i)2p_1(x_i)[p^T(x_i)a(x) - u_i] = 0,\\
&\sum_{i=1}^{n} w(x - x_i)2p_2(x_i)[p^T(x_i)a(x) - u_i] = 0,\\
&\vdots\\
&\sum_{i=1}^{n} w(x - x_i)2p_m(x_i)[p^T(x_i)a(x) - u_i] = 0.
\end{aligned} \tag{1.1.7}$$

In general, solving the interpolation problem based on the extended expansion (1.1.7) leads to a system of linear equations of the form

$$Aa = Bu, \tag{1.1.8}$$

Where the matrix $A(x)$ and $B(x)$ are defined respectively by

$$A(x) = \sum_{i=1}^{n} \omega_i(x)p(x_i)p^T(x_i), \tag{1.1.9}$$

$$B(x) = [w_1(x)p(x_1), w_2(x)p(x_2), ..., w_n(x)p(x_n)]. \tag{1.1.10}$$

Solving a(x) from Eq. (1.1.8), then we obtain

$$a(x) = A^{-1}Bu, \tag{1.1.11}$$

substituting Eq. (1.1.11) into Eq. (1.1.5), the MLS approximations can be defined as

$$u^h(x) = \sum_{i=1}^{N} \Phi_i(x)u_i = \Phi(x)u,$$

where Φ is the MLS shape functions given by

$$\Phi(x) = p^T(x)[A(x)]^{-1}B(x). \tag{1.1.12}$$

The matrix A(x) is often called moment matrix, it is of size $m \times m$. This matrix must be inverted wherever the MLS shape functions are to be evaluated. Obviously, this fact is the major drawback of the MMs because of computational cost and the possibility that this moment matrix is singular matrix.

Consider a linear basis in one dimension, the moment matrix then becomes

$$A(X) = \omega(x - x_1)\begin{bmatrix} 1 & x_1 \\ x_1 & x_1^2 \end{bmatrix} + \omega(x - x_2)\begin{bmatrix} 1 & x_2 \\ x_2 & x_2^2 \end{bmatrix} + ... + \omega(x - x_n)\begin{bmatrix} 1 & x_n \\ x_n & x_n^n \end{bmatrix}.$$

It is clear from this equation, if $n = 1$ i.e., the point x is covered by only one nodal support while the basis is linear (m = 2), then the matrix is singular and can not be inverted. Therefore, the condition for moment matrix to be invertible is that $n \geq m$.

1.1.3.4 MLS weighting function characteristics

An important ingredient of meshless methods is the weight function which are denoted by w_i; $\Omega \longrightarrow \mathbb{R}$. The weight functions have compact support. This weight function should be non zero only over a small neighborhood of node i which is called domain of influence of this node or support of weight function, in order to generate sparse discrete equations. Although any choice of the support shape may be possible, in practice, circles, rectangles are most frequently used and given by

- Circular support

$$\omega(x - x_i) = w\left(\frac{\|x - x_i\|}{d_i}\right). \tag{1.1.13}$$

- Rectangular support

$$\omega(x - x_i) = w\left(\frac{|x - x_i|}{d_i}\right) w\left(\frac{|y - y_i|}{d_i}\right). \tag{1.1.14}$$

The weight function should be continuous and positive in its support. The continuity of the shape function will be determined solely by the continuity of the weight function. For example, Generally, if $p \in C^m(W)$ and $w \in C^l(W)$, then the shape function $\phi \in C^{min(m,l)}(W)$. Some commonly-used weight functions are

- The Gaussian weight functions:

$$w(s) = \begin{cases} \dfrac{e^{-(\frac{s}{\alpha})^2} - e^{-(\frac{1}{\alpha})^2}}{1 - e^{-(\frac{1}{\alpha})^2}}, & \text{si } |s| \leq 1. \\ 0, & \text{si } |s| > 1. \end{cases} \tag{1.1.15}$$

  α is a constant controlling the shape of the weight function.

- The cubic spline weight function

$$w(s) = \begin{cases} \dfrac{2}{3} - 4s^2 + 4s^3, & \text{si } |s| \leq \dfrac{1}{2}. \\ \dfrac{4}{3} - 4s + 4s^2 - \dfrac{4}{3}s^3, & \text{si } \dfrac{1}{2} < |s| \leq 1. \\ 0, & \text{si } |s| > 1. \end{cases} \tag{1.1.16}$$

- The quartic spline weight function

$$w(s) = \begin{cases} 1 - 6s^2 + 8s^3 - 3s^4, & \text{si } |s| \leq 1. \\ 0, & \text{si } |s| > 1. \end{cases} \tag{1.1.17}$$

1.1.3.5 Domains of influence

We call domain of influence of node i, the adherence of the set $\Omega_i = \{x/x - x_i \geq 0\}$. Two different strategies are possible for implementing the radius of influence using in the equations (1.1.13)-(1.1.14).

1. Strategy 1 : At each evaluation point we take into account k closest nodes.
2. Strategy 2 : The domains of influence are arbitrarily fixed by assigning a radius of influence to each node.

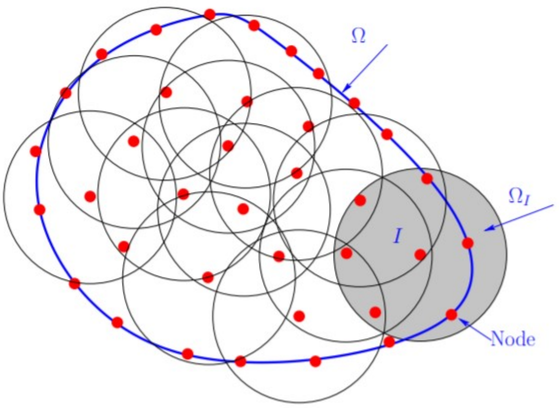
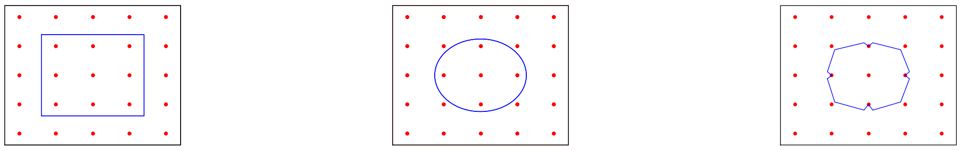


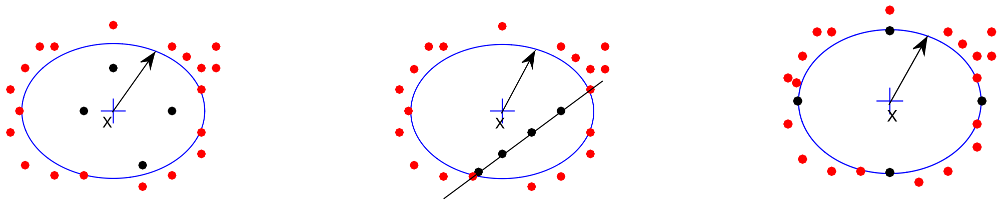
Figure 1.2: Disretization using meshless methods: nodes, domains of influence (circular shape)



Strategy 1 Strategy 1 Strategy 2

Figure 1.3: Different domains of influence

Figure 1.3 show different forms of domains of influence of the central node using various definitions of the radius of influence. In all three cases $n_l = 3$ and the radius of influence is chosen in such a way that at least 4 closest neighbors are selected. n_l is the number of terms of the polynomial basis vector l. A regular 2D grid is used in first two figures. The existence of the approximation requires a number of nodes at least equal to n_l at each evaluation point. When $n = n_l$, MLS degenerates to polynomial Lagrange interpolation and the weights have no longer effect. So, in order to guarantee the continuity, the size of the domains of influence must be adjusted. In a general case, at least $n_l + dim$ nodes are recommended at each point of the domain, where dim is the space dimension.



Will conditioned pattern with linear basis

Ill conditioned pattern with linear basis

Ill conditioned pattern with linear basis

Figure 1.4: Different cases with a linear basis for a random distribution of nodes

Figure 1.4 presents a well-conditioned case with a linear basis for a random distribution of nodes. This figure show also a particular cases where matrix A becomes singular, respectively with collinear points with a linear basis or with co circular points with a quadratic basis. These pathological situations are the limit cases in which the approximation cannot be performed. The patterns close to these singular ones may lead to ill-conditioned matrices and therefore spoil the convergence. We remark that these results do not depend on the choice of the weighting function.

1.1.3.6 MLS shape function characteristics

The properties of MLS shape functions are different then finite element method shape functions:

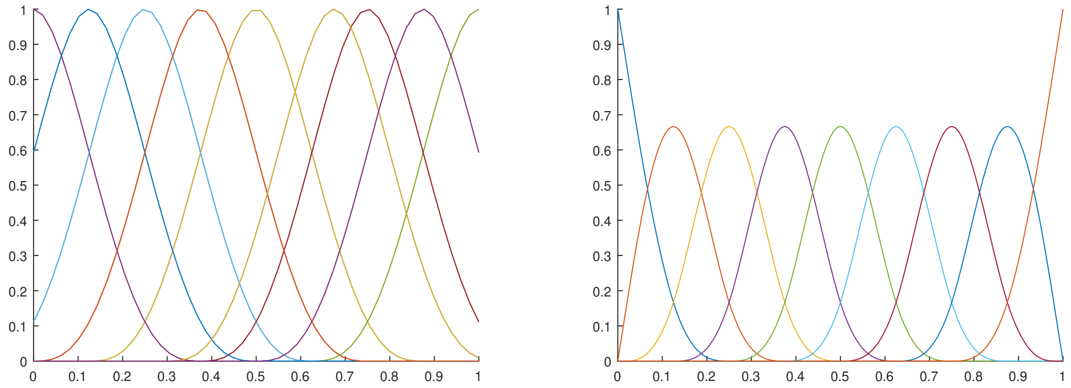
- **Absence of interpolation**; MLS shape functions do not have the Kronecker-delta property
$$\Phi_I(x_j) \neq \delta_{i,j},$$
that is why MLS approximation is definitely an approximation and not an interpolation
$$u^h(x_I) \neq u_I,$$
this lack of kronecker delta property is reason making the imposition of boundary conditions much complicated than in the FEM.

- **Consistency**: Any functions appearing in the basis can be exactly reproduced, i.e.$u^h(x) = u(x)$.
In fact, a function in the basis has the form
$$u(x) = \sum_i \alpha_i p_i(x).$$
Then if we let $a_i(x) = \alpha_i$, J in Eq. (1.1.6) will vanish and it will be a minimum since J is positive definite. Thus
$$u^h(x) = \sum_i a_i p_i(x) = \sum_i \alpha_i p_i(x) = u(x).$$

- **Partition of unity**: A partition of unity is a paradigm in which a domain is divided into overlapping patches, or subdomains Ω_I, each of which is associated with a function $\Phi_I(x)$ which is nonzero only in Ω_I and has the following partition of unity property
$$\sum_{I=1}^{n} \Phi_I(x) = 1 \quad in\, \Omega. \tag{1.1.18}$$
The zero order function $u(x) = 1$ can be reproduced exactly, this gives us
$$u^h(x) = \sum_{I=1}^{n} \Phi_I(x) u_I = \sum_{I=1}^{n} \Phi_I(x) = 1.$$
Thus, MLS shape functions are also partitions of unity.

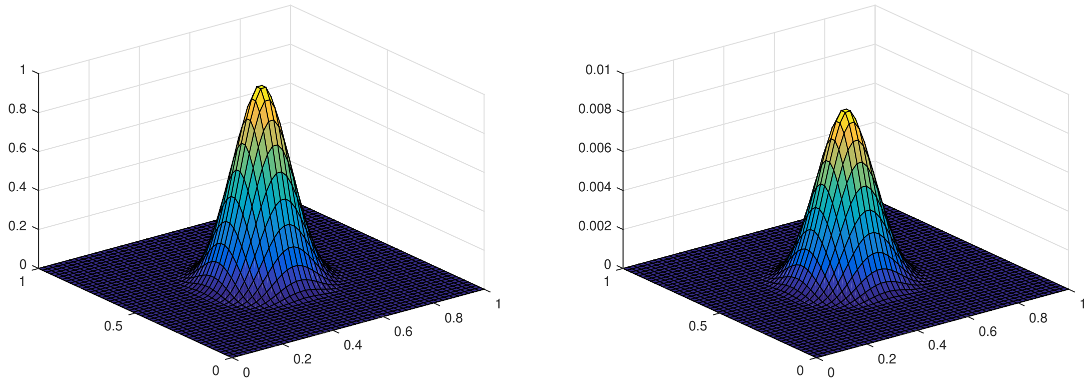
- **Continuity**: Generally, if $p \in C^m(\Omega)$ and $w \in C^l(\Omega)$, then the shape function $\phi \in C^{min(m,l)}(\Omega)$.

As an example, Fig. 3.4 shows shape and weight functions in a one dimensional domain $W = [0, 1]$ with 11 equally distributed nodes.



(a) (b)

Figure 1.5: Comparison between weight (a) and shape (b) functions



(a) (b)

Figure 1.6: Comparison between weight (a) and shape function of the central node

1.1.3.7 MLS shape function derivatives

Using Eqs. (1.1.9), (1.1.10), (1.1.12), the MLS shape function can be reformulated in the form:

$$\phi_i(x) = n^T(x)w_i(x)p(x_i), \tag{1.1.19}$$

where

$$n(x) = A(x)^{-1}p(x),$$

The first derivatives of shape functions are given by

$$\phi_{i,k}(x) = n^T_{,k}(x)w_i(x)p(x_i) + n^T(x)p(x_i)w_{i,k}(x), \tag{1.1.20}$$

where

$$n_{,k}(x) = A^{-1}_{,k}(x)p(x) + A^{-1}(x)p_{,k}(x). \tag{1.1.21}$$

The derivative of Eq.(1.1.8) implies

$$A_{,k}a + Aa_{,k} = B_{,k}u,$$

then we obtain

$$a_{,k} = A^{-1}B_{,k}u - A^{-1}A_{,k}a. \tag{1.1.22}$$

Combining Eqs (1.1.8 and (1.1.22), then we have

$$a_{,k} = A^{-1}B_{,k}u - A^{-1}A_{,k}A^{-1}Bu. \tag{1.1.23}$$

The derivative of Eq. (1.1.11) implies

$$a_{,k} = A^{-1}B_{,k}u + A^{-1}_{,k}Bu. \tag{1.1.24}$$

Finally compare Eqs. (1.1.23) and (1.1.24), then we conclude that

$$A^{-1}_{,k} = -A^{-1}A_{,k}A^{-1}. \tag{1.1.25}$$

Substituting Eq. (1.1.25) in Eq. (1.1.21), then we have

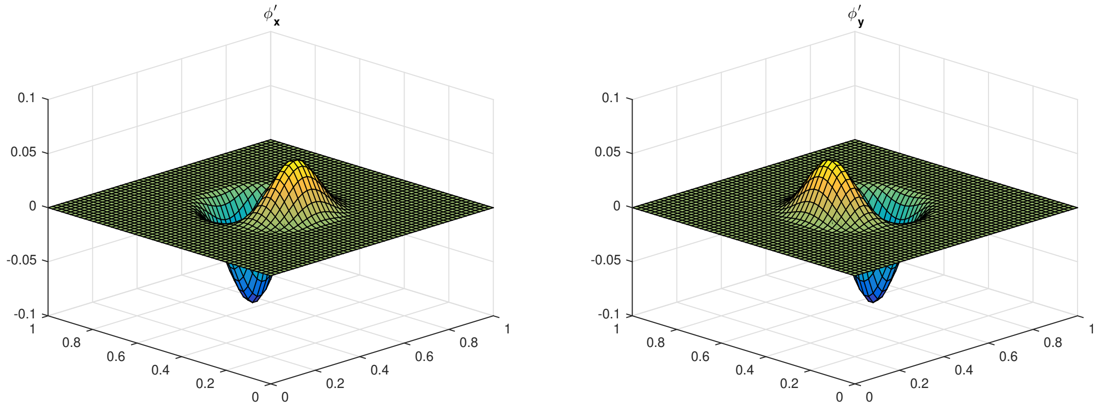
$$\begin{aligned} n_{,k}(x) &= -A^{-1}(x)A_{,k}(x)A^{-1}(x)p(x) + A^{-1}(x)p_{,k}(x), \\ &= A^{-1}(x)[-A_{,k}(x)n(x) + p_{,k}(x)], \\ &= A^{-1}(x)b_k. \end{aligned}$$

and

$$A_{i,k}(x) = \sum_{i=1}^{n} w_{i,k}(x)p(x_i)p^T(x_i),$$

$w_{i,k}$ are the derivatives of the weight functions and given by

$$w_{,k}(r) = w_{,r}(r)r_{,k} = w_{,r}\frac{x_k - x_{Ik}}{rd_I^2}.$$



(a) (b)

Figure 1.7: Shape function first derivatives of the central node: (a) $\phi_{,x}$ and (b) $\phi_{,y}$

Second derivatives

Take the derivative of (1.1.20) with respect to the lth dimension, we obtain

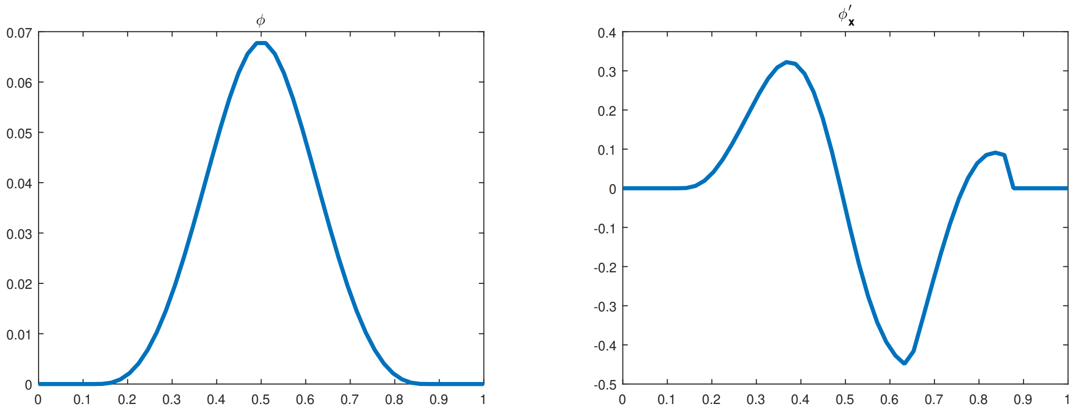
$$\begin{aligned} \phi^T_{,kl}(x) =& p^T_{,kl}(x)A^{-1}B + p^T_{,k}(x)A^{-1}_{,l}B + p^T_{,k}(x)A^{-1}B_{,l} \\ &+ p^T_{,l}(x)A^{-1}_{,k}B + p^T(x)A^{-1}_{,kl}B + p^T(x)A^{-1}_{,k}B_{,l} \\ &+ p^T_{,l}(x)A^{-1}B_{,k} + p^T(x)A^{-1}_{,l}B_{,k} + p^T(x)A^{-1}B_{,kl}. \end{aligned} \tag{1.1.26}$$

Taking the derivative of Eq. (1.1.25) with respect to the lth dimension gives

$$A^{-1}_{,kl} = -A^{-1}_{,l}A_kA^{-1} - A^{-1}A_{,kl}A^{-1} - A^{-1}A_{,k}A^{-1}_{,l}, \tag{1.1.27}$$

Using Eq. (1.1.25), substitute $A^{-1}_{,l}$ into Eq. (1.1.27) to get

$$A^{-1}_{,kl} = A^{-1}A_{,l}A^{-1}A_{,k}A^{-1} - A^{-1}A_{,kl}A^{-1} + A^{-1}A_{,k}A^{-1}A_{,l}A^{-1}.$$



(a) (b)

Figure 1.8: Comparison between shape function and shape function first derivatives of the central node: (a) ϕ and (b) ϕ_x

1.1.4 Error estimates of MLS method

This section covers the error bound which ensures the convergence analysis of the proposed method. In [77] Levin presented the error estimates in the uniform norm for a particular weight function in N dimensions, but, he did not obtain error estimates for the derivatives. In [4] Armentano and Duron studied the MLS method for the function and its derivatives in the one dimension obtaining error estimate in L^∞. Armentano in [3] presented the error estimates in L^∞ and L^2 norms for one and N dimensions which generalizes the result given in [4]. Zuppa in [129] proved the error estimates for derivatives of shape function by the condition numbers of the star of nodes in the normal equation. Authors of [103], obtained the error estimates in Sobolev space when $u(x, y) \in C^{m+1}(D)$, and $u(x, y) \in W^{m+1,q}(D)$, respectively.

Definition 1.1.1. [39] The fill distance of the nodal points $X = \{(x_1, y_1), ..., (x_n, y_n)\}$ in D can be presented as

$$h_{X,D} = \underset{(x,y)\in D}{sup} \underset{0\leq i\leq n}{min} \sqrt{(x - x_i)^2 + (y - y_i)^2}.$$

The separation distance of $X = \{(x_1, y_1), ..., (x_n, y_n)\}$ is given by

$$h_{X,D} = \frac{1}{2} \underset{i\neq j}{min} \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}.$$

The set X is considered as quasi-uniform with respect to a constant $c > 0$ if

$$q_X \leq h_{X,D} \leq c q_X.$$

Definition 1.1.2. We define the semi-norm given by

$$|u|_{q+1} = \underset{|\alpha|=q+1}{max} \|D^\alpha u\|_\infty.$$

Assuming all these definitions, Wendland proved the following Theorem about the error bound for approximating a function using the MLS approximation.

Theorem 1.1.1. *[130] Assume that $s \in C^{q+1}(D)$ and $X = \{(\mathrm{x}_1, \mathrm{y}_1), ..., (\mathrm{x}_n, \mathrm{y}_n)\}$ is a quasi-uniform set on D. Then there exists a constant $C > 0$ that can be computed explicitly such that the approximation error is defined as follows*

$$\|u - z_{u,X}\|_\infty \leq C h_X^{q+1} |u|_{q+1}, \quad h_X \leq h_0.$$

Algorithm 1 The MLS method

```
Require: Scattered points: $(x_1, x_2, x_3, ..., x_n)$
  Degree of polynomials: q
  Weight functions: $\omega_i(x), \quad i = 1, ..., n$
  for $j = 1$ to q do
    $F_j \leftarrow \sum_{k=1}^{q} a_k \sum_{i=1}^{n} \omega_i(x) p_k(x_i) p_j(x_i)$
  end for
  $\{F_j \leftarrow p_j(x), ..., F_n = p_n(x)\}$ for the unknown $\{a_1, a_2, ..., a_q\}$
  for $i = 1$ to $n$ do
    $\Phi_i(x) \leftarrow \omega_i(x) \sum_{k=1}^{q} a_k p_k(x)$
  end for
  $r(x) \leftarrow 0$
  for $i = 1$ to $n$ do
    $r(x) \leftarrow r(x) + u(x_i)\Phi_i(x)$
  end for
Ensure: $r(x)$
```

We introduce the collocation projection operator $P_n : C([0,1]) \to C_n([0,1])$ as

$$P_n u(t) = \sum_{i=0}^{n} \lambda_i \phi_i(t),$$

where $C_n([0,1]) = Span(\phi_1, ..., \phi_n)$ and the vector $V = [\lambda_1, ..., \lambda_n]$ is determined by solving the linear system

$$P_n u(t_i) = u(t_i), \quad i = 1, ..., n. \tag{1.1.28}$$

Let $l_j \in C_n([0,1])$ be elements that satisfy the interpolation conditions

$$l_j(t_i) = \delta_{ij}, \quad i = 1, ..., n.$$

There exist a unique set of functions l_j which are called Lagrange basis functions, and the set $\{l_0, ..., l_n\}$ is a new basis for $C_n([0,1])$. With this new basis we can write

$$P_n u(t) = \sum_{j=1}^{n} u(t_j) l_j(t).$$

Furthermore, in the collocation projection we need to assume that

$$\sum_{j=1}^{n} |l_j(t)| < \infty.$$

By definition of the operator P_n we have

$$\|P_n\| = \max_{t \in [0,1]} \sum_{j=0}^{n} |l_j(t)|,$$

Then P_n is uniformly bounded.

We introduced the following Theorem about the discrete projection operator with the MLS shape functions.

Theorem 1.1.2. *[130] Using the assumptions of Theorem 1.1.1. Assume the family $\{P_n, n \geq 1\}$ is uniformly bounded and $\|P_n\| \leq \delta \leq \infty$. If $u \in C^{q+1}(D)$ then $P_n u$ converge to u as $n \longrightarrow \infty$ and*

$$\|P_n u - u\|_\infty \leq (1+\delta) C h_{X,D}^{q+1} |s|_{q+1},$$

where η, C are constants.

Remark 1. The proof of uniformly bounded for the operators P_n has been discussed in [8].

1.1.5 Procedures of Meshless methods

This section will present several steps that are used for implementation of meshless methods.

- **Node distribution:** Node distribution can be performed arbitrarily over the problem domain provided the nodes are placed in the regions where the solution is required. nodes are created to discretize the domain, where randomly or uniformly distributed field nodes are arranged within the domain

- **Creating shape function of the node distribution:** The approximation function is an essential feature of the meshless method. The construction of shape functions is the central issue in meshless methods. The nodes within a created local domain or the whole domain are used for the field variable interpolation at the concerned node by suitable interpolation function as

$$u^h = \sum_{i=1}^{n} \phi_i u_i.$$

- **Disctretized system equations:** The discrete equations involved in a meshless method are formulated using the shape functions and the strong or weak form of the system equations.

- **Solving global system of equation:** These equations are often written in a nodal matrix form and are assembled into the global system matrices for the entire problem domain. The solution is obtained using the available established numerical techniques used for solving systems of algebraic equations, including direct or iterative methods such as the Gauss elimination or the QR method etc.

1.2 Review on integral equations

Many natural phenomena are studied and described mathematically by integral equations. Many initial and boundary value problems can be transformed into integral equations, they are used as mathematical models applications in various fields of science and engineering, such as diffraction theory [104], axisymmetric contact problems for bodies with complex rheology [84], the image deblurring problem and its regularization [46].

Integral equation is the equation in which the unknown function $u(x)$ appears inside an integral sign. The most standard type of integral equation in $u(x)$ is of the form

$$u(x) = f(x) + \lambda \int_{v(x)}^{w(x)} K(x, y, u(y)) dy,$$

where $v(x)$ and $w(x)$ are the limits of integration, λ is a constant parameter, and $K(x, y)$ is a known function, called the kernel of the integral equation. The unknown function $u(x)$ that will be determined appears inside the integral sign. In many other cases, the unknown function $u(x)$ appears inside and outside the integral sign. The functions $f(x)$ and $K(x, y)$ are given in advance.

1.2.1 Integral operators

First we recall the basic definitions regarding operator

Definition 1.2.1. Let $X; Y$ be normed linear spaces, and let $A : X$ be a linear operators

1. A is continuous at a point $\varphi \in X$ if $\varphi_n \longrightarrow \varphi$ in X implies $A\varphi_n \longrightarrow A\varphi$ in Y.
2. A is continuous if it is continuous at every point, i.e. If $\varphi_n \longrightarrow \varphi$ in X implies $A\varphi_n \longrightarrow A\varphi$ in Y: for every φ in X.
3. A is bounded if there exists a finite $c > 0$ such that
$$\varphi \in X, \ \|A\varphi\|_Y \leq c\|\varphi\|_X.$$
4. The operator norm of A
$$\|A\| = \sup_{\|\varphi\|=1} \|A\varphi\|.$$
5. We denote by $B(X; Y)$ the set of all bounded linear operators mapping X into Y
$$B(X, Y) = \{A : XY : A \textit{ is bounded and linear operator}\}.$$

Definition 1.2.2. Let X and Y be two normed linear spaces and $A : X \longrightarrow Y$ a linear map between X and Y. A is called a compact operator if for all bounded sets $U \subseteq X$, $A(U)$ is relatively compact in Y

Theorem 1.2.1. *Let X and Y be two normed linear spaces; suppose $A : X \longrightarrow Y$, is a linear operator. Then the following are equivalent.*

1. *A is compact.*
2. *The image of the open unit ball under A is relatively compact in Y .*
3. *For any bounded sequence φ_n in X, there exist a subsequence $\{A\varphi_{n_k}\}$ of $\{A\varphi_n\}$ that converges in Y.*

Definition 1.2.3. Let $\Omega \in \mathbb{R}^n$ a compact subset, K a continuous function from $\Omega \times \Omega$ into $\mathbb{R}$ then the linear operator defined from $C(\Omega)$ into itself by

$$A\varphi(x) = \int_\Omega K(x, y)\varphi(y)dy, \quad x \in \Omega,$$

is called integral operator, and $K(x, y)$ is the kernel of the integral operator.

A particularly class of integral operators is operators with degenerate kernels. These kernels can be decomposed into a finite sum of separable functions i.e we can write it in the form

$$K(x, y) = \sum_{i=1}^{n} a_i(x)b_i(y).$$

Theorem 1.2.2. *Let A be the integral operator from $C(\Omega)$ into itself with continuous kernel $K(x, y)$, then A is compact.*

Theorem 1.2.3. *Let*

$$A\varphi(x) = \int_\Omega K(x, y)\varphi(y)dy, \quad x \in \Omega,$$

with kernel $K \in L^2(\Omega \times \Omega)$. Then A is continuous compact operator from $L^2(\Omega)$ into itself.

Proof. First we prove that the linear operator A is bounded (continuous), let $\varphi \in L^2(\Omega)$ by the Cauchy-Schwartz inequality we get

$$\begin{aligned}\int_\Omega (A\varphi(x))^2 dx &\leq \int_\Omega \Big(\int_\Omega \mid K(x,y) \mid^2 dy\Big)\Big(\int_\Omega \mid \varphi(y) \mid^2 dy\Big) dx,\\ &\leq M \int_\Omega \mid \varphi(y) \mid^2 dy < \infty,\end{aligned}$$

with $\quad M = \int\int_{]a,b[\times]a,b[} \mid K(x,y) \mid^2 dydx.$ □

It remains to prove that A is compact, we recall that the space $L^2(\Omega)$ has a countable orthonormal basis, so we can write the kernel K as a sum of degenerate kernels $(K_n)_{n\in\mathbb{N}}$ such that

$$\lim_{x\to\infty} \|K - K_n\| = 0,$$

so we get a sequence operators

$$(A_n\varphi)(x) = \int_\Omega K(x,y)\varphi(y)dy, \quad x \in \Omega.$$

Obviously, A maps $L^2(\Omega)$ into a finite-dimensional subspace of $L^2(\Omega)$. The range of A_n is finite dimensional and hence A_n is compact. We find

$$\begin{aligned}\|(A_n - A)\varphi\| &= \int_\Omega (\int_\Omega (K_n(x,y) - K(x,y))\varphi(y)dy)^2)dx,\\ &\leq \int_{\Omega\times\Omega} |K_n(x,y) - K(x,y)|^2 dy)dx\|\varphi\|^2,\\ &= \|K_n - K\|_{L^2(\Omega\times\Omega)}\|\varphi\|^2,\end{aligned}$$

the above expression has to go to zero as $n \longrightarrow \infty$. We conclude that $A \longrightarrow A_n$ so A is compact.

Theorem 1.2.4. *Let A an integral operator with the kernel $K(x,y)$; assuming that $\|K(x,y)\| < \infty$ for $1 \leq p \leq \infty$; then A maps L^p into itself, furthermore we have*

$$\|A\varphi\|_p \leq \|K\|_p\|\varphi\|_p.$$

Proof. Let $p \in]1,\infty[$, by applying the Holder's inequality we get

$$\begin{aligned}\int_\Omega \Big(\int_\Omega |K(x,y)||\varphi(y)|dy\Big)^p dx &\leq \int_\Omega \Bigg[\Big(\int_\Omega |K(x,y)|^p dy\Big)^{\frac{p}{q}} \|\varphi\|_p^p\Bigg] dx,\\ &= \|K\|_p^p\|\varphi\|_p^p.\end{aligned}$$

then the operator A exists with

$$\|A\varphi\|_p \leq \|K\|_p\|\varphi\|_p.$$

Now we discuss the case when $p = 1$; and $p = \infty$ respectively

$$\int_\Omega\int_\Omega |K(x,y)||\varphi(y)|dydx \leq \int_\Omega\int_\Omega \sup_{y\in\Omega} ess|K(x,y)|dx \int_\Omega |\varphi(y)|dy,$$

so

$$\|A\varphi\|_1 \leq \|K\|_1\|\varphi\|_1.$$

From another side

$$\begin{aligned}\sup_{x\in\Omega} ess|A\varphi(x)| &= \sup_{x\in\Omega} ess\left|\int_\Omega K(x,y)\varphi(y)dy\right|,\\ &\leq \sup_{y\in\Omega} ess|\varphi(y)| \sup_{x\in\Omega} ess\int_\Omega |K(x,y)|dy,\end{aligned}$$

then

$$\|A\varphi\|_\infty \leq \|K\|_\infty \|\varphi\|_\infty.$$

□

1.2.2 Classification

Integral equations are classified according to many characteristics

- The limits of integration
  - The integral equation is called a Fredholm integral equation if the limits of integration are constant
  $$u(x) = f(x) + \lambda \int_a^b K(x,y,u(y))dy,$$
  - The equation is called a Volterra integral equation if one of the limits of integration is a variable
  $$u(x) = f(x) + \lambda \int_a^x K(x,y,u(y))dy,$$
- The linearity of the kernel
  - The integral equation is called linear equation if $K(x;y;u(y))$ is linear with respect to the third variable
  $$K(x,y,u(y)) = K(x,y)u(y).$$
- The integral equation is called nonlinear equation if $K(x,y,u(y))$ is nonlinear with respect to the third variable, in this case, the integral equation has two form
  - Urysohn form $\quad u(x) = f(x) + \lambda \int_a^x K(x,y,u(y))dy.$
  - Hammerstein form $\quad u(x) = f(x) + \lambda \int_a^x K(x,y)G(y,u(y))dy.$
- Placement of unknown function
  - The integral equation is called a first kind integral if the unknown function $u(x)$ appears only under the integral
  - The integral equation is called a second kind integral if the unknown function $u(x)$ appears both inside and outside integral
  - Nature of known function $u(x)$: integral equations called homogeneous integral equation if $f(x)$ is identically zero given by
  $$u(x) = \int_{v(x)}^{w(x)} K(x,y)u(y)dy.$$

1.2.2.1 Volterra-Fredholm integral equations

The Volterra-Fredholm integral equations appear in the literature in two forms

$$u(x) = f(x) + \int_x^y K(x,y)u(y)dy + \int_a^b K(x,y)u(y)dy, \tag{1.2.1}$$

and

$$u(x) = f(x) + \int_x^y \int_a^b K(x,y)u(y)dy, \tag{1.2.2}$$

where $f(x)$ and $K(x,t)$ are analytic functions. It is interesting to note that Eq.(1.2.1) contains disjoint Volterra and Fredholm integrals, whereas Eq.(1.2.2) contains mixed Volterra and Fredholm integrals.

1.2.2.2 Singular integral equation

A singular integral is where the integrand (the function being integrated) has an infinite value at one or more points within the given bounds of integration. Singular integrals appear in many areas, including fluid and solid mechanics and acoustic / electromagnetic wave scattering. Formally, a one dimensional singular integral is defined as

$$\int_a^b \frac{u(t)}{(t-s)^p}dt, \quad s \in (a,b), p > 0. \tag{1.2.3}$$

Singular integrals are classified by the order of singularity (a measure of the nature of unboundedness of the kernel).

Weakly singular: $p < 1$. A value for the integral exists and is continuous at the singularity.

Strongly singular: $p = 1$. The integrand and the integral are singular.

Hyper-singular: $p > 1$.

An important class of singular integral is the one with a Cauchy kernel, which crops up in contact and fracture problems in solid mechanics. It is defined as

$$K(s,t) = \frac{1}{s-t}, \quad s \neq t.$$

1.2.2.3 A stochastic differential equation

A stochastic differential equation (SDE) is a differential equation in which one or more of the terms is a stochastic process, resulting in a solution which is also a stochastic process. Stochastic differential equations originated in the theory of Brownian motion, in the work of Albert Einstein and Smoluchowski. Stochastic differential equations for Brownian motion were introduced by Japanese mathematician Kiyosi Itô [57], who introduced the concept of stochastic integral and initiated the study of nonlinear stochastic differential equations. The stochastic differential equation (SDE) appear in the literature in many forms

- Fredholm stochastic integral equation

$$u(x) = f(x) + \int_a^b K_1(x,t)u(t)dt + \int_c^d K_2(x,t)u(t)dB(t), \quad x \in D. \tag{1.2.4}$$

- Fredholm-Volterra stochastic integral equation

$$u(x) = f(x) + \int_a^b K_1(x)s(t)dt + \int_a^t K_2(x,t)u(t)dB(t), \quad x \in D. \tag{1.2.5}$$

- Fredholm-Volterra stochastic integro-differential equation

$$\frac{du}{dx}(x) = f(x) + \int_a^b K_1(x, yt)u(t)dt + \int_a^t K_2(xt)u(t)dB(t), \quad x \in D. \tag{1.2.6}$$

- Stratonovich Volterra integral equation

$$\frac{du}{dx}(x) = f(x) + \int_a^b K_1(x, yt)u(t)dt + \int_a^t K_2(xt)u(t)odB(t), \quad x \in D. \tag{1.2.7}$$

  the symbol o between integrand and the stochastic differential is used to indicate Stratonovich integrals. Stratonovich integrals are defined such that the chain rule of ordinary calculus holds.

Existence solutions of integral equation

Many researchers have been produced to study existence and uniqueness solutions of integral equation. many others used fixed point theory to prove the existence and uniqueness of the solution [48] [124]. Riesz theory and Fredholm alternative are also used to demonstrate the existence of solutions of integral equations. Firstly, we introduce the following corollaries as

Corollary 1.2.1. *Let A a compact operator of a normed space X into itself, for $\lambda \neq 0$ the nonhomogeneous equation*

$$T\varphi = \varphi - A\varphi = f,$$

has a unique solution $\varphi \in X$ for all $f \in X$, if and only if the homogeneous equation

$$T\varphi = \varphi - A\varphi = 0,$$

has the trivial solution $\varphi = 0$.

Proof. if the first equation has a solution for all $f \in X$, then T is surjective, as a result T injective what proves that the second equation has the unique solution $\varphi = 0$.
Furthermore if $\varphi = 0$ is not a solution for the homogeneous equation, then the homogeneous equation has a finite number $m \in N$ of solutions linearly independent, in this case either the nonhomogeneous equation is unsolvable, or has a solution given in the form

$$\varphi = \tilde{\varphi} + \sum_{i=1}^{m} \alpha_i \varphi_i,$$

where $\varphi_1, \varphi_2, ..., \varphi_m$ are arbitrary complex numbers and $\tilde{\varphi}$ a particular solution of the inhomogeneous equation. □

Corollary 1.2.2. *Let $\Omega \in \mathbb{R}^n$, and let $K(x, y)$ a continuous function. Then either homogeneous integral equations*

$$\varphi(x) - \int_\Omega K(x, y)\varphi(y)dy = 0, \quad x \in \Omega.$$

$$\psi(x) - \int_\Omega K(x, y)\psi(y)dy = 0, \quad x \in \Omega.$$

have only the trivial solution $\varphi = 0$ and $\psi = 0$ and in this case the nonhomogeneous equation

$$\varphi(x) - \int_\Omega K(x, y)\varphi(y)dy = f(x), \quad x \in \Omega.$$

$$\psi(x) - \int_\Omega K(x, y)\psi(y)dy = f(x), \quad x \in \Omega.$$

have a unique solution $\varphi \in C(\Omega)$ and $\psi \in C(\Omega)$ respectively for any $f \in C(\Omega)$ and $g \in C(\Omega)$ or the homogeneous integral equations have the same finite number $m \in N$ of solutions linearly independent, and in this case the nonhomogeneous integral equations are solvable if and only

$$\int_{\Omega} K(x,y)\varphi(y)dy = \int_{\Omega} K(x,y)\psi(y)dy = 0,$$

for all ψ solution of the adjoint homogeneous equation and for all φ solution of the homogeneous equation.

Now we consider the nonlinear integral equation, in the next theorem we well use the fixed point theory to prove that nonlinear integral equation of the second kind with bounded kernel has a unique solution for sufficiently small $|\lambda|$.

Theorem 1.2.5. *Consider nonlinear integral equation of the second kind*

$$\psi(x) - \int_{\Omega} K(x,y,\psi(y))dy = f(x), \quad x \in \Omega. \tag{1.2.8}$$

such that A is a bounded integral operator and satisfies the Lipschitz condition

$$\|A\varphi_1 - A\varphi_2\| \le c\|\varphi_1 - \varphi_2\|, \quad c \ge 0,$$

with

$$|\lambda|c < 1,$$

then Eq.(1.2.8) has a unique solution

Proof. Rewrite the nonlinear integral equation of the second kind in the form

$$\varphi = T\varphi, \tag{1.2.9}$$

with

$$T\varphi = \lambda A\varphi + f.$$

Then

$$\begin{aligned}\|T\varphi_2 - T\varphi_1\| &= \|\lambda A\varphi_2 + f - (\lambda A\varphi_1 + f)\|,\\ &= \|\lambda A\varphi_2 - \lambda A\varphi_1\|,\\ &= \|\lambda\|\|A\varphi_2 - A\varphi_1\|,\\ &\leqslant |\lambda|c|\varphi_2 - \varphi_1|.\end{aligned}$$

when $|\lambda|c < 1$, the operator T is a contraction and according to Banach fixed point theorem, there exist a unique fixed point of Eq.(1.2.9). This unique fixed point is also a solution of the nonlinear integral. □

1.3 Numerical integration

Numerical integration is one of the basic contents in numerical mathematics, and it always plays a vital role in engineering and science calculation. Numerical integration methods are introduced in detail [21, 20, 125], numerical integration is always carried out by mechanical quadrature and its basic scheme ([17]) is as follows

$$\int_a^b h(x) = \sum_{i=0}^{n} A_k h(x_i), \tag{1.3.1}$$

where $A_k > 0, k = 0, 1, ..., n$, and $x \in [a, b], k = 0, 1, ..., n$ are called coefficients and nodes for mechanical quadrature, respectively. Once the coefficients and nodes are set down, the scheme 1.3.1 can be determined.
The beginnings of numerical integration have its roots in antiquity. A prime example of how ancient these methods are is the Greek quadrature of the circle by means of inscribed and circumscribed regular polygons. This process led Archimedes to an upper bound and lower bound for the value Pi. These methods were used widely due to the lack of formal calculus. The method of the sum of an infinitesimal area over a finite range was unknown until the sixteenth century when Newton formalized the concepts of what we know now know as calculus. The earliest forms of numerical integration are similar to that of the Greek method of inscribing regular polygons into curved functions. This process broken down was taking a known area and overlapping it with an unknown area to approximate the area of the unknown shape. One could improve accuracy by choosing a better fitting shape. Later methods decided to improve upon estimating area under a curve decided to use more polygons but smaller in area. Such an example is the use of rectangles evenly spaced under a curve to estimate the area. Even further improvements saw the use of trapezoids instead of rectangles to better fit the curvature of the function being analyzed. Today, Quadrature methods are better than using any regular polygon inscribed in a function to approximate area under the curve that have a very small error, in this particular example Gaussian Quadrature

Gaussian quadrature rules

Gauss quadrature formula will only be an accurate approximation to the integral above if $f(x)$ is well-approximated by a polynomial of degree $2n - 1$ or less on $[-1, 1]$ and it can be written in the form

$$\int_{-1}^{1} h(x)dx = \sum_{i=0}^{n} w_i h(x_i),$$

where the x_i are zeros of the Legendre polynomial $P_n(x)$ of order N and the w_i are non-zero constants called "quadrature weights" defined as

$$w_i = \int_{-1}^{1} \prod_{k=1, k \neq i}^{n} \frac{x - x_i}{x_k - x_i} dx.$$

The Legendre polynomials can be defined via the recursive relation

$$P_{k+1}(x) = \frac{2k+1}{k+1} x P_k(x) - \frac{k}{k+1} P_{k-1}(x), \quad P_0(x) = 1, P_1(x) = x.$$

Some loworder rules for solving the integration problem are listed below
For integrating over a general interval $[a, b]$, a change of interval can be applied to convert the

n	Roots of $P_n(x)$	Weights w_i
2	$\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}$	1, 1
3	$-\sqrt{\frac{3}{5}}, 0, \sqrt{\frac{3}{5}}$	$\frac{5}{9}, \frac{8}{9}, \frac{5}{9}$
4	$\pm\sqrt{\frac{3-2\sqrt{\frac{6}{5}}}{7}}, \pm\sqrt{\frac{3+2\sqrt{\frac{6}{5}}}{7}}$	$\frac{18 \pm \sqrt{30}}{36}, \frac{18 \pm \sqrt{30}}{36}$

Table 1.3: Gauss–Legendre nodes and coefficients

problem to one of integrating over $[-1,1]$ in the following way:

$$\int_a^b h(t)dt = \frac{b-a}{2}\int_{-1}^{1} h(\frac{b-a}{2}s + \frac{b+a}{2})ds.$$

Applying n point Gaussian quadrature (s,w) rule then results in the following approximation

$$\int_a^b h(t)dt = \frac{b-a}{2}\sum_{i=1}^{n} w_i h(\frac{b-a}{2}s_i + \frac{b+a}{2}).$$

Composite Gauss-Legendre formulas

The singular integral in Eq.(1.2.3) can not be estimated with common numerical integration as gauss-legendre quadrature, for this reason, we use two types of composite m_n point Gauss-Legendre rules relative to the coefficients δ_p and weights ω_p in the interval [-1, 1] as follows:

- Suppose $s \in C^{2m_n[a,b]}$, for any given integer $M > 0$, we have

  $$\int_a^b h(s)ds = \frac{\Delta s}{2}\sum_{p=1}^{m_n} w_p \sum_{l=1}^{M} h(\delta_p^l),$$

  where

  $$\Delta s = \frac{b-a}{M}, \qquad \delta_p^l = \frac{\Delta s}{2}\sigma_p + (q-\frac{1}{2})\Delta s.$$

- Let $h(s)$ defined on $[a,b]$ such that

  $$|h^{2m_n}(s)| < C(a-s)^{-\alpha-2m_n}, for\ all\, t \in [0,1],\ \ and\ some\ \alpha \in [0,1].$$

  the function u only at the point $s = a$ is not continuously differentiable up to order $2m_n$ and at other points is several times continuously differentiable. Suppose δ_p are the m_n roots of the Legendre polynomial of degree m_n and weights w_p. Then, for any given integer $M > 0$, we have

  $$\int_a^b h(s)ds = \sum_{p=1}^{m_n} w_p \sum_{l=1}^{M} \frac{\Delta s_l}{2} h(\delta_p^l). \tag{1.3.2}$$

  where

  $$\delta_p^l = \frac{\Delta s_l}{2}\sigma_p + \bar{s}_l, \qquad \Delta s_l = t(s_l - s_{l-1}), \qquad \bar{s}_l = \frac{s_l + s_{l-1}}{2},$$
  $$s_l = \left(\frac{l}{M}\right)^e, and \quad e = \frac{2m_1+1}{1-\alpha}.$$

1.4 Basics for stochastic and fractional calculus

1.4.1 Basics for stochastic calculus

Stochastic calculus has come to play an important role in many branches of science and technology where day by day more and more mathematician have encountered in this field. Stochastic calculus is the area of mathematics that deals with processes containing a stochastic component and thus allows the modeling of random systems. The best-known stochastic process to which stochastic calculus is applied is the Wiener process (named in honor of Norbert Wiener), which is used for modeling Brownian motion as described by Louis Bachelier. Brownian motion $B(t)$ is a stochastic process with the following properties:

Definition 1.4.1. [65]

- $B_0 = 0$;
- The process B_t has stationary increments; for any $0 \leq s \leq t, B_t - B_s \sim B_{t-s}$.
- The process B_t has independent increments; for any $0 = t_0 \leq t_1 \leq .... \leq t_n$, $B_{t_{i+1}} - B_{t_i}$ are independents;
- The increment $B_{t+s} - B_s$ has the normal $(0, t)$ distribution; for any $t > 0, B_t \sim N(0, t)$.
- The function $t \to B(t)$ is a continuous function of t.

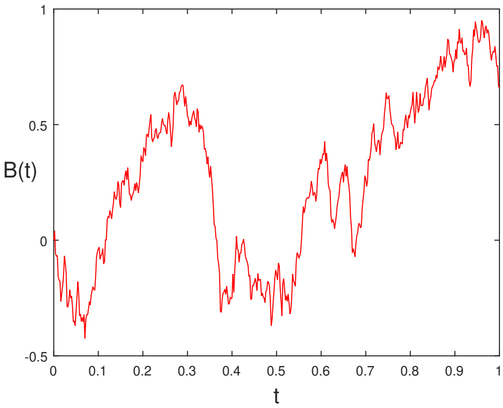


Figure 1.9: Discretized Brownian path $B(t)$

Definition 1.4.2. The stochastic Itô integral is defined as $\int_0^T u(t)dB$, and it can be approximated by Riemann sums, then the integral is given by

$$\int_0^T u(t)dB = \sum_{i=0}^{n-1} u(t_i)\Big(B(t_{i+1}) - B(t_i)\Big), \tag{1.4.1}$$

where

$t_i = idt, dt = \dfrac{T}{n}$, n is a sufficiently large number. and the approximation of the Itô integral is obtained at the left endpoint of $[t_{i+1}, t_i]$

By using the integration by parts, we have

Theorem 1.4.1. *Suppose u_s is a continuous function on $[0, T]$ with bounded variation. Then*

$$\int_0^T u_s dB_s = u_T B_T - \int_0^T B_s du_s, \tag{1.4.2}$$

where the second integral is the Stieltjes integral.

Lemma 1.4.1. *for all $T \geq 0$, we have*

$$\int_0^T B_s dB_s = \frac{1}{2}(B_T^2 - T).$$

1.4.2 Basics for fractional calculus

Fractional calculus owes its origin to a question of whether the meaning of a derivative to an integer order could be extended to still be valid when is not an integer. This question was first raised in a letter of L'Hospital to Leibniz. The prophetical answer of Leibniz to that deep question attracted the interest of many well-known mathematicians, including Riemann, Letnikov, Euler, Liouville, Laplace, and many others fractional calculus is a generalization of ordinary calculus, where derivatives and integrals of arbitrary real or complex order are defined. These fractional operators may model more efficiently certain real world phenomena, especially when the dynamics is affected by constraints inherent to the system. There exist several definitions for fractional derivatives and fractional integrals, like the Riemann-Liouville, Caputo, Hadamard, Riesz, Griinwald-Letnikov, Marchaud, etc. (see [66], [100]). In fact, one could argue that real world processes are fractional order systems in general. The main reason for the success of fractional calculus applications is that these new fractional-order models are often more accurate than integer-order ones, i.e., there are more degrees of freedom in the fractional order model than in the corresponding classical one. The applications of fractional calculus are very wide nowadays. It is safe to say that almost no discipline of modern engineering and science in general, remains untouched by the tools and techniques of fractional calculus. For example, wide and fruitful applications can be found in rheology, viscoelasticity, acoustics, optics, chemical and statistical physics, robotics, control theory, electrical and mechanical engineering, bio-engineering, etc.

1.4.2.1 The Gamma function

The Gamma function $\Gamma(x)$ is one of basic functions which plays an important role on the theory of differentiation and generalizes the ordinary definition of factorial of an integer number n and allows n to take also any non- negative integer.

The integral transform definition for the $\Gamma(x)$ is given by

$$\Gamma(x) = \int_0^\infty y^{x-1} e^{-y}\, dy, \qquad Re(x) > 0.$$

The following are the most important properties of Gamma function:

1. $\Gamma(1) = 1$.
2. $\Gamma(x+1) = x\Gamma(x)$, $\quad X$ is a non- negative integer.
3. $\Gamma(n+1) = n$.

1.4.2.2 Fractional integrals

Let defined the differentiation operator D and the integration operator J respectively as

$$Du(x) = \frac{d}{dx}u(x), \tag{1.4.3}$$

$$Ju(x) = \int_0^x u(s)\, ds. \tag{1.4.4}$$

Formally the real order generalization is introduced as follows:

$$D^\alpha$$

for every real number α in such a way that, when α takes an integer value $n \in Z$, it coincides with the usual n-fold differentiation D if $n > 0$, and with the n-th power of J when $n < 0$.

Repeating the process 1.4.4 gives

$$(J^2 u)(x) = \int_0^x \Big(Ju(t)\Big) dt = \int_0^x \Big(\int_0^t u(s)\, ds\Big) dt$$

The Cauchy formula for repeated integration, namely

$$\Big(J^n u\Big)(x) = \frac{1}{(n-1)!}\int_0^x (x-t)^{n-1}u(t)dt. \tag{1.4.5}$$

For direct use in 1.4.5 , n is restricted to be an integer. The primary restriction is the use of the fractional which in essence has no meaning for non- integer values. The gamma function is however an analytic expansion of the fractional for all reals, and thus can be used by replacing the fractional expression for its gamma function equivalent, we can generalize 1.4.5 for all $\alpha \in \mathbb{R}+$, as shown

Definition 1.4.3. The Riemann-Liouville fractional integral operator J^α of order $\alpha \geq 0$

$$J^\alpha u(t) = \frac{1}{\Gamma(\alpha)}\int_0^t \frac{u(s)}{(t-s)^{1-\alpha}}ds, t > 0.$$

with $\quad u^0(t) = u(t)$

1.4.2.3 Fractional derivatives

Let us assume that $u(x)$ is a monomial of the form

$$u(x) = x^k\,.$$

The first derivative is as usual

$$u'(x) = \frac{d}{dx}u(x) = kx^{k-1}\,.$$

Repeating this gives the more general result that

$$\frac{d^a}{dx^\alpha}x^k = \frac{k!}{(k-\alpha)!}x^{k-\alpha}\,,$$

which, after replacing the factorials with the gamma function, leads to

$$\frac{d^\alpha}{dx^\alpha}x^k = \frac{\Gamma(k+1)}{\Gamma(k-\alpha+1)}x^{k-\alpha}, \qquad k > 0.$$

In the same fashion, as in the definition of fractional integral and for a general function $u(x)$ and $0 < \alpha < 1$, the complete fractional derivative is

$$D^\alpha u(x) = \frac{1}{\Gamma(1-\alpha)}\frac{d}{dx}\int_0^x \frac{u(t)}{(x-t)^\alpha}dt$$

which is called the Riemann-Liouville fractional derivative of u of order α.
Another option for computing fractional derivatives is the Caputo fractional derivative. In contrast to the Riemann-Liouville fractional derivative, when solving differential equations using Caputo's definition, it is not necessary to define the fractional order initial conditions. Caputo's definition is illustrated as follows

Definition 1.4.4. [66] The Caputo fractional derivative of order α is defined as

$$D^\alpha u(t) = \frac{1}{\Gamma(k-\alpha)}\int_0^t \frac{u^k(s)}{(t-s)^{1+\alpha-k}}ds, \alpha > 0, \quad k-1 < \alpha < k, t > 0. \tag{1.4.6}$$

The linear property of the Caputo fractional derivative hold similar to the integer order differentiation, that is for constant numbers λ and γ, we have

$$D^{\alpha}(\lambda u(t)+\gamma g(t))=\lambda D^{\alpha}u(t)+\gamma D^{\alpha}g(t).$$

Some of most important properties of fractional operators are as follows

$$\begin{aligned}
J^{\alpha}J^{\beta}u(t) &= J^{\alpha+\beta}u(t),\\
J^{\alpha}J^{\beta}u(t) &= J^{\beta}J^{\alpha}u(t),\\
D^{\alpha}D^{\beta}u(t) &= D^{\alpha+\beta}u(t),\\
D^{\alpha}J^{\alpha}u(t) &= u(t),\\
J^{\alpha}D^{\alpha}u(t) &= u(t)-\sum_{k=0}^{n-1}\frac{t^{k}}{k!}u^{k}(0^{+}), \quad n-1<\alpha<n, n\in\mathbb{N}.
\end{aligned}$$

Part I

Modified moving least squares approach for a class of integral equations on 2D and 3D domains

Chapter 2

The numerical solution of linear and nonlinear Fredholm-Volterra integral equations on 2D regular and irregular domains

This chapter presents a comparison between moving least squares (MLS) and modified moving least squares (MMLS) approximations for solving linear and nonlinear mixed integral equations on 2D regular and irregular domains. The MMLS scheme is proposed for handling a singular moment matrix in the context of meshfree methods based on moving least squares (MLS) approximation. This modification allows quadratic base functions to be used with the same size of the support domain as linear base functions, resulting in better approximation capability. These meshfree methods don't require any background mesh or cell structures and so they are independent of the form of the consideration domain. The convergence analysis of this technique is provided. A valuable advantage of applied this new technique is that the results converge more quickly to the exact solution by using a small support domain, and it is more flexible because it allows an easy adaptation of the nodal density in non-rectangular domain. Some several numerical tests are presented and compared with the classical MLS method to prove the applicability and the reliability of this new approach.

The results obtained in this chapter are presented in the research paper [29, 30], in collaboration with Rachid El Jid and Abdelkarim Hajjaj.

2.1 Introduction

The motivation of this chapter is to solve the following class of nonlinear mixed integral equations of the second kind using the moving least squares and modified moving least squares methods, namely

- Fredholm-Volterra integral equation of the second kind on 2D regular domain

$$\begin{aligned} u(x,y) &+ \int_c^d \int_a^b K_1(x,y,s,v)\Phi_1(s,v,u(s,v))\,\mathrm{d}s\mathrm{d}v \\ &+ \int_c^y \int_a^x K_2(x,y,s,v)\Phi_2(s,v,u(s,v))\,\mathrm{d}s\mathrm{d}v = g(x,y). \end{aligned} \tag{2.1.1}$$

The functions $g(x,y)$, $K_1(x,y,s,v)$, $K_2(x,y,s,v)$ are assumed to be given smooth real valued functions on $(x,y) \in [a,b] \times [c,d]$ and $D = \{(x,y,s,v), a \leq s \leq x \leq b, c \leq v \leq y \leq$

$d\}$, respectively and $u(x, y)$ is the solution to be determined. For these types of integral equations, it is usually difficult to obtain analytical solutions and then numerical solutions have to be studied.

- Fredholm-Hammerstein integral equation of the second kind on 2D irregular domain

$$u(x,y) + \int_D K(x,y,\mu,\theta)\Phi(\mu,\theta,u(\mu,\theta))\,\mathrm{d}\mu d\theta = g(x,y). \quad x,y \in D \tag{2.1.2}$$

where u is the solution to be determined, the functions g and K are assumed to be given and D is a two-dimensional non-rectangular domain defined as

$$D = \{(\mu,\theta) \in \mathbb{R}^2, 0 \le \theta \le 1,\ \upsilon_1(\theta) \le \mu \le \upsilon_2(\theta)\},$$

where $\upsilon_1(\theta)$ and $\upsilon_2(\theta)$ are continues functions of θ. We can separate a domain to finite numbers of sub-domains

$$D = D_1 \cup_2 .... \cup D_d,$$

where $\quad D_e,\ 1 \le e \le d\quad$ are disjoint domains and then apply the method in each subdomain described as

$$D_e = \{(\mu,\theta) \in \mathbb{R}^2, 0 \le \theta \le 1,\ \upsilon_{1,e}(\theta) \le \mu \le \upsilon_{2,e}(\theta)\}, \quad e = 1, ..., d. \tag{2.1.3}$$

Throughout the paper, the following assumption is made on Φ

$$|\Phi(x,y,u_1) - \Phi(x,y,u_2)| \le C_1|u_1 - u_2|, \tag{2.1.4}$$

where Φ is Lipschitz with respect to the third variable.

These types of integral equations seem to be a good way for modeling different applications such as: plasma physics [38], approximation of implicit surfaces [108], diffraction theory [104], simulations [14, 78], and computational biomechanics [53, 60]. Some integral equations cannot be solved by the exact methods. Thus, it is desirable to present numerical methods with high performance to solve these equations numerically. It is worth remembering that, analytical and numerical analysis of one-dimensional mixed integral equations have been discussed by numerous authors [68, 85, 128]. Recently, the two-dimensional integral equation on a non-rectangular domain has been studied in only a few papers, the author in [40] applied a new spectral meshless radial point interpolation (SMRPI) method to solve two-dimensional Fredholm integral equations on general domains with error analysis, in [6] the author used a numerical method for solving linear integral equations of the second kind on the non-rectangular domains based on the moving least square MLS method.

Meshless methods have been widely used in many branches such as neural networks, surface construction, artificial intelligence, function approximation. The moving least squares (MLS) approximation as a generalization of Shepard's method [107] is developed by Lancaster and Salkauskas [76]. Among meshless methods, the MLS method has been used in many different non-rectangular domains because it does not depend on the geometry of the domain, and it does not require domain elements or background cells. For that, the distribution of nodes could be selected regularly or randomly in the analyzed domain. The new method is more adaptable and efficient to approximate the unknown function for most classes of integral equations on 2D irregular domains. The moment matrix in MLS method may be singular when the number of points in the local support domain is few. So finding good support is an open interesting research problem.

Some methods have recently been proposed in MLS based methods to avoid the singular matrix. Wang et al. [126] proposed the regularized MLS (RMLS) and the regularized IIMLS (RIIMLS) for handling a singular moment matrix in the MLS based methods, the regularization technique is based on the Tikhonov-Miller regularization for ill-posed least squares problems. Wang et al. [127] proposed an adaptive orthogonal improved interpolating moving least-square (AO-IIMLS) method. Based on IIMLS approximation, a weighted orthogonal basis functions are applied to obtain a diagonal moment matrix, which can avoid the direct inverse of the moment matrix.

In this work, a modified MLS method with nonsingular moment matrix is proposed by adding additional constraints and using convenable support. Here, the quadratic base functions (m = 6) will be used with the same size of the support domain as linear base functions (m = 3). So, the results of MMLS approximation converge more quickly to the exact solution and give better accuracy than that of MLS approximation. The first use of MMLS method was proposed in [61] for smoothing and approximating scattered. The basic advantage of the proposed method does not require any adaptation of the nodal density on a non-rectangular domain. The new technique is examined in various integral equations. We obtain the error bound and the rate of convergence.

The outline of this chapter is as follows: In section 2.2, two-dimensional classical MLS approximation is discussed. In section 2.3, the modified MLS approximation with second order polynomial basis is presented. In section 2.4, we present a computational method for solving linear and nonlinear mixed integral equations on 2D regular and irregular domains. Section 2.5 is devoted to the error estimate of the applied method. The numerical experiments are carried out in section 2.6, which will be used to verify the theoretical results obtained in section 2.5. Finally, this chapter is ended in Section 2.7.

2.2 Two-dimensional classical MLS approximation

The moving least squares (MLS) approximation as a generalization of Shepard's method [107] is developed by Lancaster and Salkauskas [76]. It is one of the meshless methods since it is based on a set of scattered points instead of interpolation on elements. We use this method to approximate two variable functions $X = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$. Let $u : D \to \mathbb{R}$ be a continuous real function and the points (x_i, y_i, u_i), $i = 1, 2, ..., n$ are known. The main point of this meshless method is to estimate a function $u(x, y)$ for every point $(x, y) \in D$ based on the weighted least square.

Let P_q be the space of polynomials of degree $q \ll n$. The MLS approximation $u^h(x, y)$ of $u(x, y)$, $\forall(x, y) \in \overline{D}$, can be given as

$$u^h(x,y) = p^T(x,y)a(x,y), \quad \forall(x,y) \in \overline{D}, \tag{2.2.1}$$

where

$$p(x,y) = [p_0(x,y), p_1(x,y), ..., p_m(x,y)]^T,$$

and $\{p_i(x, y)\}_{i=0}^m$ is a complete basis of P_q of order m and,

$$a(x,y) = [a_0(x,y), a_1(x,y), ..., a_m(x,y)],$$

are unknown coefficients to be determined. In this paper we use monomials and Chebyshev polynomials as a basis. The MLS method presents the approximate function $u^h(x, y)$ in a particularized class of differentiable functions which minimize the quantity

$$J(x,y) = \sum_{i=1}^{n} \omega_i(x,y)(p^T(x_i,y_i)a(x,y) - u_i)^2 \tag{2.2.2}$$

$$= [P.a(x,y) - u]^T.W.[P.a(x,y) - u],$$

where $\omega_i(x, y)$ is the weight function associated with node i, (x_i, y_i) denotes the value of (x, y) at node i, n is the number of nodes in $\overline{D}$ with $w_i(x, y) > 0$ and u_i are the fictitious nodal values, but not the nodal values of the unknown trial function $u^h(x, y)$ i.e. $u^h(x_i, y_i) \neq u_i$. The matrix P and W are defined as

$$\begin{aligned} P &= [p^T(x_1, y_1), p^T(x_2, y_2), ..., p^T(x_n, y_n)]^T_{n\times(m+1)} \\ W &= diag(\omega_i(x, y)), \quad i = 1, 2, ...n. \end{aligned}$$

A necessary condition for $J(x, y)$ to be minimized is $\nabla J = 0$, which implies the following normal equation

$$\sum_{i=1}^{n} \omega_i(x, y)p(x_i, y_i)p^T(x_i, y_i)a(x, y) = \sum_{i=1}^{n} \omega_i(x, y)p(x_i, y_i)u_i. \tag{2.2.3}$$

Using the moment matrix

$$A(x, y) = \sum_{i=1}^{n} \omega_i(x, y)p(x_i, y_i)p^T(x_i, y_i),$$

and setting

$$u = [u_1, u_2, ..., u_n]^T,$$

and

$$\begin{aligned} B(x, y) = [w_1(x, y)p(x_1, y_1), w_2(x, y)p(x_2, y_2), \\ ..., w_n(x, y)p(x_n, y_n)], \end{aligned}$$

(2.2.3) becomes as follows

$$A(x, y)a(x, y) = B(x, y)u, \tag{2.2.4}$$

and by selecting the nodal points such that A(x) is nonsingular, (2.2.4) can be written as

$$a(x, y) = A^{-1}(x, y)B(x, y)u. \tag{2.2.5}$$

Substituting (2.2.5) into (2.2.1) we obtain

$$u^{\rho}(x, y) = p(x, y)^T A^{-1}(x, y)B(x, y)u = \sum_{i=1}^{n} \phi_i(x, y)u_i, \tag{2.2.6}$$

where

$$\phi_i(x, y) = \sum_{k=1}^{m} p_k(x, y)[A^{-1}(x, y)B(x, y)]_{ki},$$

$\phi_i(x, y)$ are called the shape functions of the MLS approximation, corresponding to the nodal point (x_i, y_i). If $w_i(x, y) \in C^r(D)$ and $p_k(x, y) \in C^s(D), i = 1, ..., n, \ k = 1, ..., m$ then $\phi_i(x, y) \in C^{min(r,s)}(D)$. In two-dimensional case, if we choose $n = 2$ that means the point is covered by only two nodal support while the basis is linear (m = 3), then the matrix can not be inverted. Therefore, the condition for moment matrix to be not singular is that $n > m$. Now, for the quadratic basis(m=6), the moment matrix is non-singular if there are at least 6 non-collinear nodes in the support domain. Nevertheless, some nodal distributions can still lead to singular moment matrices even if enough nodes are included in the support domain, for example, if the nodes are collinear or the nodes are distributed on two parallel lines in 2D. For these reasons, the nodes have to be arranged in different coordinate directions, also we can enlarge the support domains to include more nodes. Despite that, this solution leads to higher approximation error. In this sense, Joldes [61] proposed a modified MLS with second order polynomial basis.

2.3 Two-dimensional modified MLS approximation

The proposed MMLS method will avoid singular moment matrix in the context of MLS based on meshless methods. This modification allows quadratic base functions to be used with the same size of the support domain as linear base functions by adding additional terms based on the coefficients of the polynomial base functions, leading to have a better approximation capability.

- The coefficients of the quadratic basis is defined as $p(x) = [1, x, y, xy, x^2, y^2]^T$.
- The unknown coefficients a_j are defined as $a(x) = [a_0, a_x, a_y, a_{xy}, a_{x^2}, a_{y^2}]^T$.

The moment matrix is singular means that the classical MLS minimization has multiple solutions. So, the functional (2.2.2) does not include sufficient constraints to assure a unique solution for the given node distribution. To overcome this problem, we propose to add additional constraints to the functional (2.2.2). Therefore, we obtain

$$\bar{J}(x, y) = \sum_{i=1}^{n} \omega_i(x, y)[u^\rho(x, y) - u_i]^2 + \nu_{x^2} a_{x^2}^2 + \nu_{xy} a_{xy}^2 + \nu_{y^2} a_{y^2}^2 \tag{2.3.1}$$

where $\nu = [\nu_{x^2} \quad \nu_{xy} \quad \nu_{y^2}]$ is a vector of positive weights for the additional constraints. The choice of the additional constraints ensures that, when the classical MLS moment matrix is singular, we choose the solution having the coefficients for the higher order monomials in the bases equal to zero. It assures that a solution with zero coefficients for some higher order monomials is selected when the classical MLS moment matrix has multiples solutions. By selecting ν as a small positive number we can guarantee that the classical MLS solution is little changed and the moment matrix is nonsingular. The modified matrix and the matrix form are

$$\bar{J}(x, y) = [P.a - u_i]^T.W.[P.a - u_i] + a^T M a, \quad i = 1, 2, ..., n \tag{2.3.2}$$

and

$$\mathbf{M} = \begin{bmatrix} \mathbf{O}_{3,3} & \mathbf{O}_{3,3} \\ \mathbf{O}_{3,3} & diag(\nu) \end{bmatrix}$$

where $\mathbf{O}_{3,3}$ is the zero matrix and the last three diagonal entries equal to ν.
By minimizing the functional (2.3.2), the coefficients $a(x, y)$ will be determined by

$$\bar{A}(, y)a(x, y) = B(x, y)u_i,$$

where

$$\bar{A} = P^T.W.P + M. \tag{2.3.3}$$

The modified approximation can be written as follows

$$\bar{u}^h(x, y) = \sum_{i=1}^{n} \bar{\phi}_i(x, y)u_i, \quad (x, y) \in \bar{D}, \tag{2.3.4}$$

with the MMLS shape functions defined by

$$\bar{\Phi}(x, y) = [\bar{\phi}_1(x, y), \bar{\phi}_2(x, y)...\bar{\phi}_n(x, y)] = p^T(x, y)(P^T.W.P + M)^{-1}B(x, y).$$

Lemma 2.3.1. *The moment matrix $\bar{A}$ defined in (2.3.3) is nonsingular.*

Proof. (2.3.3) can be formulated as follows

$$\bar{A} = \bar{P}^T \bar{W} \bar{P}, \tag{2.3.5}$$

where the matrices $\bar{W}$ and $\bar{P}$ are defined as

$$\bar{W} = \begin{bmatrix} W & \mathbf{O}_{3,3} \\ \mathbf{O}_{3,3} & diag(\nu) \end{bmatrix}, \tag{2.3.6}$$

$$\bar{P} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1 y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n^2 & y_n^2 & x_n y_n \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.3.7}$$

(2.3.5) can be rewritten in the following form

$$\bar{A} = \bar{P}^T D^T D \bar{P} = Y^T Y, \tag{2.3.8}$$

where

$$D = sqrt(\bar{W}), \qquad Y = D\bar{P}.$$

$\bar{W}$ is a diagonal matrix positive and with non-zeros diagonal elements, then by using the matrix rank properties, we conclude

$$rank(Y) = rank(\bar{P}). \tag{2.3.9}$$

From (2.3.8) and (2.3.9), we get

$$rank(\bar{A}) = rank(Y^T Y) = rank(Y). \tag{2.3.10}$$

Using (2.3.9) and (2.3.10), we have

$$rank(\bar{A}) = rank(\bar{P}). \tag{2.3.11}$$

(2.3.11) confirm that the matrix $\bar{A}$ will be nonsingular if the matrix $\bar{P}$ have full rank (m=6). In this case, we just need the following matrix

$$P^* = \begin{bmatrix} 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix}$$

to have full rank. The last condition is the same one for the the classical moment matrix A with linear bases. □

Lemma 2.3.2. *(See [61]) Let $w_i(x,y,z) \in C^r(D)$. If ν is a constant vector and the moment matrix $\bar{A}$ is invertible at every point of D, then $\bar{u}^\rho(x,y,z) \in C^r(D)$.*

Generally, it is clear that if $w_i(x,y) \in C^r(D)$ and $p_k(x,y) \in C^s(D)$, $i = 1, ..., n$, $k = 0, ..., m$ then $\bar{\phi}_i(x,y) \in C^{min(r,s)}(D)$, since the monomials bases have C^∞ continuity, then the order of continuity of the shape functions is determined by the weight function and u. we knew that ν is a constant vector ($\nu \in C^\infty$). As a result, the order of continuity of the approximation function is determined by that of the weight functions.

2.4 Numerical scheme

2.4.1 Numerical scheme on rectangular domain

2.4.1.1 2-D linear Fredholm-Volterra integral equation

Consider the following two-dimensional Fredholm-Volterra integral equation

$$u(x,y) + \int_c^d \int_a^b K_1(x,y,s,v) u(s,v)\,\mathrm{d}s\mathrm{d}v + \int_c^y \int_a^x K_2(x,y,s,v) u(s,v)\,\mathrm{d}s\mathrm{d}v = g(x,y),$$

where $(x,y) \in [a,b] \times [c,d]$, the intervals $[a,x],[c,y]$ are converted respectively to the fixed intervals $[a,b],[c,d]$ by the following linear transformations

$$s(x,\delta) = \frac{x-a}{b-a}\delta + \frac{b-x}{b-a}a, \quad v(y,\beta) = \frac{y-c}{d-c}\beta + \frac{d-y}{d-c}c. \tag{2.4.1}$$

Therefore, the equation takes the following form

$$u(x,y) + \int_c^d \int_a^b K_1(x,y,s,v)u(s,v)\,\mathrm{d}s dv + \int_c^d \int_a^b \bar{K}_2(x,y,s(x,\delta),v(y,\beta))u(s(x,\delta),v(y,\beta))\,\mathrm{d}\delta\mathrm{d}\beta$$
$$= g(x,y),$$

where

$$\bar{K}_2(x,y,s(x,\delta),v(y,\beta)) = \frac{x-a}{b-a}\frac{y-c}{d-c}K_2(x,y,s(x,\delta),v(y,\beta)).$$

If we replace $u(x,y)$ by $u^h(x,y)$ we obtain

$$u^h(x,y) + \int_c^d \int_a^b K_1(x,y,s,v)u^h(s,v)\,\mathrm{d}s dv + \int_c^d \int_a^b \bar{K}_2(x,y,\xi(x,\delta),v(y,\beta))u^h(s(x,\delta),v(y,\beta))\,\mathrm{d}\delta\mathrm{d}\beta = g(x,y),$$

or equivalently

$$\sum_{j=1}^{n} \Big[\phi_j(x,y) + \int_c^d \int_a^b K_1(x,y,s,v)\phi_j(s,v)\,\mathrm{d}s\,\mathrm{d}v$$
$$+ \int_c^d \int_a^b \bar{K}_2(x,y,s(x,\delta),v(y,\beta))\phi_j(s(x,\delta),v(y,\beta))\mathrm{d}\delta\mathrm{d}\beta\Big]u_j = g(x,y).$$

Assume that this equation holds at (x_i,y_i)

$$\sum_{j=1}^{n} \Big[\phi_j(x_i,y_i) + \int_c^d \int_a^b K_1(x_i,y_i,s,v)\phi_j(s,v)\,\mathrm{d}s\,\mathrm{d}v$$
$$+ \int_c^d \int_a^b \bar{K}_2(x_i,y_i,s(x_i,\delta),v(y_i,\beta))\phi_j(s(x_i,\delta),v(y_i,\beta))\mathrm{d}\delta\mathrm{d}\beta\Big]u_j = g(x_i,y_i),$$

where $i = 1,2,...,n$, we compute integrals numerically by using m_1 points quadrature formula with the quadrature points $\{s_k\},\{\delta_k\},\{v_p\},\{\beta_p\}$ and the quadrature weights $\{w_k\},\{w_p\}$. Therefore, the above equation can be written as follows

$$\sum_{j=1}^{n} F_{i,j}\hat{u}_j = g(x_i,y_i), \quad i = 1,2,...,n,$$

where $\hat{u}_j$ are the approximate quantities of u_j and F is a n by n matrix defined by

$$F_{i,j} = \phi_j(x_i,y_i) + \sum_{p=1}^{m_1}\sum_{k=1}^{m_1} K_1(x_i,y_i,s_k,v_p)\phi_j(s_k,v_p)\omega_k\omega_p$$
$$+ \sum_{p=1}^{m_1}\sum_{k=1}^{m_1} \bar{K}_2(x_i,y_i,s(x_i,\delta_k),v(y_i,\beta_p))\phi_j(s(x_i,\delta_k),v(y_i,\beta_p))\omega_k\omega_p.$$

Let's note

$$\hat{u} = [\hat{u}_1,\hat{u}_2,......,\hat{u}_n]^T, \qquad g = [g_1,g_2,....,g_n]^T.$$

Then we have the following linear system of equations

$$F\hat{u} = g. \tag{2.4.2}$$

Solving (2.4.2), we can approximate $u(x,y)$ as in (2.2.6) by

$$u^h(x,y) = \sum_{j=1}^{n} \phi_j(x,y)\hat{u}_j, \quad (x,y) \in [a,b] \times [c,d].$$

2.4.1.2 2-D nonlinear Fredholm-Volterra integral equation

In this section, MLS and MMLS approximations are used to solve two dimensional nonlinear Fredholm-Volterra integral equations of the second kind. Firstly, we transform the intervals $[a,x],[c,y]$ in to $[a,b],[c,d]$ by the last linear transformations (2.4.1), then (2.1.1) takes the following form

$$\begin{aligned}u(x,y)+&\int_c^d\int_a^b K_1(x,y,s,v)\Psi_1(s,v,u(s,v))\,\mathrm{d}s dv \\ &+\int_c^d\int_a^b \bar{K}_2(x,y,s(x,\delta),v(y,\beta))\Psi_2(s(x,\delta),v(y,\beta),u(s(x,\delta),v(y,\beta))\mathrm{d}\delta\mathrm{d}\beta=g(x,y),\end{aligned} \tag{2.4.3}$$

where

$$\bar{K}_2(x,y,s(x,\delta),v(y,\beta))=\frac{x-a}{b-a}\frac{y-c}{d-c}K_2(x,y,s(x,\delta),v(y,\beta)).$$

We estimate the unknown function $u(x,y)$ as

$$u^h(x,y)=\sum_{j=1}^n\alpha_j\phi_j(x,y).$$

If in (2.4.3) we replace $u(x,y)$ by $u^h(x,y)$, we obtain

$$\begin{aligned}\sum_{j=1}^n\alpha_j\phi_j(x,y)+&\int_c^d\int_a^b K_1(x,y,s,v)\Psi_1\left(s,v,\sum_{j=1}^n\alpha_j\phi_j(s,v)\right)\mathrm{d}s\,\mathrm{d}v \\ &+\int_c^d\int_a^b \bar{K}_2(x,y,s(x,\delta),v(y,\beta))\Psi_2\left(s(x,\delta),v(y,\beta),\sum_{j=1}^n\alpha_j\phi_j(s(x,\delta),v(y,\beta))\right)\mathrm{d}\delta\mathrm{d}\beta \\ &=g(x,y).\end{aligned}$$

If this equation holds at the collocation points (x_i,y_i) we have

$$\begin{aligned}\sum_{j=1}^n\alpha_j\phi_j(x_i,y_i)+&\int_c^d\int_a^b K_1(x_i,y_i,s,v)\Psi_1\left(s,v,\sum_{j=1}^n\alpha_j\phi_j(s,v)\right)\mathrm{d}s\,\mathrm{d}v \\ &+\int_c^d\int_a^b \bar{K}_2(x_i,y_i,s(x_i,\delta),v(y_i,\beta))\Psi_2\left(s(x_i,\delta),v(y_i,\beta),\sum_{j=1}^n\alpha_j\phi_j(s(x_i,\delta),v(y_i,\beta))\right)\mathrm{d}\delta\mathrm{d}\beta \\ &=g(x_i,y_i).\end{aligned} \tag{2.4.4}$$

Using a m_1 points quadrature formula with the points $\{s_k\},\{\delta_k\},\{v_p\},\{\beta_p\}$ and weights $\{w_k\},\{w_p\}$ for numerical integration, we obtain

$$\begin{aligned}\sum_{j=1}^n\bar{\alpha}_j\phi_j(x_i,y_i)+&\sum_{p=1}^{m_1}\sum_{k=1}^{m_1}w_kw_pK_1(x_i,y_i,s_k,v_p)\Psi_1\left(s_k,v_p,\sum_{j=1}^n\bar{\alpha}_j\phi_j(s_k,v_p)\right) \\ &+\sum_{p=1}^{m_1}\sum_{k=1}^{m_1}w_kw_p\bar{K}_2(x_i,y_i,s(x,\delta_k),v(y,\beta_p))\Psi_2\left(s(x_i,\delta_k),v(y_i,\beta_p),\sum_{j=1}^n\bar{\alpha}_j\phi_j(s(x_i,\delta_k),v(y_i,\beta_p))\right) \\ &=g(x_i,y_i).\end{aligned} \tag{2.4.5}$$

The unknowns $\bar{\alpha}_j$ can be found by solving the nonlinear system of algebraic equations which can be solved by any nonlinear solver; in this work we have used the fsolve command of Matlab. So the values of $u(x,y)$ at any point $(x,y)\in[a,b]\times[c,d]$ can be approximated by

$$u^h(x,y)=\sum_{j=1}^n\bar{\alpha}_j\phi_j(x,y)$$

2.4.2 Numerical scheme on non rectangular domain

2.4.2.1 2-D linear Fredholm integral equation

Consider the following two-dimensional Fredholm integral equation

$$u(x,y) - \lambda \int_D K(x,y,\mu,\theta)u(\mu,\theta)\,\mathrm{d}\mu d\theta = g(x,y), \quad (x,y) \in D. \tag{2.4.6}$$

To apply to method in each sub-domain, (2.4.6) take the folowing form

$$u(x,y) - \lambda \sum_{e=1}^{d} \int_0^1 \int_{\upsilon_{1,e}(\theta)}^{\upsilon_{2,e}(\theta)} K(x,y,\mu,\theta)u(\mu,\theta)\,\mathrm{d}\mu d\theta = g(x,y), \quad e = 1,...,d, \ (x,y) \in D. \tag{2.4.7}$$

The intervals $[\upsilon_{1,e}(\theta), \upsilon_{2,e}(\theta)]$, $e = 1,...,d$ are converted to the fixed interval $[0,1]$ by using the change of variable given by

$$s = \mu(\theta,\sigma) = \big(\upsilon_{2,e}(\theta) - \upsilon_{1,e}(\theta)\big)\sigma + \upsilon_{1,e}(\theta) \tag{2.4.8}$$

Therefore, (2.4.7) takes the following form

$$u(x,y) - \lambda \sum_{e=1}^{d} \int_0^1 \int_0^1 \Big(\upsilon_{2,e}(\theta) - \upsilon_{1,e}(\theta)\Big) K(x,y,s,\theta)u(s,\theta)\,\mathrm{d}\sigma\mathrm{d}\theta = g(x,y).$$

If we replace $u(x,y)$ by $\bar{u}^h(x,y)$ we obtain

$$\bar{u}^h(x,y) - \lambda \sum_{e=1}^{d} \int_0^1 \int_0^1 \Big(\upsilon_{2,e}(\theta) - \upsilon_{1,e}(\theta)\Big) K(x,y,s,\theta)\bar{u}^\rho(s,\theta)\,\mathrm{d}\sigma\mathrm{d}\theta = g(x,y),$$

or equivalently

$$\sum_{j=1}^{n} \left[\bar{\phi}_j(x,y) - \lambda \sum_{e=1}^{d} \int_0^1 \int_0^1 \Big(\upsilon_{2,e}(\theta) - \upsilon_{1,e}(\theta)\Big) K(x,y,s,\theta)\bar{\phi}_j(x,y,s,\theta)\,\mathrm{d}\sigma d\theta \right] u_j = g(x,y).$$

If this equation holds at (x_i, y_i). Then, we have

$$\sum_{j=1}^{n} \left[\bar{\phi}_j(x_i,y_i) - \lambda \sum_{e=1}^{d} \int_0^1 \int_0^1 \Big(\upsilon_{2,e}(\theta) - \upsilon_{1,e}(\theta)\Big) K(x_i,y_i,s,\theta)\bar{\phi}_j(s,\theta)\,\mathrm{d}\sigma d\theta \right] u_j = g(x_i,y_i),$$

where $i = 1,2,...,n$, we use m_1 point quadrature formula with the quadrature points s_h, θ_p and the quadrature weights $\{w_h\}, \{w_p\}$ for numerical integration. Therefore, the above equation can be reformulated as follows

$$\sum_{j=1}^{n} F_{i,j}\hat{u}_j = g(x_i,y_i),$$

where $\hat{u}_j$ are the approximate quantities of u_j and F is a matrix defined by

$$F_{i,j} = \bar{\phi}_j(x_i,y_i) - \lambda \sum_{e=1}^{d} \sum_{p=1}^{m_1} \sum_{h=1}^{m_1} w_h w_p \Big(\upsilon_{2,e}(\theta_p) - \upsilon_{1,e}(\theta_p)\Big) K(x_i,y_i,s_h,\theta_p)\bar{\phi}_j(s_h,\theta_p).$$

Let's note

$$\hat{u} = [\hat{u}_1, \hat{u}_2, ..., \hat{u}_n]^T, \qquad g = [g_1, g_2, ..., g_n]^T.$$

Then we have the following linear system of equations

$$F\hat{u} = g. \tag{2.4.9}$$

Solving (2.4.9), we can approximate $u(x,y)$ as in (2.3.4) by

$$\bar{u}^h(x,y) = \sum_{j=1}^{n} \bar{\phi}_j(x,y)\hat{u}_j, \quad (x,y) \in D.$$

2.4.2.2 2-D Fredholm-Hammerstein integral equation

We use Kumar and Sloan technique [67] for finding the approximate solution of the equation (2.1.2).

$$z(x,y) = \Psi(x,y,u(x,y)) = \Psi\left(x,y,g(x,y)+\lambda\int_D K(x,y,\mu,\theta)z(\mu,\theta)d\mu d\theta\right). \tag{2.4.10}$$

To apply the method, we estimate the unknown function $u(x,y)$ by the MLS approximation as

$$z(x,y) = \sum_{j=1}^{n} \alpha_j \bar{\phi}_j(x,y). \tag{2.4.11}$$

By substituting (2.4.11) into (2.4.10) and interpolating the collocation points $(x_i, y_i), i = 0, ..., n.$ we have

$$\sum_{j=1}^{n} \alpha_j \bar{\phi}_j(x_i,y_i) = \Psi\left(x_i,y_i,g(x_i,y_i)+\lambda\int_D K(x_i,y_i,\mu,\theta)\sum_{j=1}^{n}\alpha_j\bar{\phi}_j(\mu,\theta)d\mu d\theta\right). \tag{2.4.12}$$

Now the domain D can be replaced with their expressions given in (2.1.3). (2.4.12) becomes

$$\sum_{j=1}^{n} \alpha_j \bar{\phi}_j(x_i,y_i) = \Psi\left(x_i,y_i,g(x_i,y_i)+\lambda\sum_{e=1}^{d}\int_0^1\int_{\upsilon_{1,e}(\theta)}^{\upsilon_{2,e}(\theta)} K(x_i,y_i,\mu,\theta)\sum_{j=1}^{n}\alpha_j\bar{\phi}_j(\mu,\theta)d\mu d\theta\right). \tag{2.4.13}$$

We transform the intervals $[\upsilon_{1,e}(\theta), \upsilon_{2,e}(\theta)]$ into $[0,1]$ by the linear transformations (2.4.8). Then (2.4.13) take the following form

$$\begin{aligned}\sum_{j=1}^{n} \alpha_j \bar{\phi}_j(x_i,y_i) = \Psi\Bigg(x_i,y_i,g(x_i,y_i)+\lambda\sum_{e=1}^{d}\int_0^1\int_0^1 \Big(\upsilon_{2,e}(\theta)-\upsilon_{1,e}(\theta)\Big)\\ K(x_i,y_i,s,\theta)\sum_{j=1}^{n}\alpha_j\bar{\phi}_j(s,\theta)d\sigma d\theta\Bigg).\end{aligned} \tag{2.4.14}$$

Using a m_1-point quadrature formula with the points$\{s_h\}, \{\theta_p\}$ and weights $\{w_h\}, \{w_p\}$ for numerical integration, we obtain

$$\begin{aligned}\sum_{j=1}^{n} \bar{\alpha}_j \bar{\phi}_j(x_i,y_i) = \Psi\Bigg(x_i,y_i,g(x_i,y_i)+\lambda\sum_{e=1}^{d}\sum_{p=0}^{m_1}\sum_{h=0}^{m_1} w_h w_p\Big(\upsilon_{2,e}(\theta_p)-\upsilon_{1,e}(\theta_p)\Big)\\ K(x_i,y_i,s_h,\theta_p)\sum_{j=0}^{n}\bar{\alpha}_j\bar{\phi}_j(s_h,\theta_p)\Bigg).\end{aligned} \tag{2.4.15}$$

Finding unknowns $\bar{\alpha}_j$ by solving the nonlinear system of algebraic equations which can be solved by any nonlinear solver. In this work we have used the fsolve command of Matlab. The following approximate solution is given by

$$\bar{z}(x,y) = \sum_{j=1}^{n} \bar{\alpha}_j \bar{\phi}_j(x,y),$$

the approximation solution $\bar{u}_n(x,y)$ of $u_n(x,y)$ is obtained by

$$\bar{u}_n(x,y) = g(x,y)+\lambda\sum_{e=1}^{d}\sum_{p=0}^{m_1}\sum_{h=0}^{m_1} w_h w_p\Big(\upsilon_{2,e}(\theta_p)-\upsilon_{1,e}(\theta_p)\Big)K(x_i,y_i,s_h,\theta_p)\bar{z}(s_h,\theta_p).$$

2.5 Error estimates

(2.1.2) can be represented in abstract form as

$$(I-\mathcal{K})u = g,$$

where

$$\mathcal{K}u = \int_0^1 \int_0^1 K_1(x,y,s,v)\Psi_1(s,v,u(s,v))\,\mathrm{d}s\,\mathrm{d}v + \int_0^y \int_0^x K_2(x,y,s,v)\Psi_2(s,v,u(s,v))\,\mathrm{d}s\,\mathrm{d}v.$$

We define the collocation operator $P_n : C(D) \to G_n$ by

$$P_n u(x,y) = \sum_{i=1}^{n} \alpha_i \Phi_i(x,y), \quad (x,y) \in D,$$

where $G_n = span\{\Phi_1, \Phi_2, ..., \Phi_n\}$ and the coefficients α_i can be determined by solving the linear system

$$P_n u(x_i, y_i) = u(x_i, y_i), \quad i = 1, 2, ..., n.$$

we note that (2.4.4) can be written in the abstract form

$$(I - P_n\mathcal{K})u_n = P_n g. \tag{2.5.1}$$

Let the operator $\mathcal{K}$ be defined as

$$\begin{aligned}\mathcal{K}_n u =& \sum_{p=1}^{m_1}\sum_{k=1}^{m_1} w_k w_p K_1(x,y,s_k,v_p)\Psi_1\left(s_k, v_p, u(s_k, v_p)\right)\\ &+ \sum_{p=1}^{m_1}\sum_{k=1}^{m_1} w_k w_p \bar{K}_2(x,y,s(x,\delta_k),v(y,\beta_p))\Psi_2\left(s(x,\delta_k), v(y,\beta_p), u(s(x,\delta_k), v(y,\beta_p))\right).\end{aligned}$$

we can write (2.4.5) in the operator form

$$(I - P_n\mathcal{K}_n)\hat{u}_n = P_n g. \tag{2.5.2}$$

Let $\{T_n, \hat{T}_n\}$ be the operators defined respectively by

$$T_n u = P_n\mathcal{K}u + P_n g, \quad \hat{T}_n u = P_n\mathcal{K}_n u + P_n g.$$

So (2.5.1), (2.5.2) can be written as

$$u_n = T_n u_n, \qquad \hat{u}_n = \hat{T}_n \hat{u}_n.$$

We present the following Theorem from Vainikko [121] used to obtain the error analysis of the proposed method.

Theorem 2.5.1. *Let T and $\hat{T}$ be continuous over an open set D in Banach space X. Let the equation*

$$u = \hat{T}u,$$

has an isolated solution $\hat{u}_0$ in D and let the following conditions be satisfied:

- *The operator T is Frechet differentiable in some neighborhood of the point $\hat{u}_0$ while the linear operator $I - T^{'}(\hat{u}_0)$ is continuously invertible,*
- *For some $\eta > 0$ and $0 < q < 1$ the following inequalities are valid (the number $\eta > 0$ assumed to be so small that the sphere $\|u - \hat{u}_0\| \leq \eta$ is contained within D).*

$$\sup_{\|u-\hat{u}_0\|\leq\eta} \|(I - T^{'}(\hat{u}_0))^{-1}(T^{'}(u) - T^{'}(\hat{u}_0))\| \leq q, \tag{2.5.3}$$

$$\alpha = \|(I - T^{'}(\hat{u}_0))^{-1}(T(\hat{u}_0) - \hat{T}(\hat{u}_0))\| \leq \eta(1-q), \tag{2.5.4}$$

then the equation $u = Tu$ has in the sphere $\|u - \hat{u}_0\| \leq \eta$ a unique solution u_0. Moreover, the inequality

$$\frac{\alpha}{1+q}\|u_0 - \hat{u}_0\| \leq \frac{\alpha}{1-q},$$

is valid.

Theorem 2.5.2. *let $u_0 \in C([0,1]\times[0,1])$ be an isolated solution of*

$$u = \mathcal{K}u + g,$$

Assume that 1 is not an eigenvalue of the linear operator $T^{'}(u_0)$. Then for sufficiently large n, the operator $(I-\hat{T}_{\mathrm{n}}^{'}(u_0))^{-1}$ is invertible and there exists constant $L>0$ independent of n such that $\|(I-\hat{T}_{\mathrm{n}}^{'}(u_0))^{-1}\| \leq L$.

Theorem 2.5.3. *let $u_0 \in C([0,1]\times[0,1])$ be an isolated solution of*

$$u = \mathcal{K}u + g.$$

Assume that 1 is not an eigenvalue of the linear operator $T^{'}(u_0)$. Then for sufficiently large n, the approximate solution $\hat{u}_n$ of (2.5.2) is unique in $B(u_0,\eta) = \{u : \|u-u_0\| \leq \eta\}$ for some $\eta > 0$. Moreover, there exists a constant $0<q<1$ independent of n such that

$$\frac{\varphi_{\mathrm{n}}}{1+q} \leq \|u_0 - \hat{u}_{\mathrm{n}}\| \leq \frac{\varphi_{\mathrm{n}}}{1-q},$$

where $\qquad \varphi_{\mathrm{n}} = \|(I-\hat{T}_{\mathrm{n}}^{'}(u_0))^{-1}(\hat{T}_{\mathrm{n}}(u_0) - T(u_0))\|.$

Proof. Applying Theorem 2.5.2 we have $(I-\hat{T}_{n}^{'}(u_0))^{-1}$. exists and it is uniformly bounded i.e, there exists a constant $L>0$ such that

$$\|(I-\hat{T}_{n}^{'}(u_0))^{-1}\| \leq L.$$

Assume that $\|P_n\| < p$ and $\|\mathcal{K}_n'\| < M$, then

$$\begin{aligned}
\|\hat{T}_{n}^{'}(u) - \hat{T}_{n}^{'}(u_0)\| &= \|P_{n}{}_{n}^{'}u - P_n\mathcal{K}_{n}^{'}u_0\| \\
&\leq \|P_n\|\|\mathcal{K}_{n}^{'}u - \mathcal{K}_{n}^{'}u_0\| \\
&\leq pM\eta, \quad \forall u \in B(u_0,\eta).
\end{aligned}$$

Thus, we obtain

$$\sup_{\|u-\hat{u}_0\|\leq\eta} \|(I-\hat{T}_{n}^{'}(u_0))^{-1}(\hat{T}_{n}^{'}(u) - \hat{T}_{n}^{'}(u_0))\| \leq LpM\eta \leq q,$$

where $0<q<1$, which demonstrates (2.5.3) for η sufficiently small. Also we have

$$\begin{aligned}
\varphi_n &= \|(I-\hat{T}_{n}^{'}(u_0))^{-1}(\hat{T}_n(u_0) - T(u_0))\| \\
&\leq \|(I-\hat{T}_{n}^{'}(u_0))^{-1}\|\|(\hat{T}_n(u_0) - T(u_0))\| \\
&\leq L\|(\hat{T}_n(u_0) - T(u_0))\|.
\end{aligned}$$

Here, we will prove that $\|(\hat{T}_n(u_0) - T(u_0))\| \longrightarrow 0$ as $n \longrightarrow \infty$. Now consider

$$\begin{aligned}
\|(\hat{T}_n(u_0) - T(u_0))\| &= \|P_n\mathcal{K}_n u_0 + P_n f - \mathcal{K}u_0 - g\| \\
&\leq \|P_n\{\mathcal{K}_n u_0 + g - \mathcal{K}u_0 - g\}\| + \|(P_n - I)(\mathcal{K}u_0 + g)\| \\
&\leq p\|\mathcal{K}_n - \mathcal{K}\|\|u_0\| + \|P_n u_0 - u_0\| \longrightarrow 0 \quad as \quad n \longrightarrow \infty
\end{aligned} \tag{2.5.5}$$

For sufficient large n we have $\beta_n \leq \eta(1-q)$. Since (2.5.4) is satisfied. Then from Theorem 2.5.1 we have

$$\frac{\varphi_n}{1+q}\|\hat{u}_n - u_0\| \leq \frac{\varphi_n}{1-q}.$$

□

We complete the error estimate by the following Theorem.

Theorem 2.5.4. *let $u_0 \in C^{q+1}(D)$ be an isolated solution of the equation $u = \mathcal{K}u + g$ and $\hat{u}_n$ be the discrete MLS collocation method of u_0. Then we have*

$$\|\hat{u}_n - u_0\|_{L^\infty(D)} \leq \frac{Lp}{1-q}\|\mathcal{K}_n - \mathcal{K}\|_{L^\infty(D)}\|u_0\|_{L^\infty(D)} + \frac{L}{1-q}(1+\delta)Ch_{X,D}^{q+1}|u_0|_{C^{q+1}(D)}. \tag{2.5.6}$$

Proof. We have

$$\|\hat{u}_n - u_0\| \ \leq \frac{\varphi_n}{1-q}$$

Using Eq. (2.5.5) and Theorems 2.5.3, 1.1.2, we obtain the estimate

$$\begin{aligned}
\|\hat{u}_n - u_0\|_{L^\infty(D)} &\leq \frac{\varphi_n}{1-q} \\
&\leq \frac{\|(I - \hat{T}_n'(u_0))^{-1}(\hat{T}_n(u_0) - T(u_0))\|_{L^\infty(D)}}{1-q} \\
&\leq \frac{L}{1-q}\|(\hat{T}_n(u_0) - T(u_0))\|_{L^\infty(D)} \\
&\leq \frac{L}{1-q}\|P_N\|_{L^\infty(D)}\|\mathcal{K}_n - \mathcal{K}\|_{L^\infty(D)}\|u_0\|_{L^\infty(D)} + \frac{L}{1-q}\|P_n u_0 - u_0\|_{L^\infty(D)} \\
&\leq \frac{Lp}{1-q}\ \|\mathcal{K}_n - \mathcal{K}\|_{L^\infty(D)}\|u_0\|_{L^\infty(D)} + \frac{L}{1-q}(1+\delta)Ch_{X,D}^{q+1}|u_0|_{C^{q+1}(D)}.
\end{aligned}$$

□

2.6 Numerical results

To show the validity of the method as a numerical tool. Linear and nonlinear Fredholm-Volterra integral equations are solved. For numerical implementation we put $h_X = \dfrac{1}{n-1}$, then in computations of the MLS method we put for linear case $\rho_i = 2 \times h_X$, for quadratic case $\rho_i = 2.5 \times h_X$, for degree 3 case $\rho_i = 3 \times h_X$, and for degree 4 case $\rho_i = 4 \times h_X$, where h_X is the distance between two consecutive nodes. When we use the MMLS method, we take for quadratic case $\rho_i = 2 \times h_X$. Also, we use the 5-points Gauss-Legendre quadratic rule for numerical integration and spline weight functions for approximating integrals in the scheme. Furthermore, for computing shape function in MMLS method, we take $\nu_e = 10^{-9}$; with $e = 1; 2; 3$ as weights of additional coefficients for MMLS, it should be pointed out that, this value was selected experimentally. Accuracy of the numerical solutions can be worked out by measuring the $\|e\|_\infty$ and $e(x,y)$ norms which are defined by

$$\begin{aligned}
\|e\|_\infty &= \max|u_{ex}(x,y) - \hat{u}(x,y)|, \quad && (x,y) \in D, \\
e(x,y) &= |u_{ex}(x,y) - \hat{u}(x,y)|, && (x,y) \in D,
\end{aligned}$$

where $\hat{u}$ is the approximate solution of the exact solution u_{ex}. The rate convergence presented in this work is defined as

$$Ratio = \frac{\ln(\|e_n\|_\infty) - \ln(\|e_{n'}\|_\infty)}{\ln(h_X) - \ln(h_{X'})},$$

$(\|e_n\|_\infty, \|e_{n'}\|_\infty)$ are the maximum errors of the previous and the current row respectively. The "Fsolve" command is employed to solve the nonlinear system of algebraic equations. All calculations are done by Matlab.

2.6.1 Applications on rectangular domain

Example 1. As the first example, consider the two-dimensional linear Fredholm-Volterra integral equation

$$\begin{aligned}
u(x,y) &+ \int_0^1\int_0^1 cos(x-s)\exp(v)u(s,v)\,\mathrm{d}s\,\mathrm{d}v \\
&+ \frac{1}{2}\int_0^y\int_0^x sin(x-s)\exp(v-y)u(s,v)\,\mathrm{d}s\,\mathrm{d}v = g(x,y). \quad (x,y) \in [0,1],
\end{aligned}$$

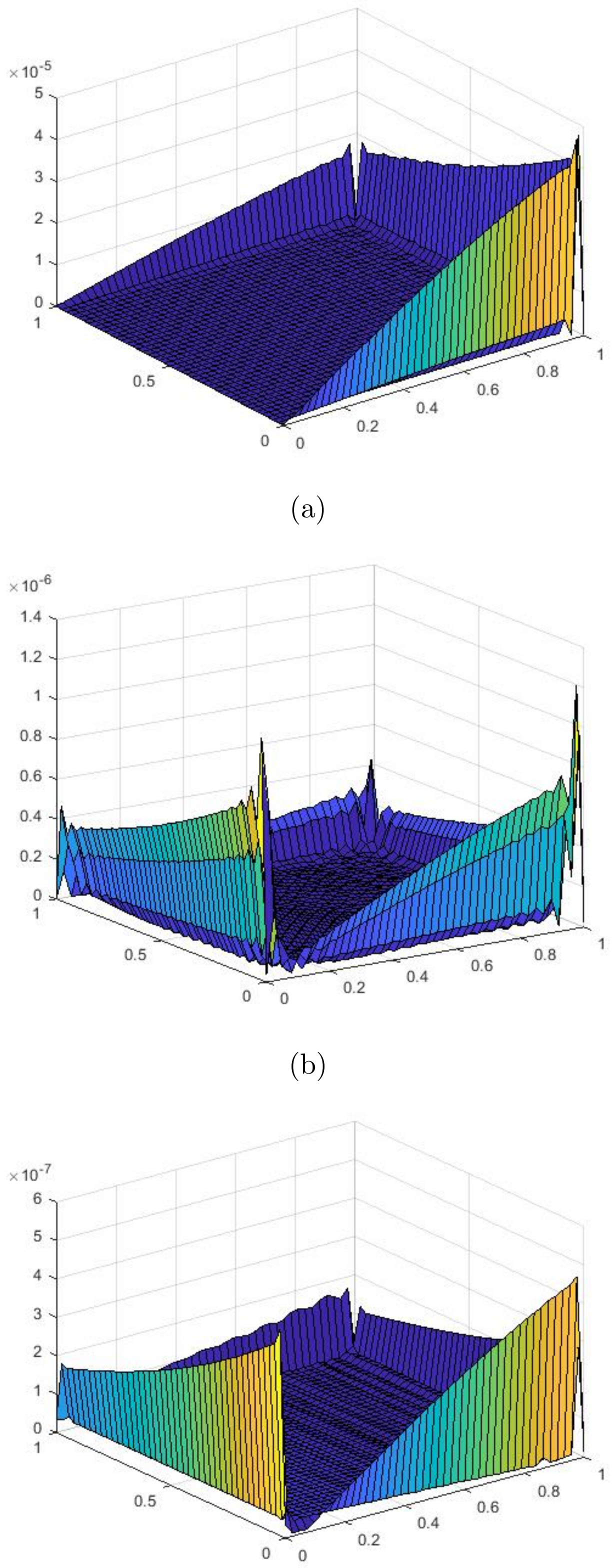
where

$$\begin{aligned}
g(x,y)) = sin(x)\exp(-y) &+ \frac{y}{4}\exp(-y)\Big(sin(x) - xcos(x)\Big) \\
&+ \frac{1}{4}\Big(2sin(x) - cos(2-x) + cos(x)\Big), \quad 0 \leq x, y \leq 1.
\end{aligned}$$

Table 2.1: Maximum errors using different values of n for Example 1

n	h_X	MLS approximation error				MMLS approximation error	
		Linear basis	Ratio	Quadratic basis	Ratio	Quadratic basis	Ratio
3×3	0.50	1.43×10^{-2}	-	7.47×10^{-3}	-	2.83×10^{-3}	-
5×5	0.25	4.41×10^{-3}	1.69	1.30×10^{-3}	2.52	4.77×10^{-4}	2.56
9×9	0.12	1.22×10^{-3}	1.85	1.73×10^{-4}	2.90	6.36×10^{-5}	2.90
19×19	0.05	2.54×10^{-4}	1.93	1.55×10^{-5}	2.97	7.08×10^{-6}	2.70
37×37	0.02	6.46×10^{-5}	1.97	1.94×10^{-6}	2.99	9.17×10^{-7}	2.94
43×43	0.02	4.75×10^{-5}	1.99	1.21×10^{-6}	3.06	5.36×10^{-7}	3.48

The function $g(x, y)$ has been chosen such that the exact solution of the integral equation is $u_{ex}(x, y) = sin(x)\exp(-y)$. The integral equation is solved on the square domain $D = [0, 1] \times [0, 1]$. Numerical results are presented in Table 2.1 in terms of $\|e\|_\infty$ at different numbers of n, and Figure 2.1 is depicting the absolute error with $(n = 43 \times 43)$ points. As we expected, from Figure 2.1, the error near the boundary increases which effects on the global error. As we can see, the results gradually converge to the exact values as the number of nodes increases, and the obtained results by the MMLS method are better than the results given by the MLS method. According to the Theorem 2.5.4, the ratio of error remains approximately constant for the linear case $O(h_X^2)$ and for the quadratic case $O(h_X^3)$ as $n \to \infty$.



(a)

(b)

(c)

Figure 2.1: Approximation error :(a) Linear basis(MLS), (b) Quadratic basis(MLS), (c) Quadratic basis(MMLS) of Example 1

Example 2. Consider the following nonlinear Volterra-Fredholm integral equation by using the shifted Chebyshev polynomials as basis

$$u(x) - \int_0^x sin(x-s)cos(u(x))\,\mathrm{d}s \; - \frac{1}{8}\int_0^1 (x-s)u(x)\,\mathrm{d}s = g(x), \quad x \in [0,1],$$

where

$$g(x) = \frac{49}{48} - \frac{17x}{16} - \frac{1}{2}\bigg(sin(x)(xcos(1) + sin(1)) - xsin(1)cos(x)\bigg).$$

The exact solution of this equation is $u_{ex}(x) = 1 - x$.

Table 2.2: Maximum errors using different values of degree basis m for Example 2

n	h_X	$m=1$	$m=2$	$m=3$	$m=4$
5	0.25	8.65×10^{-4}	3.56×10^{-5}	3.65×10^{-6}	1.37×10^{-6}
9	0.12	2.89×10^{-4}	2.88×10^{-6}	1.35×10^{-6}	3.95×10^{-8}
15	0.07	1.36×10^{-4}	1.12×10^{-6}	1.55×10^{-7}	2.65×10^{-9}
21	0.05	9.51×10^{-5}	2.50×10^{-7}	4.59×10^{-8}	4.43×10^{-10}
25	0.04	7.61×10^{-5}	1.30×10^{-7}	2.78×10^{-8}	2.78×10^{-10}

Table 2.2 shows $\|e\|_\infty$ at the different numbers of nodes that are regularly employed in the segment. According to the table, when the degree basis increases, the numerical results show the higher performance of the MLS method with the Chebyshev basis, and also the results converge to the exact values as the number of nodes increases.

Example 3. Consider the following two-dimensional nonlinear Fredholm-Volterra integral equation defined as

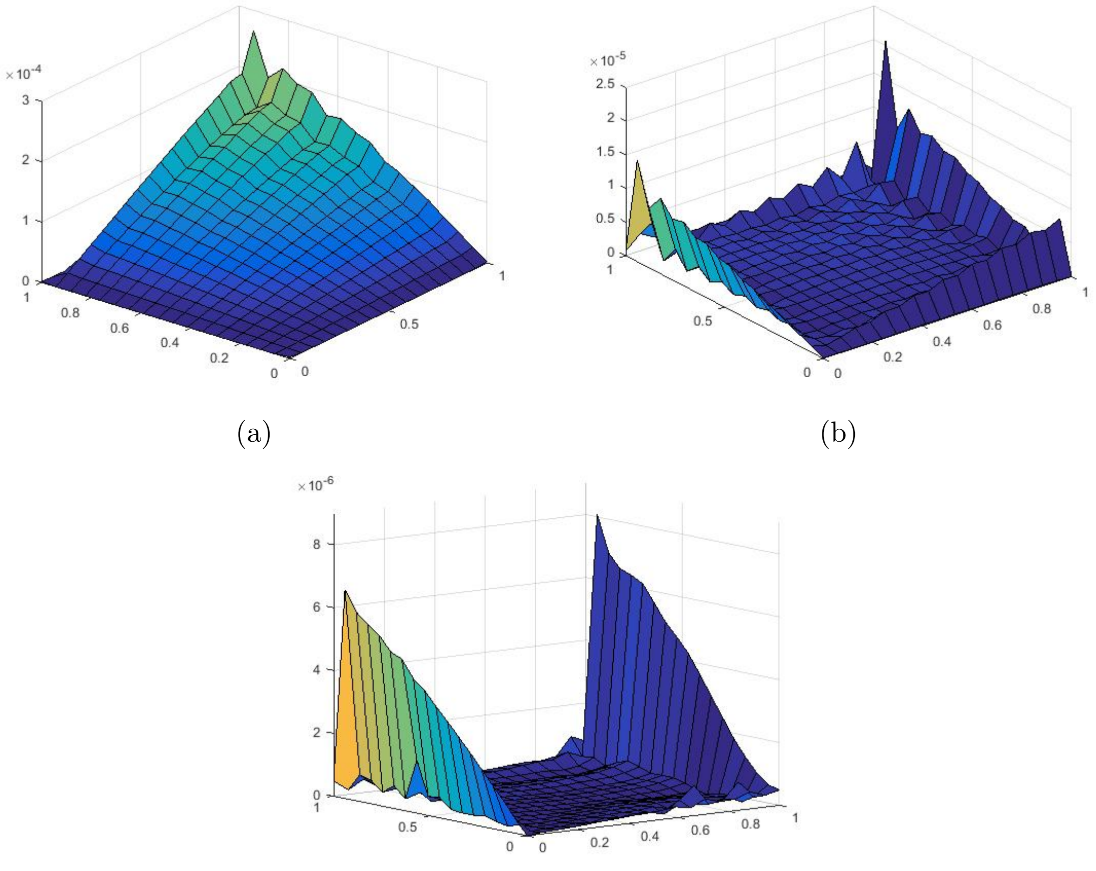
$$\begin{aligned} u(x,y) - &\int_0^1\int_0^1 (v-s)sin(x-y)u(s,v)\,\mathrm{d}s\,\mathrm{d}v \\ &- \int_0^y\int_0^x (xs^2 + cos(v))u^2(s,v)\,\mathrm{d}s\,\mathrm{d}v = g(x,y), \quad 0 \le x,y \le 1, \end{aligned}$$

where

$$\begin{aligned} g(x,y) = xsin(y)&\bigg(1 - \frac{1}{9}x^2sin^2(y)\bigg) + \frac{1}{10}x^6\bigg(\frac{1}{2}sin(2y) - y\bigg) \\ &+ \frac{1}{6}sin(x-y)\bigg(2 - 3sin(1) + cos(1)\bigg). \end{aligned}$$

Table 2.3: Maximum errors using different values of n for Example 3

n	h_X	MLS approximation error				MMLS approximation error	
		Linear basis	Ratio	Quadratic basis	Ratio	Quadratic basis	Ratio
3×3	0.50	2.23×10^{-2}	-	1.08×10^{-2}	-	7.51×10^{-3}	-
4×4	0.33	1.09×10^{-2}	1.76	3.89×10^{-3}	2.51	1.55×10^{-3}	3.89
5×5	0.25	6.59×10^{-3}	1.74	1.56×10^{-3}	3.17	4.93×10^{-4}	3.98
9×9	0.12	1.91×10^{-3}	1.78	1.89×10^{-4}	3.04	6.35×10^{-5}	2.95
12×12	0.09	8.43×10^{-4}	2.56	8.04×10^{-5}	2.68	2.80×10^{-5}	2.57
18×18	0.05	2.66×10^{-4}	2.64	2.07×10^{-5}	3.11	8.07×10^{-6}	2.43



(a) (b)

(c)

Figure 2.2: Approximation error :(a) Linear basis(MLS), (b) Quadratic basis(MLS), (c) Quadratic basis(MMLS) of Example 3

The exact solution is $u_{ex}(x,y) = xsin(y)$. Table 2.3 shows the maximum errors for different values of m and n that are regularly employed in the unit square domain, and the absolute error for $(n = 18 \times 18)$ is graphically shown in Figure 2.2. We can see that when the values of n increases, the maximum errors decrease, the results of MMLS approximation converge more quickly to the true solution than that of MLS approximation. The ratio of error remains constant for the linear case (≈ 2) and the quadratic case (≈ 3). So the numerical results show that the proposed method will be of $O(h_X^{q+1})$ as it is expected in Theorem 2.5.4.

Example 4. As the final example, we consider the following two-dimensional nonlinear Fredholm-Volterra integral equation

$$
\begin{aligned}
u(x,y) &- \int_0^1 \int_0^1 \exp(-x-y-4) u^2(s,v)\,\mathrm{d}s\,\mathrm{d}v \\
&- \frac{1}{4}\int_0^y \int_0^x (ys+xv)u(s,v)\,\mathrm{d}s\,\mathrm{d}v = g(x,y),\ (x,y) \in [0,1]\times[0,1],
\end{aligned}
$$

where

$$
\begin{aligned}
g(x,y) = \frac{1}{2}\exp(y) + xy - \frac{1}{72}\exp(-x-y-4)\Bigg(35 \\
+ 9\exp(2)\Bigg) - \frac{x^2}{48}\Bigg(4y^3x - 3y + \exp(y)(9y-6) + 6\Bigg).
\end{aligned}
$$

The analytic solution of this problem is $u_{ex}(x,y) = \dfrac{1}{2}exp(y) + xy$. The maximum errors using MLS and

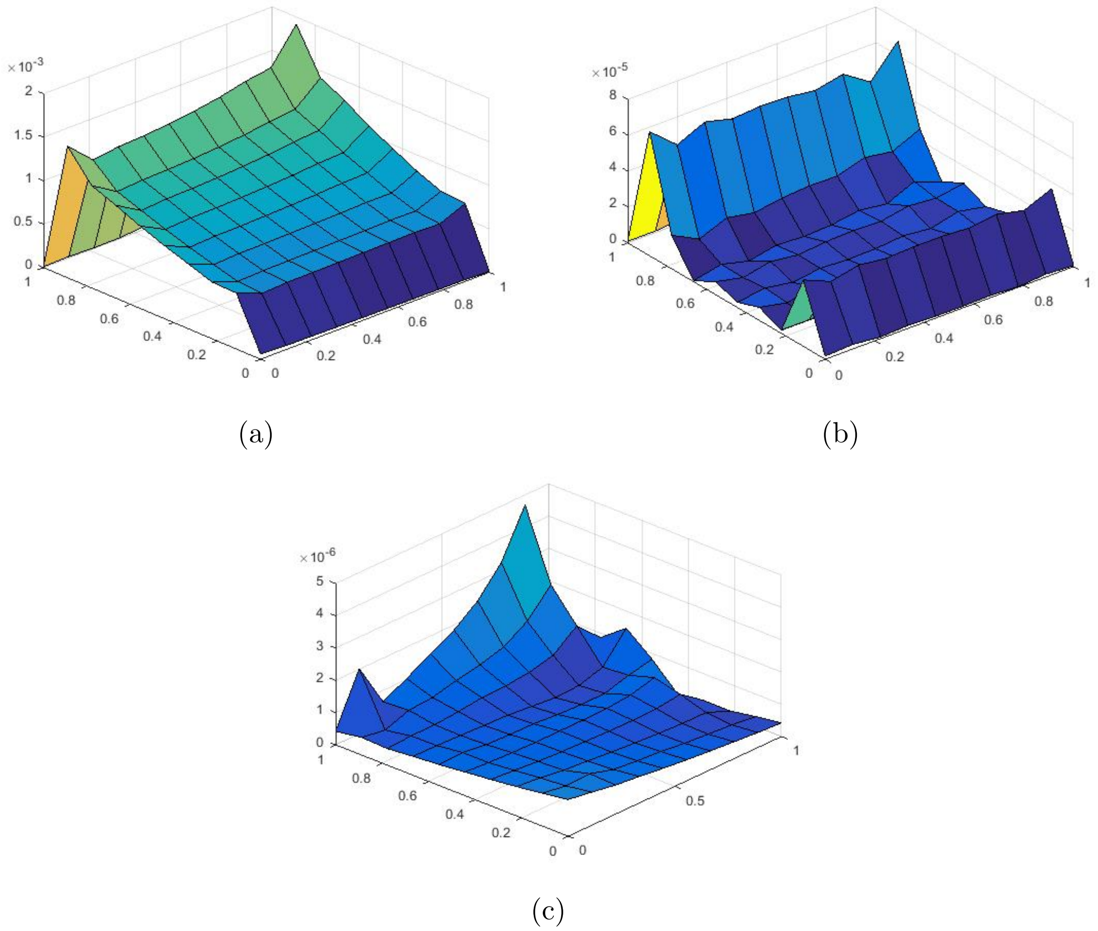
Table 2.4: Maximum errors using different values of n for Example 4

n	h_X	MLS approximation error		MMLS approximation error
		Linear basis	Quadratic basis	Quadratic basis
3×3	0.50	2.44×10^{-2}	1.03×10^{-3}	5.39×10^{-4}
5×5	0.25	7.99×10^{-3}	6.20×10^{-4}	6.31×10^{-5}
6×6	0.20	5.60×10^{-3}	3.60×10^{-4}	2.19×10^{-5}
10×10	0.11	1.91×10^{-3}	6.91×10^{-5}	4.68×10^{-6}

Table 2.5: Maximum errors using different values of ν and $n = 5 \times 5$ for Example 4

ν	10^{-6}	10^{-8}	10^{-9}	10^{-10}
$\|e\|_\infty$	1.42×10^{-4}	6.34×10^{-5}	6.31×10^{-5}	6.31×10^{-5}

MMLS approximation are shown in Table 2.4, the absolute error with ($n = 10 \times 10$) points is plotted in figure 2.3. The results, presented in Table 2.5, support the maximum errors using different values of ν when $n = 5 \times 5$, this value was selected experimentally, we can see that MMLS approximation gives better accuracy when we put ν as small positive numbers. As shown, from Table 2.4, the faster method is modified moving least squares method, also as n (the number of nodes) increases, the error term decreases in both MLS and MMLS approximation. Apparently, the method provides accurate numerical solutions for mixed integral equations.



(a)

(b)

(c)

Figure 2.3: Approximation error: (a) Linear basis(MLS), (b) Quadratic basis(MLS), (c) Quadratic basis(MMLS) of Example 4

2.6.2 Applications on non rectangular domain

Example 5. (see [40]) As the first example, consider (2.4.6) with the following

$$K(x,y,\mu,\theta) = (1-\theta-\mu)e^{-(\mu+\theta+x+y)}, \quad \lambda = 1, \tag{2.6.1}$$

$$g(x,y) = x^2 + y^2 - \frac{1}{3}(98-36e)e^{-x-y-1}, \tag{2.6.2}$$

$$D = \{ \quad (\mu,\theta): \quad 0 \le \theta \le 1, \quad 0 \le \mu \le -\theta + 1 \quad \}. \tag{2.6.3}$$

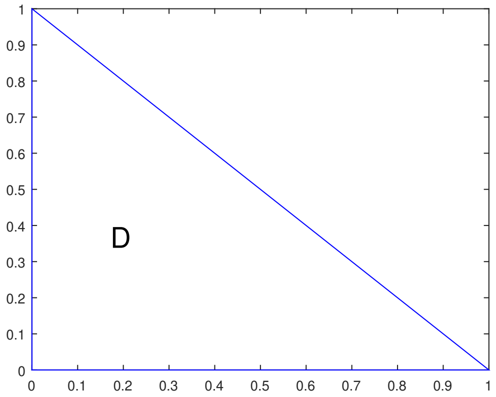


Figure 2.4: The consideration domain D for Example 5

Table 2.6: Numerical results for Example 5 with $m_1 = 2$

(x,y)	$e(x,y)$ of MMLS method			$e(x,y)$ of method [40]
	$n = 11$	$n = 20$	$n = 67$	$n = 1326$
$(0.0,\ 0.0)$	4.09×10^{-5}	1.04×10^{-8}	1.70×10^{-10}	7.23×10^{-3}
$(0.1, 0.1)$	3.37×10^{-5}	8.57×10^{-9}	1.39×10^{-10}	5.92×10^{-3}
$(0.2, 0.2)$	2.75×10^{-5}	7.02×10^{-9}	1.14×10^{-10}	4.84×10^{-3}
$(0.3, 0.3)$	2.24×10^{-5}	5.75×10^{-9}	9.32×10^{-11}	3.96×10^{-3}
$(0.4, 0.4)$	1.82×10^{-5}	4.70×10^{-9}	7.71×10^{-11}	3.25×10^{-3}
$(0.5, 0.5)$	1.50×10^{-5}	3.85×10^{-9}	6.35×10^{-11}	2.66×10^{-3}

The exact solution is $u_{ex}(x,y) = x^2 + y^2$. Domain D is shown in Figure 2.4. Table 2.6 gives absolute errors at different numbers of n when $m_1 = 2$ using the quadratic spline weight function. We have compared the errors of the present method with that in [40]. Figure 2.5 shows the domain for a different number of nodal points. The results show that better accuracy is obtained with the present method by increasing a number of collocation points.

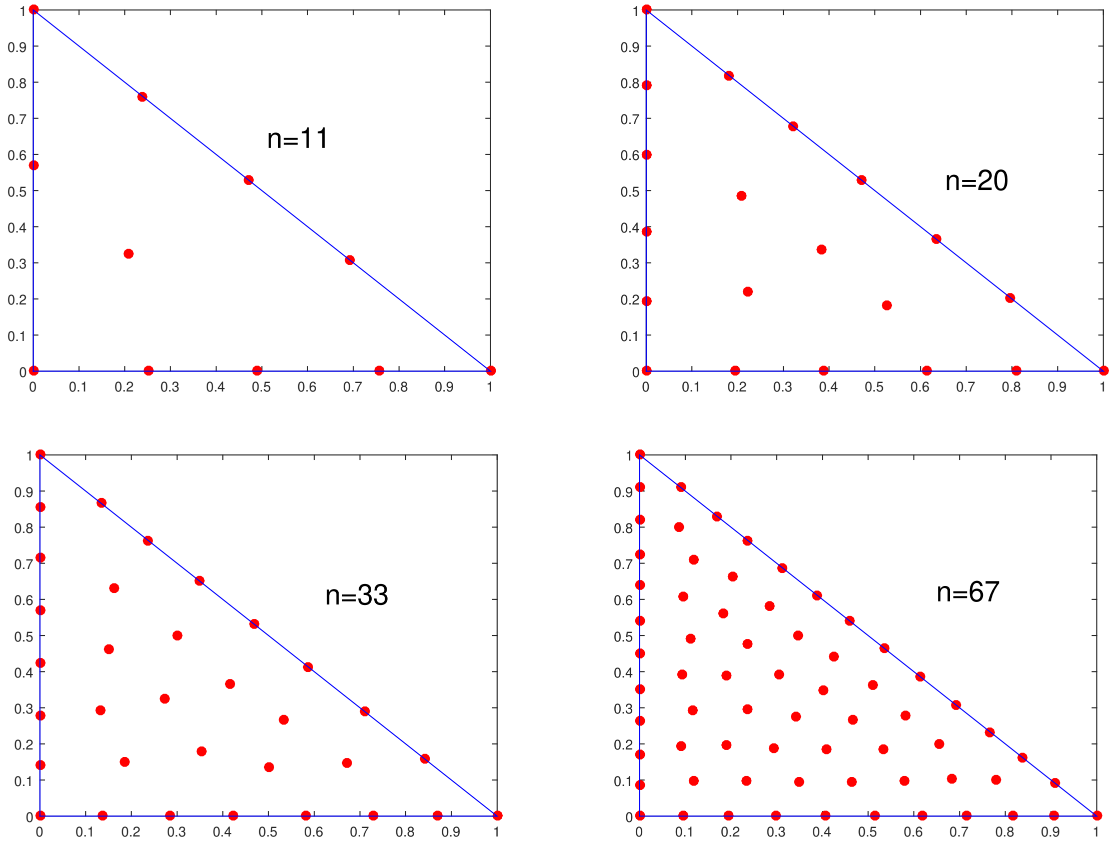


Figure 2.5: Node distribution for Example 5

Example 6. (See [6]) Consider (2.4.6) with the following

$$K(x,y,\mu,\theta) = \mu sin(\theta) + 1 \quad \lambda = 1, \tag{2.6.4}$$

$$g(x,y) = x\cos(y) - 0.2804057157, \tag{2.6.5}$$

$$D = \left\{ \quad (\xi,\eta): \quad 0 \leq \theta \leq 1, \quad \sqrt{\frac{1}{4} - (\eta - \frac{1}{2})^2} \leq \mu \leq \sqrt{\left(1 - 4(\eta - \frac{1}{2})^2\right)} \quad \right\}. \tag{2.6.6}$$

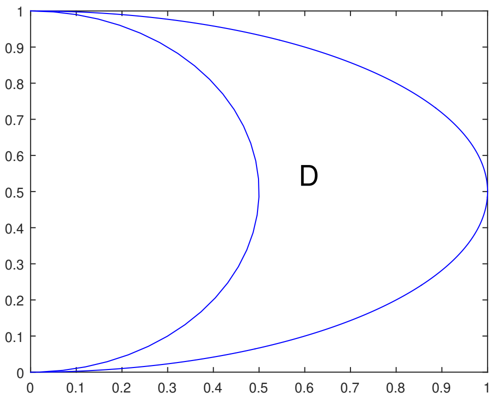


Figure 2.6: The consideration domain D for Example 6

Table 2.7: Numerical results for Example 6 with different values of n and $m_1 = 2$

n	$\|e\|_\infty$ of MMLS method		$\|e\|_\infty$ of method [6]
	Quadratic weight functions	Cubic weight functions	GA weight functions
22	1.79×10^{-4}	1.36×10^{-4}	2.68×10^{-3}
33	6.14×10^{-5}	4.87×10^{-5}	1.65×10^{-3}
47	5.80×10^{-5}	4.45×10^{-5}	1.10×10^{-3}
68	2.48×10^{-5}	1.75×10^{-5}	7.49×10^{-4}
90	1.22×10^{-5}	8.68×10^{-6}	4.73×10^{-4}

The exact solution is $u_{ex}(x, y) = xcos(y)$. D is the crescent domain which is drawn in Figure 2.6. The numerical results of this example are reported in Table 2.7. These results are presented in terms of $\|e\|_\infty$ by using the quadratic and cubic spline weight functions. Also, these results are compared with the method in [6] based on the use of the moving least squares (MLS) approximation. MMLS results converge more quickly to the true solution than that of MLS approximation. The results confirm the high accuracy of the present method.

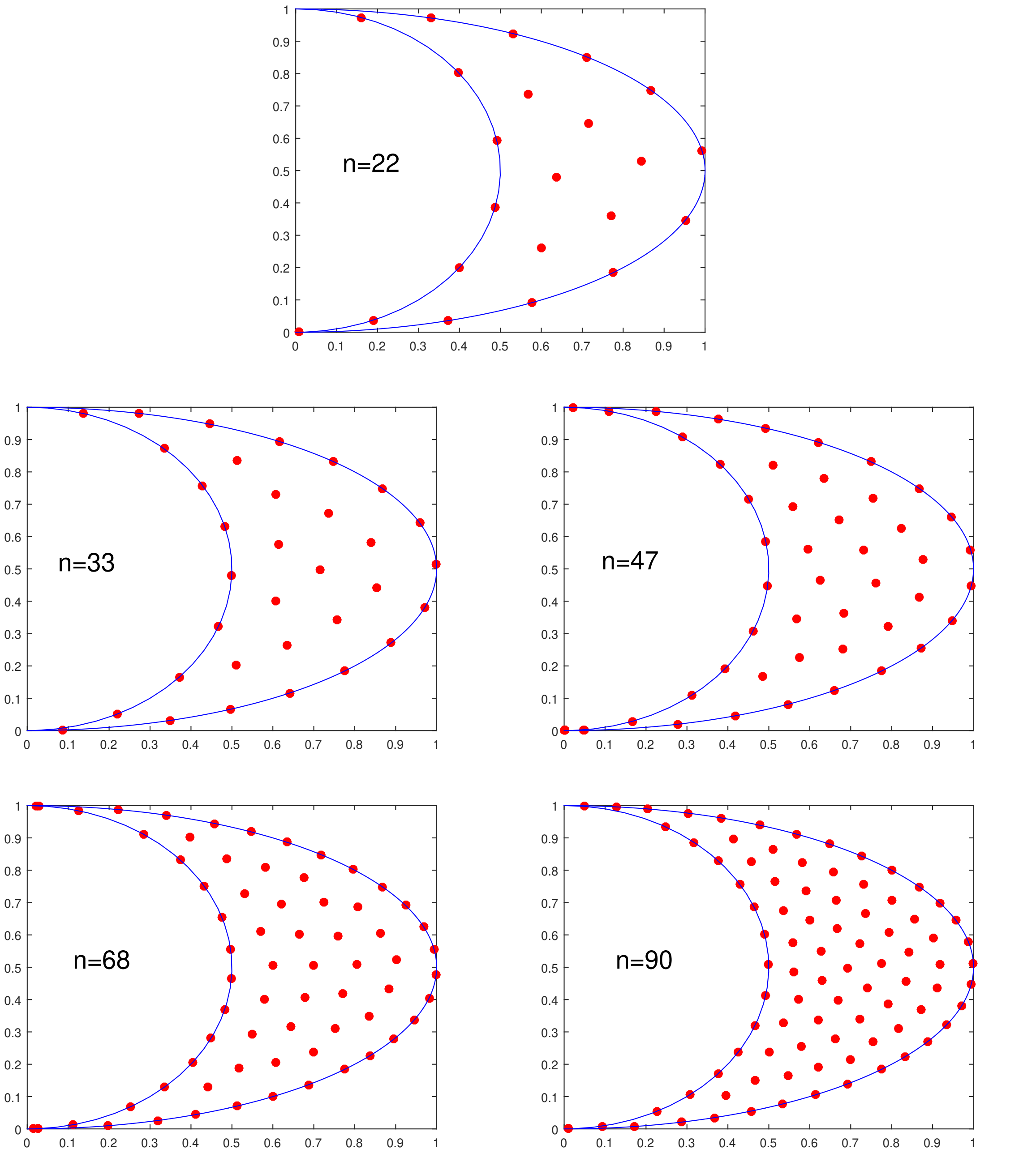


Figure 2.7: Node distribution for Example 6

Example 7. Consider (2.4.6) with the following

$$K(x,y,\mu,\theta) = \frac{e^{-y}sin(-x)}{1+y+2e^{x}} \quad \lambda = 1, \tag{2.6.7}$$

$$g(x,y) = xe^{-y} + y - \frac{e^{-y}sin(-x)}{1+y+2e^{x}}(0.190821 + 0.287982), \tag{2.6.8}$$

$$D_1 = \left\{ (\mu,\theta) : 0 \le \theta \le 1, \ -\sqrt{\frac{1}{4} - (\theta - \frac{1}{2})^2} + \frac{1}{2} < \mu < -\frac{1}{8}\sqrt{1 - 4(\theta - \frac{1}{2})^2} + \frac{1}{2} \right\}, \tag{2.6.9}$$

$$D_2 = \left\{ (\mu,\theta) : 0 \le \theta \le 1, \ \frac{1}{8}\sqrt{1 - 4(\theta - \frac{1}{2})^2} + \frac{1}{2} \le \mu \le \sqrt{\frac{1}{4} - (\theta - \frac{1}{2})^2} + \frac{1}{2} \right\}. \tag{2.6.10}$$

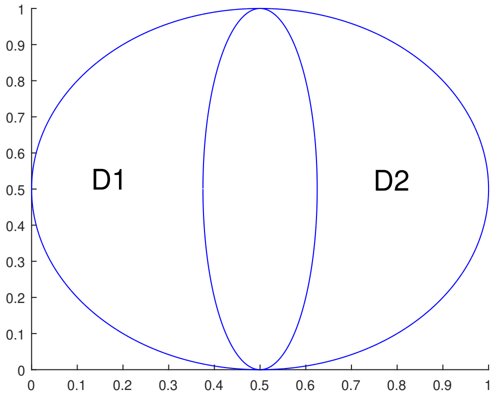


Figure 2.8: The consideration domain for Example 7

Table 2.8: Numerical results for Example 7 with different values of n and $m_1 = 2$

n	$\|e\|_\infty$ of MLS method		$\|e\|_\infty$ of MMLS method	
	Quadratic weight function	Cubic weight function	Quadratic weight function	Cubic weight function
8	$7.54\ \times 10^{-3}$	6.43×10^{-3}	$1.19\ \times 10^{-3}$	$1.56\ \times 10^{-3}$
19	$4.16\ \times 10^{-3}$	3.29×10^{-3}	$1.04\ \times 10^{-4}$	$2.08\ \times 10^{-5}$
35	321×10^{-3}	$2.65\ \times 10^{-3}$	2.28×10^{-5}	4.71×10^{-6}
49	6.53×10^{-4}	4.18×10^{-4}	4.06×10^{-6}	2.01×10^{-6}

The exact solution is $u(x, y) = xe^{-y} + y$. $D = D_1 \cup D_2$ is drawn in Figure 2.8. Figure 2.10 shows the domain for different number of nodal points. The maximum errors using MLS and MMLS approximation are shown in Table 2.8. The obtained errors for the different numbers of nodes are drawn in the logarithmic mode in Figure 2.9. In this example, we use cubic and quadratic spline weight functions. For numerical integration we choose $m_1 = 2$. The numerical results confirm that the MMLS results converge more quickly to the exact solution than that of MLS approximation.

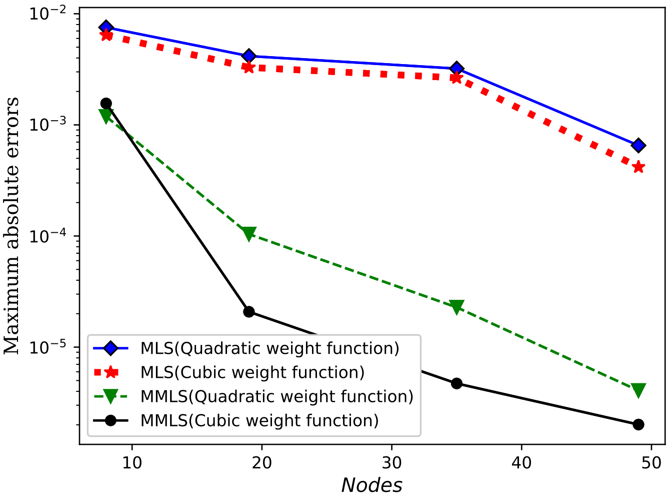


Figure 2.9: Comparison between MLS and MMLS methods for Example 7

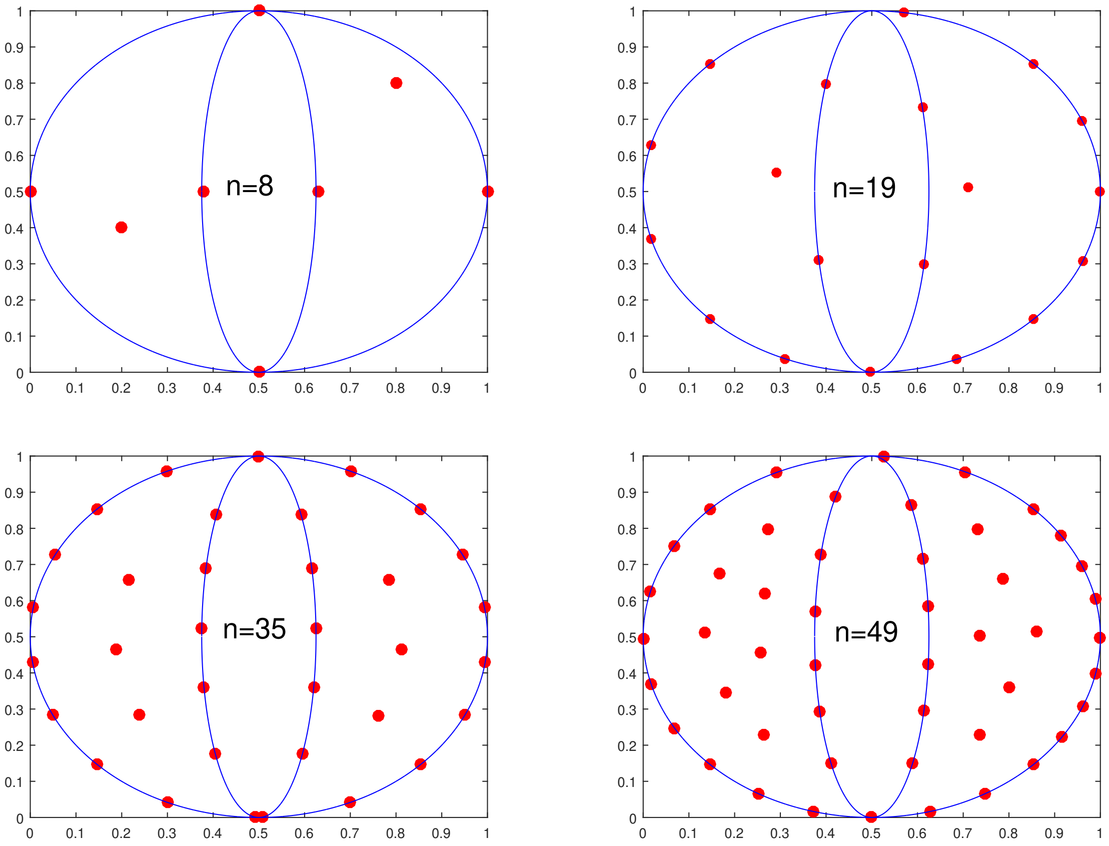


Figure 2.10: Node distribution for Example 7

Example 8. Consider (2.4.6) with the following

$$K(x,y,\mu,\theta) = \frac{\theta e^{-y-\mu} sin(-x)}{1+y+2e^x} \quad \lambda = 1, \tag{2.6.11}$$

$$g(x,y) = xe^{-y} + y - 0.0691965\Big(\frac{e^{-y} sin(-x)}{1+y+2e^x}\Big), \tag{2.6.12}$$

$$D = \Bigg\{ (\mu,\theta) : 0 \le \theta \le 1, \sqrt{\frac{1}{4} - (\theta - \frac{1}{2})^2} < \mu < \sqrt{1-\theta^2} \Bigg\}. \tag{2.6.13}$$

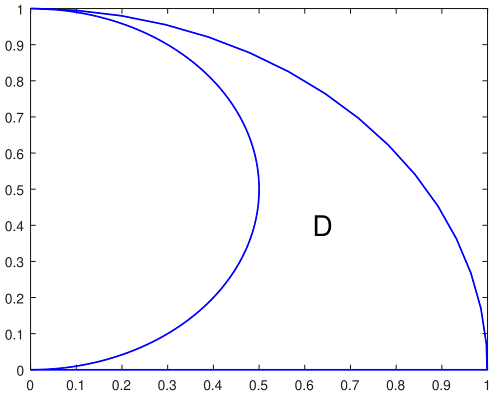


Figure 2.11: The consideration domain D for Example 8

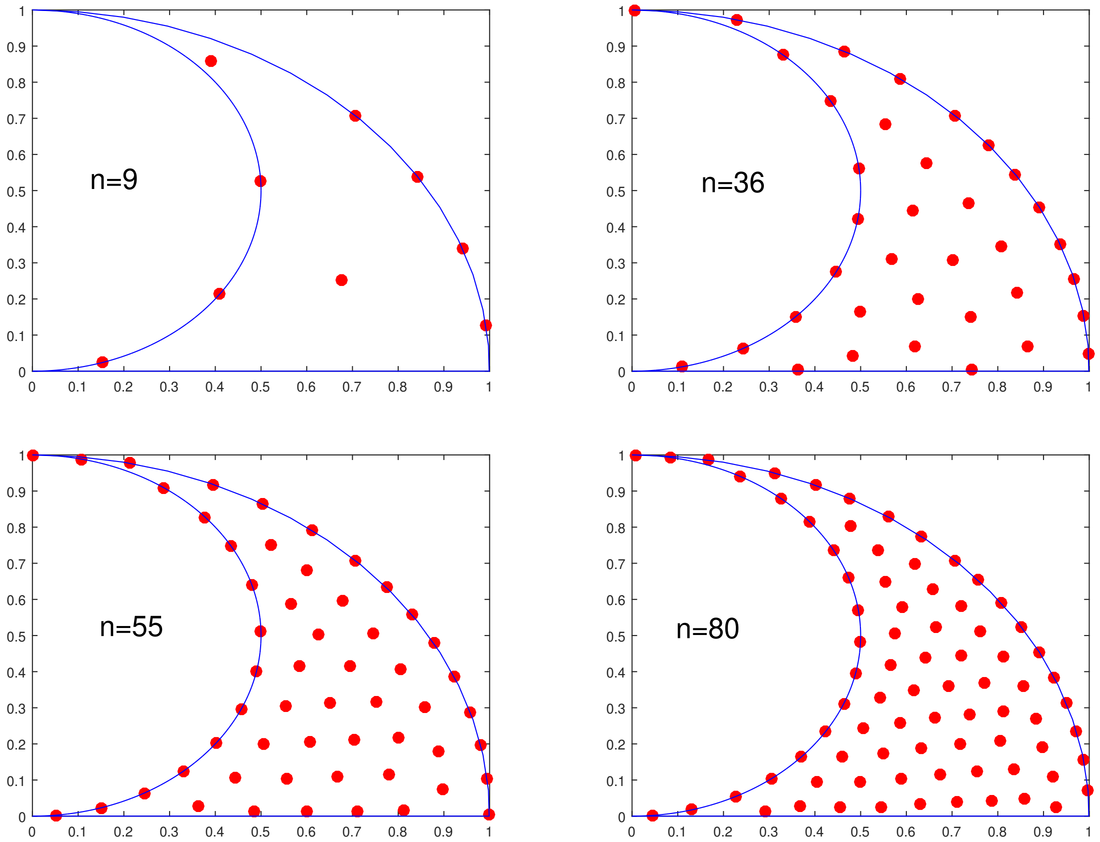


Figure 2.12: Nodes distribution for Example 8

Table 2.9: Maximum error and rate of convergence for Example 8 using MMLS method and $m_1 = 2$

n	h	Quadratic spline weight function		Cubic spline weight function		RBF weight function	
		$\|e\|_\infty$	Ratio	$\|e\|_\infty$	Ratio	$\|e\|_\infty$	Ratio
9	0.50	2.13×10^{-3}	-	2.14×10^{-3}	-	9.51×10^{-4}	-
36	0.20	2.12×10^{-4}	2.51	1.59×10^{-4}	2.83	5.85×10^{-5}	3.04
55	0.15	5.89×10^{-5}	5.13	4.30×10^{-5}	5.24	2.04×10^{-5}	4.22
80	0.12	4.17×10^{-5}	1.61	2.90×10^{-5}	1.84	1.15×10^{-5}	2.68

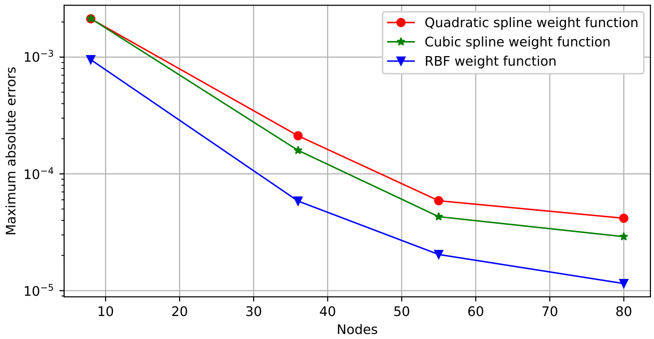


Figure 2.13: A comparison between weight functions for Example 8

The exact solution is $u_{ex}(x, y) = xe^{-y} + y$. The integral equation is solved on the irregular domain depicted in figure 2.11. Figure 2.12 shows the node distribution. Table 4.1 presented the maximum error at some particular points for $m_1 = 2$. Here we put $h = \dfrac{1}{\sqrt{n} - 1}$. A graphical comparison between spline and RBF weight functions is shown in figure 2.13. Numerical MMLS results show that the better results are obtained with RBF weight functions which converge more quickly to the exacte solution.

Example 9. Consider (2.1.1) with the following

$$K(x, y, \mu, \theta) = e^{-x-y-4} \quad \lambda = 1, \tag{2.6.14}$$

$$\Phi(\mu, \theta, u(\mu, \theta)) = u^2(\mu, \theta), \tag{2.6.15}$$

$$g(x, y) = \frac{1}{2}e^{y} + xy - e^{-x-y-4}(0.275604 + 0.499492), \tag{2.6.16}$$

$$D_1 = \left\{ \quad (\mu, \theta): \quad 0 \leq \theta \leq 1, \quad -\frac{1}{2}\sqrt{1 - (\theta - 1)^2} + \frac{1}{2} < \mu < \frac{1}{2}\sqrt{1 - \theta^2} \quad \right\}, \tag{2.6.17}$$

$$D_2 = \left\{ \quad (\mu, \theta): \quad 0 \leq \theta \leq 1, \quad -\frac{1}{2}\sqrt{1 - \theta^2} + 1 \leq \mu \leq \frac{1}{2}\sqrt{1 - (\theta - 1)^2} + \frac{1}{2} \quad \right\}. \tag{2.6.18}$$

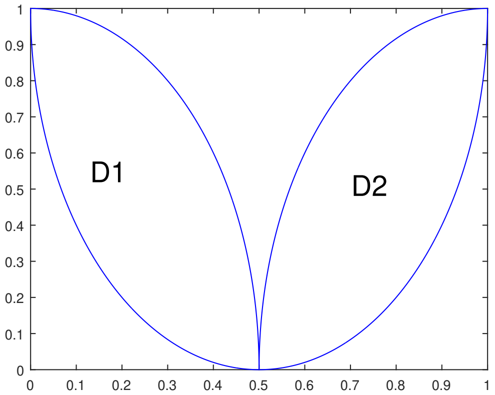


Figure 2.14: The consideration domain for Example 9

Table 2.10: Numerical results for Example 9 with different values of n and $m_1 = 2$

n	$\|e\|_\infty$ of MLS method		$\|e\|_\infty$ of MMLS method	
	Quadratic weight function	Cubic weight function	Quadratic weight function	Cubic weight function
11	5.48×10^{-3}	4.31×10^{-3}	8.97×10^{-4}	5.91×10^{-4}
21	1.96×10^{-3}	1.51×10^{-3}	1.47×10^{-4}	8.51×10^{-5}
39	6.42×10^{-4}	7.60×10^{-4}	9.94×10^{-5}	6.05×10^{-5}
71	2.23×10^{-4}	1.72×10^{-4}	2.19×10^{-5}	1.83×10^{-5}

The exact solution is $u_{ex}(x, y) = \dfrac{1}{2}e^y + xy$. $D = D_1 \cup D_2$ is drawn in Figure 2.14. Table 2.10 illustrates the maximum error for different values of nodes by using MMLS and MLS approximations. The obtained errors are drawn in the logarithmic mode in Figure 2.15; here we use $m_1 = 2$ for numerical integration. Figure 2.16 presented the distribution of nodal points. As shown, from Table 2.10, the faster method is modified moving least squares method.

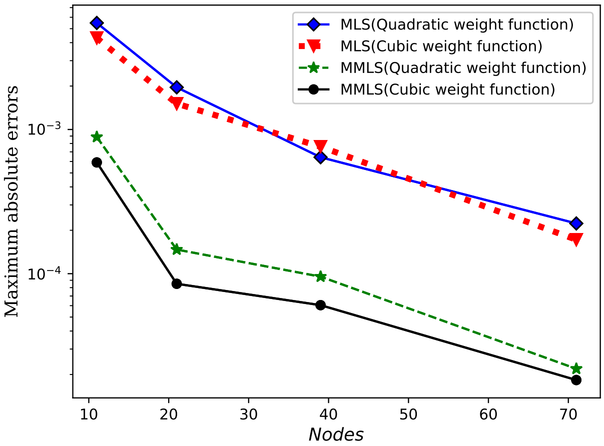


Figure 2.15: Comparison between MLS and MMLS methods for Example 9

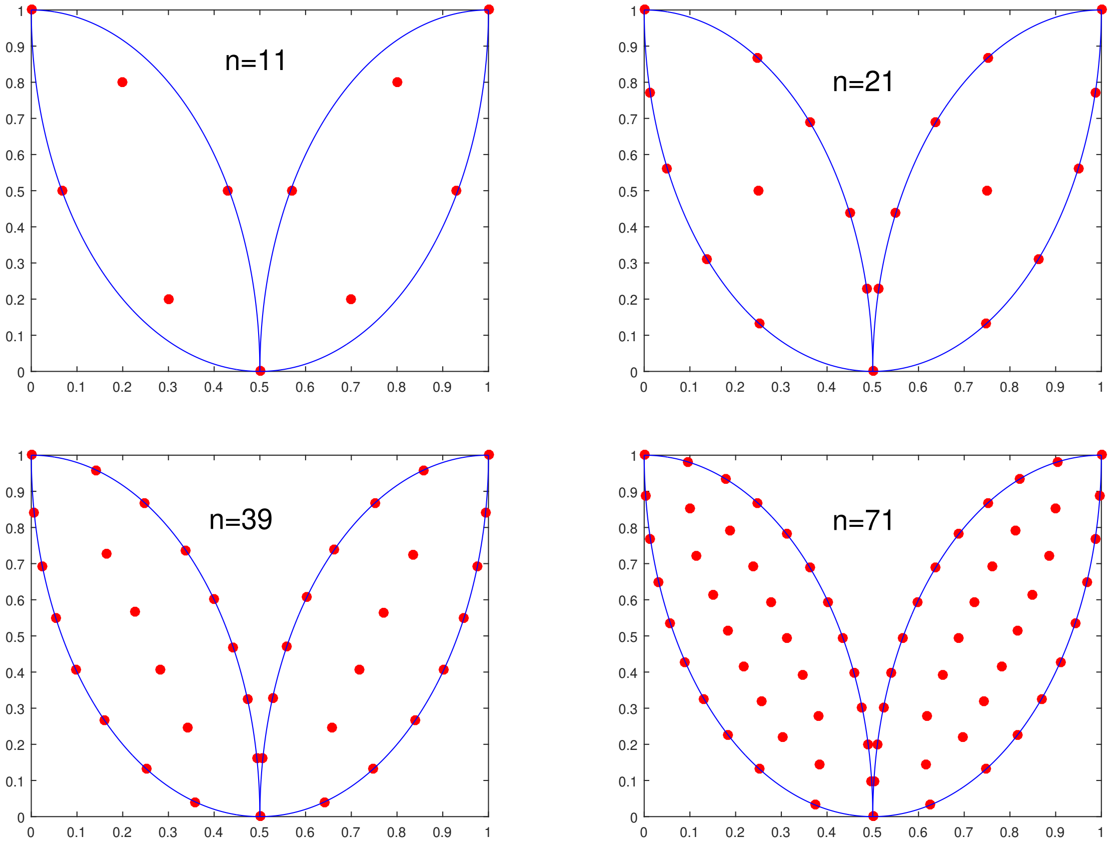


Figure 2.16: Node distribution for Example 9

Example 10. As the final example, consider the following two dimensional Fredholm-Hammerstein integral equation (2.1.1) with the following

$$K(x,y,\mu,\theta) = \theta \quad \lambda = 1, \tag{2.6.19}$$

$$\Phi(\mu,\theta,u(\mu,\theta)) = u^2(\mu,\theta), \tag{2.6.20}$$

$$g(x,y) = xsin(y) - (0.0340488 + 0.00193779), \tag{2.6.21}$$

$$D_1 = \left\{ (\mu,\theta): \quad 0 \leq \theta \leq 1, \quad \frac{-1}{4}(\theta-1) < \mu < \frac{-1}{4}(\theta-2) \right\}, \tag{2.6.22}$$

$$D_2 = \left\{ (\mu,\theta): \quad 0 \leq \theta \leq 1, \quad \frac{1}{4}(\theta+2) \leq \mu \leq \frac{1}{4}(\theta+3) \right\}. \tag{2.6.23}$$

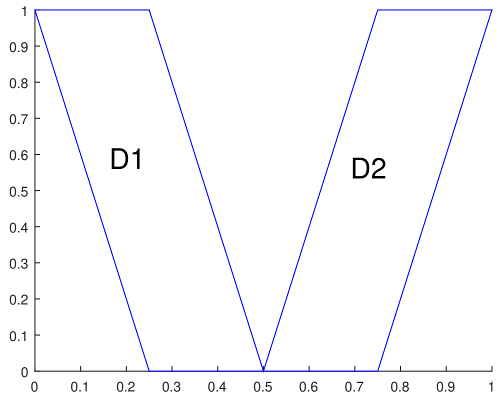


Figure 2.17: The consideration domain D for Example 10

Table 2.11: Numerical results for Example 10 with different values of n and with $m_1 = 2$

N	$\|e\|_\infty$ of MMLS method	
	Quadratic weight function	Cubic weight function
9	$2.07\ \times 10^{-3}$	1.53×10^{-3}
28	$4.44\ \times 10^{-4}$	4.09×10^{-4}
39	5.60×10^{-5}	4.52×10^{-5}
65	2.57×10^{-5}	$2.17\ \times 10^{-5}$

The exact solution is $u_{ex}(x, y) = xsin(y)$. $D = D_1 \cup D_2$ is drawn in Figure 2.17. The maximum error by using the MMLS method is displayed in Table 2.11 for different values of n. The distribution of nodal points is plotted in Figure 2.18. In this example, we put $m_1 = 2$. The results, plotted in Table 2.12, support the maximum error for different values of ν when $n = 39$, this value was selected experimentally. So it is clear that MMLS approximations give better accuracy when we put ν as a small positive number.

Table 2.12: Numerical results for Example 10 using different values of ν and $n = 39$

ν	10^{-5}	10^{-6}	10^{-8}	10^{-9}	10^{-10}
$\|e\|_\infty$	3.76×10^{-3}	5.68×10^{-4}	5.60×10^{-5}	5.49×10^{-5}	5.55×10^{-5}

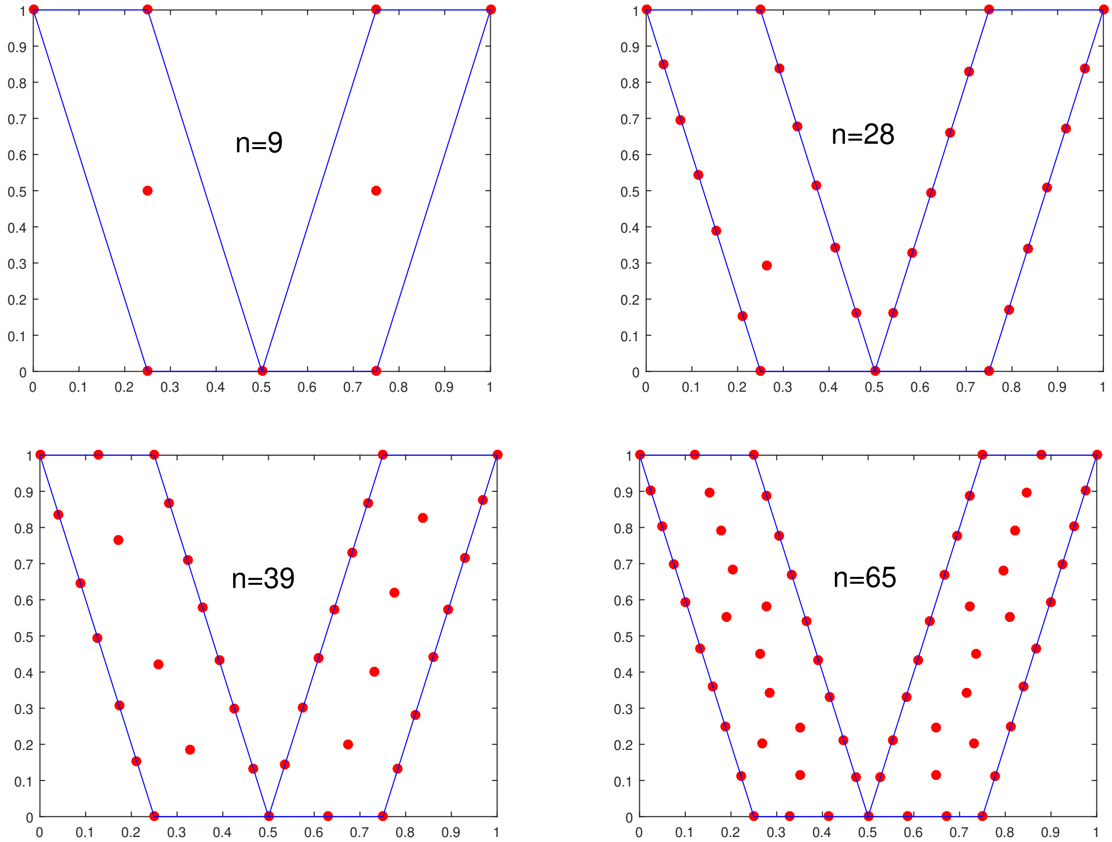


Figure 2.18: Node distribution for Example 10

2.7 Conclusion

In this chapter, two meshless approaches called moving least squares and modified moving least-squares approximation are applied for solving linear and nonlinear Fredholm-Volterra integral equations on 2D regular and irregular domains. These approaches don't need any background interpolation or approximation cells and they don't depend to the geometry of the domain. Comparing the results obtained by these methods with the results obtained by the exact solution shows the efficiency and good accuracy of the MMLS scheme for solving a system of functional equations.

Chapter 3

An implementation of modified moving least square for solving multidimensional mixed integral equations in 3D hypercube domains

In this chapter, the three-dimensional (3-D) modified moving least square method is developed for solving the three-dimensional linear and nonlinear Volterra-Fredholm integral equations of the second kind. This approach is very convenient for solving integral equations in high dimensions and it does not require any need for mesh connectivity. The size of the support used is the only factor that has a significant effect on the maximum errors of the MLS method. To overcome this problem, the MMLS approach with a nonsingular moment matrix is applied to obtain better results than MLS approximation on using the best support, then applying the method in three dimensions can be easily achieved. The numerical experiments of the MMLS and classical MLS techniques are presented to show the difference between both methods for multi-dimensional problems. The convergence analysis is provided and some numerical tests are given to prove the applicability of this technique.

The results obtained in this chapter are presented in the research article [31], in collaboration with Rachid El Jid and Abdelkarim Hajjaj.

3.1 Introduction

The multi-dimensional integral equations can be arisen in many branches of sciences and provide an important tool for modeling many problems in mathematics, physics, and engineering. These equations appear in many branches of applied science such as, determination of elastic waves [21], methods for solving electromagnetic problems [20, 125], applications to heat and mass transfer problems [58], predicting water structure around molecules [79], the reformulation of an asexual population in a changing environment [118] and the investigation of population growth in a closed model [114]. Three-dimensional integral equations are generally difficult to be solved analytically, for this reason. In this research, we expand the MMLS method for solving a class of multi-dimensional integral equation on the form

- Three-dimensional linear Volterra-Fredholm integral equations of the second kind

$$\begin{aligned} u(x,y,z) - \lambda_1 \int_0^x \int_0^y \int_0^z K_1(x,y,z,\mu,\theta,\nu) u(\mu,\theta,\nu)\, \mathrm{d}\nu d\theta d\mu \\ - \lambda_2 \int_0^1 \int_0^1 \int_0^1 K_2(x,y,z,\mu,\theta,\nu) u(\mu,\theta,\nu)\, \mathrm{d}\nu d\theta d\mu = g(x,y,z), \qquad (3.1.1) \end{aligned}$$

  where $\lambda_1,\ \lambda_2$ are real numbers, also $K_1,\ K_2$ and g defined respectively on $D = [0,1]\times[0,1]\times[0,1]$, u is an unknown function and $\ N = \{(x,y,z,\mu,\theta,\nu),\ 0 \le \mu \le x \le 1,\ 0 \le \theta \le y \le 1,\ 0 \le \nu \le z \le 1\}$.

- Three-dimensional nonlinear mixed Volterra-Fredholm integral equations of the second kind

$$u(x,y,z) - \lambda \int_0^x \int_0^1 \int_0^1 K(x,y,z,\mu,\theta,\nu) \Psi\Big(\mu,\theta,\nu,u(\mu,\theta,\nu)\Big)\, \mathrm{d}\nu d\theta d\mu = g(x,y,z), \qquad (3.1.2)$$

λ is a real number, g, K and Ψ are assumed to be given. u is an unknown function and $S(D) = \{(x, y, z, \mu, \theta, \nu), 1 \leq \mu \leq x \leq 1, (y, z, \theta, \nu) \in [0, 1]\}$. The following assumption is made:

$$|\Psi(x, y, z, u_1) - \Psi(x, y, z, u_2)| \leq C_1|u_1 - u_2|, \tag{3.1.3}$$

where Ψ is lipschitz with respect to the fourth variable.

To avoid any surface discretization or mesh connectivity, the meshless methods have gained more attention, particularly moving least squares (MLS) method, it has been applied in many branches of modern sciences, such as surface construction [2], function approximation [77] and meshless (meshfree) methods [6, 68, 83]. Nevertheless, the moment matrix in MLS method may be singular because of the small number of points used in the local support. In this work, we propose the modified moving least square method (MMLS) to overcome this difficulty and finding the best support, then the results converge more quickly to the exact solution and give better computational efficiency and accuracy than that of MLS approximation.

Computational complexity of mathematical operations seems to be the most principal difficulty to approximate integral equations in higher dimensions. Then a few numerical methods can be efficient and applicable to solve the high dimensional problems. In [28], the meshfree method based on the use of radial basis functions was applied to the solution of three-dimensional Fredholm integral equations. In [91] the modified block-pulse functions were investigated to the three-dimensional Volterra-Fredholm integral equations, and in [1] a new pseudo spectral technique was presented for the three-dimensional integral equations (3D-IEs). The new idea in this paper is to expand MMLS method to approximate 3D nonlinear Volterra-Fredholm integral equations, this method does not arise the difficulties for higher dimensional problems because of the simple adaption of the MMLS and it is more flexible for most classes of multi-dimensional integral equations.

This work is organized as follows: We briefly describe the three-dimensional MLS method in Section 3.2. In the Section 3.3, the three-dimensional modified moving least square approximation is presented. The computational method for solving linear and nonlinear three-dimensional Volterra-Fredholm integral equations are introduced in Section 3.4. Section 3.5 investigates the error estimate of the present method. Some numerical examples are presented in Section 3.6. We ended the chapter in Section 3.7.

3.2 Three-dimensional MLS approximation

In this work, the moving least squares approximation is used to estimate a function of three variables $X = \{(x_1, y_1, z_1), (x_2, y_2, z_2), ..., (x_n, y_n, z_n)\}$ on the cubic form $D = [0, 1] \times [0, 1] \times [0, 1]$. Let $u : D \to \mathbb{R}$ be a continuous real function and points $(x_i, y_i, z_i), \quad i = 1, 2, ..., n$ are known. The main point of this meshless method is to approximate a function $u(x, y, z)$ for every point $(x, y, z) \in D$ based on the weighted least square. Let P_q be the space of polynomials of degree $q \ll n$. The MLS approximation $u^h(x, y, z)$ of $u(x, y, z)$, $\forall(x, y, z) \in \overline{D}$, can be given as

$$u^h(x, y, z) = p^T(x, y, z)a(x, y, z), \quad \forall(x, y, z) \in \overline{D}, \tag{3.2.1}$$

where

$$p^T(x, y, z) = [p_0(x, y, z), p_1(x, y, z), ..., p_q(x, y, z)],$$

and $\{p_i(x, y, z)\}_{i=0}^q$ is a complete basis of P_q and

$$a(x, y, z) = [a_0(x, y, z), a_1(x, y, z), ..., a_q(x, y, z)],$$

are unknown coefficients to be determined. The MLS method presents the approximate function $u^h(x, y, z)$ in a particularized class of differentiable functions which minimizes the quantity

$$J(x, y, z) = \sum_{i=1}^{n} \omega_i(x, y, z)(p^T(x_i, y_i, z_i)a(x, y, z) - u_i)^2 \tag{3.2.2}$$

$$= [P.a(x, y, z) - u]^T.W.[P.a(x, y, z) - u],$$

Where $\omega_i(x, y, z)$ is the weight function associated with node i, (x_i, y_i, z_i) denotes the value of (x, y, z) at node i, n is the number of nodes in $\overline{D}$ with $w_i(x, y, z) > 0$ and u_i are the fictitious nodal values. The matrix P and W are defined as

$$\begin{aligned} P &= [p^T(x_1, y_1, z_1), p^T(x_2, y_2, z_2), ..., p^T(x_n, y_n, z_n)]^T_{n\times(q+1)}, \\ W &= diag(\omega_i(x, y, z)), \quad i = 1, 2, ..., n. \end{aligned}$$

A necessary condition for $J(x,y,z)$ to be minimized is $\nabla J = 0$, which implies the following normal equation

$$\sum_{i=1}^{n} \omega_i(x,y,z)p(x_i,y_i,z_i)p^T(x_i,y_i,z_i)a(x,y,z) = \sum_{i=1}^{n} \omega_i(x,y,z)p(x_i,y_i,z_i)u_i, \tag{3.2.3}$$

Using the moment matrix

$$A(x,y,z) = \sum_{i=1}^{n} \omega_i(x,y,z)p(x_i,y_i,z_i)p^T(x_i,y_i,z_i),$$

and setting

$$u = [u_1, u_2, ..., u_n]^T,$$

and

$$B(x,y,z) = [w_1(x,y,z)p(x_1,y_1,z_1), w_2(x,y,z)p(x_2,y_2,z_2), ..., w_n(x,y,z)p(x_n,y_n,z_n)],$$

(3.2.3) becomes as follows

$$A(x,y,z)a(x,y,z) = B(x,y,z)u, \tag{3.2.4}$$

and by selecting the nodal points such that $A(x)$ is nonsingular, (3.2.4) can be written as

$$\mathbf{a}(x,y,z) = A^{-1}(x,y,z)B(x,y,z)u. \tag{3.2.5}$$

Substituting (3.2.5) into (3.2.1) we obtain

$$u^\rho(x,y,z) = p(x,y,z)^T A^{-1}(x,y,z)B(x,y,z)u = \sum_{i=1}^{n} \phi_i(x,y,z)u_i, \quad (x,y,z) \in \overline{D}, \tag{3.2.6}$$

where

$$\phi_i(x,y,z) = \sum_{k=1}^{q} p_k(x,y,z)[A^{-1}(x,y,z)B(x,y,z)]_{ki},$$

$\phi_i(x,y,z)$ are called the shape functions of the MLS approximation.

3.3 Three-dimensional modified MLS approximation

A necessary condition for the moment matrix to be nonsingular is that $n \geq q$ at any node. For three-dimensional domains, for example on a linear basis, each point is covered by at least 3 non-aligned node supports and 9 non-aligned node supports for a quadratic basis. Practically, this condition imposes a lower limit on the size of the support domain; the supports must be large enough so that each point is sufficiently covered. Therefore, the computation of the MLS shape functions needs the inverse of the moment matrix which seems more difficult in 3-D space. In this case, Joldes [61] suggested a modified MLS method with an invertible matrix. In this chapter, we restrict ourselves to expand this technique in 3D surfaces. Additional constraints are added in this modification and the coefficients are defined respectively as

- The linear basis ($q = 3$) $\qquad p(x) = [1, x, y, z]^T$,
- The quadratic basis ($q = 9$) $\qquad p(x) = [1, x, y, z, xy, xz, yz, x^2, y^2, z^2]^T$.

The unknown coefficients a_j are defined as

$$a(x) = [a_1, a_x, a_y, a_z, a_{xy}, a_{zy}, a_{xz}, a_{x^2}, a_{y^2}, a_{z^2}]^T.$$

The moment matrix is singular means that the classical MLS minimization has multiple solutions, so the functional (3.2.2) does not include enough constraints to assure a unique solution for the given node distribution. To overcome this problem, we propose to add additional constraints to the functional (3.2.2). Therefore, (3.2.2) can be represented as follows

$$\bar{J}(x,y,z) = \sum_{i=1}^{n} \omega_i(x,y,z)[u^h(x,y,z) - u_i]^2 + \nu_{x^2}a_{x^2}^2 + \nu_{y^2}a_{y^2} + \nu_{z^2}a_{z^2}^2 + \nu_{xy}a_{xy}^2 + \nu_{zy}a_{zy}^2 + \nu_{xz}a_{xz}^2 \tag{3.3.1}$$

where $\nu = [\nu_{x^2}\ \nu_{y^2}\ \nu_{z^2}\ \nu_{xy}\ \nu_{zy}\ \nu_{xz}]$ is a vector of positive weights for the additional constraints. By choosing ν as a small positive number we can assure that the classical MLS solution is small changed and the moment matrix is invertible.
The modified matrix and the matrix form of (3.3.1) are

$$\bar{J}(x,y,z) = [P.a - u_i]^T.W.[P.a - u_i] + a^T Ma, \quad i = 1,2,...,n \tag{3.3.2}$$

and

$$M = \begin{bmatrix} \mathbf{O}_{4,4} & \mathbf{O}_{4,6} \\ \mathbf{O}_{6,4} & diag(\nu) \end{bmatrix}$$

O is the zero matrix and the last six diagonal entries equal to ν.
By minimizing the functional (3.3.2), the coefficients $a(x,y,z)$ defined by

$$\bar{A}(x,y,z)a(x,y,z) = B(x,y,z)u_i,$$

where

$$\bar{A} = P^T.W.P + M.$$

The modified approximation can be determined as follows

$$\bar{u}^h(x,y,z) = \sum_{i=1}^{n} \bar{\phi}_i(x,y,z)u_i, \tag{3.3.3}$$

where the MMLS shape functions is given by

$$\begin{aligned}\bar{\Phi}(x,y,z) &= [\bar{\phi}_1(x,y,z), \bar{\phi}_2(x,y,z), ..., \bar{\phi}_n(x,y,z)] \\ &= p^T(x,y,z)(P^T.W.P + M)^{-1}B(x,y,z).\end{aligned}$$

Remark 2. The proof of nonsingular matrix $\bar{A}$ is shown in [30].

In this work, we use the following different weight functions
The cubic spline weight function

$$w_i(x,y,z) = \begin{cases} \frac{2}{3} - 4\left(\frac{d_i}{\rho_i}\right)^2 + 4\left(\frac{d_i}{\rho_i}\right)^3, & \text{si } 0 \le d_i \le \frac{1}{2}\rho_i, \\ \frac{4}{3} - 4\left(\frac{d_i}{\rho_i}\right) + 4\left(\frac{d_i}{\rho_i}\right)^2 - \frac{4}{3}\left(\frac{d_i}{\rho_i}\right)^3, & \text{si } \frac{1}{2}\rho_i < d_i \le \rho_i, \\ 0, & \text{si } d_i > \rho_i. \end{cases}$$

The quadratic spline weight function

$$w_i(x,y,z) = \begin{cases} 1 - 6\left(\frac{d_i}{\rho_i}\right)^2 + 8\left(\frac{d_i}{\rho_i}\right)^3 - 3\left(\frac{d_i}{\rho_i}\right)^4, & \text{si } 0 \le d_i \le \rho_i, \\ 0, & \text{si } d_i > \rho_i. \end{cases}$$

The RBF weight function

$$w_i(x,y,z) = \begin{cases} (1-d_i)^6\left(6 + 36d_i + 82d_i^2 + 72d_i^3 + 30d_i^4 + 5d_i^5\right), & \text{si } 0 \le d_i \le \rho_i, \\ 0, & \text{si } d_i > \rho_i. \end{cases}$$

where $d_i = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}$ (the Euclidean distance between nodes), ρ_i is the size of the support domain.

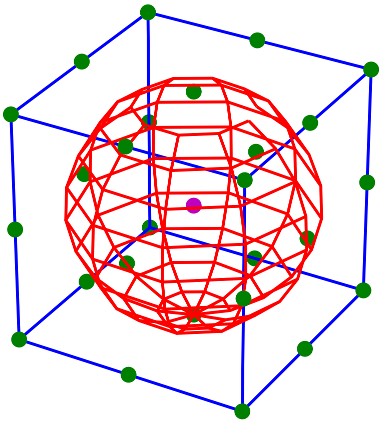


Figure 3.1: $3 \times 3 \times 3$ uniformly distributed nodes on the cub domain and the red sphere is the support of the central node

3.4 Numerical scheme

3D linear mixed Volterra-Fredholm integral equation

Consider the following three-dimensional Volterra-Fredholm integral equation on the form

$$\begin{aligned} u(x,y,z) - \lambda_1 \int_0^x \int_0^y \int_0^z K_1(x,y,z,\mu,\theta,\nu) u(\mu,\theta,\nu)\, \mathrm{d}\nu d\theta d\mu \\ - \lambda_2 \int_0^1 \int_0^1 \int_0^1 K_2(x,y,z,\mu,\theta,\nu) u(\mu,\theta,\nu)\, \mathrm{d}\nu d\theta d\mu = g(x,y,z), \quad (x,y,z) \in D. \end{aligned}$$

The intervals $[0,x], [0,y], [0,z]$ are converted respectively to the fixed intervals $[0,1], [0,1], [0,1]$ by the following linear transformations

$$\mu(x,\delta) = x\delta, \quad \theta(y,\sigma) = y\sigma, \quad \nu(z,\alpha) = z\alpha,$$

then the above equation can be reformulated as

$$\begin{aligned} u(x,y,z) - \lambda_1 \int_0^1 \int_0^1 \int_0^1 \bar{K}_1\Big(x,y,z,\mu(x,\delta),\theta(y,\sigma),\nu(z,\alpha)\Big) u\Big(\mu(x,\delta),\theta(y,\sigma),\nu(z,\alpha)\Big)\, \mathrm{d}\alpha d\sigma d\delta \\ - \lambda_2 \int_0^1 \int_0^1 \int_0^1 K_2(x,y,z,\mu,\theta,\nu) u(\mu,\theta,\nu)\, \mathrm{d}\nu d\theta d\mu = g(x,y,z), \qquad (3.4.1) \end{aligned}$$

where

$$\bar{K}_1\Big(x,y,z,\mu(x,\delta),\theta(y,\sigma),\nu(z,\alpha)\Big) = xyzK_1\Big(x,y,z,\mu(x,\delta),\theta(y,\sigma),\nu(z,\alpha)\Big).$$

If we replace $u(x,y,z)$ by $\bar{u}^h(x,y,z)$, we obtain

$$\begin{aligned} \bar{u}^h(x,y,z) - \lambda_1 \int_0^1 \int_0^1 \int_0^1 \bar{K}_1\Big(x,y,z,\mu(x,\delta),\theta(y,\sigma),\nu(z,\alpha)\Big) \bar{u}^h\Big(\mu(x,\delta),\theta(y,\sigma),\nu(z,\alpha)\Big)\, \mathrm{d}\alpha d\sigma d\delta \\ - \lambda_2 \int_0^1 \int_0^1 \int_0^1 K_2(x,y,z,\mu,\theta,\nu) \bar{u}^h(\mu,\theta,\nu)\, \mathrm{d}\nu d\theta d\mu = g(x,y,z), \qquad (3.4.2) \end{aligned}$$

or equivalently

$$\sum_{j=1}^{n}\Bigg[\bar{\phi}_j(x,y,z)-\lambda_1\int_0^1\int_0^1\int_0^1\bar{K}_1\Big(x,y,z,\mu(x,\delta),\theta(y,\sigma),\nu(z,\alpha)\Big)$$

$$\bar{\phi}_j\Big(\mu(x,\delta),\theta(y,\sigma),\nu(z,\alpha)\Big)\,\mathrm{d}\alpha d\sigma d\delta-\lambda_2\int_0^1\int_0^1\int_0^1 K_2(x,y,z,\mu,\theta,\nu)$$

$$\bar{\phi}_j(\mu,\theta,\nu)\,\mathrm{d}\nu d\theta d\mu\Bigg]u_j=g(x,y,z). \tag{3.4.3}$$

If this equation holds at (x_i,y_i,z_i), so we obtain

$$\sum_{j=1}^{n}\Bigg[\bar{\phi}_j(x_i,y_i,z_i)-\lambda_1\int_0^1\int_0^1\int_0^1\bar{K}_1\Big(x_i,y_i,z_i,\mu(x_i,\delta),\theta(y_i,\sigma),\nu(z_i,\alpha)\Big)$$

$$\bar{\phi}_j\Big(\mu(x_i,\delta),\theta(y_i,\sigma),\nu(z_i,\alpha)\Big)\,\mathrm{d}\alpha d\sigma d\delta-\lambda_2\int_0^1\int_0^1\int_0^1 K_2\big(x_i,y_i,z_i,\mu,\theta,\nu\big)$$

$$\bar{\phi}_j\big(\mu,\theta,\nu\big)\,\mathrm{d}\nu d\theta d\mu\Bigg]u_j=g(x_i,y_i,z_i).\quad i=1,2,...,n. \tag{3.4.4}$$

We utilize m_1 point quadrature formula with the quadrature points $\delta_h,\sigma_p,\alpha_k,\mu_h,\theta_p,\nu_k$ and the quadrature weights $\{w_h\},\{w_k\},\{w_p\}$ for numerical integration. Then the equation can be represented as follows

$$\sum_{j=1}^{n}F_{i,j}\hat{u}_j=g(x_i,y_i,z_i),\quad i=1,2,...,n,$$

where $\hat{u}_j$ are the approximate quantities of u_j and F is a matrix given by

$$F_{i,j}=\Bigg[\bar{\phi}_j(x_i,y_i,z_i)-\lambda_1\sum_{h=1}^{m_1}\sum_{p=1}^{m_1}\sum_{k=1}^{m_1}w_hw_kw_p\bar{K}_1\Big(x_i,y_i,z_i,\mu(x_i,\delta_h),\theta(y_i,\sigma_p),\nu(z_i,\alpha_k)\Big)$$

$$\bar{\phi}_j\Big(\mu(x_i,\delta_h),\theta(y_i,\sigma_p),\nu(z_i,\alpha_k)\Big)-\lambda_2\sum_{h=1}^{m_1}\sum_{p=1}^{m_1}\sum_{k=1}^{m_1}w_hw_kw_pK_2(x_i,y_i,z_i,\mu_h,\theta_p,\nu_k)$$

$$\bar{\phi}_j(\mu_h,\theta_p,\nu_k)\Bigg].$$

where

$$\hat{u}=[\hat{u}_1,\hat{u}_2,...,\hat{u}_n]^T,$$

$$g=[g_1,g_2,...,g_n]^T.$$

So the linear system of equations can be reformulated as

$$F\hat{u}=g. \tag{3.4.5}$$

Solving (3.4.5), we can approximate $u(x,y,z)$ as in (3.3.3) by

$$\bar{u}^h(x,y,z)=\sum_{j=1}^{n}\bar{\phi}_j(x,y,z)\hat{u}_j,\quad (x,y,z)\in D.$$

3D nonlinear mixed Volterra-Fredholm integral equation

We utilize Kumar and Sloan approach [67] to solve three-dimensional nonlinear mixed Volterra-Fredholm integral equations of the second kind (3.1.2).

$$\begin{aligned}s(x,y,z)&=\Psi\Big(x,y,z,u(x,y,z)\Big)\\&=\Psi\Bigg(x,y,z,g(x,y,z)+\lambda\int_0^x\int_0^1\int_0^1 K(x,y,z,\mu,\theta,\nu)s(\mu,\theta,\nu)\,\mathrm{d}\nu d\theta d\mu\Bigg).\end{aligned} \tag{3.4.6}$$

To approximate the solution of this integral equation, we replace the unknown function $s(x, y, z)$ by the modified approximation given by

$$s(x, y, z) = \sum_{j=1}^{n} \alpha_j \bar{\phi}_j(x, y, z). \tag{3.4.7}$$

By replacing (3.4.7) into (3.4.6) and interpolating the collocation points (x_i, y_i, z_i);
$i = 0, ..., n.$ we obtain

$$\begin{aligned}\sum_{j=1}^{n} \alpha_j \bar{\phi}_j(x_i, y_i, z_i) &= \Psi(x_i, y_i, z_i, u(x_i, y_i, z_i)) \\ &= \Psi\Bigg(x_i, y_i, z_i, g(x_i, y_i, z_i) + \lambda \int_0^x \int_0^1 \int_0^1 K(x_i, y_i, z_i, \mu, \theta, \nu) \\ &\quad \sum_{j=1}^{n} \alpha_j \bar{\phi}_j(\mu, \theta, \nu)\, \mathrm{d}\nu d\theta d\mu\Bigg).\end{aligned} \tag{3.4.8}$$

The interval [0, x] is transformed to the fixed interval [0, 1] by the linear transformation as follows

$$\mu(x, \delta) = x\delta.$$

Equation becomes

$$\begin{aligned}\sum_{j=1}^{n} \alpha_j \bar{\phi}_j(x_i, y_i, z_i) = \Psi\Bigg(x_i, y_i, z_i, g(x_i, y_i, z_i) + \lambda \int_0^1 \int_0^1 \int_0^1 \bar{K}(x_i, y_i, z_i, \mu(x, \delta), \theta, \nu) \\ \sum_{j=1}^{n} \alpha_j \bar{\phi}_j(\mu(x, \delta), \theta, \nu)\, \mathrm{d}\nu d\theta d\delta\Bigg),\end{aligned} \tag{3.4.9}$$

where

$$\bar{K}\Big(x, y, z, \mu(x, \delta), \theta, \nu\Big) = xK\Big(x, y, z, \mu(x, \delta), \theta, \nu\Big).$$

We use m_1 point quadrature formula with the quadrature points $\{\delta_h\}, \{\theta_p\}, \{\nu_k\}$ and the quadrature weights $\{w_h\}, \{w_k\}, \{w_p\}$ for numerical integration. we have

$$\begin{aligned}\sum_{j=1}^{n} \widehat{\alpha}_j \bar{\phi}_j(x_i, y_i, z_i) = \Psi\Bigg(x_i, y_i, z_i, g(x_i, y_i, z_i) + \lambda \sum_{h=1}^{m_1} \sum_{p=1}^{m_1} \sum_{k=1}^{m_1} w_h w_k w_p \\ \bar{K}(x_i, y_i, z_i, \mu(x_i, \delta_h), \theta_p, \nu_k) \sum_{j=1}^{n} \widehat{\alpha}_j \bar{\phi}_j(\mu(x_i, \delta_h), \theta_p, \nu_k)\Bigg),\end{aligned} \tag{3.4.10}$$

the unknowns $\widehat{\alpha}_j$ can be found by solving the nonlinear system of algebraic equations with any nonlinear solver as the fsolve command of Matlab. So the values of $s(x, y, z)$ can be approximated by

$$\bar{s}(x, y, z) = \sum_{j=1}^{n} \widehat{\alpha}_j \bar{\phi}_j(x, y, z).$$

The approximation solution $\bar{u}_n(x, y, z)$ of $u_n(x, y, z)$ is given by

$$\bar{u}(x, y, z) = g(x, y, z) + \lambda \sum_{h=1}^{m_1} \sum_{p=1}^{m_1} \sum_{k=1}^{m_1} w_h w_k w_p \bar{K}\big(x, y, z, \mu(x, \delta_h), \theta_p, \nu_k\big) \bar{s}((\mu(x, \delta_h), \theta_p, \nu_k\big). \tag{3.4.11}$$

3.5 Error estimates

In this section, we present the rate of convergence and the error estimate of the proposed method. This analysis is based on those results given in [8, 30, 121, 125].
Let $P_n : C(D) \to G_n$ be the the collocation operator given by

$$P_n u(x, y, z) = \sum_{i=1}^{n} \alpha_i \bar{\phi}_i(x, y, z), \quad (x, y, z) \in D,$$

where $G_n = span\{\bar{\phi}_1, \bar{\phi}_2, ..., \bar{\phi}_n\}$. Let Γ be the operator $\Gamma : X \to X$ defined by

$$\Gamma(\bar{\phi})(x, y, z) = \Psi(x, y, z, \bar{\phi}(x, y, z)),$$

So we can write (3.4.6) in abstract form as

$$s(x, y, z) = \Gamma(\mathcal{K}s + g)(x, y, z).$$

where

$$(\mathcal{K}s)(x, y, z) = \int_0^x \int_0^1 \int_0^1 K(x, y, z, \mu, \theta, \nu)s(\mu, \theta, \nu)\, \mathrm{d}\nu d\theta d\mu$$

Let

$$\Upsilon(u)(x, y, z) = \Gamma(\mathcal{K}u + g)(x, y, z), \tag{3.5.1}$$

then (3.5.1) can be written as

$$s(x, y, z) = \Upsilon(s)(x, y, z).$$

We define the approximate solution for (3.4.8) by

$$s_n(x, y, z) = P_n\Gamma(\mathcal{K}s_n + g)(x, y, z),$$

The discrete collocation method for (3.4.10) is written as

$$\bar{s}_n(x, y, z) = P_n\Gamma(\mathcal{K}_n\bar{s}_n + g)(x, y, z). \tag{3.5.2}$$

Let $\{\Upsilon_n, \bar{\Upsilon}_n\}$ be the operators defined respectively by

$$\Upsilon_n(u)(x, y, z) = P_n\Gamma(\mathcal{K}u + g)(x, y, z),$$
$$\bar{\Upsilon}_n(u)(x, y, z) = P_n\Gamma(\mathcal{K}_nu + g)(x, y, z).$$

So (3.4.6) (3.4.8), (3.4.10) can be written as

$$s = \Upsilon s, \qquad s_n = \Upsilon_n s_n, \qquad \bar{s}_n = \bar{\Upsilon}_n\bar{s}_n.$$

We present the following theorems that are used to obtain the error estimate of the method.

Theorem 3.5.1. *[121] let $s_0 \in C(D)$ be an isolated solution of*

$$s = \Psi(x, y, z, \mathcal{K}s + g).$$

Suppose that the operator Υ is Frechet differentiable in some neighborhood of the point s_0 and assume that 1 is not an eigenvalue of the linear operator $\Upsilon^{'}(s_0)$.Then for sufficiently large n, the operator $(I - \bar{\Upsilon}^{'}_{\mathrm{n}}(s_0))^{-1}$ is invertible and there exists constant $L > 0$ independent of n such that $\|(I - \bar{\Upsilon}^{'}_{\mathrm{n}}(s_0))^{-1}\| \leq L$. The approximate solution $\bar{s}_n$ of (3.5.2) is unique in $B(s_0, \delta) = \{s : \|s - s_0\| \leq \delta\}$ for some $\delta > 0$. Moreover, there exists a constant $0 < q < 1$ independent of n *such that*

$$\frac{\alpha_{\mathrm{n}}}{1 + q} \leq \|s_0 - \bar{s}_{\mathrm{n}}\| \leq \frac{\alpha_{\mathrm{n}}}{1 - q},$$

where

$$\alpha_{\mathrm{n}} = \|(I - \bar{\Upsilon}^{'}_{\mathrm{n}}(s_0))^{-1}(\bar{\Upsilon}_{\mathrm{n}}(s_0) - \Upsilon(s_0))\|.$$

Theorem 3.5.2. *Let $\mathcal{K}_n$ be bounded operator on C(D) to C(D) and $\|\mathcal{K}_n\| \leq C_2$, u_0 is an isolated solution of the equation $u = \mathcal{K}s + g$ corresponding to s_0 and $\bar{u}_n$ is the MLS collocation approximation of u_0, then we have*

$$\|\bar{u}_n - u_0\| \leq \left(\frac{C_1C_2L\delta}{1 - q} + 1\right)\|\mathcal{K}_n - \mathcal{K}\|\|s_0\| + \frac{L}{1 - q}(1 + m)Ch^{q+1}_{X,D}|s_0|_{C^{q+1}}.$$

Proof. Using Theorem 3.5.1, we can write

$$\begin{aligned}
\|\bar{u}_n - u_0\| = \|\mathcal{K}_n\bar{s}_n - \mathcal{K}s_0\| &\leq \|\mathcal{K}_n(\bar{s}_n - s_0)\| + \|(\mathcal{K}_n - \mathcal{K})s_0\| \\
&\leq C_2\|\bar{s}_n - s_0\| + \|(\mathcal{K}_n - \mathcal{K})s_0\| \\
&\leq \frac{C_2\alpha_n}{1 - q} + \|(\mathcal{K}_n - \mathcal{K})s_0\|. \\
&\leq \frac{C_2}{1 - q}\|(I - \bar{\Upsilon}^{'}_n(s_0))^{-1}(\bar{\Upsilon}_n(s_0) - \Upsilon(s_0))\| + \|(\mathcal{K}_n - \mathcal{K})s_0\| \\
&\leq \frac{C_2L}{1 - q}\|(\bar{\Upsilon}_n(s_0) - \Upsilon(s_0))\| + \|(\mathcal{K}_n - \mathcal{K})\|\|s_0\| \qquad (3.5.3)
\end{aligned}$$

Here, we show that $\|\bar{\Upsilon}_n(s_0) - \Upsilon(s_0)\| \longrightarrow 0$ as $n \longrightarrow \infty$.

Using lipschitz condition on Ψ and let consider the following decomposition

$$\begin{aligned}\|\bar{\Upsilon}_n(s_0) - \Upsilon(s_0)\| &= \|P_n\Gamma(\mathcal{K}_n s_0 + g) - \Gamma(\mathcal{K}s_0 + g)\| \\ &\leq \|P_n\Gamma(\mathcal{K}_n s_0 + g) - P_n\Gamma(\mathcal{K}s_0 + g)\| + \|P_n\Gamma(\mathcal{K}s_0 + g) - \Gamma(\mathcal{K}s_0 + g)\| \\ &\leq \|P_n\Big(\Gamma(\mathcal{K}_n s_0 + g) - \Gamma(\mathcal{K}s_0 + g)\Big)\| + \|(P_n - I)\Gamma(\mathcal{K}s_0 + g)\| \\ &\leq \|P_n\|\|(\Gamma(\mathcal{K}_n s_0 + g) - \Gamma(\mathcal{K}s_0 + g))\| + \|(P_n - I)s_0\| \\ &\leq \|P_n\|\|\Psi(x, y, z, \mathcal{K}_n s_0 + g) - \Psi(x, y, z, \mathcal{K}s_0 + g)\| + \|(P_n - I)s_0\| \\ &\leq C_1\|P_n\|\|(\mathcal{K}_n - \mathcal{K})s_0\| + \|P_n s_0 - s_0\| \longrightarrow 0 \quad as\, n \longrightarrow \infty. \end{aligned} \tag{3.5.4}$$

Using Theorem 1.1.2 and combining Eq. (3.5.3) and (3.5.4), we obtain

$$\begin{aligned}\|\bar{u}_n - u_0\| &\leq \frac{C_1C_2L}{1-q}\|P_n\|\|\mathcal{K}_n - \mathcal{K}\|\|s_0\| + \frac{C_2L}{1-q}\|P_n s_0 - s_0\| + \|(\mathcal{K}_n - \mathcal{K})\|\|s_0\| \\ &\leq \left(\frac{C_1C_2L\delta}{1-q} + 1\right)\|\mathcal{K}_n - \mathcal{K}\|\|s_0\| + \frac{C_2L}{1-q}(1+\delta)Ch_{X,D}^{q+1}|s_0|_{C^{q+1}}. \end{aligned} \tag{3.5.5}$$

Using the fact that $h_{X,D}^{q+1} \to 0$ as $n \to \infty$ and the pointwise convergence of $\mathcal{K}_n$ to $\mathcal{K}$, we deduce from Eq. (3.5.5) that $\|\bar{u}_n - u_0\| \to 0$ as $n \to \infty$. □

The error of the proposed approach depends on the error of the quadrature formula and the MMLS approximation error. So, for sufficiently large integration nodes the error of the MMLS is dominated over the error of the numerical integration method and the rate of convergence of the proposed method is of order $O(h_{X,D}^{q+1})$. In the rest of paper, we put $h_{X,D}^{q+1} \equiv h$ for simplicity in notations.

3.6 Numerical results

In this section, some examples are provided to show the strength of the proposed method in approximating the solution of multi-dimensional linear and nonlinear Volterra-Fredholm integral equations. For the numerical implementation, we put $h = \dfrac{1}{\sqrt[3]{n} - 1}$ where h is the distance between two consecutive nodes, for MMLS method we put $\rho_i = 1.5 \times h$. When we use the MLS method, we put for linear case $\rho_i = 2 \times h$ and for quadratic case $\rho_i = 2.5 \times h$. Also we use the Gauss-Legendre quadrature rule for numerical integration. Moreover, for computing MMLS shape function, we choose $\nu_e = 10^{-9}$; with $e = 1, ..., 6$ as weights of additional coefficients for MMLS, it should be pointed out that, this value was selected experimentally. We applied the absolute errors $e(x, y, z)$ and the maximum errors $\|e\|_\infty$ between the exact solution and the present solution given by

$$e(x, y, z) = |u_{ex}(x, y, z) - \hat{u}(x, y, z)|, \qquad \|e\|_\infty = \max_{(x,y,z)\in D} |u_{ex}(x, y, z) - \hat{u}(x, y, z)|.$$

where $\hat{u}$ is the approximate solution of the exact solution u_{ex}. All calculations are done by Matlab. In order to prove the rate convergence of this technique, the values of ratio are calculated by

$$Ratio = \frac{ln(\|e\|_\infty)_n - ln(\|e\|_\infty)_{n-1}}{ln(h)_n - ln(h)_{n-1}}$$

Example 11. Consider the three-dimensional Fredholm integral equation [28]

$$u(x, y, z) - \frac{1}{2}\int_0^1\int_0^1\int_0^1 xyzu(\nu, \theta, \mu)\,\mathrm{d}\nu\,\mathrm{d}\theta\mathrm{d}\mu = xyz\exp(-x^2 - y^2 - z^2) - \frac{1}{16}\Big(1 - \frac{1}{\exp(1)}\Big)^3 xyz. \tag{3.6.1}$$

The exact solution is $u_{ex}(x, y, z) = xyz\exp(-x^2 - y^2 - z^2)$. The integral equation is solved on the cub domain $D = [0, 1] \times [0, 1] \times [0, 1]$ depicted in Figure 3.3. Table 3.1 presented the absolute errors at some particular points for $m_1 = 2$, $n = 15$ and $n = 27$. MMLS approximation results are shown in Table 3.2

Table 3.1: Numerical results for Example 11 with $m_1 = 2$

x	y	z	$e(x,y,z)$ of MMLS method	
			n=15	n=27
0.1	0.1	0.1	3.27×10^{-2}	8.17×10^{-4}
0.01	0.1	0.1	2.17×10^{-2}	1.02×10^{-4}
0.01	0.01	0.1	8.16×10^{-3}	1.05×10^{-5}
0.01	0.01	0.01	8.60×10^{-3}	1.09×10^{-6}
0.001	0.01	0.01	7.13×10^{-3}	1.33×10^{-7}
0.001	0.001	0.01	5.66×10^{-3}	1.28×10^{-8}
0.001	0.001	0.001	5.67×10^{-3}	9.39×10^{-10}

Table 3.2: Maximum error and rate of convergence for Example 11 with different values of n

n	h	MMLS method			RBF(IQ) method [28]
		$\|e\|_\infty$	Ratio	CPU time	$\|e\|_\infty$
8	1.00	1.35×10^{-2}	-	0.13	-
27	0.50	2.53×10^{-3}	2.41	0.72	$8.11\ \times10^{-3}$
64	0.33	8.58×10^{-4}	2.66	3.36	$1.63\ \times10^{-3}$
125	0.25	6.04×10^{-5}	9.22	19.92	$5.46\ \times10^{-4}$

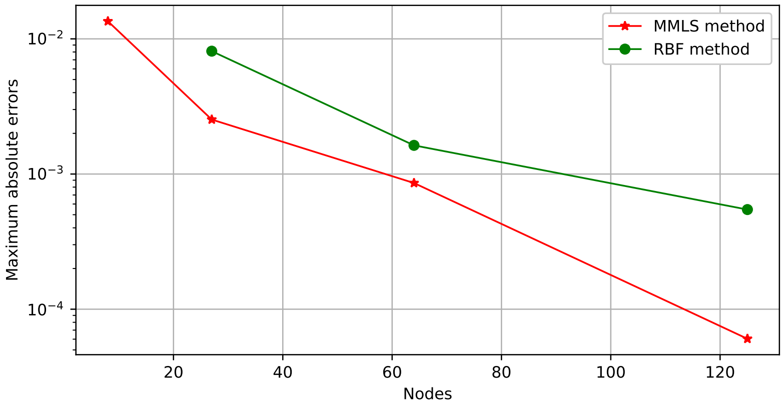


Figure 3.2: A comparison between MMLS and RBF(IQ) methods for Example 11

in terms of $\|e\|_\infty$ at various values of n and $m_1 = 10$. Figure 3.2 presented a comparison between MMLS and RBF(IQ) [28] methods, The results show that better accuracy is obtained with the present method. By increasing the number of nodes the rate errors increase and the maximum errors decrease which confirm that the new approach converge more quickly to the exact solution.

Example 12. Consider the following three-dimensional linear Volterra-Fredholm integral equation defined as [91]

$$u(x,y,z) + \frac{1}{20}\int_0^x\int_0^y\int_0^z zu(\mu,\theta,\nu)\,\mathrm{d}\nu\,\mathrm{d}\theta\mathrm{d}\mu - \frac{1}{10}\int_0^1\int_0^1\int_0^1 (x+\mu)u(\mu,\theta,\nu)\,\mathrm{d}\nu\,\mathrm{d}\theta\mathrm{d}\mu = g(x,y,z),$$

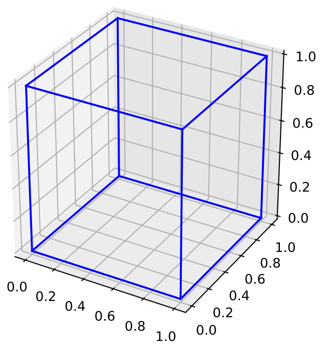
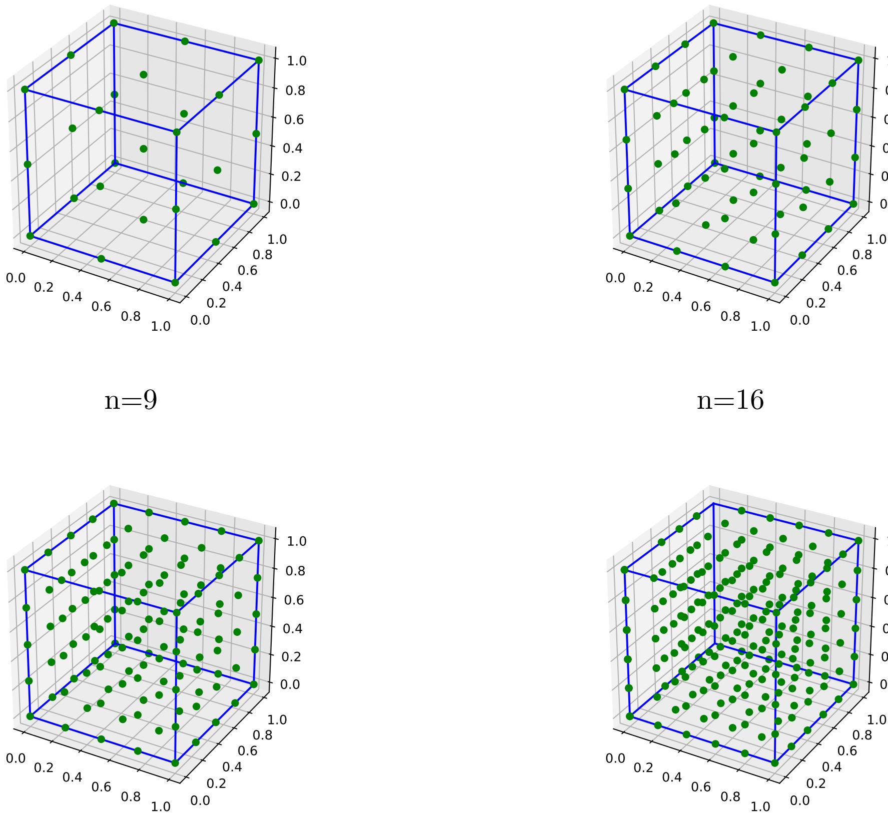


Figure 3.3: The consideration domain D for Examples 11,12,13,14,15



n=9

n=16

n=125

n=216

Figure 3.4: Regular nodes distribution for Examples 11,12,13,14,15

where

$$
\begin{aligned}
g(x,y,z) = sin(y+z) - \frac{1}{10}\Big((2x+1)sin(1) - (x+1)sin(2) - 1 - cos(2) + 2cos(1)\Big) \\
+ \frac{xz}{20}\Big(sin(y) + sin(z) - sin(y+z)\Big).
\end{aligned}
$$

The exact solution is $u_{ex}(x,y,z) = sin(y+z)$. The absolute errors of MMLS method are displayed

Table 3.3: Numerical results for Example 12 with different values of n and m_1

x	y	z	$e(x,y,z)$ of MMLS method			
			n=8, $m_1 = 2$	n=8, $m_1 = 10$	n=64, $m_1 = 2$	n=64, $m_1 = 10$
0.1	0.1	0.1	4.68×10^{-2}	4.70×10^{-2}	5.52×10^{-3}	6.09×10^{-3}
0.01	0.1	0.1	4.48×10^{-2}	4.49×10^{-2}	5.55×10^{-2}	6.04×10^{-3}
0.01	0.01	0.1	2.56×10^{-2}	2.57×10^{-2}	2.07×10^{-3}	6.04×10^{-3}
0.01	0.01	0.01	1.07×10^{-2}	1.08×10^{-2}	3.47×10^{-4}	8.41×10^{-4}
0.001	0.01	0.01	1.04×10^{-2}	1.06×10^{-2}	3.47×10^{-4}	8.33×10^{-4}
0.001	0.001	0.01	9.05×10^{-3}	9.15×10^{-3}	1.53×10^{-4}	6.39×10^{-4}
0.001	0.001	0.001	7.59×10^{-3}	7.69×10^{-3}	1.58×10^{-5}	4.70×10^{-4}

Table 3.4: Maximum error and rate of convergence for Example 12 with different values of n and $m_1 = 2$

n	h	$\|e\|_\infty$ of MLS method						$\|e\|_\infty$ of MMLS method		
		Linear basis	Ratio	CPU time	Quadratic basis	Ratio	CPU time	Quadratic basis	Ratio	CPU time
8	1.00	2.17×10^{-2}	-	0.10	2.14×10^{-2}	-	0.15	*2.10*$\times 10^{-2}$	-	0.14
27	0.50	2.04×10^{-2}	0.08	0.65	1.02×10^{-2}	1.06	0.99	*7.49*$\times 10^{-3}$	1.48	0.92
64	0.33	1.75×10^{-2}	0.37	3.11	4.51×10^{-3}	2.01	6.18	*4.80*$\times 10^{-4}$	6.77	4.66
125	0.25	1.03×10^{-2}	1.84	17.73	2.03×10^{-3}	2.77	32.54	*6.31*$\times 10^{-5}$	7.05	24.49
216	0.20	6.69×10^{-3}	1.93	91.29	1.06×10^{-3}	3.56	177.03	*2.98*$\times 10^{-5}$	3.36	160.29

Table 3.5: Numerical results for Example 12 using MMLS method for different values of ν and $n = 125$

ν	10^{-5}	10^{-6}	10^{-8}	10^{-9}	10^{-10}
$\|e\|_\infty$	8.36×10^{-4}	1.42×10^{-4}	6.34×10^{-5}	6.31×10^{-5}	6.31×10^{-5}

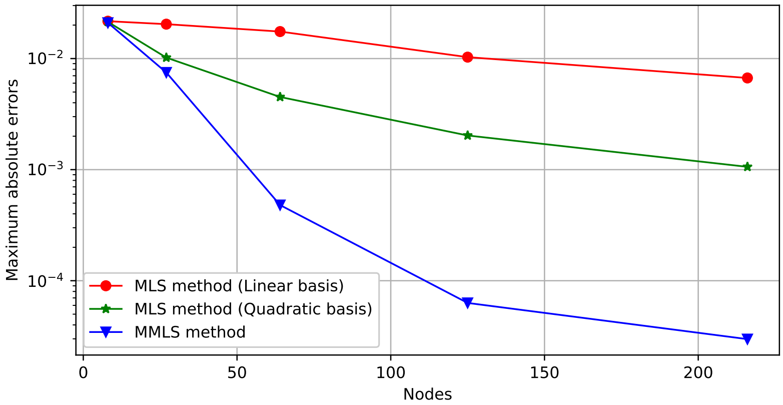


Figure 3.5: A comparison between MLS and MMLS methods for Example 12

in Table 3.3. The maximum errors using MLS and MMLS approximations are shown in Table 3.4. The results, introduced in Table 3.5, show the maximum errors using different values of ν when $n = 125$, we can see that MMLS approximations give better accuracy when we put ν as a small positive number. These tables show that the errors decrease as n increases which confirm the convergence of the method. A graphical comparison is shown in Figure 3.5 between MMLS and MLS methods. As $n \to \infty$, the ratio of MLS error remains approximately constant for the linear case $O(h^2)$ and for the quadratic case $O(h^3)$.

For large n, the rate of MMLS error seems to be a big value which confirms that the new approach is very fast in comparison with classical MLS method. Apparently, the MMLS method provides accurate numerical solutions for mixed integral equations.

Example 13. In this example we consider the following three-dimensional linear Volterra-Fredholm integral equation [1]

$$u(x,y,z) - \frac{1}{2}\int_0^x \int_0^1 \int_0^1 (yz + \mu\nu)u(\mu,\theta,\nu)\,\mathrm{d}\mu\,\mathrm{d}\theta\mathrm{d}\nu = g(x,y,z),$$

where $\quad g(x,y,z) = x^2y^2sin(z) - \dfrac{1}{72}\Big(-4yzcos(x) + 3sin(x) - 3xcos(x) + 4yz\Big).$

Table 3.6: Numerical results for Example 13 with different values of n and $m_1 = 2$

x	y	z	$e(x,y,z)$ of MMLS method		
			n=8	**n=27**	**n=64**
0.1	0.1	0.1	$1.29\ \times 10^{-3}$	$2.24\ \times 10^{-4}$	$9.08\ \times 10^{-5}$
0.01	0.1	0.1	$1.29\ \times 10^{-3}$	$1.61\ \times 10^{-5}$	$5.58\ \times 10^{-6}$
0.01	0.01	0.1	$1.30\ \times 10^{-4}$	$1.47\ \times 10^{-5}$	$1.96\ \times 10^{-7}$
0.01	0.01	0.01	$1.30\ \times 10^{-4}$	$1.44\ \times 10^{-5}$	$1.12\ \times 10^{-7}$
0.001	0.01	0.01	$1.30\ \times 10^{-5}$	$1.45\ \times 10^{-6}$	$1.54\ \times 10^{-8}$
0.001	0.001	0.01	$1.29\ \times 10^{-5}$	$1.44\ \times 10^{-6}$	$1.03\ \times 10^{-8}$
0.001	0.001	0.001	$1.29\ \times 10^{-5}$	$1.43\ \times 10^{-6}$	$9.57\ \times 10^{-9}$

Table 3.7: Maximum error and rate of convergence for Example 13 with different values of n and $m_1 = 2$

n	h	$\Vert e\Vert_\infty$ of MLS method						$\Vert e\Vert_\infty$ of MMLS method		
		Linear basis	Ratio	CPU time	Quadratic basis	Ratio	CPU time	Quadratic basis	Ratio	CPU time
27	0.50	$4.45\ \times 10^{-2}$	-	0.48	$5.39\ \times 10^{-2}$	-	0.80	$1.37\ \times 10^{-2}$	-	0.67
64	0.33	$3.08\ \times 10^{-2}$	0.90	2.31	$1.35\ \times 10^{-2}$	3.41	4.38	$1.07\ \times 10^{-3}$	6.28	3.28
125	0.25	$1.06\ \times 10^{-2}$	3.70	12.78	$6.49\ \times 10^{-3}$	2.54	28.34	$1.46\ \times 10^{-4}$	6.92	17.07
216	0.20	$7.00\ \times 10^{-3}$	1.85	50.01	$3.23\ \times 10^{-3}$	3.12	87.03	$4.31\ \times 10^{-5}$	5.46	60.25

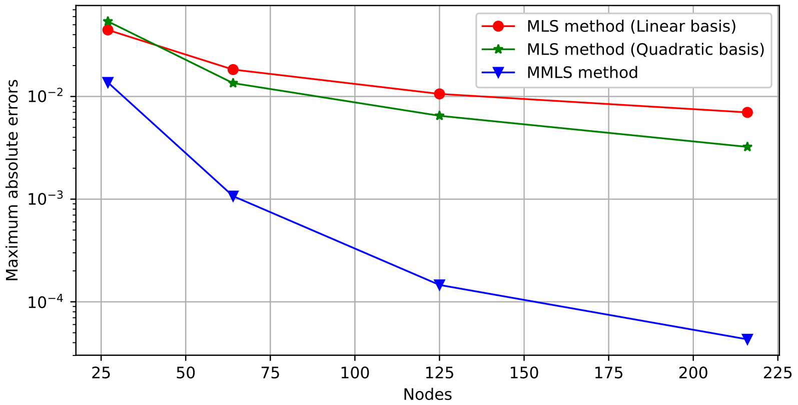


Figure 3.6: A comparison between MLS and MMLS methods for Example 13

The exact solution is $\quad u_{ex}(x,y,z) = x^2y^2sin(z)$. The distribution of nodes is depicted in Figure 3.3. Table 3.6 illustrates the absolute errors at some particular points. Table 3.7 shows $\Vert e\Vert_\infty$ at the different number of the nodes that are regularly employed in a cub domain. A graphical comparison is shown in Figure 3.6 between MMLS and MLS methods. As $n \to \infty$, the ratio of MLS error remains approximately constant for the linear case $O(h^2)$ and for the quadratic case $O(h^3)$. For large n, the rate of MMLS error seems to be a big value which confirms the high accuracy of this method. In these tables, we see that the results gradually converge to the exact values along with the increase of the nodes.

Example 14. Consider the following three-dimensional nonlinear mixed Volterra-Fredholm integral equation [83]

$$u(x,y,z) - \int_0^x \int_0^1 \int_0^1 x^2 yz\mu\theta sin(\nu) u(\nu,\theta,\mu)^3 \,\mathrm{d}\mu\,\mathrm{d}\theta\mathrm{d}\nu = \frac{yz}{100}\Big(x^2 cos(x)^4 - x^2 + 100cos(x)\Big).$$

Table 3.8: Numerical results for Example 14 with $m_1 = 2$

x	y	z	$e(x,y,z)$ of MMLS method
			$n = 8$
0.1	0.1	0.1	4.13 $\times 10^{-5}$
0.01	0.1	0.1	1.83 $\times 10^{-5}$
0.01	0.01	0.1	4.26 $\times 10^{-6}$
0.01	0.01	0.01	5.88 $\times 10^{-7}$
0.001	0.01	0.01	1.86 $\times 10^{-7}$
0.001	0.001	0.01	2.89 $\times 10^{-7}$
0.001	0.001	0.001	1.55 $\times 10^{-8}$

Table 3.9: Maximum error and rate of convergence for Example 14 using different values of n and $m_1 = 2$

n	h	MMLS method		
		$\|e\|_\infty$	Ratio	CPU time
8	1.00	1.41 $\times 10^{-2}$	-	1.20
27	0.50	5.15 $\times 10^{-3}$	1.45	37.56
64	0.33	3.09 $\times 10^{-4}$	6.93	141.07

Table 3.10: Maximum error for Example 14 using MMLS method with different values of m_1 and $n = 27$

m_1	2	3	5	7	10
$\|e\|_\infty$	5.15×10^{-3}	3.08×10^{-3}	3.56×10^{-3}	2.31×10^{-3}	2.75×10^{-3}

The exact solution for this equation is $u_{ex}(x,y,z) = zycos(x)$. The absolute errors with $m_1 = 2$ for this example are presented in Table 3.8. The maximum errors are shown in Table 3.9 for different values of n. As $n \longrightarrow \infty$ the rate error increases which confirms that the new approach converges more quickly to the exact solution. As we expected from Table 3.10, using bigger values of m_1 has no significant effect on maximum errors which confirm that the error of MMLS approximation is dominated over the error of integration.

Example 15. As the final example, we consider the following three-dimensional nonlinear Volterra-Fredholm integral equation [91]

$$u(x,y,z) - \frac{1}{2}\int_0^x \int_0^1 \int_0^1 x^2\theta(yz + \mu\nu) u(\nu,\theta,\mu)^3 \,\mathrm{d}\mu\,\mathrm{d}\theta\mathrm{d}\nu = g(x,y,z).$$

The exact solution for this equation is $u_{ex}(x, y, z) = zysin(x)$. The absolute errors of MMLS approximation and the method based on three-dimensional block-pulse functions approximation of [91] are presented in Table 3.11. Table 3.12 displayed the maximum error for different values of n. It is observed that the results of MMLS approximation converge more quickly to the true solution along with the increase of the nodes. As $n \longrightarrow \infty$ the rate error increases which confirms that the new approach converge more quickly to the exact solution. We can summarize that MMLS method is simple and efficient for solving high-dimensional nonlinear mixte integral equations.

Table 3.11: Numerical results for Example 15 with $m_1 = 2$ and $n = 27$

$x = y = z = 2^{-l}$	3D-BFs method [91]	MMLS method
	$e(x, y, z)$	$e(x, y, z)$
$l = 1$	$5{,}32\times10^{-4}$	2.64×10^{-6}
$l = 2$	$1{,}59\times10^{-4}$	1.78×10^{-5}
$l = 3$	$2{,}63\times10^{-3}$	5.43×10^{-5}
$l = 4$	$4{,}34\times10^{-3}$	6.02×10^{-5}
$l = 5$	$4{,}55\times10^{-3}$	5.11×10^{-5}
$l = 6$	$4{,}58\times10^{-3}$	2.38×10^{-5}

Table 3.12: Maximum error and rate of convergence for Example 15 with different values of n and $m_1 = 2$

n	h	MMLS method		
		$\|e\|_\infty$	Ratio	CPU time
8	1	$3.34\ \times10^{-2}$	-	1.31
27	0.5	$7.42\ \times10^{-3}$	2.17	38.95
64	0.33	$2.59\ \times10^{-3}$	2.59	153.23
125	0.25	$5.81\ \times10^{-5}$	13.19	765.02

3.7 Conclusion

In this chapter, we extended the modified moving least squares (MMLS) method for solving the three-dimensional linear and nonlinear Volterra-Fredholm integral equations of the second kind. An important benefit of the proposed method is the ability to provide an approximation for cases when classical MLS with quadratic base functions fails due to a singular moment matrix. Additionally, through the comparison with exact solutions we show that the MMLS method have good reliability and efficiency and it can be extended without difficulties to the three-dimensional problems because of the simple adaption of MMLS method for the 3-D space.

Part II

A meshless methods based on moving least squares and regularised moving least squares approaches for solving stochastic integral equations and fractional stochastic differential equations

Chapter 4

Regularized moving least squares scheme for two-dimensional stochastic Fredholm integral equations

In this chapter, a new adaptive meshless scheme based on the regularized moving least squares approximation(RMLS) combined with It$\hat{o}$ approximation is employed to solve two-dimensional stochastic integral equations. This approach is proposed for handling a singular moment matrix in the context of meshfree methods based on moving least squares (MLS) approximation. A valuable advantage of applied this technique is that the results converge more quickly to the exact solution by using a small support domain, and it is more flexible because it allows an easy adaptation of the nodal density. The computational complexity is presented to measure the usage time of the proposed approach. The convergence rate of the new method is provided. The numerical test problems are presented and compared with the results obtained by other meshless methods to verify the efficiency and accuracy of the proposed scheme.

The results obtained in this chapter are presented in the research paper [32] in collaboration with Rachid El Jid and Abdelkarim Hajjaj.

4.1 Introduction

Non deterministic phenomena appear increasingly in many fields such as engineering, physical and biological sciences. These phenomena can be naturally modeled by stochastic integral equations which are usually dependent on time and a random factor. Since solving analytically stochastic integral equations is very difficult and not available in some cases, it is essential to give their numerical solutions. Recently, some numerical methods have been used to solve different types of one dimensional stochastic integral equations such as stochastic operational matrix [86], spline interpolation methods [109], Euler polynomial [93], generalized hat basis functions [50], Taylor series [63], Chebyshev wavelet [96]. For instance, the number of research articles on the numerical solution of 2D dimensional stochastic integral equations is very few; in [36] a new numerical method based on Haar wavelet is introduced for solving numerical solution of two-dimensional linear stochastic Volterra integral equation, [35] the block-pulse is employed to solve two-dimensional linear stochastic Volterra-Fredholm integral equation. Also, a numerical meshless scheme based on radial basis functions is developed to find numerical solutions of two-dimensional stochastic Fredholm integral equations [110]. In this work, an accurate numerical technique is used to solve two-dimensional stochastic integral equation

$$u(x,y) = g(x,y) + \int_c^d \int_a^b K_1(x,y,r,t)u(r,t)dtdr + \int_c^d \int_a^b K_2(x,y,r,t)s(r,t)dB(t)dB(r),\ (x,y) \in D, \tag{4.1.1}$$

where $D = [a,b] \times [c,d]$ is a rectangular domain, $u(x,y)$ is the unknown function to be determined and the functions K_1, K_2 and g are assumed to be given. In literature, the MLS method has been used to approximate many various types of integral equations [68, 6, 83]. However, the moment matrix of the traditional MLS method may be singular. In this context, some techniques have recently been suggested to avoid the singular matrix; Joldes et al [61] developed a modified moving least squares (MMLS) method for handling a singular moment matrix in the context of MLS method. Also, Wang et al [127] presented an adaptive orthogonal IIMLS (AO-IIMLS) approximation to avoid the singular matrix by using the weighted orthogonal basis function.

In this chapter, the regularized MLS approach is used to solve stochastic integral equation (4.1.1) with nonsingular matrix. The advantage of this method is using the quadratic base functions with a small support domain by adding only some constraints and also this modification reduces the cost of calculating the shape functions. The moment matrix still nonsingular and gives better accuracy more than MLS approximation. The computational complexity of the method is studied to measure the usage time of the proposed algorithm which is confirmed by numerical tests given. The first use of RMLS method was proposed in [126] for smoothing and approximating scattered.

The chapter is organized as follows: Section 4.2 is devoted to present the regularized moving least-square approximation. In Section 4.3, we introduce the description of numerical scheme based on regularised MLS approximation. In Section 4.4, we provide the error estimate of the method. Tests problems are given in Section 4.5. we ended the chapter in Section 4.6.

4.2 Regularized moving least-square approximations

Given data coordinates of the function $u(x, y)$ at certain data nodes $X = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\} \subset D$, the MLS approximate $\hat{u}(x, y)$ of $u(x, y)$, can be given as

$$\hat{u}(x,y) = \sum_{k=1}^{m} a_k(x,y)p_k(x,y) = P^T(x,y)a(x,y), \tag{4.2.1}$$

where $\{p_k(x, y)\}_{k=1}^{m}$ is a basis function of P_q which is the linear space of polynomials of total degree less than or equal to q and $a_k(x, y)$ are the unknown coefficients.

The MLS approximation is obtained using the solution of

$$J(a) = \sum_{k=1}^{n} \Big| a^T(x_k, y_k)p(x,y) - u_k \Big|^2 w_k(x,y) = [Pa(x,y) - u]^T W[Pa(x,y) - u], \tag{4.2.2}$$

where

$$\begin{array}{rcl} P & = & [p^T(x_1,y_1), p^T(x_2,y_2), ..., p^T(x_n,y_n)]^T \\ W & = & diag(\omega_i(x,y)), \quad i = 1,2,...,n. \end{array}$$

A regularized technique is employed to modify MLS moment matrix by adding additional constraints to the functional (4.2.2). Then Eq. (4.2.2) can be reformulated as follows

$$J(a) = [Pa - u]^T W[Pa - u] + aHa, \tag{4.2.3}$$

where

$$H = diag(\varepsilon_i), \quad i = 1,2,...,m,$$

ε_i are the constants and positifs values, then by set $\dfrac{\partial J}{\partial a}$ we obtain

$$[P^TWP + H]a - P^TWu = 0. \tag{4.2.4}$$

We put

$$R = P^TWP + H, \tag{4.2.5}$$

Eq. (4.2.4) becomes as follows

$$Ra = P^TWu. \tag{4.2.6}$$

Remark 3. The moment matrix R is invertible.

Proof. Eq. (4.2.5) can be represented as follows

$$R = \bar{P}^T\bar{W}\bar{P}, \tag{4.2.7}$$

where

$$\bar{W} = \begin{bmatrix} W & 0 \\ 0 & R \end{bmatrix}, \quad \bar{P} = \begin{bmatrix} P \\ I_m \end{bmatrix}, \quad \textit{with} \qquad P = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n^2 & y_n^2 & x_ny_n \end{bmatrix}. \tag{4.2.8}$$

Eq. (4.2.7) can be reformulated as

$$R = \bar{P}^T N^T N \bar{P} = D^T D, \tag{4.2.9}$$

where

$$N = sqrt(\bar{W}), \qquad\qquad D = N\bar{P}.$$

As we knew that $\bar{W}$ is a diagonal positive matrix with non-zeros diagonal values, thus by utilizing the matrix rank properties, we obtain

$$rank(D) = rank(\bar{P}). \tag{4.2.10}$$

Utilizing Eq. (4.2.9) and Eq. (4.2.10), we concluded that

$$rank(R) = rank(D^T D) = rank(D) = rank(\bar{P}). \tag{4.2.11}$$

Eq. (4.2.11) proved that the matrix R is invertible if the matrix $\bar{P}$ have full rank (m=6). □

Now Eq. (4.2.6) can be written as

$$a = R^{-1} P^T W u, \tag{4.2.12}$$

substituting Eq. (4.2.12) into Eq. (4.2.1), then we obtain

$$\hat{u} = P^T R^{-1} P^T W u,$$

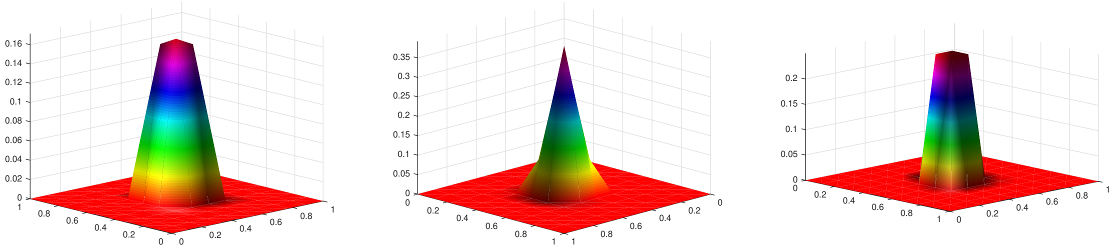
we put

$$\Psi = P^T R^{-1} P^T W.$$

$\Psi = \{\phi_1, \phi_2, ..., \phi_m\}$ are called the shape function of the RMLS approximation, the expressions P, W and H are continuous and the moment matrix R is invertible, it implies the shape functions of regularized MLS are continuous. If the weight functions $w(x, y)$ and the monomials bases P have compact support domains, then the shape functions have compact support domains.

Finally, the approximation of $u(x, y)$ can be represented by

$$\hat{u}(x, y) = \sum_{k=1}^{n} \phi_k(x, y) u_k = \Psi u. \tag{4.2.13}$$



RMLS shape function for $\varepsilon = 0.1$

RMLS shape function for $\varepsilon = 10^{-6}$

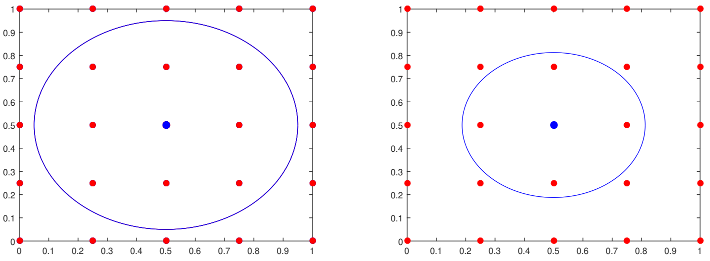
MMLS shape function [31] for $\varepsilon = 0.1$

Figure 4.1: RMLS and MMLS shape functions of the central node

RMLS and MMLS shape functions of the central node with 25 uniformly distributed nodes are illustrated in Fig. 4.1 on the square domain $[0, 1] \times [0, 1]$. We use quadratic bases with the small support domain $R = 0.3125$. In this case, the moment matrix of MLS approximation is singular but those of RMLS and MMLS approximations are still nonsingular. The shape functions influences when we decrease the value of the weight parameter ε, this small value allows the singular moment matrix to be little changed and the moment matrix becomes nonsingular. The effect of choosing ε as a small value will be also treated in the numerical part.

Computational complexity of the proposed method

Considering a 5 × 5 uniformly distributed nodes depicted in Fig. 4.2. Here, a comparison between the MLS and RMLS support domain of the central node is given. The quadratic basis is used in both methods. Generally, the condition for the MLS moment matrix to be invertible is that $n \geq m$. For this reason, we put the size of MLS circular support larger (R=0.45) and more than six nodes are included in the support domain. For RMLS approximation, we put the size of circular support smaller (R=0.3125) and only five nodes are including in the support domain. Therefore, the MLS shape function contains nine nonzero elements while the RMLS shape function contains only five nonzero elements, which means the computation time is normalized by the number of nodes included in the support domain. Then, it concludes that the cost of calculating shape functions for the MLS method is expensive compared to the RMLS method.



MLS circular support RMLS circular support

Figure 4.2: MLS and RMLS circular support of the central node

4.3 Numerical scheme

In this section, we use a meshfree approach based on regularized moving least-square method for solving stochastic integral equations (4.3.1). We approximate the unknown function $u(x, y)$ by $\hat{u}(x, y)$ which is defined in Eq. (4.2.13), then we obtain

$$\hat{u}(x,y) = g(x,y) + \int_c^d \int_a^b K_1(x,y,r,t)\hat{u}(r,t)dtdr + \int_c^d \int_a^b K_2(x,y,r,t)\hat{u}(r,t)dB(t)dB(r), \tag{4.3.1}$$

if we collocate equation at collocation nodes (x_i, y_i), then we have

$$\sum_{k=1}^{n} \left[\phi_k(x_i,y_i) + \int_c^d \int_a^b K_1(x_i,y_i,r,t)\phi_k(r,t)dtdr + \int_c^d \int_a^b K_2(x_i,y_i,r,t)\phi_k(r,t)dB(t)dB(r) \right] u_k$$
$$= g(x_i, y_i). \tag{4.3.2}$$

We use m_1 point quadrature formula with the quadrature points r_p, u_j and the quadrature weights w_p, w_j for numerical integration of the first integral. At first, we compute the integrals over $[-1, 1]$ by using the following linear transformations

$$r = \gamma(\tau) = \frac{b-a}{2}\tau + \frac{b+a}{2}, \quad t = \varepsilon(\eta) = \frac{d-c}{2}\eta + \frac{d+c}{2},$$

then we obtain

$$\int_c^d \int_a^b K_1(x_i,y_i,r,t)\phi_k(r,t)dtdr = \frac{(b-a)^2}{4}\int_{-1}^{1}\int_{-1}^{1} K_1(x_i,y_i,\gamma(\tau),\varepsilon(\eta))\phi_k(\gamma(\tau),\varepsilon(\eta)dtdr \tag{4.3.3}$$
$$= \frac{(b-a)^2}{4}\sum_{p=1}^{m_1}\sum_{j=1}^{m_1} K_1(x_i,y_i,\gamma(\tau_p),\varepsilon(\eta_j))\phi_k(\gamma(\tau_p),\varepsilon(\eta_j))w_p w_j.$$

The second integral can be approximated by using the Riemann sum introduced in Eq. (1.4.1) as

$$\int_c^d \int_a^b K_2(x_i, y_i, r, t)\phi_k(r,t)dB(t)dB(r) = \sum_{p=0}^{n-1}\sum_{j=0}^{n-1} K_2(x_i, y_i, r_p, t_j)\phi_k(r_p, t_j)$$
$$(B(t_{j+1}) - B(t_j))(B(r_{p+1}) - B(r_p)), \tag{4.3.4}$$

by substituting Eqs. (4.3.3)-(4.3.4) in Eq. (4.3.2), we have the following system

$$\sum_{k=1}^{n}\Bigg[\phi_k(x,y) + \frac{(b-a)^2}{4}\sum_{p=1}^{m_1}\sum_{j=1}^{m_1} K_1(x_i, y_i, \gamma(\tau_p), \varepsilon(\eta_j))\phi_k(\gamma(\tau_p), \varepsilon(\eta_j))w_p w_j \tag{4.3.5}$$
$$+\sum_{p=0}^{n-1}\sum_{j=0}^{n-1} K_2(x_i, y_i, r_p, t_j)\phi_k(t_j, r_p)(B(t_{j+1}) - B(t_j))(B(r_{p+1}) - B(r_p))\Bigg]\hat{u}_k = g(x,y),$$

where $\hat{u}_k$ are the approximations for u_k. After solving the linear system, we find the unknown u_k and then the values of $u(x,y)$ can be approximated by

$$\hat{u}(x,y) = \sum_{k=1}^{n}\phi_k(x,y)u_k, \quad (x,y) \in [a,b]\times[c,d].$$

4.4 Error estimates

Firstly, we remind some basic defintions which are useful to estimate the error of proposed method [39, 37].

Equation (4.3.1) can be reformulated as

$$(I - \mathcal{K}_1 - \mathcal{K}_2)u = f,$$

where

- $(\mathcal{K}_1 u)(x,y) = \displaystyle\int_c^d\int_a^b K_1(x,y,r,t)u(r,t)dtdr,$
- $(\mathcal{K}_2 u)(x,y) = \displaystyle\int_c^d\int_a^b K_2(x,y,r,t)u(r,t)dB(t)dB(r),$

Using Eq. (1.1.28), then Eq. (4.3.2) can be written as the abstract form

$$(I - P_n\mathcal{K}_1 - P_n\mathcal{K}_2)u_n = P_n f.$$

Let $\mathcal{K}_{1,n}, \mathcal{K}_{2,n}$ be the numerical integration operators defined respectively by

- $(\mathcal{K}_{1,n}s)(x,y) = \displaystyle\sum_{p=1}^{m_1}\sum_{j=1}^{m_1} K_1(x,y,r_p,t_j)u(r_p,t_j)w_p w_j,$
- $(\mathcal{K}_{2,n}s)(x,y) = \displaystyle\sum_{p=0}^{n-1}\sum_{j=0}^{n-1} K_2(x,y,r_p,t_j)u(r_p,t_j)(B(t_{j+1}) - B(t_j))(B(r_{p+1}) - B(r_p)).$

Now, Eq. (4.3.5) can be reformulated as the operator form

$$(I - P_n\mathcal{K}_{1,n} - P_n\mathcal{K}_{2,n})\hat{u}_n = P_n f, \tag{4.4.1}$$

we give the following iterated discrete collocation solution

$$\bar{u}_n = f + P_n\mathcal{K}_{1,n}\hat{u}_n + P_n K_{2,n}\hat{u}_n. \tag{4.4.2}$$

Adding P_n on both sides of Eq. (4.4.2) and using $P_n\hat{u}_n = \hat{u}_n$, we have

$$P_n\bar{u}_n = P_n f + P_n\mathcal{K}_{1,n}\hat{u}_n + P_n\mathcal{K}_{2,n}\hat{u}_n. \tag{4.4.3}$$

Now using Eq. (4.4.1) and Eq. (4.4.3), we obtain

$$P_n\bar{u}_n = \hat{u}_n. \tag{4.4.4}$$

Lemma 4.4.1. *[37] Let $\mathcal{K}_1, \mathcal{K}_2$ be bounded operators on $C(D)$ to $C(D)$, assume that $I - \mathcal{K}_1 - \mathcal{K}_2 : C(D) \to C(D)$, and*

$$\|\mathcal{K}_1 + \mathcal{K}_2 - P_n(\mathcal{K}_{1,n} + \mathcal{K}_{2,n})\| \longmapsto 0, \quad as \quad n \longmapsto \infty.$$

For all sufficiently large n, the operator $\big(I - P_n(K_{1,n} + K_{2,n})\big)^{-1}$ existe and is uniformly bounded, then there exists a natural number N such that

$$sup_{n \geq N} \|\big(I - P_n(\mathcal{K}_{1,n} + \mathcal{K}_{2,n})\big)^{-1}\| < C_0 < \infty.$$

Let u_0 be a unique solution of Eq. (4.3.1) and for the solution $\bar{s}_n$ we have

$$\|u_0 - \bar{u}_n\| \leq \|\big(I - P_n(\mathcal{K}_{1,n} + \mathcal{K}_{2,n})\big)^{-1}\| \|\mathcal{K}_1 u_0 - \mathcal{K}_{1,n} P_n u_0 + \mathcal{K}_2 u_0 - \mathcal{K}_{2,n} P_n u_0\|.$$

Theorem 4.4.1. *Let the assumptions of Lemma 4.4.1 are satisfied, let u_0 be a unique solution of the integral equation (4.3.1), the proposed method in the current paper has a unique solution, we have*

$$\|u_0 - \hat{u}_n\| \leq \Big(1 +_0 (C_1 + C_2)(1 + \delta)\Big) C h_{X,D}^{q+1} |u_0|_{q+1} + \delta C_0 \Big(\|\mathcal{K}_1 u_0 - \mathcal{K}_{1,n} u_0\| + \|\mathcal{K}_2 u_0 - \mathcal{K}_{2,n} u_0\|\Big), \quad (4.4.5)$$

where C_0, C_1, C_2 and η are constant numbers.

Proof. From Eq. (4.4.4), we see that

$$u_0 - \hat{u}_n = u_0 - P_n \overline{u}_n = u_0 - P_n u_0 + P_n(u_0 - \overline{u}_n).$$

Therefore

$$\|u_0 - \hat{u}_n\| \leq \|u_0 - P_n u_0\| + \delta \|u_0 - \overline{u}_n\|$$

Using Lemma 4.4.1, we find that

$$\begin{aligned} \|u_0 - \bar{u}_n\| &\leq \|u_0 - P_n u_0\| + \delta \|\big(I - P_n(\mathcal{K}_{1,n} + \mathcal{K}_{2,n})\big)^{-1}\| \|\mathcal{K}_1 u_0 - \mathcal{K}_{1,n} P_n u_0 + \mathcal{K}_2 u_0 - \mathcal{K}_{2,n} P_n u_0\| \quad (4.4.6) \\ &\leq \|u_0 - P_n u_0\| + \delta C_0 \|\mathcal{K}_1 u_0 - \mathcal{K}_{1,n} P_n u_0 + \mathcal{K}_2 u_0 - \mathcal{K}_{2,n} P_n u_0\|. \end{aligned}$$

It is clear that

$$\begin{aligned} \|\mathcal{K}_1 u_0 + \mathcal{K}_2 u_0 - \mathcal{K}_{1,n} P_n u_0 - \mathcal{K}_{2,n} P_n u_0\| &\leq \|\mathcal{K}_1 u_0 - \mathcal{K}_{1,n} u_0\| + \|\mathcal{K}_2 u_0 - \mathcal{K}_{2,n} u_0\| \quad (4.4.7) \\ &+ \|\mathcal{K}_{1,n} u_0 - \mathcal{K}_{1,n} P_n u_0\| + \|\mathcal{K}_{2,n} u_0 - \mathcal{K}_{2,n} P_n u_0\| \\ &\leq \|\mathcal{K}_1 u_0 - \mathcal{K}_{1,n} u_0\| + \|\mathcal{K}_2 u_0 - \mathcal{K}_{2,n} u_0\| \\ &+ \|\mathcal{K}_{1,n}(u_0 - P_n u_0)\| + \|\mathcal{K}_{2,n}(u_0 - P_n u_0)\|. \end{aligned}$$

By using the pointwise convergence of $\mathcal{K}_{1,n}$ and $\mathcal{K}_{2,n}$, we can assume that

$\mathcal{K}_{1,n} \leq C_1 < \infty$,

$\mathcal{K}_{2,n} \leq C_2 < \infty$.

Hence, it is concluded that

$$\|\mathcal{K}_1 u_0 + \mathcal{K}_2 u_0 - \mathcal{K}_{1,n} P_n u_0 - \mathcal{K}_{2,n} P_n u_0\| \leq \|\mathcal{K}_1 u_0 - \mathcal{K}_{1,n} u_0\| + \|\mathcal{K}_2 u_0 - \mathcal{K}_{2,n} u_0\| + (C_1 + C_2)\|u_0 - P_n u_0\|. \quad (4.4.8)$$

Combining Eq. (4.4.6) and Eq. (4.4.8), we obtain

$$\begin{aligned} \|u_0 - \hat{u}_n\| &\leq \|u_0 - P_n u_0\| + \delta C_0 \|\mathcal{K}_1 u_0 - \mathcal{K}_{1,n} u_0\| + \delta C_0 \|\mathcal{K}_2 u_0 - \mathcal{K}_{2,n} u_0\| \\ &+ \delta C_0 (C_1 + C_2) \|(s_0 - P_n u_0)\| \\ &\leq (1 + \delta C_0 (C_1 + C_2)) \|u_0 - P_n u_0\| + \delta C_0 \Big(\|\mathcal{K}_1 u_0 - \mathcal{K}_{1,n} u_0\| + \|\mathcal{K}_2 u_0 - \mathcal{K}_{2,n} u_0\|\Big). \end{aligned}$$

From Lemma 1.1.2, we get

$$\|u_0 - \hat{u}_n\| \leq \Big(1 + \delta C_0 (C_1 + C_2)\Big)(1 + \delta) C h_{X,D}^{q+1} |u_0|_{q+1} + \delta C_0 \Big(\|\mathcal{K}_1 u_0 - \mathcal{K}_{1,n} u_0\| + \|\mathcal{K}_2 u_0 - \mathcal{K}_{2,n} u_0\|\Big), \quad (4.4.9)$$

for every $u_0 \in C^{q+1}(D)$, using the fact that $h_{X,D}^{q+1} \to 0$ as $n \to \infty$ and the pointwise convergence of $\mathcal{K}_{1,n}$ and $\mathcal{K}_{2,n}$ to $\mathcal{K}_1$ and $\mathcal{K}_2$ respectively, we deduce from Eq. (4.4.9) that $\|u_0 - \hat{u}_n\| \to 0$ as $n \to \infty$. □

In the rest of chapter, we put $h_{X,D}^{q+1} \equiv h$ for simplicity in notations.

4.5 Numerical results

In this section, two dimensional stochastic integral equations are solved and compared to the computational efficiency of other meshless methods : MLS, MMLS, and RBF methods. Quadratic basis functions and spline weight functions are applied in illustrative tests. For computational details, we put $R = 2.5 \times h$ for classical MLS method and $R = 1.5 \times h$ for RMLS and MMLS methods. For computing shape function, we put $\varepsilon_e = 10^{-8}$, with $e = 1, ..., 6$ as weights of additional coefficients for the RMLS method, this value was selected experimentally. To show the accuracy and convergence of the proposed method, the absolute error $e(x, y)$ and maximum absolute error $\|e(x, y)\|_\infty$ are given as follows

$$\begin{aligned} e(x, y) &= |u_0(x, y) - \hat{u}_n(x, y)|, \\ \|e\|_\infty &= \max_{(x,y)\in D} |u_0(x, y) - \hat{u}_n(x, y)|, \end{aligned}$$

where $\hat{u}_n(x, y)$ is the approximate solution of $u_0(x, y)$. All codes are written in Matlab.

Example 16. We consider the following stochastic integral equation [110]

$$u(x, y) = g(x, y) + \int_0^1 \int_0^1 xyrtu(r, t)dtdr + \int_0^1 \int_0^1 (x + y)u(r, t)dB(t)dB(r),$$

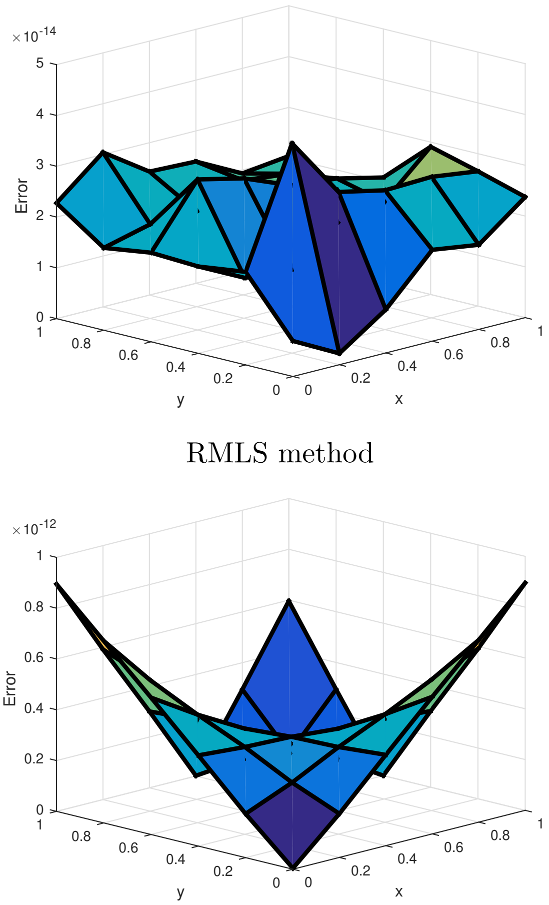
where

$$g(x, y) = 1 - \frac{1}{4}xy - B^2(1)(x + y),$$

and the exact solution is $u_0(x, y) = 1$.

Table 4.1: Maximum error for Example 16 using RMLS and MMLS methods

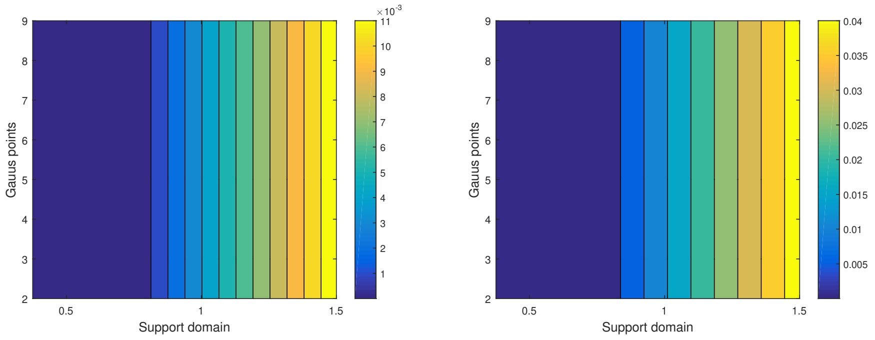
n	h	RMLS method	MMLS method	RBF method [110]
		$\|e\|_\infty$	$\|e\|_\infty$	$\|e\|_\infty$
4	1.00	1.19×10^{-2}	4.32×10^{-2}	1.58×10^{-1}
9	0.50	5.82×10^{-8}	5.80×10^{-7}	$\mathit{1.45} \times 10^{-2}$
25	0.25	3.57×10^{-13}	1.91×10^{-11}	3.46×10^{-4}
36	0.20	4.13×10^{-14}	8.95×10^{-13}	1.33×10^{-4}



RMLS method

MMLS method

Figure 4.3: Absolute error $e(x, y)$ of MMLS and RMLS methods ($n = 36$) for Example 16



RMLS method

MMLS method

Figure 4.4: Contour plots of maximum error for MMLS and RMLS methods for Example 16

The maximum error of the numerical results is shown in Table 4.1 for different values of n. As we expected, from Theorem 4.4.1, the results gradually converge to the exact values along with the increase of the nodes. The absolute errors for $n = 36$ are represented in Fig. 4.3. The comparison of obtained errors of RMLS, MLS and RBF methods are reported in Table 4.1. The 2-point Gauss-Legendre quadrature rule is used for numerical integration of the first integral. Contour plots given in Fig. 4.4 reveal that the RMLS method is more accurate and the efficiency of this method is concluded from using a small support domain.

Example 17. We consider the following stochastic integral equation [110]

$$u(x,y) = g(x,y) + \int_0^1 \int_0^1 \frac{xy}{t} u(r,t)dtdr + \int_0^1 \int_0^1 cos(xy)u(r,t)dB(t)dB(r),$$

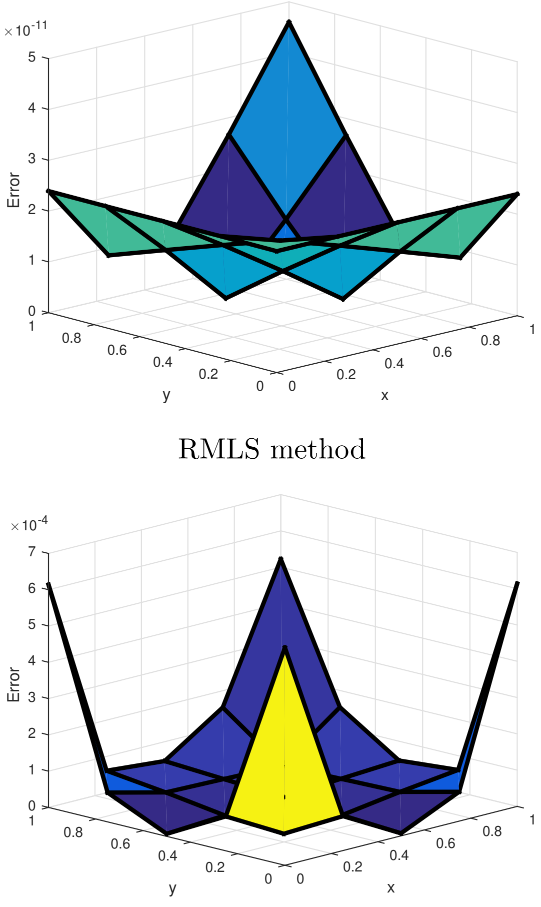
where

$$g(x,y) = \frac{xy}{2} - cos(xy)\left[B(1) - \int_0^1 B(t)dt\right]^2.$$

and the exact solution is $u_0(x,y) = xy$,

Table 4.2: Maximum error and CPU time for Example 17 using RMLS and MLS methods

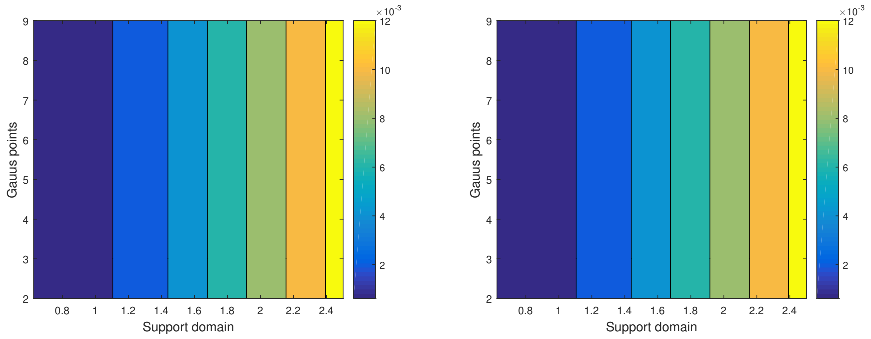
n	h	RMLS method		MLS method		RBF method [110]
		$\|e\|_\infty$	CPU time	$\|e\|_\infty$	CPU time	$\|e\|_\infty$
4	1.00	1.19 $\times 10^{-2}$	0.02	1.29 $\times 10^{-2}$	0.07	1.58 $\times 10^{-1}$
9	0.50	4.99 $\times 10^{-4}$	0.65	2.42 $\times 10^{-3}$	0.87	*1.45* $\times 10^{-2}$
25	0.25	4.60 $\times 10^{-11}$	63.42	6.14 $\times 10^{-4}$	112.48	1.33 $\times 10^{-4}$



RMLS method

MLS method

Figure 4.5: Absolute error $e(x,y)$ of MMLS and RMLS methods ($n = 36$) for Example 17



RMLS method

MMLS method

Figure 4.6: Contour plots of maximum error for MMLS and RMLS methods for Example 17

The numerical results in terms of $\|e\|_\infty$ are given in Table 4.2 at different numbers of n. Theorem 4.4.1 concludes that the results gradually converge to the exact values as the number of nodes increases. We employ the 2-point Gauss-Legendre quadrature formula for numerical integration of the first integral. Fig. 4.5 shows the absolute error for $n = 25$. Three meshless schemes are compared and the results of the RMLS method converge more quickly to the exact solution and give better accuracy than that of the classical MLS and RBF methods. CPU time presented confirms the computational cost of the moment matrix which is decreased compared with the MLS method. Contour plots given in Fig. 4.6 reveal that the RMLS method is more accurate and the efficiency of this method is concluded from using a small support domain.

Example 18. We consider the following stochastic integral equation

$$u(x,y) = g(x,y) + \int_0^1 \int_0^1 (xr + yt)u(r,t)dtdr + \int_0^1 \int_0^1 exp(-x-y-t-4)u(r,t)dB(t)dB(r),$$

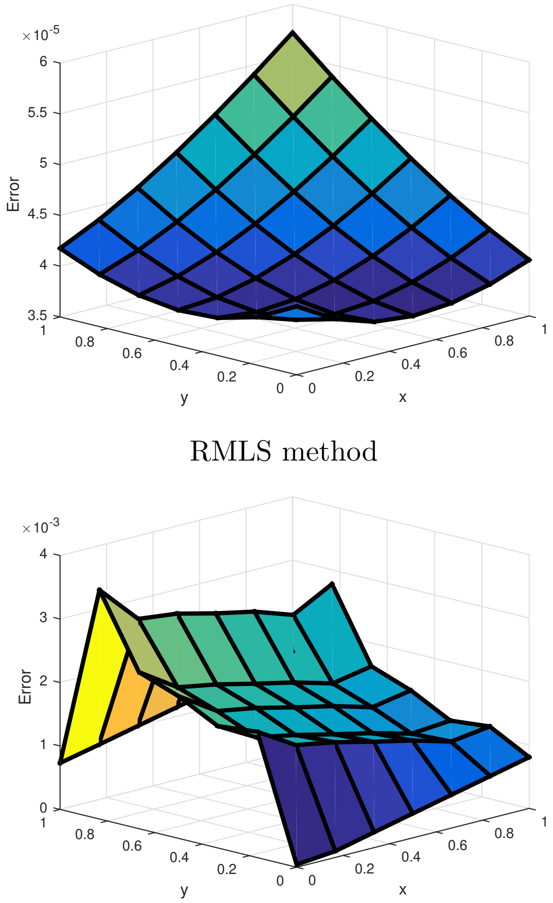
where

$$g(x,y) = \frac{1}{2}ex(y) + \frac{x}{4} - \frac{y}{12}\Big(3exp(1)+1\Big) - exp(-x-y-4)\Bigg(\frac{1}{2}B(1)^2 + \Big(B(1) - \int_0^1 B(s)ds\Big)\Big(exp(-1)B(1) - \int_0^1 exp(-s)B(s)ds\Big)\Bigg), \tag{4.5.1}$$

and the exact solution is $u_0(x,y) = \frac{1}{2}exp(y) + x$.

Table 4.3: Maximum error and CPU time for Example 18 using RMLS and MLS methods

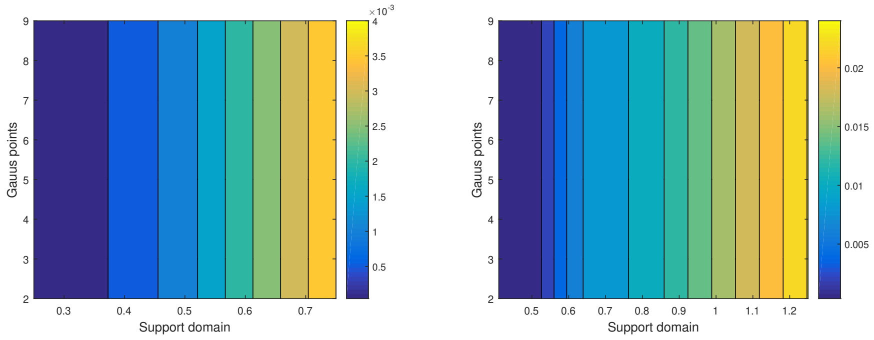
		RMLS method				MLS method	
		$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-7}$	$\varepsilon = 10^{-8}$			
n	h	$\|e\|_\infty$	$\|e\|_\infty$	$\|e\|_\infty$	CPU time	$\|e\|_\infty$	CPU time
9	0.50	7.30 $\times 10^{-3}$	4.50 $\times 10^{-3}$	4.00 $\times 10^{-3}$	0.78	2.41$\times 10^{-2}$	0.89
16	0.33	2.66 $\times 10^{-3}$	1.44 $\times 10^{-3}$	*1.27* $\times 10^{-3}$	7.00	1.11$\times 10^{-2}$	11.12
25	0.25	0.11 $\times 10^{-3}$	5.77 $\times 10^{-4}$	5.11 $\times 10^{-4}$	52.42	7.77$\times 10^{-3}$	132.82
36	0.20	9.69 $\times 10^{-4}$	2.29 $\times 10^{-4}$	1.21 $\times 10^{-4}$	209.82	4.98$\times 10^{-3}$	396.88
47	0.16	7.06 $\times 10^{-4}$	8.01 $\times 10^{-5}$	4.16 $\times 10^{-5}$	725.64	3.60$\times 10^{-3}$	989.79



RMLS method

MLS method

Figure 4.7: Absolute error $e(x, y)$ of MLS and RMLS methods $(n = 47)$ for Example 18



RMLS method

MMLS method

Figure 4.8: Contour plots for MMLS and RMLS methods for Example 18

Table 4.3 reported the maximum error for different values of ε compared with the classical MLS method. As we can see in Theorem 4.4.1, the errors decrease as $N \to \infty$. In this example, we put $m_1 = 5$. The absolute error for $n = 47$ is plotted in Figs. 4.7. The numerical results confirm that RMLS approximations give better accuracy when we put ε as a small positive number. Since the support domain used is smaller, the computational cost of the moment matrix decrease compared with the MLS method which is confirmed by CPU time presented in Table 4.3. Contour plots given in Fig. 4.8 reveal that the RMLS method is more accurate and the efficiency of this method is concluded from using a small support domain.

4.6 Conclusion

In this chapter, an adaptive meshless approach based on the regularized moving least squares approximation (RMLS) combined with It$\hat{o}$ approximation is applied to solve two-dimensional stochastic integral equation. More accurate results are obtained by using a small support domain and this makes it suitable to study real world problems. Also, the computational complexity of the method is studied to measure the usage time of the proposed algorithm which is confirmed by numerical tests given.

Chapter 5

On the numerical solution of fractional stochastic integro-differential equations of fractional order using moving least square approximation

Fractional calculus is used to model real problems that occur in real life, the aim of this chapter is to extend the meshless method based on moving least square approximation (MLS) to solve stochastic fractional integro-differential equations (SFIDEs). To establish the scheme; we apply the composite Gauss-Legendre integration rule to compute the singular-fractional integral appearing in the scheme and the Riemann sum for estimating Itô integral. We have also compared different basis in terms of CPU time. The error bound of the numerical method is given. The major advantage of this approach is that the results converge more quickly to the exact solution by using a small number of points and basis functions, and it is more flexible because it allows an easy adaptation of the nodal density, then the computational cost of the moment matrix is reduced. This method is very convenient for solving fractional stochastic integro-differential since it does not require any need for mesh connectivity. Several numerical tests are reported and compared with the results obtained by other methods which verified the theoretical findings.

The results obtained in this chapter are presented in the research paper [33] in collaboration with Rachid El Jid and Elham Taghizadeh.

5.1 Introduction

In the last few decades, the study of fractional calculus has been an interesting research area because of its applicapility in various fields such us solid mechanics [105] control theory [15], fluid-dynamic traffic [49], and economics [10]. Accordingly, many researchers have been produced to approximate solution of fractional differential and integral equations [55], [51], [7].
Recently, stochastic models are employed in diverse areas for modeling problems occur in real life such us the study of biological population growth [64], the stochastic formulation of problems in reactor dynamics [22], the model of rainfall-runof [54]. Solving the stochastic equations analytically are not available or usually difficult because of the presence of random factors in mathematical models. Thus, some numerical approaches are used to obtain their approximate solutions [63],[110].
A fractional stochastic integro-differential equation is a stochastic integro-differential equation where order of derivative is non integer and it is a generalization of the fractional Fokker-Planck equation which describes the random walk of a particle [26]. Many phenomena in science that have been modeled by fractional differential equations have some additive noise and uncertainty, for this reason, there is a great tendency to approximate fractional stochastic integro-differential equations. For instance, few studies have been done and discussed about the numerical solutions of SFDE. Galerkin method based on orthogonal polynomials [97], spectral collocation method based on the shifted Legendre polynomials

[111] and Cubic B-spline approximation [92].
The MLS method is one of the most effective meshless approaches have implemented a significant role in numerical analysis field, it does not require domain elements or background cells. This method allows an easy adaptation of the nodal density, then the distribution of nodes could be chosen regularly or randomly in the consideration domain. This approach has been applied for solving the most classes of integral equations: 1D and 2D linear and nonlinear integral equations on rectangular and non-rectangular domains [68, 88], nonlinear 1D integro-differential equations [25], stochastic Volterra-Fredholm integral equations [90]. We obtain the error bound and the rate of convergence for the new technique. The computational complexity of the method is studied to measure the usage time of the proposed algorithm which is confirmed by numerical tests presented.
In this chapter, we focus on the numerical solution of the following stochastic fractional integro-differential equations SFIDEs by applying meshfree method based on moving least square method.

$$\begin{aligned} &D^\alpha u(x) + u(x) = g(x) + \int_0^x K_1(x,s)u(s)ds + \sigma \int_0^x K_2(x,s)u(s)dB(s), \\ &u(0) = u_0, \end{aligned} \tag{5.1.1}$$

where $g(x), K_1(x,s)$ and $K_2(x,s)$ for $x, s \in [0,1]$ are given functions, σ is real number and $u(x)$ is the unknown function to be determined, D^α denotes the Caputo fractional derivative of order α which will be introduced later. B is a Brownian motion defined on the probability space (Ω, A, P) and $\int_0^x K_2(x,s)u(s)dB(s)$ is the Itô integral. This chapter is organized as follows: The computational method for solving fractional stochastic integro-differential equations is introduced in Section 5.2. In Section 5.3, we provide the error estimate and rate convergence of the method. Several tests are presented in Section 5.4. We ended the chapter in Section 5.5.

5.2 Numerical scheme

In this section, we applied the MLS approximation for solving fractional stochastic integro-differential equation (5.1.1) for special case $0 \leq \alpha \leq 1$ and $\sigma = 1$. By using formulation of Caputo fractional derivative operator Eq. (1.4.6), we obtain

$$\frac{1}{\Gamma(k-\alpha)} \int_0^x \frac{u^k(s)}{(x-s))^{1+\alpha-k}} ds + u(x) = g(x) + \int_0^x K_1(x,s)u(s)ds + \int_0^x K_2(x,s)u(s)dB(s), \tag{5.2.1}$$

Substituting $u^h(x) = \sum_{q=1}^n \Phi_q(x)u_q$ as the MLS approximation of $u(x)$, we get

$$\begin{aligned} &\frac{1}{\Gamma(k-\alpha)} \int_0^x (x-s)^{k-\alpha-1} \sum_{q=1}^n \Phi_q^k(s)u_q ds + \sum_{q=1}^n \Phi_q(x)u_q = g(x) \\ &+ \int_0^x K_1(x,s)\Big[\sum_{q=1}^n \Phi_q(s)u_q\Big]ds + \int_0^x K_2(x,s)\Big[\sum_{q=1}^n \Phi_q(s)u_q\Big]dB(s), \end{aligned} \tag{5.2.2}$$

We collocate Eq. (5.2.2) at points x_i, then we have

$$\begin{aligned} \sum_{q=1}^n \Bigg[&\frac{1}{\Gamma(k-\alpha)} \int_0^{x_i} (x_i-s)^{k-\alpha-1} \Phi_q^k(s)ds + \Phi_q(x_i) - \int_0^{x_i} K_1(x_i,s)\Phi_q(s)ds \\ &- \int_0^{x_i} K_2(x_i,s)\Phi_q(s)dB(s) \Bigg] u_q = g(x_i). \end{aligned} \tag{5.2.3}$$

Eq. (5.2.3) can be approximated using the composite m_1-point Gauss-Legendre formula with M uniform subdivisions relative to the quadrature points δ_p and weights w_p, we have

$$\int_0^{x_i} (x_i-s)^{k-\alpha-1}\Phi_q^k(s)ds = \frac{\Delta s}{2} \sum_{l=1}^M \sum_{p=1}^{m_1} w_p (x_i - \delta_p^l)^{k-\alpha-1} \Phi_q^k((\delta_p^l), \tag{5.2.4}$$

where

$$\delta_p^l = \frac{\Delta s}{2} s_p + (l-1)\Delta s, \quad \textit{where} \quad \Delta s = \frac{x_i}{M}.$$

To estimate the second integral in Eq. (5.2.3) we use the m_1-point Gauss-Legendre formula with the quadrature points ξ_p and weights w_p as follows

$$\int_0^{x_i} K_1(x_i,s)\Phi_q(s)ds = \int_0^1 x_i K_1(x_i, x_i\xi)\Phi_q(x_i\xi)d\xi = \sum_{p=0}^{m_1} x_i w_p K_1(x_i, x_i\xi_p)\Phi_q(x_i\xi_p), \tag{5.2.5}$$

The third integral can be approximated by using the Riemann sum introduced in Eq. (1.4.1) as

$$\int_0^{x_i} K_2(x_i,s)\Phi_q(s)dB(s) = \sum_{p=0}^{n-1} K_2(x_i,s_p)\Phi_q(s_p)\Big(B(s_{p+1}) - B(s_p)\Big), \tag{5.2.6}$$

Substituting Eqs. (7.2)-(5.2.5)-(5.2.6) in Eq. (5.2.3), we get

$$\sum_{q=1}^{n}\Bigg[\frac{1}{\Gamma(k-\alpha)}\sum_{p=1}^{m_1} w_p \sum_{l=1}^{M}\frac{\Delta s}{2}(x_i-\delta_p^l)^{k-\alpha-1}\Phi_q^k(\delta_p^l) + \phi_q(x_i) \tag{5.2.7}$$
$$-\sum_{p=0}^{n} x_i w_p K_1(x_i, x_i\xi_p)\Phi_q(x_i\xi_p) - \sum_{p=0}^{n-1} K_2(x_i,s_p)\Phi_q(s_p)\Big(B(s_{p+1}) - B(s_p)\Big)\Bigg]\hat{u}_q = g(x_i),$$

where $\hat{u}_q$ are the approximations for u_q. After solving the linear system, we find the unknown u_q and then the values of $u(x)$ can be approximated by

$$\hat{u}(x) = \sum_{q=1}^{n}\phi_q(x)u_q.$$

5.3 Error estimates

Theorem 5.3.1. *[94] The derivatives of u are defined as*

$$D^\alpha u = \sum_{i=0}^{n} D^\alpha \phi_j u(x_i).$$

Then there exists a constant $C > 0$ such that for all $u \in C^{q+1}$ and all X with $h_{X,D} \le h_0$ there is an error bound

$$\|P_n D^\alpha(u) - D^\alpha(u)\|_\infty \le (1+\delta)Ch_{X,D}^{q+1-|\alpha|}|u|_{q+1}, \quad \textit{where} \quad q > |\alpha|.$$

We note that

$$D^\alpha P_n u = P_n D^\alpha u,$$

which is proved in [94].

Theorem 5.3.2. *[95] Assume that u and $U(x) = \sum_{j=0}^{n} U_j\phi_j(x)$ denote the exact and the MLS approximate solutions of Eq.(5.1.1) Then there exists a constant $C_n > 0$ such that for all $u \in C^{q+1}$ and all X with $h_{X,D} \le h_0$ there is an error bound*

$$\|D^\alpha U - D^\alpha u\|_\infty \le \delta C_n h_{X,D}^{q+1-|\alpha|}|u|_{q+1}, \quad \textit{where} \quad q > |\alpha|.$$

To prove error estimate of proposed method, we need the following inequality

Theorem 5.3.3. ***(Gronwall inequality)*** *[80] If a non-negative continuous function $e(x)$ satisfies*

$$e(x) \le C\int_{-1}^{x} e(\xi)d\xi + S(x), \quad x \in [0,1],$$

where $S(x)$ is a given integrable function, then

$$\|e\|_{L^\infty} \le C\|S\|_{L^\infty}.$$

In the following theorem, we estimate the error of the proposed method

Theorem 5.3.4. *Suppose that* u *and* $U(x)=\sum_{j=0}^{n}U_j\phi_j(x)$ *denote the exact and the MLS approximate solutions of Eq.(5.1.1), respectively. Also, suppose that*

$$\|K_1(x,v)\|\leq C_1, \|K_2(x,v)\|\leq C_2, \quad (x,v)\in[0,1]^2,$$
$$\left\|\frac{\partial K_1(x,v)}{\partial x}+\frac{\partial K_2(x,v)}{\partial x}\right\|\leq C_3, \quad (x,v)\in[0,1]^2,$$
$$\|B(x)\|\leq L, \qquad x\in[0,1],$$

where C_1, C_2, C_3, L *are positives constants. Then*

$$\|e\|_{L^\infty}\leq\frac{C}{1-LC_2}h_{X,D}\left((1+\delta)\Big(\big(2+h^q_{X,D}C_2L+(C_1+C_3)\big)\|e(t)\|_{L^\infty}+2\|u\|_{L^\infty}+(h^{q-|\alpha|}_{X,D}+h^q_{X,D})|u|_{q+1}\Big)+\delta h^{q-|\alpha|}_{X,D}|u|_{q+1}\right).$$

Proof. The numerical approximation equation can be reformulated as

$$U_j=g(x_i)-D^\alpha U_j(x_i)+x_i\sum_{j=1}^{m_1}w_jK_1\big(x_i,x_i\xi_j)U_j\big(x_i\xi_j\big)+\sum_{p=0}^{n-1}K_2(x_i,x_iv_p)U_j(x_iv_p)\Big(B(x_iv_{p+1})-B(x_iv_p)\Big) \tag{5.3.1}$$

Eq. (5.3.1) can be rewritten as

$$U_j=g(x_i)-D^\alpha U_j(x_i)+x_i\int_0^1K_1\big(x_i,x_i\xi)U_j\big(x_i\xi\big)d\xi+\int_0^1K_2(x_i,x_iv_p)U_j(x_iv_p)dB(x_iv)-S_1(x_i)-S_2(x_i), \tag{5.3.2}$$

where

$$S_1(x)=x\int_0^1K_1\big(x,x\xi)U_j\big(x\xi\big)d\xi-x\sum_{j=1}^{m_1}w_jK_1\big(x,x\xi_j\big)U_j\big(x\xi_j\big), \tag{5.3.3}$$

$$S_2(x)=\int_0^1K_2(x,xv)U_j(xv)dB(xv)-\sum_{p=0}^{n-1}K_2(x,xv_p)U_j(xv_p)\Big(B(xv_{p+1})-B(xv_p)\Big). \tag{5.3.4}$$

Assume that

$$\mu_1(x)=x\int_0^1K_1(x,x\xi)U_j\big(x\xi\big)d\xi=\int_0^xK_1(x,v)U_j\big(v\big)dv.$$

Then Eq.(5.3.3) can be writen as

$$\begin{aligned}S_1(x)&=\int_0^xK_1(x,v)U_j\big(v\big)ds-P_n\left(\int_0^xK_1(x,v)U_j\big(v\big)dv\right),\\&=\mu_1(x)-P_n(\mu_1(x)).\end{aligned} \tag{5.3.5}$$

Using Theorem 1.1.2 for $q=0$, we obtain

$$\begin{aligned}\|S_1(x)\|_{L^\infty}&=\|\mu_1(x)-P_n(\mu_1(x))\|_{L^\infty}\\&<(1+\delta)Ch_{X,D}\|\mu_1^{'}(x)\|_{L^\infty}\\&<(1+\delta)Ch_{X,D}\|K_1(x,x)U(x)+\int_0^x\frac{\partial K_1}{\partial x}(x,v)U(v)dv\|_{L^\infty}\\&<(1+\delta)Ch_{X,D}\|U\|_{L^\infty}\\&<(1+\delta)Ch_{X,D}(\|e(x)\|_{L^\infty}+\|u\|_{L^\infty}).\end{aligned} \tag{5.3.6}$$

Assume that

$$\mu_2(x)=\int_0^1K_2(x,xv)U_j(xv)dB(xv)=\int_0^xK_2(x,v)U_j\big(v\big)dB(v).$$

Then Eq.(5.3.4) can be writen as

$$
\begin{aligned}
S_2(x) =& \int_0^x K_2(x,v)U_j\big(v\big)dB(v) - P_n\Bigg(\int_0^x K_2(x,v)U_j\big(v\big)dB(v)\Bigg), \qquad (5.3.7)\\
&= \mu_2(x) - P_n(\mu_2(x))
\end{aligned}
$$

Using Theorem 1.1.2 for $q=0$, we obtain

$$
\begin{aligned}
\|S_2(x)\|_{L^\infty} &= \|\mu_2(x) - P_n(\mu_2(x))\|_{L^\infty}\\
&< (1+\delta)Ch_{X,D}\|\mu_2^{'}(x)\|_{L^\infty}\\
&< (1+\delta)Ch_{X,D}\|K_2(x,x)U(x) + \int_0^x \frac{\partial K_2}{\partial x}(x,v)U(v)dB(v)\|_{L^\infty}\\
&< (1+\delta)Ch_{X,D}\|U\|_{L^\infty}\\
&< (1+\delta)Ch_{X,D}(\|e(x)\|_{L^\infty} + \|u\|_{L^\infty}).
\end{aligned}
$$

Eq. (5.3.2) can be reformulated as

$$
\begin{aligned}
U_j =& g(x_i) - D^\alpha U_j(x_i) + \int_0^{x_i} K_1\big(x_i,v\big)U_j\big(v\big)dv + \int_0^{x_i} K_2(x_i,v)U_j(v)dB(v) - S_1(x_i) - S_2(x_i), \qquad (5.3.8)\\
\le& g(x_i) - D^\alpha U_j(x_i) + \int_0^{x_i} K_1\big(x_i,v\big)U_j\big(v\big)dv + K_2(x_i,x_i)U_j(x_i)B(x_i) - S_1(x_i) - S_2(x_i).
\end{aligned}
$$

By using $U = \sum_{j=0}^{n} U_j\phi_j(x)$, then from Eq.(5.3.8) we get

$$
U \le P_n g - D^\alpha U + P_n\Bigg(\int_0^t K_1\big(x,v\big)U\big(v\big)dv\Bigg) + P_n\Big(K_2(x,x)U(x)B(x)\Big) - P_n(S_1) - P_n(S_2). \qquad (5.3.9)
$$

By using $U = \sum_{j=0}^{n} U_j\phi_j(x)$, then from Eq.(5.1.1) we get

$$
\begin{aligned}
P_n u =& P_n g - D^\alpha u + P_n\Bigg(\int_0^x K_1(x,s)u(v)dv\Bigg) + P_n\Bigg(\int_0^x K_2(x,v)u(v)dB(v)\Bigg), \qquad (5.3.10)\\
\le& P_n g - P_n D^\alpha u + P_n\Bigg(\int_0^x K_1(x,v)u\big(v\big)dv\Bigg) + P_n\Big(K_2(x,x)u(x)B(x)\Big).
\end{aligned}
$$

Subtracting $U(x)$ from $u(x)$, then we obtain the error equation as

$$
\begin{aligned}
e(x) &= u(x) - U(x)\\
&= u(x) - P_n u(x) + P_n u(x) - U(x),\\
&\le u(x) - P_n u(x) + D^\alpha U - P_n D^\alpha u + P_n\Bigg(\int_0^x K_1\big(x,v\big)e\big(v\big)ds\Bigg)\\
&+ P_n\Big(K_2(x,x)e(x)B(x)\Big) + P_n(S_1) + P_n(S_2),\\
&\le \int_0^t K_1\big(x,v\big)e\big(v\big)dv + K_2(x,x)e(t)B(x) + P_n(S_1) + P_n(S_2)\\
&+ S_3(t) + S_4(x) + S_5(x) + S_6(x),
\end{aligned}
$$

where

$$
\begin{aligned}
S_3(x) &= u(x) - P_n u(x),\\
S_4(x) &= D^\alpha U - P_n D^\alpha u,\\
S_5(x) &= P_n\Bigg(\int_0^x K_1\big(x,v\big)e\big(v\big)dv\Bigg) - \int_0^x K_1\big(x,v\big)e\big(v\big)dv,\\
S_6(x) &= P_n\Big(K_2(x,x)e(x)B(x)\Big) - K_2(x,x)e(x)B(x).
\end{aligned}
$$

Then

$$|e(x)| \le \frac{1}{1-LC_2}\left(\int_0^x \big|K_1(x,v)e(v)\big|dv + \big|P_n(S_1)+P_n(S_2)+S_3(x)+S_4(x)+S_5(x)+S_6(x)\big| \right).$$

From the Gronwall inequality (5.3.3), we obtain

$$\|e\|_{L^\infty} \le \frac{1}{1-LC_2}\left(\|P_n(S_1)\|_{L^\infty} + \|P_n(S_2)\|_{L^\infty} + \|S_3\|_{L^\infty} + \|S_4\|_{L^\infty} + \|S_5\|_{L^\infty} + \|S_6\|_{L^\infty} \right). \tag{5.3.11}$$

The error bound for the MLS approximation implies

$$\begin{aligned}
\|P_n(S_1(x))\|_{L^\infty} &= \| \sum_{i=1}^{n} S_1(x_i)\phi_i(x)\|_{L^\infty} \\
&\le \underset{0\le i\le n}{max} |S_1(x_i)| \| \sum_{i=1}^{n} \phi_i(x)\|_{L^\infty} \\
&\le (1+\delta)Ch_{X,D}\Big(\|e(x)\|_{L^\infty} + \|u\|_{L^\infty}\Big), \\
\|P_n(S_2(x))\|_{L^\infty} &= \| \sum_{i=1}^{n} S_2(x_i)\phi_i(x)\|_{L^\infty} \\
&\le \underset{0\le i\le n}{max} |S_2(x_i)| \| \sum_{i=1}^{n} \phi_i(x)\|_{L^\infty} \\
&\le (1+\delta)Ch_{X,D}\Big(\|e(x)\|_{L^\infty} + \|u\|_{L^\infty}\Big), \\
\|S_3(x)\|_{L^\infty} &= \|u(x) - P_n u(x)\|_{L^\infty} \le (1+\delta)Ch_{X,D}^{q+1}|u|_{q+1}, \\
\|S_4(x)\|_{L^\infty} &= \|P_n D^\alpha u(x) - D^\alpha U(x)\|_{L^\infty} \\
&\le \|P_n D^\alpha u(x) - D^\alpha u(x)\|_{L^\infty} + \|D^\alpha u(x) - D^\alpha U(x)\|_{L^\infty} \\
&\le \delta C h_{X,D}^{q+1-|\alpha|}|u|_{q+1} + (1+\delta)Ch_{X,D}^{q+1-|\alpha|}|u|_{q+1}, \\
&\le (1+2\delta)Ch_{X,D}^{q+1-|\alpha|}|u|_{q+1}, \\
\|S_5(x)\|_{L^\infty} &\le C(1+\delta)Ch_{X,D}\|\Big(\int_0^t K_1\big(x,v)e(v)dv\Big)'\|_{L^\infty}, \\
&\le C(1+\delta)Ch_{X,D}\|K_1(x,x)e(x) + \int_0^x \frac{\partial K_1(x,x))}{\partial x} e(v)dv)\|_{L^\infty}, \\
&\le C(1+\delta)Ch_{X,D}(C_1+C_3)\|e(x)\|_{L^\infty}, \\
\|S_6(x)\|_{L^\infty} &= \|P_n\big(K_2(x,x)e(x)B(x)\big) - K_2(x,x)e(x)B(x)\|_{L^\infty} \\
&\le (1+\delta)CC_2Lh_{X,D}^{q+1}|e(x)|_{q+1}.
\end{aligned}$$

Then

$$\|e\|_{L^\infty} \le \frac{C}{1-LC_2}h_{X,D}\left((1+\delta)\Big(\big(2+h_{X,D}^{q}C_2L+(C_1+C_3)\big)\|e(x)\|_{L^\infty}+2\|u\|_{L^\infty}+(h_{X,D}^{q-|\alpha|}+h_{X,D}^{q})|u|_{q+1}\Big)+\delta h_{X,D}^{q-|\alpha|}|u|_{q+1} \right).$$

We get $h_{X,D} = \dfrac{1}{n-1}$, by increasing n and $h_{X,D} \to 0$, it implies $\|e(x)\| = \|u(x) - U(x)\| \to 0$ as $n \to \infty$. In the rest of chapter, we put $h_{X,D}^{q+1} = h$ for simplicity in notations. □

5.4 Numerical results

In this section some numerical tests problems are presented to demonstrate the high accuracy of the method. We give also a comparison between our described approach and the proposed method in [111], [97],[92]. We used spline weight functions and different basis functions as: the linear basis ($m = 1$), the quadratic basis ($m = 2$), the cubic basis ($m = 3$) and the quartic basis ($m = 4$). For computational details, we put $R = 1.5 \times h$ for $m = 1$, $R = 2.5 \times h$ for $m = 2$, $R = 3 \times h$ for $m = 3$ and $R = 4 \times h$ for $m = 4$, where $h = \dfrac{1}{n-1}$. In addition, we use the 3-points composite Gauss-Legendre quadrature rule with $M = 5$ for numerical integration. If the exact solution of SFIDE (5.1.1) is known then the absolute errors and the maximum errors are given as

$$e(x) = |u(x) - U(x)| \qquad\qquad \|e(x)\|_\infty = \max_{x\in[0,1]} |u(x) - U(x)|$$

All calculations are done by Matlab R2015a. In order to prove the rate convergence of this technique, the values of ratio are calculated by

$$Ratio = \frac{ln(\|e\|_\infty)_n - ln(\|e\|_\infty)_{n-1}}{ln(h)_n - ln(h)_{n-1}}$$

Example 19. Consider the following fractional stochastic integro-differential equation [97, 111, 92]

$$\begin{aligned}&D^\alpha u(x) = \frac{x^2}{2} + \frac{\Gamma(2)}{\Gamma(2-\alpha)} x^{1-\alpha} + \int_0^x u(s)ds + \sigma \int_0^x u(s)dB(s). \quad x \in [0,1]\\ &u(0) = 0\end{aligned}$$

The exact solution of this equation is unknown. In stochastic case ($\sigma = \alpha = 0$) the problem has the exact solution $u(x) = 2e^x - (2 + x)$, in this case, the maximum absolute errors are reported in Table 5.1 for different values of x and compared with those of Wavelet collocation method. Table 5.2 shows the approximate solutions obtained by the proposed method, Chebyshev wavelet method and Cubic B-spline method for $\sigma = 1$ and various values of α. The comparison of obtained errors of linear, quadratic and cubic basis are listed in Table 5.3 and plotted in Fig 5.1. The approximate solutions obtained by the MLS method for different values of α and $n = 10$ are plotted in Fig. 5.2. Contour plots of maximum error for $m = \{1, 2\}$ and $\sigma = \alpha = 0$ are plotted in Fig. 5.3. As we expected, from Theorem 5.3.4, Table 5.1 and Table 5.3 confirm that the results gradually converge to the exact values along with the increase of the nodes. According to Table 5.1 the numerical results show the higher performance of the MLS method which is more accurate and better than those of obtained method in [111]. From Table 5.3, we can see that the scheme consumes less CPU time and the computational cost of the moment matrix increases as the number of points increases. Also, the ratio of MLS error remains approximately constant for the linear case $O(h^2)$ and the quadratic case $O(h^3)$. For a high degree basis ($m = 3, m = 4$), the rate of MLS error seems to be a big value which confirms the great potential of the MLS method in this case.

Table 5.1: Errors using different values of n and $\sigma = 0$ for Example 19

n	h	$\|e\|_\infty$ MLS method		$\|e\|_\infty$ Wavelet collocation method [111]
		$m = 1$	$m = 2$	
4	0.33	2.86×10^{-2}	5.86×10^{-3}	-
6	0.20	9.03×10^{-3}	1.35×10^{-3}	1.95×10^{-2}
12	0.09	1.65×10^{-3}	1.34×10^{-4}	4.80×10^{-3}
24	0.04	3.90×10^{-4}	1.56×10^{-5}	1.20×10^{-3}
48	0.02	8.87×10^{-5}	4.66×10^{-6}	2.92×10^{-4}

Table 5.2: Numerical results using different values of x, α, $\sigma = 1$ and $n = 4$ for Example 19

	t	MLS method	Chebyshev wavelet method [97]	Cubic B-spline method [92]
$\alpha = 0.25$	0.10	0.0875	0.14133	0.03695
	0.30	0.2883	0.48481	0.22947
	0.50	0.5450	1.54907	0.32993
	0.70	0.8777	1.45203	0.79859
	0.90	1.3163	2.34412	1.16076
$\alpha = 0.5$	0.10	0.1027	0.11473	0.01593
	0.30	0.3171	0.42058	0.09125
	0.50	0.5736	1.40845	0.24115
	0.70	0.8988	1.22509	0.47565
	0.90	1.3281	1.90333	0.87980
$\alpha = 0.75$	0.10	0.1472	0.10199	0.00349
	0.30	0.4361	0.37445	0.03022
	0.50	0.7665	1.30037	0.09128
	0.70	1.1887	1.06846	0.23163
	0.90	1.7708	1.61821	0.42955

Table 5.3: Errors, rates and CPU times using different values of m and n for Example 19

	n	h	$\|e\|_\infty$	CPU time	Ratio
$m = 1$	4	0.33	2.85×10^{-2}	0.059 s	-
	6	0.16	9.03×10^{-3}	0.126	2.24
	7	0.14	6.05×10^{-3}	0.192	2.19
	8	0.125	4.33×10^{-3}	0.238	2.16
$m = 2$	4	0.33	5.86×10^{-3}	0.062	-
	6	0.16	*1.35* $\times 10^{-3}$	0.135	2.87
	7	0.14	7.96×10^{-4}	0.202	2.89
	8	0.125	5.08×10^{-4}	0.288	2.91
$m = 3$	4	0.33	9.36×10^{-4}	0.064	
	6	0.16	4.52×10^{-4}	0.160	1.42
	7	0.14	2.61×10^{-4}	0.339	3.01
	8	0.125	1.37×10^{-4}	0.513	4.18
$m = 4$	6	0.16	6.78×10^{-5}	0.233	-
	7	0.14	2.38×10^{-5}	0.381	5.74
	8	0.125	2.15×10^{-5}	0.653	0.65

Example 20. Consider the following fractional stochastic integro-differential equation [62]

$$\begin{aligned} &D^\alpha u(x) + u(x) = x + \frac{\Gamma(2)}{\Gamma(2-\alpha)} x^{1-\alpha} + \sigma \int_0^x dB(s), \quad x \in [0,1] \\ &u(0) = 0 \end{aligned}$$

The exact solution of this equation is unknown. The errors of our approach with $\sigma = 0$ and $\sigma = 1$ for different values of x are shown in Table 5.4 and compared with Galerkin method. In stochastic case $(\sigma = \alpha = 0)$ the problem has the exact solution $u(x) = x$, and the absolute error for different values of α are reported in the Table 5.5. Furthermore, the approximate solutions for different values of α with $x = 10$ are plotted in Fig. 5.4. As we can see, the numerical solutions converge to the exact solution as

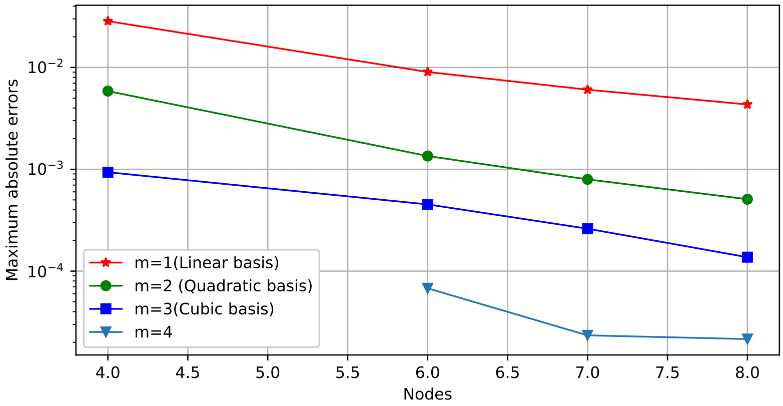


Figure 5.1: Error distributions using different basis and $\sigma = 0$ for Example 19

the value $\alpha \to 0$ which is the exact solution of this equation for the case $\sigma = \alpha = 0$. In addition, The MLS results give better accuracy than those obtained in [62].

Table 5.4: Errors using different values of x, α, σ and $n = 4$ for Example 20

x	$e(x)$ Proposed method			$e(x)$ Galerkin method [62]	
	$\sigma = \alpha = 0$	$\sigma = 1, \alpha = 0.25$	$\sigma = 1, \alpha = 0.5$	$\sigma = 1, n = 2$	$\sigma = 1, n = 4$
0.01	9.18×10^{-15}	1.29×10^{-6}	3.24×10^{-5}	1.49×10^{-2}	2.18×10^{-2}
0.20	8.32×10^{-17}	2.59×10^{-5}	6.48×10^{-4}	1.23×10^{-2}	9.00×10^{-4}
0.34	1.47×10^{-13}	4.36×10^{-5}	1.08×10^{-3}	2.89×10^{-2}	1.17×10^{-2}
0.43	1.50×10^{-13}	5.39×10^{-5}	1.32×10^{-3}	3.50×10^{-2}	1.63×10^{-2}
0.50	6.01×10^{-13}	6.17×10^{-5}	1.50×10^{-3}	3.90×10^{-2}	1.18×10^{-2}
0.65	9.37×10^{-13}	7.82×10^{-5}	1.87×10^{-3}	4.13×10^{-2}	2.14×10^{-2}
0.87	1.30×10^{-11}	1.01×10^{-4}	2.35×10^{-3}	3.23×10^{-2}	1.72×10^{-2}
1.00	3.30×10^{-11}	1.14×10^{-4}	2.62×10^{-3}	1.99×10^{-2}	9.70×10^{-3}

Table 5.5: Numerical results using different values of x, α, $\sigma = 1$ and $n = 4$ for Example 20

	$\sigma = 0$		$\sigma = 1$		
t	Exact solution	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$
0.10	0.1000	0.1000	0.1000	0.1010	1.1140
0.30	0.3000	0.3000	0.3001	0.3027	0.3321
0.50	0.5000	0.5000	0.5002	0.5042	0.5479
0.70	0.7000	0.7000	0.7003	0.7055	0.7611
0.90	0.8999	0.8999	0.9004	0.9067	0.9723

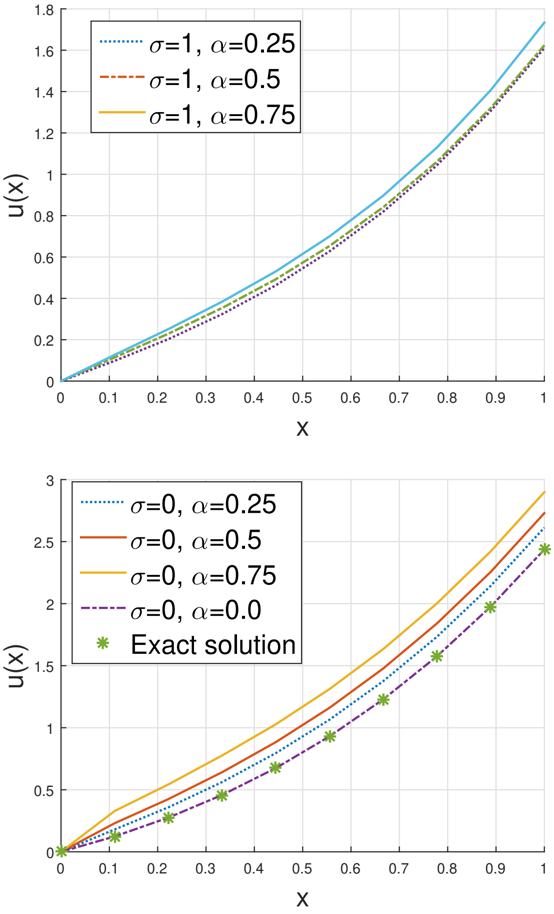


Figure 5.2: The approximate solution using different values of σ and α for Example 19

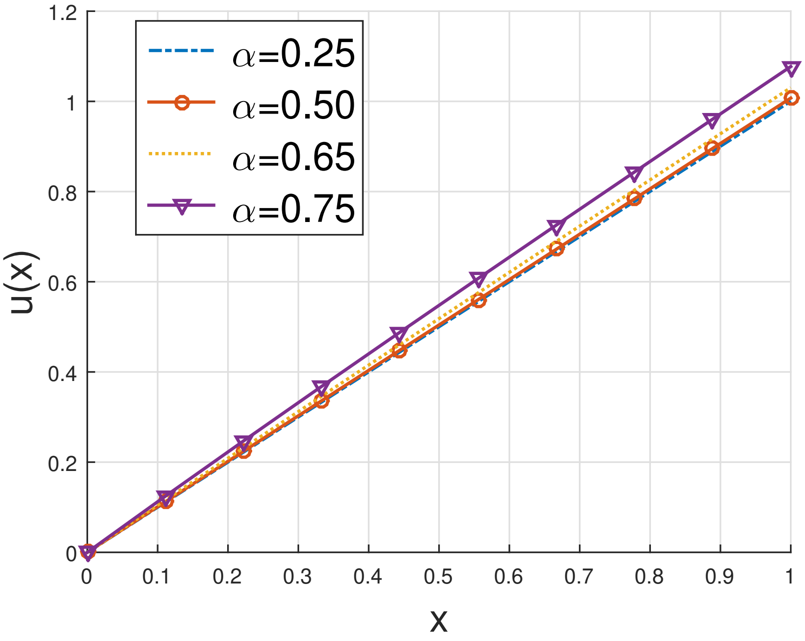
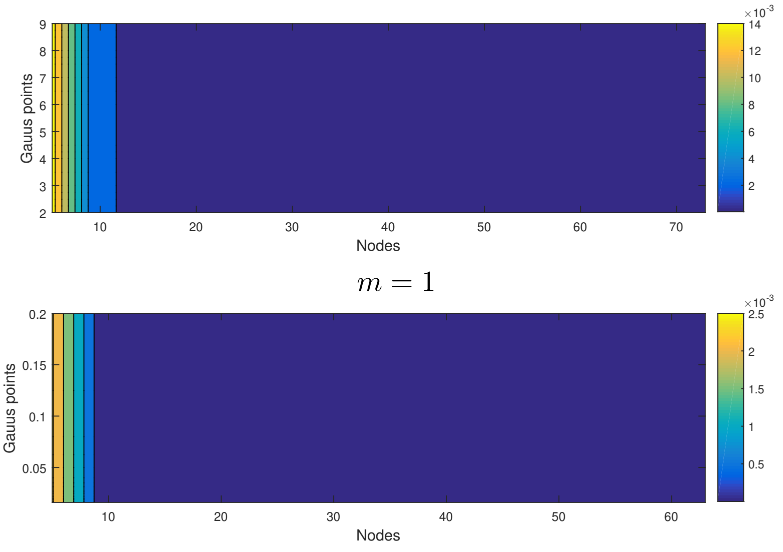


Figure 5.4: The approximate solution for different values of α and $n = 10$ for Example 20



$m = 1$

$m = 2$

Figure 5.3: Contour plots of maximum error (nodes, the distance between two consecutive nodes) for $m = \{1, 2\}$ and $\sigma = \alpha = 0$ for Example 19

Example 21. We consider the following stochastic integro-differential equation

$$\grave{u}(x) = g(x) + \int_0^1 cos(x - r)u(r)dr + \int_0^1 exp(x - r)u(r)dB(r),$$
$$u(0) = 0.$$

where

$$g(x) = xe^x + e^x - 0.25(cos(x) - e^1 sin(1 - x)) - e^x \left[B(1) - \int_0^1 B(r)dr\right].$$

and the exact solution is $u(x) = xe^x$. Absolute errors using regular and irregular points are shown in Table 5.6. Also, the Maximum error for Example 21 with different values of x and m are given in Table 5.7. Fig. 5.6 plots the graph of maximum solutions for different values of m. Figure 5.7 plots the approximate solution which is the same as the exact solution. Contour plots of maximum error for $m = \{1, 2, 3\}$ and $\sigma = \alpha = 0$ are plotted in Fig. 5.5. As we can see in Theorem 5.3.4, the results gradually converge to the exact values as the number of nodes increases. For a given CPU time. the ratio

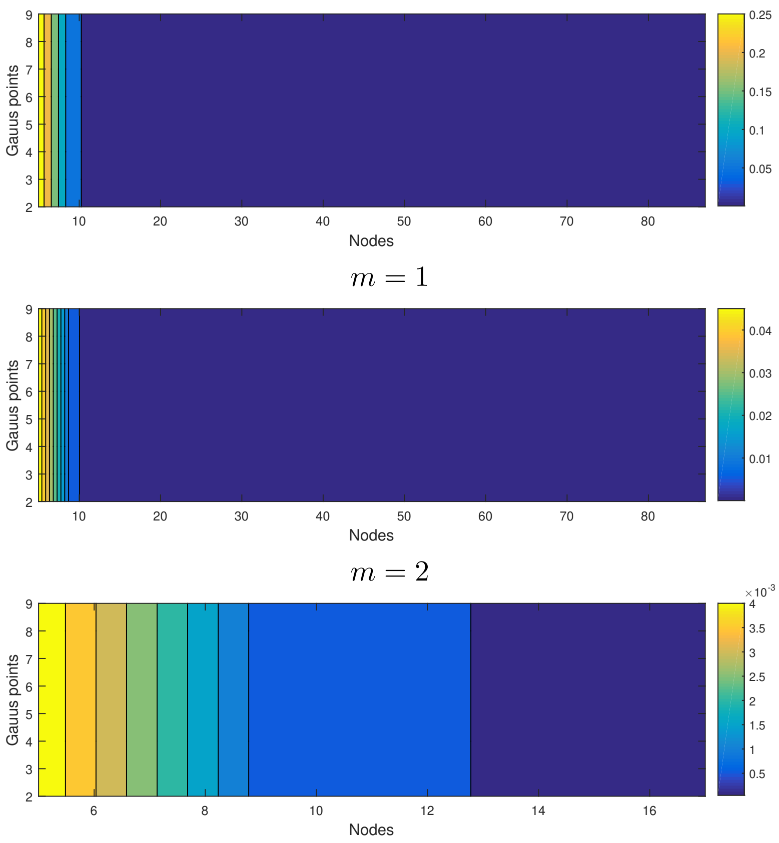
Table 5.6: Errors using regular and irregular points and $m = 3$ for Example 21

	x	$n = 4$		$n = 7$	
		$e(x)$	CPU times	$e(x)$	CPU times
Regular points	0.1	$3.04\ \times 10^{-2}$	0.076 s	$1.10\ \times 10^{-2}$	0.163 s
	0.01	$2.68\ \times 10^{-3}$	0.075	$1.91\ \times 10^{-4}$	0.165
	0.001	$2.65\ \times 10^{-4}$	0.073	$9.76\ \times 10^{-6}$	0.165
	0.0001	$2.69\ \times 10^{-5}$	0.077	$7.69\ \times 10^{-8}$	0.168
	0.00001	$3.66\ \times 10^{-6}$	0.074	$8.08\ \times 10^{-7}$	0.159
Irregular points	0.1	1.41×10^{-2}	0.075	1.11×10^{-2}	0.118
	0.01	9.86×10^{-4}	0.076	1.46×10^{-4}	0.133
	0.001	9.54×10^{-5}	0.074	5.31×10^{-6}	0.123
	0.0001	9.51×10^{-6}	0.074	4.38×10^{-7}	0.126
	0.00001	9.49×10^{-7}	0.078	4.29×10^{-8}	0.140

Table 5.7: Errors using different values of n and m for Example 21

	n	h	$\|e\|_\infty$	Ratio	CPU times
$m = 1$	5	0.25	2.88×10^{-1}	-	1.07 s
	9	0.12	6.27×10^{-2}	2.19	1.33
	12	0.09	3.23×10^{-2}	2.08	1.84
	17	0.06	1.70×10^{-2}	1.71	3.65
	25	0.04	7.03×10^{-3}	2.17	6.78
	38	0.02	2.94×10^{-3}	2.01	31.79
	87	0.01	6.69×10^{-4}	1.73	750.24
$m = 2$	5	0.25	4.93×10^{-2}	-	1.17
	9	0.12	6.62×10^{-3}	2.89	1.41
	12	0.09	1.93×10^{-3}	3.87	1.93
	17	0.06	9.01×10^{-4}	2.03	3.77
	25	0.04	4.11×10^{-4}	1.93	8.19
	38	0.02	2.08×10^{-4}	1.57	38.25
	87	0.01	2.23×10^{-5}	2.61	783.48
$m = 3$	5	0.25	4.44×10^{-3}	-	1.20
	9	0.12	8.02×10^{-4}	2.46	1.54
	12	0.09	5.84×10^{-4}	0.99	2.02
	17	0.06	4.59×10^{-5}	6.78	3.91

of MLS error remains approximately constant for the linear case $O(h^2)$ and the quadratic case $O(h^3)$, for a high degree basis ($m = 3, m = 4$), the rate of MLS error seems to be a big value which confirms the convergence of the method. Also, the scheme consumes less CPU time and the computational cost of the moment matrix increase as the number of points increases.



$m = 1$

$m = 2$

$m = 3$

Figure 5.5: Contour plots of maximum error (nodes, the distance between two consecutive nodes) for $m = \{1, 2, 3\}$ for Example 21

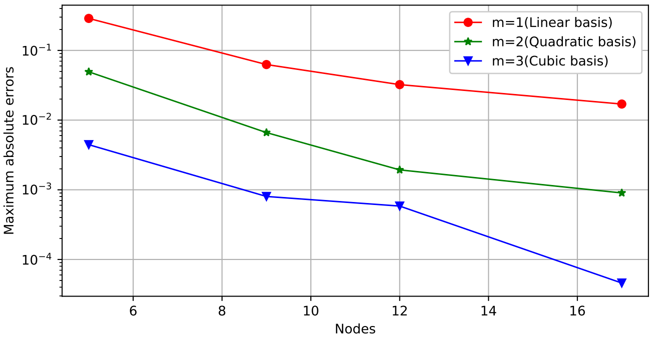


Figure 5.6: Error distributions using different basis and $\sigma = 0$ for different values of m and n for Example 21

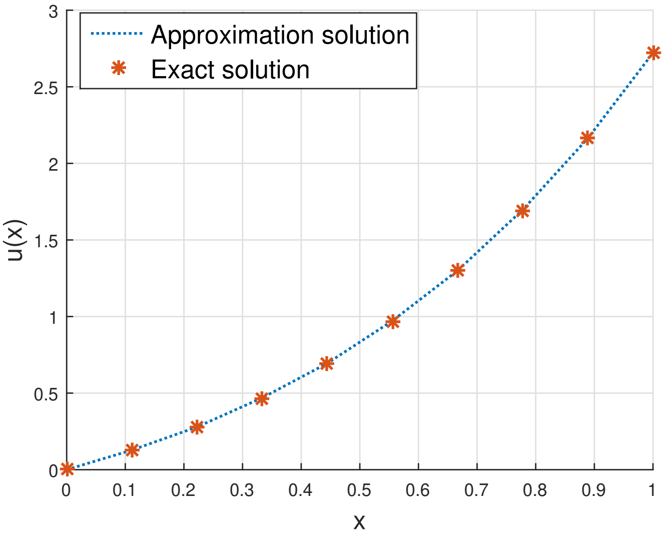


Figure 5.7: Comparison between the numerical solution and the exact solution for $\sigma = 0$ and $n = 15$ for Example 21.

5.5 Conclusion

In this chapter, the combination of MLS approximation, composit Gauss-Legendre quadrature formula, and Itô approximation are used to solve fractional stochastic integro-differential equations. The achieved results indicate that the RMLS method uses much less volume computing and time. This approach provides acceptable results with a small number of nodes and basis functions. The suggest method is practical and it can be used in cases where the exact answer to the problem is not available. Also, the method is free from choosing nodes as regular or irregular points and this makes it suitable to study real world problems.

General conclusion and perspectives

In this thesis, new adaptive meshless schemes based on the modified moving least squares approximation(MMLS) and the regularized moving least squares approximation(RMLS) have been proposed as novel numerical models, which have been applied to solve a different class of integral equations and integro-differential equations in regular and irregular domains.

The first part of this thesis is devoted to applying the modified moving least squares approach for a class of integral equations on 2D and 3D domains. More specifically, in Chapter 2, we treated the question of singular moment matrix in the context of MLS based method by adding additional terms based on the coefficients of the polynomial base functions, leading to have a better approximation capability with a nonsingular moment matrix. We presented the moving least squares (MLS) and modified moving least squares (MMLS) methods for solving two-dimensional linear and nonlinear Fredholm-Volterra integral equations and Fredholm-Hammerstein integral equations on 2D irregular domains. The numerical examples show that the approximation of the MMLS gave more accurate results than that of the classical MLS. The MMLS method can be used in various irregular domains since it does not depend on the geometry of the domain. The efficiency of the obtained solutions can be improved by taking more nodes in the domain. In Chapter 3, the three-dimensional mixed integral equations are solved by using the MMLS approach, compared with the classical MLS method. The most advantage of the MMLS method is the potential to get an approximation for cases when classical MLS with quadratic base functions fail because of a non-invertible moment matrix. It can be concluded that the MMLS method is very convenient for solving integral equations in high dimensions. Apparently, the MMLS approximation is a powerful tool for solving mixed integral equations and it can be extended without difficulties to the three-dimensional problems because of the simple adaption of the MMLS method for the 3-D space.

In the second part, we deal with two chapters on the theoretical and numerical study of the RMLS approach and its applications for solving stochastic integral equations and stochastic differential equations. More specifically, in Chapter 4, the combination of an efficient meshfree method, the regularized MLS approximation, and the It$\hat{o}$ approximation for solving two-dimensional stochastic Fredholm integral equations are applied. Numerical examples have been examined and compared with classical MLS, modified MLS, and RBF methods which confirmed the validity and the reliability of the new approach with only a few nodal points. The achieved results indicate that the RMLS method uses much less volume computing and time. This approach is free from choosing the domain and this makes it suitable to study real world problems. In Chapter 5, the combination of the MLS approximation, the Caputo fractional derivative, the composite Gauss-Legendre quadrature formula, and the Itô approximation are used to solve the fractional stochastic integro-differential equation. To illustrate the capability of the proposed method, three different test cases are solved. Obtained results indicate that this approach can be used in cases where the exact solution to the problem is not available. the method is free from choosing nodes as regular or irregular points and this makes it suitable to study real world problems. The valuable advantage of the MLS method is that it can be used in various irregular domains since it is free from choosing the domain, furthermore, it can be easily extended to the higher problems because of the simple adaption of MLS method.

Based on the work undertaken in this thesis, there are some areas which present themselves as clear opportunities for future research and development. Some of those points are listed below:

1) Application of MLS method for solving system of linear Stratonovich Volterra integral equation

Stochastic integrals are split into two parts in stochastic processes; Stratonovich integrals which is frequently used in physics and Itô integrals which are usually used in applied mathematics. Unlike the Itô calculus. the Stratonovich integral is defined to hold the chain rule in ordinary calculus. Owing to this feature, stochastic differential equations are easier to describe the differentiable manifolds in the Stratonovich sense. On many occasions, it is not possible to find an exact solution for these equations. This fact, together with an essential requirement in applied mathematics for solving them, has given rise to a heightened development of work aimed to design numerical methods for solving this type of equation. In this work, we suggest using the MLS method to get an approximate solution for the linear system of Stratonovich Volterra integral equation

$$\frac{du(x)}{dx} = g(x) + \int_a^b K_1(x,t)u(t)dt + \int_a^x K_2(x,t)u(t)odB(t), \quad x \in D, \tag{6.0.1}$$

where

$$\begin{aligned} u(x) &= [u_1(x), u_2(x), ..., u_n(x)]^T, \\ f(x) &= [g_1(x), g_2(x), ..., g_n(x)]^T, \\ K_1(x,t) &= [K1_{i,j}(x,t)], \quad i,j = 1,2,...,n. \\ K_2(x,t) &= K1_{i,j}(x,t), i,j = 1,2,...,n. \end{aligned}$$

where $g_i(x), K1_{i,j}(x,y)$ and $K2_{i,j}(x,y)$ for $i,j = 1,2,...,n$ are known functions whereas $u_i(x)$ are unknown functions. B(x) is a standard Brownian motion. Note that the symbol o between integrand and the stochastic differential is used to indicate Stratonovich integrals.

2) Finding the error analysis of the MLS method for the Volterra integral equations

In recent literature, the researches of the mathematical theory of meshless methods are much less than those of their applications. The error analysis of MLS approximation was provided by some authors, beginning with the work of Levin [77] presented the error estimates in the uniform norm for a particular weight function in N dimensions, but, he did not obtain error estimates for the derivatives. In [4] Armentano and Duron studied the MLS method for the function and its derivatives in the one dimension obtaining error estimate in L^∞. Armentano in [3] presented the error estimates in L^∞ and L^2 norms for one and N dimensions which generalize the result given in [4]. Zuppa in [129] proved the error estimates for derivatives of shape function by the condition numbers of the star of nodes in the normal equation. Authors of [103], obtained the error estimates in Sobolev space when $u(x,y) \in C^{m+1}(D)$, and $u(x,y) \in W^{m+1,q}(D)$, respectively. However finding the error analysis of the MLS method for the Volterra integral equations is an interesting research problem

3) Finding the optimum value α for the Gaussian weight function

An important ingredient of meshless methods is the weight function which is denoted by w_i, consider the following Gaussian weight function given us

$$w(s) = \begin{cases} \dfrac{e^{-(\frac{s}{\alpha})^2} - e^{-(\frac{1}{\alpha})^2}}{1 - e^{-(\frac{1}{\alpha})^2}} & \text{si } |s| \leq 1 \\ 0 & \text{si } |s| > 1 \end{cases} \tag{6.0.2}$$

α is a constant controlling the shape of the weight function

The Gaussian weight function is sensitive in respect to parameter α. Experiments show $\alpha = 0.6 \times \delta$ is an appropriate choice, where δ is the distance between two consecutive nodes. However, finding the optimum value for α is an open problem for this method.

4)Solving three-dimensional time fractional diffusion equation with variable-order derivatives

One of the important classes of fractional partial differential equations (FPDEs) [89] which is widely used nowadays is the well-known time fractional diffusion equation [45, 101]. This kind of equation can be used to explain the phenomena such as anomalous diffusion which involves long-time simulations in a better way. The time fractional diffusion equation can be written in the following form

$$\frac{\partial^{\alpha(x,t)}u(x,t)}{\partial t^{\alpha(x,t)}} - \kappa\triangle(x,t) + \nu.\nabla u(x,t) - f(x,t) = 0, \quad x = (x,y,z) \in \Omega \subset \mathbb{R}^3, \tag{6.0.3}$$

$$u(x,0) = h(x), \quad x \in \Omega \tag{6.0.4}$$

$$u(x,t) = g(x,t), \quad x \in \partial\Omega, \quad t \in [0, t_{finale}], \tag{6.0.5}$$

where ∂u and ∇u denote Laplace and gradient operators, respectively, Ω represents the computational domain in $\mathbb{R}^3$ with boundary $\partial\Omega$, κ stands for the diffusion coefficient, represents the vector of velocities, $\nu = [\nu, \nu_y, \nu_z]$ represents the vector of velocities, and $\dfrac{\partial^{\alpha(x,t^{n+1})}u(x,t^{n+1})}{\partial t^{\alpha(x,t^{n+1})}}$ stands for the fractional derivative with variable-orders [113] which can be defined as

$$\frac{\partial^{\alpha(x,t)}u(x,t)}{\partial t^{\alpha(x,t)}} = \frac{1}{\Gamma(1-\alpha(x,t))}\int_0^t \frac{\partial u(x,\xi)}{\partial \xi}\frac{1}{(t-\xi)^{\alpha(x,t)}}, \quad 0 < \alpha, (x,t) < 1.$$

The current dominant numerical method for modeling FPDE is the finite difference Method (FDM), which is based on a pre-defined grid leading to inherited issues or shortcomings including difficulty in the simulation of problems with the complex problem domain and in using irregularly distributed nodes. Because of its distinguished advantages, the meshless method has good potential in the simulation of FPDEs.

5) Solving Sobolev equation arising in fluid dynamics

The Sobolev equations have been used for modeling many phenomena in mechanical engineering and physics such as the migration of the moisture in soil [99], thermodynamics [115], and the motion of fluid in different media such as soil and rock [119]. The Sobolev equation of order can be written in the following form

$$u_t - \gamma\Delta u_t - \sigma\Delta u = g(x,t), \quad x \in \Omega \subset \mathbb{R}^2, t \in [0,T]. \tag{6.0.6}$$

The initial condition is considered as

$$u(x,0) = f(x), x \in \Omega$$

together with the boundary conditions

$$u(x,t) = h(t), \quad x \in \partial\Omega, \quad t \in [0,T]$$

where

- σ is the factor of piezo-conductivity of fissured rock,
- γ represents a new characteristic related to the fissured rocks.

$g(x,t), f(x)$ and $h(t)$ are known functions, σ and γ are positive constants, and Δ is considered as second-order derivative operator. The existence and uniqueness of the solution for the Sobolev equations have been discussed in [116].
the solutions of FPDEs cannot be achieved via usual paper–pencil work. Therefore, researchers are constantly developing various approximation methods to cope with this issue. Approximate solutions of FPDEs have been reported via finite differences scheme [34], weak Galerkin finite element method scheme [42] and Legendre spectral element scheme [24]. However, implementation of these schemes becomes a challenge as dimension of the problem domain increases. In such a situation, meshless methods perform very well due to their independence of problem geometry and dimension.

Annex

This section presents preliminary material from functional analysis which we have used in the previous chapters, and the basic numerical steps and results for implementation of MLS shape functions

7.1 Preliminary functional analysis

Normed spaces

We start this section with basic definitions, notations and results concerning the normed spaces.
Given a linear space X, we recall that a norm $\|.\|_X$ is a function from X to $\mathbb{R}$ with the following properties. Let X be a vector space over K. A norm on X is a map $\|.\| : X \longrightarrow [0, \infty)$ that satisfies the following three properties.

1. $\|u\|_X = 0 \quad implies \quad x = 0.$ (definiteness).
2. $\|\lambda x\| = \|\lambda\| \|x\|,$ for all $x \in X \quad and \quad \lambda \in K.$ (homogeneity).
3. $\|x + y\| \leq \|x\| + \|y\|,$ for all $x, y \in X.$ (triangle inequality).

A normed space is a pair $(X, \|.\|)$, where X is a vector space and $\|.\|$ is a norm on X. The vector space $\mathbb{R}^n$ with the usual addition and scalar multiplication allows for several norms, for example:

- The Euclidean norm $\|(x_1, ..., x_n)\|_{l_2} = (x_1^2 + ... + x_n^2)^{1/2}$,
- The maximum norm $\|(x_1, ..., x_n)\|_{l_\infty} = max\{|x_1|, ..., |x_n|\}$,
- The summation norm $\|(x_1, ..., x_n)\|_{l_1} = \{|x_1| + ... + |x_n|\}$.

These are all special cases of the (finite-dimensional) l_p-norm

$$\|(x_1, ..., x_n)\|_{l_p} = \left(\sum_{j=1}^{n} |x_j|^p \right)^{1/p}, \quad 1 \leq p \leq \infty.$$

Let $\|.\|^{(1)}$ and $\|.\|^{(2)}$ be two norms over a linear space X. The two norms are said to be equivalent if there exist two constants $c_1, c_2 > 0$ such that

$$c_1 \|u\|^1 \leq \|u\|^2 \leq c_2 \|u\|^1. \quad u \in X.$$

A sequence $(x_n)_{n \in N}$ in X is said to converge to $a \in X$ if

$$\forall \varepsilon > 0, \, \exists N_0 \in \mathbb{N} : \forall n \geq N_0, \, \|x_n - a\| < \varepsilon.$$

A sequence $(x_n)_{n \in N}$ in X is said to be bounded if there exists $M > 0$ such that

$$\|u_n\|_X \leq M, \quad \forall n \in \mathbb{N},$$

or, equivalently, if

$$\sup_n \|u_n\| \leq \infty.$$

To test the convergence of a sequence without knowing its limit, it is usually convenient to refer to the notion of a Cauchy sequence. Let X be a normed space. A sequence $(u_n) \in X$ is called a Cauchy sequence if

$$\|u_m - u_n\| \longmapsto 0, \; as \quad m, n \longmapsto \infty.$$

The open ball about $x \in X$ with radius $r > 0$ is the set

$$B(X, r) = \{y \in X, \; \|x - y\| < r\}.$$

Definition 7.1.1. A normed space is said to be complete if every Cauchy sequence from the space converges to an element in the space. A complete normed space is called a Banach space.

Definition 7.1.2. Given a linear space X we recall that an inner product $(X, \|.\|_X)$ is a function from $X \times X$ to $\mathbb{R}$ with the following properties.

- $(u, u)_X \geq 0, \forall u \in X$ and $(u, u)_X = 0$ if and only if $u = 0_X$,
- $(u, v)_X = (v, u)_X, \forall u, v \in X$.
- $(\alpha u + \beta v, \omega) = \alpha(u, \omega) + \beta(v, \omega), \quad \forall u, v, w \in X, \forall \alpha, \beta \in \mathbb{R}$.

The pair $(X, \|.\|_X)$ is called an inner product space.

Definition 7.1.3. A complete inner product space is called a Hilbert space.

we suppose that $a, b \in \mathbb{R}$ with $a < b$, consider the closed interval. Recall the space $B([a, b])$ (comprising all bounded functions on $[a, b]$) and consider the linear subspace $C^0[a, b]$ comprising all continuous functions $f : [a, b] \longrightarrow \mathbb{R}$. (By the Weierstrass extreme value theorem, any such function is bounded, thus we have the inclusion $C^0([a, b]) \subset B([a, b])$. The restriction of the supremum norm to $C^0([a, b])$ gives rise to a normed subspace of $B([a, b])$. We denote its norm by

$$\|f\|_{C^0([a,b])} = \sup_{x\in[a,b]} |f(x)|, \quad f \in C^0([a, b]).$$

Definition 7.1.4. Let $k \in \mathbb{N}$. Then $C^k([a, b])$ comprises all functions $f \in C^0([a, b])$ such that f is k times continuously differentiable in (a, b) and there exist $g_1, ..., g_k \in C^0([a, b])$ such that $f^i(t) = g_i(t)$ for all $t \in (a, b)$ and $i = 1, ..., k$. For $f \in C^k([a, b])$, we define

$$\|f\|_{C^k([a,b])} = \sum_{i=0}^{k} \|g_i\|_{C^0([a,b])},$$

where $g_0 = f$ and $g_1, ..., g_k \in C^0([a, b])$.

In other words, a function $f[a, b] \longrightarrow \mathbb{R}$ belongs to $C^k([a, b])$ if it is continuous on $[a, b]$ and k times continuously differentiable in (a, b) and every derivative up to order k has a continuous extension to $[a, b]$. It is common to abuse notation and write f^i for the continuous extension of the i-th derivative of f (rather than g_i). Then

$$\|f\|_{C^k([a,b])} = \sum_{i=0}^{k} \|f^i\|_{C^0([a,b])}$$

7.1.1 Support and Compact Support

Let U be an open set in $\mathbb{R}^n$, and let $f : U \longrightarrow \mathbb{R}$ be a continuous function.

Definition 7.1.5. The support of f is

$$supp = \overline{\{x \in U : f(x) \neq 0\}}.$$

Definition 7.1.6. Let $f : U \longrightarrow \mathbb{R}$, the function f is compactly supported if $suppf$ is compact.

Notation $C_0^k(U) =$ *The set of compactly supported* C^k *functions on* U.

Suppose that $f \in C_0^k(U)$. Define a new set $U_1 = (\mathbb{R}^n - suppf)$. Then $U \cup U_1 = \mathbb{R}^n$, because $suppf \subseteq U$.

7.1.2 Partitions of Unity

Let $\{U_\alpha : \alpha \in I\}$ be a collection of of open subsets of $\mathbb{R}^n$ such that $U = \cup_\alpha U_\alpha$.

Theorem 7.1.1. *There exists a sequence of rectangles $Q_i, i = 1, 2, 3, ...$ such that*

1. *Int $Q_i, i = 1, 2, 3, ...$ is a cover of U,*
2. *Each $Q_i \subset I_\alpha$ for some α,*
3. *For every point $p \in U$, there exists a neighborhood Up of p such that $U_p \cap Q_i = \phi$ for all $i > N_p$.*

The following theorem is called the Partition of Unity Theorem.

Theorem 7.1.2. *There exist functions $f_i \subseteq C^0_\infty(U)$ such that*

1. *$f_1 \geq 0$,*
2. *supp $f_i \subseteq U_\alpha$, for some α,*
3. *For every $p \in U$, there exists a neighborhood Up of p such that $U_p \cup suppf_i = \phi$ for all $i > N_p$,*
4. *$\sum f_i = 1$.*

7.1.3 Hölder spaces

Suppose $\Omega \subset \mathbb{R}^d$ is an open set and $0 < \gamma < 1$. Recall that k-Lipschitz continuous functions $f : \Omega \longrightarrow \mathbb{R}$ satisfy by definition the following estimate:

$$|f(x) - f(y)| \leq k\|x - y\|, \quad \forall x, y \in \Omega,\ k \in \mathbb{R}_+.$$

This relation provides a uniform modulus of continuity. It is often useful to consider functions f satisfying

$$|f(x) - f(y)| \leq k\|x - y\|^\gamma, \quad x, y \in \Omega,\ k \in \mathbb{R}_+.$$

Such functions are said to be Hölder continuous with exponent $\gamma \in \mathbb{R}_+$.

Definition 7.1.7. Consider $f : \Omega \longrightarrow \mathbb{R}$

- if f is bounded and continuous, we write

$$\|f\|_{C^0(\bar{\Omega})} \overset{def}{=} \underset{x \in \Omega}{sup} |f(x)|.$$

- the γ^{th}-Hölder seminorm of f is:

$$|f|_{C^{0,\gamma}(\bar{\Omega})} \overset{def}{=} \underset{\substack{x,y \in \Omega \\ x \neq y}}{sup} \left(\frac{|f(x) - f(y)|}{|x - y\|^{(\gamma}} \right),$$

and the γ^{th}-Hölder norm of f is:

$$\|f\|_{C^{0,\gamma}(\bar{\Omega})} \overset{def}{=} \|f\|_{C^0(\bar{\Omega})} + |f|_{C^{0,\gamma}(\bar{\Omega})}.$$

Definition 7.1.8. The Hölder space $C^{k,\gamma}(\bar{\Omega})$ consists of all functions $f \in C^k(\bar{\Omega})$ for which the norm

$$\|f\|_{C^{k,\gamma}(\bar{\Omega})} \overset{def}{=} \sum_{|\alpha| \leq k} \|\partial^\alpha f\|_{C^0(\bar{\Omega})} + \sum_{|\alpha| = k} |\partial^\alpha f|_{C^{0,\gamma}(\bar{\Omega})},$$

is finite.

Definition 7.1.9. The space of functions $C^{k,\gamma}(\bar{\Omega})$ is a Banach space.

L^p spaces

Definition 7.1.10. Let (X, A, μ) be a measure space and $1 \leq p < \infty$. The space $L^p(X)$ consists of equivalence classes of measurable functions $f : X \longrightarrow \mathbb{R}$ such that

$$\int |f|^p d\mu < \infty,$$

where two measurable functions are equivalent if they are equal μ-a.e. TheL^p-norm of $f \in L^p(X)$ is defined by

$$\|f\|_{L^p} = \left(\int |f|^p d\mu \right)^{1/p}.$$

The notation $L^p(X)$ assumes that the measure μ on X is understood. We say that $f_n \longrightarrow f$ in L^p if $\|f - f_n\|_{L^p} \longrightarrow 0$.

The space $L^\infty(X)$ is defined in a slightly different way. First, we introduce the notion of esssential supremum.

Definition 7.1.11. Let $f : X \longrightarrow \mathbb{R}$ be a measurable function on a measure space (X, A, μ). The essential supremum of f on X is

$$ess \sup_X f = inf\{a \in \mathbb{R} : \mu\{x \in X : f(x) > a\} = 0\}.$$

Equivalently

$$ess \sup_X f = inf\Big\{ \sup_X g : g = f pointwise\ a.e \Big\}.$$

Thus, the essential supremum of a function depends only on its μ-a.e. equivalence class. We say that f is essentially bounded on X if

$$ess \sup_X |f| < \infty$$

Definition 7.1.12. Let (X, A, μ) be a measure space. The space $L^\infty(X)$ consists of pointwise a.e.-equivalence classes of essentially bounded measurable functions $f : X \longrightarrow \mathbb{R}$ with norm

$$\|f\|_{L^\infty} = ess \sup_X |f|.$$

We state two fundamental inequalities.

Theorem 7.1.3. ***(Minkowski inequality)****. If $f, g \in L^p(X)$, where $1 \leq p \leq \infty$, then $f + g \in L^p(X)$ and*

$$\|f + g\|_p \leq \|f\|_p + \|g\|_p.$$

This inequality means, as stated previously, that $\|.\|_{L_p}$ is a norm on $L^p(X)$ for $1 \leq p \leq \infty$. If $0 < p < 1$, then the reverse inequality holds

$$\|f\|_p + \|g\|_p \leq \|f + g\|_p,$$

so $\|.\|_{L^p}$ is not a norm in that case. Nevertheless, for $0 < p < 1$ we have

$$|f + g|^p \leq |f|^p + |g|^p,$$

so $L^p(X)$ is a linear space in that case also. To state the second inequality, we define the Hölder conjugate of an exponent.

Definition 7.1.13. Let $1 \leq p \leq \infty$. The Hölder conjugate $p^{'}$ of p is defined by

$$\frac{1}{p} + \frac{1}{p^{'}} = 1, \quad if \quad 1 < p < \infty.$$

Theorem 7.1.4. ***(Hölder's inequality)****. Suppose that (X, A, μ) is a measure space and $1 \leq p \leq \infty$. If $f \in L^p(X)$ and $g \in L^{p^{'}}(X)$, then $fg \in L^1(X)$ and*

$$\int fg d\mu \leq \|f\|_{L^p} \|g\|_{L^{p^{'}}}.$$

For $p = p^{'} = 2$, this is the Cauchy-Schwartz inequality.

7.2 Implementation of MLS shape functions

All codes are implemented in Matlab language and given the results in the following listing:

7.2.1 Numerical steps for implementation of MLS shape functions on 1D

- **Node distribution**: Consider an interval $[0 \quad 4]$ which is divided into 5 equally spaced nodes depicted in Fig.7.1 and given data coordinates as

$$node = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

Figure 7.1: Node distribution

- **Define support** The domain of influence for nodes is a circle with fixed radius $d = 1.5$ (1.1.13), and finding all points including in each local support nodes

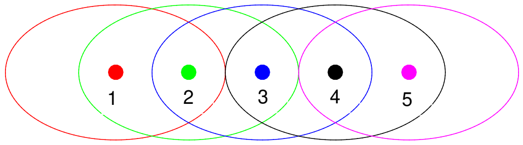


Figure 7.2: MLS circular support

$$Index(1) = \{1,2\}, \qquad Index(2) = \{1,2,3\}, \qquad Index(3) = \{2,3,4\}$$

$$Index(4) = \{3,4,5\}, \qquad Index(5) = \{4,5\}.$$

- **MLS Weight functions**: A cubic weight function (1.1.16) is chosen to compute shape function, then we have

$$\begin{aligned}
\omega(1) &= [1.0000 \quad 0.1111 \quad 0 \quad 0 \quad 0] \\
\omega(2) &= [0.1111 \quad 1.0000 \quad 0.1111 \quad 0 \quad 0], \\
\omega(3) &= [0 \quad 0.1111 \quad 1.0000 \quad 0.1111 \quad 0], \\
\omega(4) &= [0 \quad 0 \quad 0.1111 \quad 1.0000 \quad 0.1111], \\
\omega(5) &= [0 \quad 0 \quad 0 \quad 0.1111 \quad 1.0000].
\end{aligned}$$

  From those results, one can see that the weight functions are positive and non zero only over its support.

- **MLS shape functions**: A linear basis is used to compute shape function, then we have

$$\begin{aligned}
\phi(1) &= [1.0000 \quad 0.0000 \quad 0 \quad 0 \quad 0] \\
\phi(2) &= [0.0207 \quad 0.9586 \quad 0.0207 \quad 0 \quad 0], \\
\phi(3) &= [0 \quad 0.0207 \quad 0.9586 \quad 0.0207 \quad 0], \\
\phi(4) &= [0 \quad 0 \quad 0.0207 \quad 0.9586 \quad 0.0207], \\
\phi(5) &= [0 \quad 0 \quad 0 \quad 0.0000 \quad 1.0000].
\end{aligned}$$

  From those results, one can easily see that MLS shape functions satisfy the consistency property (1.1.18) of MLS approximation.

7.2.2 Numerical steps for implementation of MLS shape functions on 2D

- **Node distribution**: Let us define a vector node including 25 equidistant points and given data coordinates as follows

$$node = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

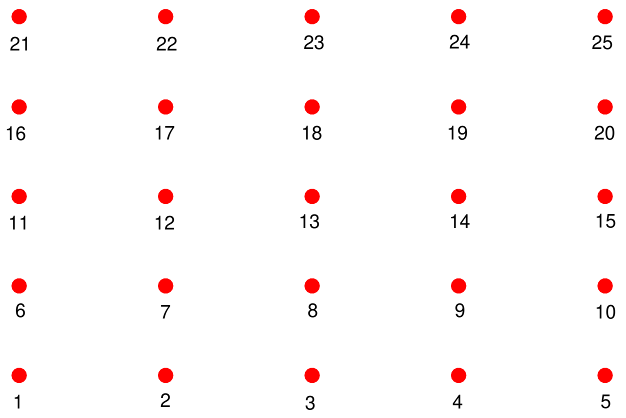


Figure 7.3: Node distribution

- **Define support**: The domain of influence for nodes is a circle with fixed radius $d = 2.5$ (1.1.13), and finding all points including in each local support node:

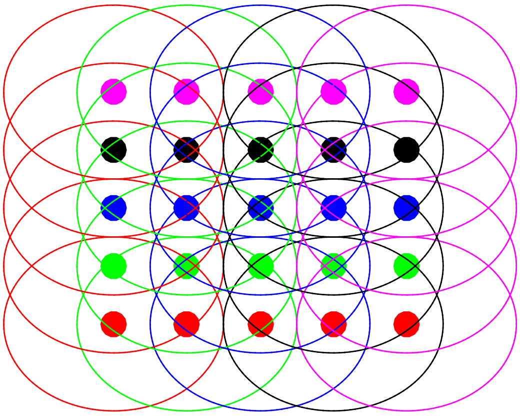


Figure 7.4: MLS circular support

$$Index(1) = \{1, 2, 6, 7\}, \qquad Index(13) = \{7, 8, 9, 12, 13, 14, 17, 18, 19\},$$
$$Index(15) = \{9, 10, 14, 15, 19, 20\}, \qquad Index(25) = \{19, 20, 24, 25\}.$$

- **MLS Weight functions**; A cubic weight function (1.1.16) is chosen to compute shape function, then we have

$$w(1) = \begin{pmatrix} 1.0000 & 0.1111 & 0 & 0 & 0 \\ 0.1111 & 0.0007 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad w(13) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0007 & 0.1111 & 0.0007 & 0 \\ 0 & 0.1111 & 1.0000 & 0.1111 & 0 \\ 0 & 0.0007 & 0.1111 & 0.0007 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$w(15) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0007 & 0.1111 \\ 0 & 0 & 0 & 0.1111 & 1.0000 \\ 0 & 0 & 0 & 0.0007 & 0.1111 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad w(25) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0007 & 0.1111 \\ 0 & 0 & 0 & 0.1111 & 1.0000 \end{pmatrix}.$$

From those results, one can see that the weight functions are positive and non zero only over its support.

- **MLS shape functions**: A linear basis is used to compute shape function, then we have

$$w(1) = \begin{pmatrix} 0.9904 & 0.0096 & 0 & 0 & 0 \\ 0.0096 & -0.0096 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad w(13) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0069 & 0.0891 & 0.0069 & 0 \\ 0 & 0.0891 & 0.6161 & 0.0891 & 0 \\ 0 & 0.0069 & 0.0891 & 0.0069 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$w(15) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0003 & 0.0207 \\ 0 & 0 & 0 & 0.012 & 0.09586 \\ 0 & 0 & 0 & 0.0009 & 0.0.0207 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad w(25) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0011 & 0.0011 \\ 0 & 0 & 0 & 0.0011 & 0.9989 \end{pmatrix}.$$

From those results, one can easily see that MLS shape functions satisfy the consistency property (1.1.18) of MLS approximation.

Bibliography

[1] M.A. Abdelkawy, E.H. Doha, A.H. Bhrawy, A.Z.A. Amin, Efficient pseudospectral scheme for 3D integral equations, Proc. Rom. Acad., Ser. A, Math. Phys. Tech. Sci. Inf. Sci, 18(3) (2017), 199–206.

[2] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C.T. Silva, Computing and rendering point set surfaces, IEEE Trans. Vis. Comput. Graph. 9(1) (2003), 3–15.

[3] M.G. Armentano, Error estimates in Sobolev spaces for moving least square approximations, SIAM J. Numer. Anal, 39(1) (2002), 38–51.

[4] M.G. Armentano and R. G. Duron,Error estimates for moving least square approximations, Appl. Numer. Math, 37(3) (2001), 397–416.

[5] P. Assari, H. Adibi, M. Dehghan, A meshless method based on the moving least squares (MLS) approximation for the numerical solution of two-dimensional nonlinear integral equations of the second kind on non-rectangular domains, Numer Algor, 67 (2014), 423-455.

[6] P. Assari, H. Adibi, M. Dehghan, A numerical method for solving linear integral equations of the second kind on the non-rectangular domains based on the meshless method, Appl. Math. Model, 37(22) (2013), 9269–9294.

[7] P. Assari, F. Asadi-Mehregan, Local radial basis function scheme for solving a class of fractional integro-differential equations based on the use of mixed integral equations, Z Angew Math Mech. 99(8) (2019), e201800236.

[8] K. Atkinson, A. Bogomolny, The discrete Galerkin method for integral equations, Math. Comp. 48(178) (1987), 595-616.

[9] P.L. Auer, C.S. Gardner, Note on singular integral equations of the Kirkwood-Riseman type, Chem. Phys. 23(8) (1955), 1545–1546.

[10] R.T. Baillie, Long memory processes and fractional integration in econometrics. J Econometrics, 73 (5) 1996, 5-59.

[11] R.P. Banaughf, W. Goldsmith, Diffraction of steady acoustic waves by surfaces of arbitrary shape, J. Aeoust. See. Am. 35, (10) (1963), 1590–1601.

[12] G.I. Barenbant, I.P. Zheltov, I.N. Kochain, Basic conception in the theory of seepage of homogeneous liquids in fissured rocks, J. Appl. Math. Mech. 24 (1960) 1286–1309.

[13] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: An overview and recent developments, Comput. Methods Appl. Mech. Eng. 139 (1996), 3–47.

[14] T. Belytschko, T.Y. Lu Y, L. Gu, Element-free Galerkin methods, Internat. J. Numer. Methods Engrg. 37(2) 1994, 229-256.

[15] G.W. Bohannan, Analog fractional order controller in temperature and motor control applications, J. Vib. Control (14) (2008), 1487-1498.

[16] M.D. Buhmann, Radial Basis functions: theory and implementations. Cambridge University Press, Cambridge, 2003.

[17] R. L. Burden, J. Douglas, Numerical Analyisis, Thomson Learning, Boston, Mass, USA, 2001.

[18] T. Canor, V. Denoël, Transient Fokker-Planck-Kolmogorov equation solved with smoothed particle hydrodynamics method, International Journal for Numerical Methods in Engineering, 94(6) 2013, 535-553.

[19] A. Chakrabarti, S.R. Manam, S. Banerjea, Scattering of surface water waves involving a vertical barrier with a gap, J. Eng. Math. 45(2) (2003), 183–194.

[20] M.V.K. Chari, S.J. Salon, Numerical methods in electromagnetism, Academic Press. 1999.

[21] W.C. Chew, M.S. Tong, B. Hu, Integral equation methods for electromagnetic and elastic waves, Morgan Claypool Publishers, 2008.

[22] P.A. Cioica, S. Dahlke, Spatial Besov regularity for semilinear stochastic partial differential equations on bounded Lipschitz domains. Int J Comput Math. 89(18) 2012, 2443-2459.

[23] R.A. Dalrymple, O. Knio, SPH modelling of water waves, In Coastal dynamics' 01, 2001, 779-787.

[24] M. Dehghan, N. Shafieeabyaneh, M. Abbaszadeh, Application of spectral element method for solving Sobolev equations with error estimation. Applied Numerical Mathematics, 158 (2020), 439-462.

[25] M. Dehghan, R. Salehi, The numerical solution of the non-linear integro-differential equations based on the meshless method. Journal of Computational and Applied Mathematics. 236(9) (2012), 2367-2377.

[26] S.I. Denisov, P. Hanggi, H. Kantz. Parameters of the fractional Fokker-Planck equation. Europhysics Letters. (2009);85(4);40007.

[27] J. Dolbow, T. Belytschko, Numerical integration of the galerkin weak form in meshfree methods, Computational mechanics, 23(3) (1999), 219–230.

[28] M. Esmaeilbeigi, F. Mirzaee, D. Moazami. A meshfree method for solving multidimensional linear Fredholm integral equations on the hypercube domains, Appl. Math. Comput. 298 (2017), 236–246.

[29] Z. El Majouti, R. El Jid, and A. Hajjaj. Numerical solution of two-dimensional Fredholm–Hammerstein integral equations on 2D irregular domains by using modified moving least-square method. International Journal of Computer Mathematics 98(8) (2021): 1574- 1593. https://doi.org/10.1080/00207160.2020.1834089

[30] Z. El Majouti, R. El Jid and A. Hajjaj. Solving two-dimensional linear and nonlinear mixed integral equations using moving least squares and modified moving least squares methods. International Journal of Applied Mathematics. 51(1) (2021).

[31] Z. El Majouti, R. El Jid and A. Hajjaj. Numerical solution for three-dimensional nonlinear mixed Volterra–Fredholm integral equations via modified moving least-square method. International Journal of Computer Mathematics (2021):1-19. https://doi.org/10.1080/00207160.2021.2014053

[32] Z. El Majouti, R. El Jid and A. Hajjaj. A meshless method for solving two-dimensional stochastic Fredholm integral equations using regularized moving least squares approximation.Applied Numerical Mathematics, (submitted paper).

[33] Z. El Majouti, R. El Jid and E. Taghizadeh. Moving least squares scheme to approximate the solution of fractional stochastic integro-differential equations. Applied Mathematics and Computation, (submitted paper).

[34] R.E. Ewing, Numerical solution of Sobolev partial differential equations, SIAM J. Numer. Anal, 12 (1974), 345–363.

[35] M. Fallahpour, M. Khodabin, K. Maleknejad, Theoretical error analysis and validation in numerical solution of two-dimensional linear stochastic Volterra-Fredholm integral equation by applying the block-pulse functions, Cogent Mathematics, 4 (1) (2017), 1296750.

[36] M. Fallahpour, K. Maleknejad, M. Khodabin, Approximation solution of two-dimensional linear stochastic Fredholm integral equation by applying the Haar wavelet, Int. J Math Model. Comput, 5 (4) (2015), 361-372.

[37] W. Fang, Y. Wang, Y. Xu. An implementation of fast wavelet Galerkin methods for integral equations of the second kind, J. Sci. Comput, 20 (2) (2004), 277-302

[38] R. Farengo, Y.C. Lee, and P.N. Guzdar, An electromagnetic integral equation: Application to microtearing modes, Phys. Fluids, 26 (12) (1983), 3515–3523.

[39] GE. Fasshauer, Meshfree methods, In: Rieth M, Schommers W (eds) Handbook of theoretical and computational nanotechnology, American Scientific Publishers, Valencia, 2005.

[40] H. Fatahi, J. Saberi-Nadjafi, E. Shivanian, A new spectral meshless radial point interpolation (SM-RPI) method for the two-dimensional Fredholm integral equations on general domains with error analysis, J. Comput. Appl. Math, 294 (2016), 196–209.

[41] T.P. Fries, H. Matthies, Classification and overview of meshfree methods. (2004).

[42] F. Gao, J. Cui, G. Zhao. Weak Galerkin finite element methods for Sobolev equation, Journal of Computational and Applied Mathematics, 317 (2017), 188-202.

[43] R.Gingold, J. Monaghan, Smoothed Particle Hydrodynamics:theory and application to non spherical stars, Astrophysical Journal 181 (1977), 275–389.

[44] A. Golbabai, S. Seifollahi, Radial basis function networks in the numerical solution of linear integro-differential equations. Applied Mathematics and Computation, 188(1) (2007), 427-432.

[45] R. Gorenflo, F. Mainardi, Fractional calculus: integral and differential equations of fractional order, arXiv preprint arXiv:0805.3823, 2008.

[46] P.C. Hansen, T.K. Jensen, Large-scale methods in image deblurring, in: Proceedings of the LNCS of Lecture Notes in Computer Science, 4699, 2007.

[47] R. Hardy, Multiquadric equations of topography and other irregular surfaces, Journal of Geophysical Research, 76 (8) (1971), 1905-1915.

[48] P.M. Hasan, N.A. Sulaiman, F. Soleymani, A. Akgül, The existence and uniqueness of solution for linear system of mixed Volterra-Fredholm integral equations in Banach space. AIMS Mathematics, 5(1) (2020), 226-235.

[49] He J.H. Some applications of nonlinear fractional differential equations and their approximations. Bull Sci Technol, 5(86) (1999), 86-90.

[50] M.H. Heydari , M.R. Hooshmandasl, C. Cattani, FM. Ghaini, An efficient computational method for solving nonlinear stochastic Itˆo-integral equations: Application for stochastic problems in physics, J Comput Phys. 283 (2015), 148-168.

[51] M. H. Heydari, H. Laeli Dastjerdi, M. Nili Ahmadabadi, An Efficient Method for the Numerical Solution of a Class of Nonlinear Fractional Fredholm Integro-Differential Equations. International Journal of Nonlinear Sciences and Numerical Simulation, 19(2) (2018), 165-173.

[52] E. Hopf, Mathematical Problems of Radiative Equilibrium, Stechert-Hafner Service Agency, NewYork, 1964.

[53] A. Horton, A. Wittek, G.R. Joldes, and K. Miller, A meshless total Lagrangian explicit dynamics algorithm for surgical simulation, Int. J. Numer. Methods Biomed. Eng. 26 (8) (2010), 977–998.

[54] T.V. Hromadka, Approximating rainfall-runoff modelling uncertainty using the stochastic integral equation method. Adv Water Resour, 12(1) (1989), 21-25.

[55] L. Huang, X.F. Li, Y. Zhao, x.Y. Duan. Approximate solution of fractional integro-differential equations by Taylor expansion method. Comput. Math. Appl, 62(3) (2011), 1127-1134.

[56] K. Hui-Hsiung, Introduction to stochastic integration, Springer Science+Business Media, Inc. 2006.

[57] K. It$\hat{o}$, Multiple Wiener integral, J. Math. Society of Japan, 3(1) (1951), 157–169.

[58] Y. Ioannou, M. M. Fyrillas, C. Doumanidis, Approximate Solution to Fredholm Integral Equations Using Linear Regression and Applications to Heat and Mass Transfer, Eng. An. Bound. El. 36 (8)(2012), 1278-1283.

[59] Y. Jianjun, Z. Jianlong. Meshless local strong-weak (MLSW) method for irregular domain problems, 49(3) (2017), 659-666.

[60] X. Jin, G.R. Joldes, K. Miller, K.H. Yang, A. Wittek, Meshless algorithm for soft tissue cutting in surgical simulation, Comput. Methods Biomech. Biomed, Eng. 17(7) (2014), 800–811.

[61] G.R. Joldes, H.A. Chowdhury, A. Wittek, B. Doyle, K. Miller, Modified moving least squares with polynomial bases for scattered data approximation. Appl. Math. Comput. 266 (2015), 893–902.

[62] M. Kamrani, Numerical solution of stochastic fractional differential equations. Numerical Algorithms, 68(1) (2015), 81-93.

[63] M. Khodabin, K. Maleknejad,T. Damercheli, Approximate solution of the stochastic Volterra integral equations via expansion method, Int J Ind Math. 6(1) (2014), 41-48.

[64] M. Khodabin, K. Maleknejad, M. Rostami, M. Nouri, Interpolation solution in generalized stochastic exponential population growth model. Appl Math Model. 36(3) 2012, 1023-1033.

[65] F.C. Klebaner. Introduction to stochastic calculus with applications. Imperial College Press. 2005.

[66] A.A. Kilbas, H.M. Srivastava, J.J. Trujillo, Theory and application of fractional differential equations. Amsterdam: Elsevier; 2006.

[67] S. Kumar, I.H. Sloan, A new collocation-type method for Hammerstein integral equations, Math. Comp. 48, (178) (1987), 585–593.

[68] H. Laeli Dastjerdi, F.M. Maalek Ghaini, Numerical solution of Volterra-Fredholm integral equations by moving least square method and Chebyshev polynomials, Appl. Math. Model, 36 (2012), 3283–3288.

[69] H. Laeli Dastjerdi, F.M. Maalek Ghaini, The discrete collocation method for Fredholm–Hammerstein integral equations based on moving least squares method. International Journal of Computer Mathematics, 93(8) (2016), 1347-1357.

[70] H. Laeli Dastjerdi, M. Nili Ahmadabadi. Moving least squares collocation method for Volterra integral equations with proportional delay. International Journal of Computer Mathematics, 94(12) (2017), 2335-2347.

[71] G.R. Liu, Mesh Free Methods: Moving Beyond the Finite Element Method, CRC Press, Boca Raton, 2003.

[72] L. Lucy, A numerical approach to the testing of fission hypothesis, Astrophysical Journal, 82 (1977), 1013–1020.

[73] G.R. Liu, Y. T. Gu, A meshfree method: meshfree weak–strong (MWS) form method, for 2-D solids. Computational Mechanics, 33(1) (2003), 2-14.

[74] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods. International journal for numerical methods in fluids, 20(8-9) (1995), 1081-1106.

[75] G.R. Liu, Y.T. Gu, An introduction to meshfree methods and their programming. Springer Science Business Media.(2005).

[76] P. Lancaster and K. Salkauskas, Surfaces generated by moving least squares methods, *Math. Comput*, 37(155) (1981), 141–158.

[77] D. Levin. The approximation power of moving least-squares, Math. Comput. 67(224) (1998), 1517–1531, 1998.

[78] G.R. Liu, Mesh Free Methods: Moving Beyond the Finite Element Method, CRC Press; 1st Edition July 29, (2002).

[79] Y. Liu, T. Ichiye, Integral equation theories for predicting water structure around molecules, Biophys. Chem. 78 (1999), 97–111.

[80] X. Li, Tang T. Convergence analysis of Jacobi spectral collocation methods for Abel Volterra integral equations of second kind. Front. Math. China.7(2012), 69-84.

[81] P. Lancaster, S. Salkauskas, Surfaces generated by moving least squares methods, Math. Comp. 37(155) 1981; 141-158.

[82] K. Mei, J. Van Bladel, Low-frequency scattering by rectangular cylinders, IEEE Trans. Anten. Propag. 11(1) (1963), 52–56.

[83] D. Mirzaei, M. Dehghan, A meshless based method for solution of integral equations, Appl. Numer. Math. 60 (2010), 245–262.

[84] A.V. Manzhirov, On a method of solving two-dimensional integral equations of axisymmetric contact problems for bodies with complex rheology, J.Appl. Math. Mech. 49 (6) (1985), 777–782.

[85] S.J. Majeed, H. H. Omran, Numerical methods for solving linear Fredholm-Volterra integral equations, Journal of Al-Nahrain University, 11(3) (2008), 131–134.

[86] K. Maleknejad, M. Khodabin, M. Rostami, Numerical solution of stochastic volterra integral equations by a stochastic operational matrix based on block pulse functions, Math Comput Model. 55 (2012), 791-800.

[87] A. Mazzia, M. Ferronato, G. Pini, and G. Gambolati, A comparison of numerical integration rules for the meshless local Petrov–Galerkin method, Numerical Algorithms, 45(1-4) (2007), 61–74.

[88] M. Matinfar, E. Taghizadeh, M. Pourabd, Application of moving least squares algorithm for solving systems of Volterra integral equations. International Journal of Nonlinear Sciences and Numerical Simulation. 22(3-4) (2021), 255-265.

[89] R. Metzler, J. Klafter, The random walk's guide to anomalous diffusion: a fractional dynamics approach, Physics reports, 339(1) (2000), 1-77.

[90] F. Mirzaee, E. Solhi E. Samadyar N. Moving least squares and spectral collocation method to approximate the solution of stochastic Volterra–Fredholm integral equations. Applied Numerical Mathematics. 161 (2021), 275-285.

[91] F. Mirzaee, E. Hadadiyan, Applying the modified block-pulse functions to solve the three-dimensional Volterra-Fredholm integral equations, Appl. Math. Comput. 265 (2015), 759–767.

[92] F. Mirzaee, S. Alipour, Cubic B-spline approximation for linear stochastic integro-differential equation of fractional order, Journal of Computational and Applied Mathematics. 366 (2020), 112440.

[93] F. Mirzaee, N. Samadyar, SF. Hoseini, Euler polynomial solutions of nonlinear stochastic It$\hat{o}$-Volterra integral equations, J Comput Appl Math. 330 (2018), 574-585.

[94] D. Mirzaei, Analysis of moving least squares approximation revisited. Journal of Computational and Applied Mathematics. 282 (2015), 237-250.

[95] D. Mirzaei, R. Schaback, M. Dehghan, On generalized moving least squares and diffuse derivatives. IMA Journal of Numerical Analysis, 32(3) (2012), 983-1000.

[96] F. Mohammadi, A wavelet-based computational method for solving stochastic It$\hat{o}$-Volterra integral equations, J Comput Phys. 298 (2015), 254-265.

[97] F. Mohammadi, Wavelet Galerkin method for solving stochastic fracthional differential equations. J. Fract. Calc. Appl. 7 (2016), 73-86.

[98] B. Nayroles, G. Touzot, P. Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. Computational mechanics, 10(5) (1992), 307-318.

[99] R.R. Nigmatulin, The realization of the generalized transfer equation in a medium with fractal geometry. Phys. Status Solidi B, 133 (1986), 425–430.

[100] I. Podlubny, Fractional differential equations, Mathematics in Science and Engineering, 198. Academic Press, Inc., San Diego, CA, 1999.

[101] I. Podlubny, Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications. Elsevier, 1998.

[102] P.W. Randles, L.D. Libersky, Smoothed particle hydrodynamics: Some recent improvements and applications. Computer Methods in Applied Mechanics and Engineering, 139(1-4) 1996, 375-408.

[103] H. Ren, K. Pei, L. Wang, Error analysis for moving least squares approximation in 2D space, Applied Mathematics and Computation, 238 (2014). 527–546.

[104] J. Radlow, A two-dimensional singular integral equation of diffraction theory, Bull. Amer. Math. Soc. 70 (4) (1964), 596–599.

[105] Y.A. Rossikhin, M.V. Shitikova, Applications of fractional calculus to dynamic problems of linear and nonlinear hereditary mechanics of solids. Appl Mech Rev, 50(15) 1997.

[106] E. Shivanian, A new spectral meshless radial point interpolation (SMRPI) method: a well-behaved alternative to the meshless weak forms, Engineering Analysis with Boundary Elements, 54(1–12), 2015.

[107] D. Shepard, A two-dimensional interpolation function for irregularly spaced points, In: Proceedings. 23rd Nat. Conf. ACM, ACM Press, New York, 1968, 517–524.

[108] C. Shen, J.F. O'Brien, J.R. Shewchuk, Interpolating and approximating implicit surfaces from polygon soup, ACM Trans. Graph. 23 (3) (2004), 896–904.

[109] M. Saffarzadeh, G.B. Loghmani, M. Heydari, An iterative technique for the numerical solution of nonlinear stochastic It$\hat{o}$-Volterra integral equations, J Comput Appl Math, 333 (2018), 74-86.

[110] N.Samadyar, F. Mirzaee, Numerical solution of two-dimensional stochastic Fredholm integral equations on hypercube domains via meshfree approach, J Comput Appl Math. 377 (2020), 112875.

[111] A.K. Singh, M. Mehra, Wavelet collocation method based on Legendre polynomials and its application in solving the stochastic fractional integro-differential equations. Journal of Computational Science, (51) (2021), 101342.

[112] D. Shepard, A two-dimensional interpolation function for irregularly spaced points. in: Proc. 23rd Nat. Conf. ACM, ACM Press. New York. 1968, 517–524

[113] S. Shen, et al. Numerical techniques for the variable order time fractional diffusion equation, Applied Mathematics and Computation 218(22) (2012), 10861-10870.

[114] R.D. Small. Population growth in a closed model. In: Mathematical modelling: classroom notes in applied mathematics, SIAM, Philadelphia. 1989.

[115] D.M. Shi, On the initial boundary value problem of the nonlinear equation of the migration of the moisture in soil, Acta Math. Appl. Sin. 13 (1990), 31–38.

[116] R.E. Showalter, Existence and representation theorems for a semilinear Sobolev equation in Banach space, SIAM J. Math. Anal. 3 (3) (1972), 527–543.

[117] A.N. Tikhonov, V.Y. Arsenin, Solution of Ill-posed Problems, Winston Sons, Washington, 1977.

[118] Q. Tang, D. Waxman, An integral equation describing an asexual population in a changing environment, Nonli. Anal. 53 (2003), 683–699

[119] T.W. Ting, A cooling process according to two-temperature theory of heat conduction, J. Math. Anal. Appl. 45 (1974), 23–31.

[120] E. Taghizadeh, M. Matinfar, Modified numerical approaches for a class of Volterra integral equations with proportional delays. Comp. Appl. Math. 38(63), 2019.

[121] G.M. Vainikko, Perturbed Galerkin method and the general theory of approximate methods for nonlinear equations, USSR Comput. Math. Math. Phys. 7(4) (1967), 1–41.

[122] X. Zhang, X.H. Liu, K.Z. Song, M.W. Lu MW, Least-square collocation meshless method. Int. J. Numer. Methods Engrg, 51 (9) (2001), 1089-1100.

[123] H. Zhang, Y. Chen, X. Nie, Solving the linear integral equations based on radial basis function interpolation, Journal of Applied Mathematics, 2014.

[124] T. E. E. R. A. W. A. T. Wongyat, W. Sintunavarat. The existence and uniqueness of the solution for nonlinear Fredholm and Volterra integral equations via adapting-ceiling distances. J. Math. Anal. 8(5) (2017), 105-118.

[125] K.F. Warnick, Numerical analysis for electromagnetic integral equations, Artech House, 2008.

[126] Q. Wang, W. Zhou, Y. Cheng, G. Ma, X. Chang, Y. Miao, E. Chen, Regularized moving least-square method and regularized improved interpolating moving least-square method with nonsingular moment matrices, Appl. Math. Comput. 325 (2018), 120–145.

[127] Q. Wang, W. Zhou, Y.T. Feng, G. Ma, Y. Cheng, X. Chang, An adaptive orthogonal improved interpolating moving least-square method and a new boundary element-free method, Appl. Math. Comput. 353 (2019), 347–370.

[128] M. Zarebnia, A numerical solution of nonlinear Volterra-Fredholm integral equations, Journal of Applied Analysis and Computation, 3(1) (2013), 95-104.

[129] C. Zuppa, Error estimates for moving least square approximations, Bull. Braz. Math. Soc. New Series, 34(2) 2003. 231–249.

[130] H. Wendland, Scattered data approximation, Cambridge University Press, 2004.